

[www.aurora-group.eu](http://www.aurora-group.eu)  
[www.nbpublish.com](http://www.nbpublish.com)

# ПРОГРАММНЫЕ СИСТЕМЫ

И

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ

*научный журнал*



## Выходные данные

Номер подписан в печать: 03-07-2024

Учредитель: Даниленко Василий Иванович, w.danilenko@nbpublish.com

Издатель: ООО <НБ-Медиа>

Главный редактор: Морозов Михаил Николаевич, кандидат технических наук,  
mikhail.n.morozov@gmail.com

ISSN: 2454-0714

Контактная информация:

Выпускающий редактор - Зубкова Светлана Вадимовна

E-mail: info@nbpublish.com

тел.+7 (966) 020-34-36

Почтовый адрес редакции: 115114, г. Москва, Павелецкая набережная, дом 6А, офис 211.

Библиотека журнала по адресу: [http://www.nbpublish.com/library\\_tariffs.php](http://www.nbpublish.com/library_tariffs.php)

## Publisher's imprint

Number of signed prints: 03-07-2024

Founder: Danilenko Vasiliy Ivanovich, w.danilenko@nbpublish.com

Publisher: NB-Media ltd

Main editor: Morozov Mikhail Nikolaevich, kandidat tekhnicheskikh nauk,  
mikhail.n.morozov@gmail.com

ISSN: 2454-0714

Contact:

Managing Editor - Zubkova Svetlana Vadimovna

E-mail: info@nbpublish.com

тел.+7 (966) 020-34-36

Address of the editorial board : 115114, Moscow, Paveletskaya nab., 6A, office 211 .

Library Journal at : [http://en.nbpublish.com/library\\_tariffs.php](http://en.nbpublish.com/library_tariffs.php)

## Редакционный совет

**Гельман Виктор Яковлевич** – доктор технических наук, профессор, профессор кафедры медицинской информатики и физики ФГБОУ ВО «Северо-Западный государственный медицинский университет им. И.И.Мечникова», 191015, Россия, г. Санкт-Петербург, ул. Кирочная, д.41, [gelm@sg2104.spb.edu](mailto:gelm@sg2104.spb.edu)

**Поляков Виктор Павлович** – доктор педагогических наук, профессор, Главный научный сотрудник лаборатории психолого-педагогического и учебно-методического обеспечения развития информатизации образования Центра информатизации образования Федерального государственного бюджетного научного учреждения «Институт управления образованием Российской академии образования», 105062, г. Москва, ул. Макаренко, д. 5/16, стр. 1Б, [polvikpal@mail.ru](mailto:polvikpal@mail.ru)

**Гармаев Баир Заятуевич** – кандидат физико-математических наук, научный сотрудник, Институт физического материаловедения Сибирского Отделения РАН, 670000, Россия, республика Бурятия, г. Улан-Удэ, ул. Сахьяновой, 6, каб. 313

**Клименко Анна Борисовна** – кандидат технических наук, научный сотрудник Научно-исследовательского института многопроцессорных вычислительных систем имени академика А.В. Каляева Южного федерального университета (НИИ МВС ЮФУ), 347935, Россия, Ростовская область, г. Таганрог, ул. 8 Переулок, 15

**Лютикова Лариса Адольфовна** – кандидат физико-математических наук, заведующая отделом Нейроинформатики и машинного обучения, Институт прикладной математики и автоматизации Кабардино-Балкарского научного центра РАН – филиал Кабардино-Балкарского научного центра РАН (ИПМА КБНЦ РАН), 360000, Россия, Республика Кабардино-Балкария, г. Нальчик, ул. Шортанова, 89а

**Мустафаев Арслан Гасанович** – доктор технических наук, профессор, Государственное автономное образовательное учреждение высшего образования "Дагестанский государственный университет народного хозяйства", кафедра «Информационные технологии и информационная безопасность», 367015, Россия, Республика Дагестан, г. Махачкала, ул. Атаева, 5, каб. 4.5

**Шестаков Александр Валентинович** – кандидат технических наук, доцент Южный Федеральный университет, кафедра вычислительной техники, 347902, Россия, Ростовская область, г. Таганрог, ул. Свободы, 24/2

**Сидоркина Ирина Геннадьевна** – доктор технических наук, профессор, декан факультета Информатики и вычислительной техники Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: [dekan\\_fivt@mail.ru](mailto:dekan_fivt@mail.ru)

**Екатерина Прасолова-Førland** - PhD, Норвежский университет науки и технологии (NTNU), Трондхейм, Норвегия E-mail: [Ekaterina.Prasolova-Forland@idi.ntnu.no](mailto:Ekaterina.Prasolova-Forland@idi.ntnu.no)

**Голенков Владимир Васильевич** - доктор технических наук, профессор, заведующий кафедрой Интеллектуальных информационных технологий Белорусского государственного университета информатики и радиоэлектроники, г. Минск, Республика Беларусь E-mail: [golen@bsuir.by](mailto:golen@bsuir.by)

**Домошницкий Александр Исаакович** - кандидат физико-математических наук, декан естественно-научного факультета Университетского центра в г.Ариэль, Израиль, Самариа E-mail: [adom@ariel.ac.il](mailto:adom@ariel.ac.il) Department of Mathematics and Computer Sciences, The Ariel



University Center of Samaria, 44837 Ariel, ISRAEL

**Коробейников Анатолий Григорьевич** - доктор технических наук, профессор «Институт земного магнетизма, ионосферы и распространения радиоволн РАН (ИЗМИРАН)», Санкт-Петербургский филиал E-mail: [korobeynikov\\_a\\_g@mail.ru](mailto:korobeynikov_a_g@mail.ru)

**Заболеева-Зотова Алла Викторовна**, доктор технических наук, профессор Волгоградского технического университета, Волгоград, Россия E-mail: [zabzot@gmail.com](mailto:zabzot@gmail.com)

**Бенкевич Леонид Владимирович** - кандидат физических наук и инженерной физики, научный сотрудник Массачусеттского Технологического Института (MIT), обсерватория Хэйстек, Бостон, США E-mail: [ibenkev@gmail.com](mailto:ibenkev@gmail.com)

**Морозов Михаил Николаевич** - кандидат технических наук, профессор, руководитель лаборатории мультимедиа, заведующий кафедрой Информатики и системного программирования Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: [mikhail.n.morozov@gmail.com](mailto:mikhail.n.morozov@gmail.com)

**Олзоева Сэсэг Ивановна** - доктор технических наук, профессор, Восточно-Сибирский государственный университет технологий и управления (г. Улан-Уде) E-mail: [sseseg@yandex.ru](mailto:sseseg@yandex.ru)

**Курейчик Владимир Викторович** - доктор технических наук, профессор, заведующий кафедрой Систем автоматизации проектирования Технологического института «Южного федерального университета» в г.Таганрог, Россия E-mail: [ykur@tsure.ru](mailto:ykur@tsure.ru)

**Филатова Наталья Николаевна** - доктор технических наук, профессор, Тверской государственный технический университет, Тверь, Россия E-mail: [nfilatova99@mail.ru](mailto:nfilatova99@mail.ru)

**Песошин Валерий Андреевич** - член-корреспондент Академии наук Республики Татарстан, заслуженный деятель науки Республики Татарстан и Российской Федерации, доктор технических наук, профессор, заслуженный деятель науки и техники Республики Татарстан. Заведующий кафедрой Компьютерных систем Казанского национальный исследовательский университет им. А.Н. Туполева, Казань, Россия E-mail: [pesoshin@evm.kstu-kai.ru](mailto:pesoshin@evm.kstu-kai.ru)

**Краснов Сергей Викторович** - доктор технических наук, профессор, проректор по научно-исследовательской работе, заведующий кафедрой Информатика и системы управления Волжского университета им. Татищева, Тольятти, Россия E-mail: [krasnovtlt@mail.ru](mailto:krasnovtlt@mail.ru)

**Горохов Алексей Витальевич** - доктор технических наук, профессор кафедры Прикладной математики и информационных технологий Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: [agv64@mail.ru](mailto:agv64@mail.ru)

**Галанина Наталья Андреевна** - доктор технических наук, профессор, Чувашский государственный университет им. И.Н.Ульянова, Чебоксары, Россия E-mail: [galaninacheb@mail.ru](mailto:galaninacheb@mail.ru)

**Сюзов Владимир Васильевич** - доктор технических наук, профессор, заведующий кафедрой Компьютерные системы и сети Московского государственного технического университета им. Н. Э. Баумана, Москва, Россия E-mail: [v.suzev@bmstu.ru](mailto:v.suzev@bmstu.ru)

**Леухин Анатолий Николаевич** - доктор физико-математических наук, профессор, заведующий кафедрой Информационной безопасности Поволжского государственного технологического университета, Йошкар-Ола, Россия E-mail: [code@volgatech.net](mailto:code@volgatech.net)

**Гвинианидзе Темур Николаевич** - Доктор технических наук, профессор,  
Государственный университет им. Ак. Церетели Грузия, г. Кутаиси, пр. Тamar-мепе 59.  
П.и 4600. [temuri1951@mail.ru](mailto:temuri1951@mail.ru)

## Council of Editors

**Gelman Viktor Yakovlevich** – Doctor of Technical Sciences, Professor, Professor of the Department of Medical Informatics and Physics of the I.I.Mechnikov Northwestern State Medical University, 41 Kirochnaya Str., St. Petersburg, 191015, Russia, [gelm@sg2104.spb.edu](mailto:gelm@sg2104.spb.edu)

**Polyakov Viktor Pavlovich** – Doctor of Pedagogical Sciences, Professor, Chief Researcher of the Laboratory of Psychological, Pedagogical and Educational methodological support for the development of Informatization of Education of the Center for Informatization of Education of the Federal State Budgetary Scientific Institution "Institute of Education Management of the Russian Academy of Education", 105062, Moscow, Makarenko str., 5/16, p. 1B, [polvikpal@mail.ru](mailto:polvikpal@mail.ru)

**Garmaev Bair Zayatuevich** – Candidate of Physical and Mathematical Sciences, Researcher, Institute of Physical Materials Science of the Siberian Branch of the Russian Academy of Sciences, 670000, Russia, Republic of Buryatia, Ulan-Ude, Sakhyanova str., 6, room 313

**Klimenko Anna Borisovna** – Candidate of Technical Sciences, Researcher at the Research Institute of Multiprocessor Computing Systems named after Academician A.V. Kalyaev of the Southern Federal University (Research Institute of the Ministry of Internal Affairs of the Southern Federal University), 347935, Russia, Rostov region, Taganrog, ul. 8 Lane, 15

**Lyutikova Larisa Adolfofna** – Candidate of Physical and Mathematical Sciences, Head of the Department of Neuroinformatics and Machine Learning, Institute of Applied Mathematics and Automation of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences - branch of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences (IPMA KBSC RAS), 360000, Russia, Republic of Kabardino-Balkaria, Nalchik, 89a Shortanova str.

**Mustafayev Arslan Hasanovich** – Doctor of Technical Sciences, Professor, State Autonomous Educational Institution of Higher Education "Dagestan State University of National Economy", Department of "Information Technologies and Information Security", 367015, Russia, Republic of Dagestan, Makhachkala, Ataeva str., 5, office 4.5

**Alexander V. Shestakov** – Candidate of Technical Sciences, Associate Professor, Southern Federal University, Department of Computer Engineering, 24/2 Svobody str., Taganrog, Rostov Region, 347902, Russia

**Sidorkina Irina Gennadievna** - Doctor of Technical Sciences, Professor, Dean of the Faculty of Computer Science and Computer Engineering of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: [dekan\\_fivt@mail.ru](mailto:dekan_fivt@mail.ru)

**Ekaterina Prasolova-Forland** - PhD, Norwegian University of Science and Technology (NTNU), Trondheim, Norway E-mail: [Ekaterina.Prasolova-Forland@idi.ntnu.no](mailto:Ekaterina.Prasolova-Forland@idi.ntnu.no)

**Golenkov Vladimir Vasilyevich** - Doctor of Technical Sciences, Professor, Head of the Department of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus E-mail: [golen@bsuir.by](mailto:golen@bsuir.by)

**Domoshnitsky Alexander Isaakovich** - Candidate of Physical and Mathematical Sciences, Dean of the Faculty of Natural Sciences of the University Center in Ariel, Israel, Samaria E-mail: [adom@ariel.ac.il](mailto:adom@ariel.ac.il) Department of Mathematics and Computer Sciences, The Ariel University Center of Samaria, 44837 Ariel, ISRAEL

**Korobeynikov Anatoly Grigorievich** - Doctor of Technical Sciences, Professor, Institute of

Terrestrial Magnetism, Ionosphere and Radio Wave Propagation of the Russian Academy of Sciences (IZMIRAN), St. Petersburg Branch E-mail: [korobeynikov\\_a\\_g@mail.ru](mailto:korobeynikov_a_g@mail.ru)

**Zaboleeva-Zotova Alla Viktorovna**, Doctor of Technical Sciences, Professor of Volgograd Technical University, Volgograd, Russia E-mail: [zabzot@gmail.com](mailto:zabzot@gmail.com)

**Leonid V. Benkevich** - Candidate of Physical Sciences and Engineering Physics, Researcher at the Massachusetts Institute of Technology (MIT), Haystack Observatory, Boston, USA E-mail: [lbenkev@gmail.com](mailto:lbenkev@gmail.com)

**Mikhail N. Morozov** - Candidate of Technical Sciences, Professor, Head of the Multimedia Laboratory, Head of the Department of Computer Science and System Programming of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: [mikhail.n.morozov@gmail.com](mailto:mikhail.n.morozov@gmail.com)

**Olzoeva Seseg Ivanovna** - Doctor of Technical Sciences, Professor, East Siberian State University of Technology and Management (Ulan-Ude) E-mail: [sseseg@yandex.ru](mailto:sseseg@yandex.ru)

**Kureychik Vladimir Viktorovich** - Doctor of Technical Sciences, Professor, Head of the Department of Design Automation Systems of the Technological Institute of the Southern Federal University in Taganrog, Russia E-mail: [vkur@tsure.ru](mailto:vkur@tsure.ru)

**Natalia Filatova** - Doctor of Technical Sciences, Professor, Tver State Technical University, Tver, Russia E-mail: [nfilatova99@mail.ru](mailto:nfilatova99@mail.ru)

**Pesoshin Valery Andreevich** - Corresponding member of the Academy of Sciences of the Republic of Tatarstan, Honored Scientist of the Republic of Tatarstan and the Russian Federation, Doctor of Technical Sciences, Professor, Honored Worker of Science and Technology of the Republic of Tatarstan. Head of the Department of Computer Systems of Kazan National Research University named after A.N. Tupolev, Kazan, Russia E-mail: [pesoshin@evm.kstu-kai.ru](mailto:pesoshin@evm.kstu-kai.ru)

**Krasnov Sergey Viktorovich** - Doctor of Technical Sciences, Professor, Vice-Rector for Research, Head of the Department of Computer Science and Control Systems of the Volga State University. Tatishcheva, Togliatti, Russia E-mail: [krasnovtlt@mail.ru](mailto:krasnovtlt@mail.ru)

**Gorokhov Alexey Vitalievich** - Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Information Technologies of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: [agv64@mail.ru](mailto:agv64@mail.ru)

**Galanina Natalia Andreevna** - Doctor of Technical Sciences, Professor, I.N.Ulyanov Chuvash State University, Cheboksary, Russia E-mail: [galaninacheb@mail.ru](mailto:galaninacheb@mail.ru)

**Vladimir V. Syuzev** - Doctor of Technical Sciences, Professor, Head of the Department of Computer Systems and Networks of the Bauman Moscow State Technical University, Moscow, Russia E-mail: [v.suzev@bmstu.ru](mailto:v.suzev@bmstu.ru)

**Leukhin Anatoly Nikolaevich** - Doctor of Physical and Mathematical Sciences, Professor, Head of the Department of Information Security of the Volga State Technological University, Yoshkar-Ola, Russia E-mail: [code@volgatech.net](mailto:code@volgatech.net)

**Gvinianidze Temur Nikolaevich** - Doctor of Technical Sciences, Professor, Ak. Tsereteli State University Georgia, Kutaisi, 59 Tamar-mepe Ave., and 4600. [temuri1951@mail.ru](mailto:temuri1951@mail.ru)





## Требования к статьям

Журнал является научным. Направляемые в издательство статьи должны соответствовать тематике журнала (с его рубрикатором можно ознакомиться на сайте издательства), а также требованиям, предъявляемым к научным публикациям.

Рекомендуемый объем от 12000 знаков.

Структура статьи должна соответствовать жанру научно-исследовательской работы. В ее содержании должны обязательно присутствовать и иметь четкие смысловые разграничения такие разделы, как: предмет исследования, методы исследования, апелляция к оппонентам, выводы и научная новизна.

Не приветствуется, когда исследователь, трактуя в статье те или иные научные термины, вступает в заочную дискуссию с авторами учебников, учебных пособий или словарей, которые в узких рамках подобных изданий не могут широко излагать свое научное воззрение и заранее оказываются в проигрышном положении. Будет лучше, если для научной полемики Вы обратитесь к текстам монографий или диссертационных работ оппонентов.

Не превращайте научную статью в публицистическую: не наполняйте ее цитатами из газет и популярных журналов, ссылками на высказывания по телевидению.

Ссылки на научные источники из Интернета допустимы и должны быть соответствующим образом оформлены.

Редакция отвергает материалы, напоминающие реферат. Автору нужно не только продемонстрировать хорошее знание обсуждаемого вопроса, работ ученых, исследовавших его прежде, но и привнести своей публикацией определенную научную новизну.

Не принимаются к публикации избранные части из диссертаций, книг, монографий, поскольку стиль изложения подобных материалов не соответствует журнальному жанру, а также не принимаются материалы, публиковавшиеся ранее в других изданиях.

В случае отправки статьи одновременно в разные издания автор обязан известить об этом редакцию. Если он не сделал этого заблаговременно, рискует репутацией: в дальнейшем его материалы не будут приниматься к рассмотрению.

Уличенные в плагиате попадают в «черный список» издательства и не могут рассчитывать на публикацию. Информация о подобных фактах передается в другие издательства, в ВАК и по месту работы, учебы автора.

Статьи представляются в электронном виде только через сайт издательства <http://www.e-notabene.ru> кнопка "Авторская зона".

Статьи без полной информации об авторе (соавторах) не принимаются к рассмотрению, поэтому автор при регистрации в авторской зоне должен ввести полную и корректную информацию о себе, а при добавлении статьи - о всех своих соавторах.

Не набирайте название статьи прописными (заглавными) буквами, например: «ИСТОРИЯ КУЛЬТУРЫ...» — неправильно, «История культуры...» — правильно.

При добавлении статьи необходимо прикрепить библиографию (минимум 10–15 источников, чем больше, тем лучше).

При добавлении списка использованной литературы, пожалуйста, придерживайтесь следующих стандартов:

- [ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание. Общие требования и правила составления.](#)
- [ГОСТ 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления](#)

В каждой ссылке должен быть указан только один диапазон страниц. В теле статьи ссылка на источник из списка литературы должна быть указана в квадратных скобках, например, [1]. Может быть указана ссылка на источник со страницей, например, [1, с. 57], на группу источников, например, [1, 3], [5-7]. Если идет ссылка на один и тот же источник, то в теле статьи нумерация ссылок должна выглядеть так: [1, с. 35]; [2]; [3]; [1, с. 75-78]; [4].... А в библиографии они должны отображаться так:

[1]

[2]

[3]

[4]....

Постраничные ссылки и сноски запрещены. Если вы используете сноску, не содержащую ссылку на источник, например, разъяснение термина, включите сноску в текст статьи.

После процедуры регистрации необходимо прикрепить аннотацию на русском языке, которая должна состоять из трех разделов: Предмет исследования; Метод, методология исследования; Новизна исследования, выводы.

Прикрепить 10 ключевых слов.

Прикрепить саму статью.

Требования к оформлению текста:

- Кавычки даются уголками (« ») и только кавычки в кавычках — лапками (" ").
- Тире между датами дается короткое (Ctrl и минус) и без отбивок.
- Тире во всех остальных случаях дается длинное (Ctrl, Alt и минус).
- Даты в скобках даются без г.: (1932–1933).
- Даты в тексте даются так: 1920 г., 1920-е гг., 1540–1550-е гг.
- Недопустимо: 60-е гг., двадцатые годы двадцатого столетия, двадцатые годы XX столетия, 20-е годы XX столетия.
- Века, король такой-то и т.п. даются римскими цифрами: XIX в., Генрих IV.
- Инициалы и сокращения даются с пробелом: т. е., т. д., М. Н. Иванов. Неправильно: М.Н. Иванов, М.Н. Иванов.

**ВСЕ СТАТЬИ ПУБЛИКУЮТСЯ В АВТОРСКОЙ РЕДАКЦИИ.**

**По вопросам публикации и финансовым вопросам** обращайтесь к администратору Зубковой Светлане Вадимовне

E-mail: [info@nbpublish.com](mailto:info@nbpublish.com)

или по телефону +7 (966) 020-34-36

**Подробные требования к написанию аннотаций:**

Аннотация в периодическом издании является источником информации о содержании статьи и изложенных в ней результатах исследований.

Аннотация выполняет следующие функции: дает возможность установить основное

содержание документа, определить его релевантность и решить, следует ли обращаться к полному тексту документа; используется в информационных, в том числе автоматизированных, системах для поиска документов и информации.

Аннотация к статье должна быть:

- информативной (не содержать общих слов);
- оригинальной;
- содержательной (отражать основное содержание статьи и результаты исследований);
- структурированной (следовать логике описания результатов в статье);

Аннотация включает следующие аспекты содержания статьи:

- предмет, цель работы;
- метод или методологию проведения работы;
- результаты работы;
- область применения результатов; новизна;
- выводы.

Результаты работы описывают предельно точно и информативно. Приводятся основные теоретические и экспериментальные результаты, фактические данные, обнаруженные взаимосвязи и закономерности. При этом отдается предпочтение новым результатам и данным долгосрочного значения, важным открытиям, выводам, которые опровергают существующие теории, а также данным, которые, по мнению автора, имеют практическое значение.

Выводы могут сопровождаться рекомендациями, оценками, предложениями, гипотезами, описанными в статье.

Сведения, содержащиеся в заглавии статьи, не должны повторяться в тексте аннотации. Следует избегать лишних вводных фраз (например, «автор статьи рассматривает...», «в статье рассматривается...»).

Исторические справки, если они не составляют основное содержание документа, описание ранее опубликованных работ и общеизвестные положения в аннотации не приводятся.

В тексте аннотации следует употреблять синтаксические конструкции, свойственные языку научных и технических документов, избегать сложных грамматических конструкций.

**Гонорары за статьи в научных журналах не начисляются.**

#### **Цитирование или воспроизведение текста, созданного ChatGPT, в вашей статье**

Если вы использовали ChatGPT или другие инструменты искусственного интеллекта в своем исследовании, опишите, как вы использовали этот инструмент, в разделе «Метод» или в аналогичном разделе вашей статьи. Для обзоров литературы или других видов эссе, ответов или рефератов вы можете описать, как вы использовали этот инструмент, во введении. В своем тексте предоставьте prompt - командный вопрос, который вы использовали, а затем любую часть соответствующего текста, который был создан в ответ.

К сожалению, результаты «чата» ChatGPT не могут быть получены другими читателями, и хотя невозстановимые данные или цитаты в статьях APA Style обычно цитируются как личные сообщения, текст, сгенерированный ChatGPT, не является сообщением от человека.

Таким образом, цитирование текста ChatGPT из сеанса чата больше похоже на совместное использование результатов алгоритма; таким образом, сделайте ссылку на автора алгоритма записи в списке литературы и приведите соответствующую цитату в тексте.

Пример:

На вопрос «Является ли деление правого полушария левого полушария реальным или метафорой?» текст, сгенерированный ChatGPT, показал, что, хотя два полушария мозга в некоторой степени специализированы, «обозначение, что люди могут быть охарактеризованы как «левополушарные» или «правополушарные», считается чрезмерным упрощением и популярным мифом» (OpenAI, 2023).

#### **Ссылка в списке литературы**

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].  
<https://chat.openai.com/chat>

Вы также можете поместить полный текст длинных ответов от ChatGPT в приложение к своей статье или в дополнительные онлайн-материалы, чтобы читатели имели доступ к точному тексту, который был сгенерирован. Особенно важно задокументировать точный созданный текст, потому что ChatGPT будет генерировать уникальный ответ в каждом сеансе чата, даже если будет предоставлен один и тот же командный вопрос. Если вы создаете приложения или дополнительные материалы, помните, что каждое из них должно быть упомянуто по крайней мере один раз в тексте вашей статьи в стиле APA.

Пример:

При получении дополнительной подсказки «Какое представление является более точным?» в тексте, сгенерированном ChatGPT, указано, что «разные области мозга работают вместе, чтобы поддерживать различные когнитивные процессы» и «функциональная специализация разных областей может меняться в зависимости от опыта и факторов окружающей среды» (OpenAI, 2023; см. Приложение А для полной расшифровки). .

#### **Ссылка в списке литературы**

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].  
<https://chat.openai.com/chat> Создание ссылки на ChatGPT или другие модели и программное обеспечение ИИ

Приведенные выше цитаты и ссылки в тексте адаптированы из шаблона ссылок на программное обеспечение в разделе 10.10 Руководства по публикациям (Американская психологическая ассоциация, 2020 г., глава 10). Хотя здесь мы фокусируемся на ChatGPT, поскольку эти рекомендации основаны на шаблоне программного обеспечения, их можно адаптировать для учета использования других больших языковых моделей (например, Bard), алгоритмов и аналогичного программного обеспечения.

Ссылки и цитаты в тексте для ChatGPT форматируются следующим образом:

OpenAI. (2023). ChatGPT (версия от 14 марта) [большая языковая модель].  
<https://chat.openai.com/chat>

Цитата в скобках: (OpenAI, 2023)

Описательная цитата: OpenAI (2023)

Давайте разберем эту ссылку и посмотрим на четыре элемента (автор, дата, название и

источник):

Автор: Автор модели OpenAI.

Дата: Дата — это год версии, которую вы использовали. Следуя шаблону из Раздела 10.10, вам нужно указать только год, а не точную дату. Номер версии предоставляет конкретную информацию о дате, которая может понадобиться читателю.

Заголовок. Название модели — «ChatGPT», поэтому оно служит заголовком и выделено курсивом в ссылке, как показано в шаблоне. Хотя OpenAI маркирует уникальные итерации (например, ChatGPT-3, ChatGPT-4), они используют «ChatGPT» в качестве общего названия модели, а обновления обозначаются номерами версий.

Номер версии указан после названия в круглых скобках. Формат номера версии в справочниках ChatGPT включает дату, поскольку именно так OpenAI маркирует версии. Различные большие языковые модели или программное обеспечение могут использовать различную нумерацию версий; используйте номер версии в формате, предоставленном автором или издателем, который может представлять собой систему нумерации (например, Версия 2.0) или другие методы.

Текст в квадратных скобках используется в ссылках для дополнительных описаний, когда они необходимы, чтобы помочь читателю понять, что цитируется. Ссылки на ряд общих источников, таких как журнальные статьи и книги, не включают описания в квадратных скобках, но часто включают в себя вещи, не входящие в типичную рецензируемую систему. В случае ссылки на ChatGPT укажите дескриптор «Большая языковая модель» в квадратных скобках. OpenAI описывает ChatGPT-4 как «большую мультимодальную модель», поэтому вместо этого может быть предоставлено это описание, если вы используете ChatGPT-4. Для более поздних версий и программного обеспечения или моделей других компаний могут потребоваться другие описания в зависимости от того, как издатели описывают модель. Цель текста в квадратных скобках — кратко описать тип модели вашему читателю.

Источник: если имя издателя и имя автора совпадают, не повторяйте имя издателя в исходном элементе ссылки и переходите непосредственно к URL-адресу. Это относится к ChatGPT. URL-адрес ChatGPT: <https://chat.openai.com/chat>. Для других моделей или продуктов, для которых вы можете создать ссылку, используйте URL-адрес, который ведет как можно более напрямую к источнику (т. е. к странице, на которой вы можете получить доступ к модели, а не к домашней странице издателя).

### **Другие вопросы о цитировании ChatGPT**

Вы могли заметить, с какой уверенностью ChatGPT описал идеи латерализации мозга и то, как работает мозг, не ссылаясь ни на какие источники. Я попросил список источников, подтверждающих эти утверждения, и ChatGPT предоставил пять ссылок, четыре из которых мне удалось найти в Интернете. Пятая, похоже, не настоящая статья; идентификатор цифрового объекта, указанный для этой ссылки, принадлежит другой статье, и мне не удалось найти ни одной статьи с указанием авторов, даты, названия и сведений об источнике, предоставленных ChatGPT. Авторам, использующим ChatGPT или аналогичные инструменты искусственного интеллекта для исследований, следует подумать о том, чтобы сделать эту проверку первоисточников стандартным процессом. Если источники являются реальными, точными и актуальными, может быть лучше прочитать эти первоисточники, чтобы извлечь уроки из этого исследования, и перефразировать или процитировать эти статьи, если применимо, чем использовать их интерпретацию модели.

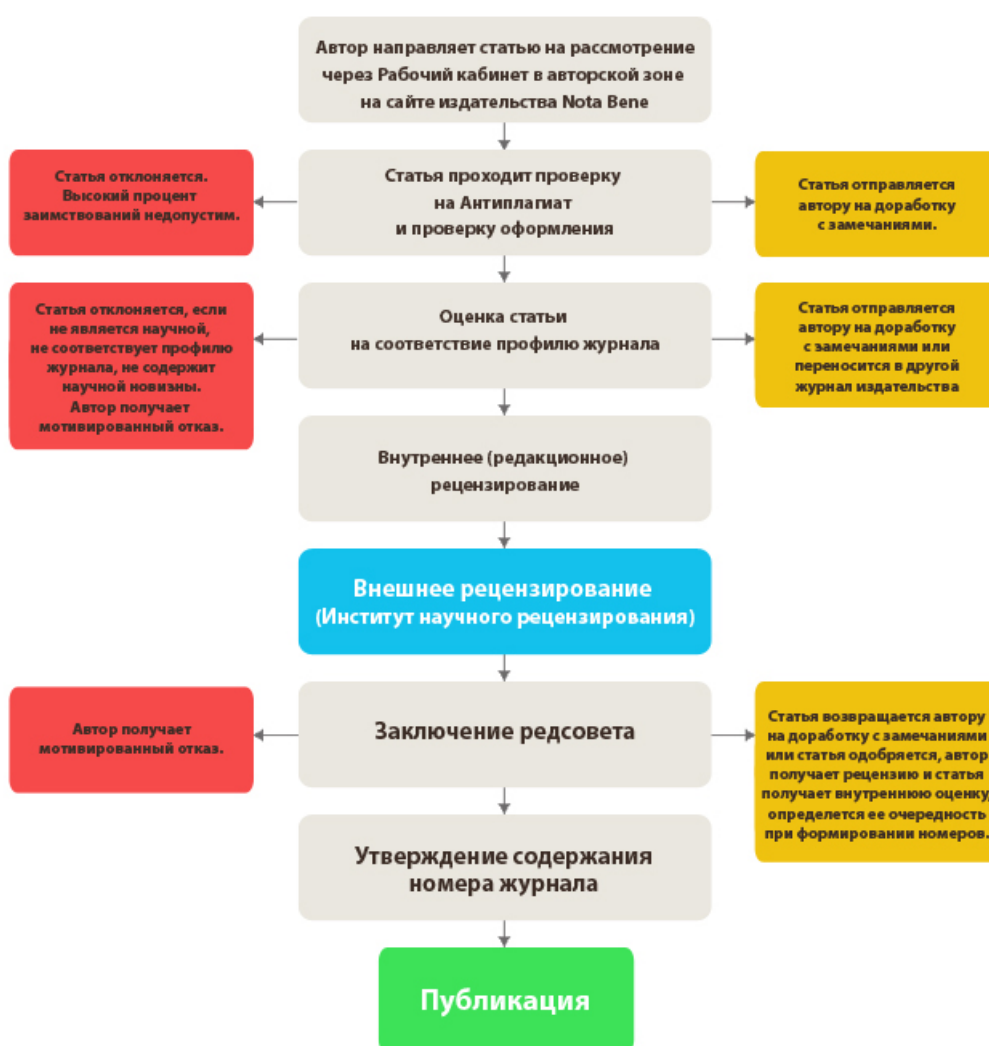


Материалы журналов включены:

- в систему Российского индекса научного цитирования;
- отображаются в крупнейшей международной базе данных периодических изданий Ulrich's Periodicals Directory, что гарантирует значительное увеличение цитируемости;
- Всем статьям присваивается уникальный идентификационный номер Международного регистрационного агентства DOI Registration Agency. Мы формируем и присваиваем всем статьям и книгам, в печатном, либо электронном виде, оригинальный цифровой код. Префикс и суффикс, будучи прописанными вместе, образуют определяемый, цитируемый и индексируемый в поисковых системах, цифровой идентификатор объекта — digital object identifier (DOI).

[Отправить статью в редакцию](#)

### Этапы рассмотрения научной статьи в издательстве NOTA BENE.



## Содержание

|   |     |
|---|-----|
| Алпатов А.Н., Богатырева А.А. Формат хранения данных для аналитических систем на основе метаданных и графов зависимостей между CSV и JSON   | 1   |
| Шарипов Р.Р., Юсупов Б.З., Мартынов А.М., Зарипова Р.С. Разработка методики эффективного размещения охранно-пожарной системы  | 15  |
| Зубов Д.В., Лебедев Д.А. Диагностика отказов технологического оборудования химических производств с помощью искусственного интеллекта   | 30  |
| Филиппова К.А., Аюшеев Т.В., Дамдинова Т.Ц., Цыдыпов Ц.Ц. Исследование напряженно-деформированного состояния композитной лопасти в ANSYS WorkBench                                      | 41  |
| Малахов С.В., Якупов Д.О. Исследование стохастических моделей генерации пакетов в компьютерных сетях  | 53  |
| Вальков В.А., Столяров Е.П., Корчагин А.А., Ермишин М.В., Якупов Д.О. Сравнение методов оптимизации скорости чтения/записи накопителей  | 73  |
| Мамадаев И.М., Минитаева А.М. Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS | 86  |
| Демидов Н.А., Выгоняйло К.В., Маняев А.А., Ефимов Д.А., Баженов А.Э. Сравнительный анализ Wine и PortProton: Перекрёстные платформы в контексте эмуляции Windows приложений             | 99  |
| Гусенко М.Ю. Создание обобщенной нотации программного интерфейса процессоров x86 для автоматизированного построения дизассемблера   | 119 |
| Англоязычные метаданные   | 147 |

## Contents

|   |     |
|---|-----|
| Alpatov A.N., Bogatireva A.A. Data storage format for analytical systems based on metadata and dependency graphs between CSV and JSON   | 1   |
| Sharipov R.R., Yusupov B.Z., Martynov A.M., Zaripova R.S. Developing the Methodology for the Effective Placement of Security and Fire Alarm Systems                                   | 15  |
| Zubov D.V., Lebedev D.A. Diagnostics of failures of technological equipment of chemical industries using artificial intelligence  | 30  |
| Filippova K.A., Ayusheev T.V., Damdinova T.T., Tsidipov T.T. Investigation of the stress-strain state of a composite blade in ANSYS WorkBench   | 41  |
| Malakhov S.V., Yakupov D.O. Investigation of stochastic models of packet generation in computer networks  | 53  |
| Val'kov V.A., Stolyarov E.P., Korchagin A.A., Ermishin M.V., Yakupov D.O. Comparison of methods for optimizing the speed of reading/writing drives                                    | 73  |
| Mamadaev I.M., Minitaeva A.M. Performance optimization of machine learning-based image recognition algorithms for mobile devices based on the iOS operating system                    | 86  |
| Demidov N.A., Vygonyailo K.V., Manyayev A.A., Efimov D.A., Bazhenov A.E. Comparative analysis of Wine and PortProton: Cross platforms in the context of Windows application emulation | 99  |
| Gusenko M.Y. Creating a common notation of the x86 processor software interface for automated disassembler construction   | 119 |
| Metadata in english   | 147 |

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Алпатов А.Н., Богатырева А.А. Формат хранения данных для аналитических систем на основе метаданных и графов зависимостей между CSV и JSON // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70229 EDN: TVEPRE URL: [https://nbpublish.com/library\\_read\\_article.php?id=70229](https://nbpublish.com/library_read_article.php?id=70229)

## Формат хранения данных для аналитических систем на основе метаданных и графов зависимостей между CSV и JSON

**Алпатов Алексей Николаевич**

ORCID: 0000-0001-8624-1662

доцент, кафедра ИиППО, МИРЭА- Российский технологический университет

119454, Россия, г. Москва, ул. Проспекте Вернадского, 78

✉ [aleksej01-91@mail.ru](mailto:aleksej01-91@mail.ru)



**Богатырева Анна Алексеевна**

студент, кафедра инструментального и прикладного программного обеспечения, МИРЭА-Российский технологический университет

111033, Россия, г. Москва, пр-д Танковый, 4А, кв. 24

✉ [pecherni@gmail.com](mailto:pecherni@gmail.com)



---

[Статья из рубрики "Математическое и программное обеспечение новых информационных технологий"](#)

### DOI:

10.7256/2454-0714.2024.2.70229

### EDN:

TVEPRE

### Дата направления статьи в редакцию:

25-03-2024

### Дата публикации:

01-04-2024

**Аннотация:** В современном информационном обществе объемы данных постоянно растут, и эффективная их обработка становится ключевой для предприятий. Передача и хранение этих данных также играет критическую роль. Большие данные, которые

используются в системах аналитики, чаще всего передаются в одном из двух популярных форматов: CSV для структурированных данных и JSON для неструктурированных данных. Однако существующие форматы файлов могут оказаться неэффективными или недостаточно гибкими для определенных задач анализа данных. Например, они могут не поддерживать сложные структуры данных или не предоставлять достаточного контроля над метаданными. Или же аналитические задачи могут требовать дополнительной информации о данных, такой как метаданные, схема данных и т.д. Исходя из вышеназванного, предметом данного исследования является формат данных, основанный на совместном использовании CSV и JSON для обработки и анализа больших объемов информации. Предлагается вариант совместного использования обозначенных типов данных для реализации нового формата данных. Для этого введены обозначения для структуры данных, включающей CSV-файлы, JSON-файлы, метаданные и граф зависимостей. Описаны различные типы функций, такие как агрегирующие, преобразующие, фильтрующие и т.д. Приведены примеры применения этих функций к данным. Предложенный подход представляет собой методику, которая может значительно облегчить процессы анализа и обработки информации. В её основе лежит формализованный подход, который позволяет установить четкие правила и процедуры для работы с данными, что способствует их более эффективной обработке. Другим аспектом предложенного подхода является определение критерия выбора наиболее подходящего формата хранения данных. Этот критерий основан на математических принципах теории информации и энтропии. Введение критерия выбора формата данных на основе энтропии позволяет оценить информационную содержательность и компактность данных. Этот подход основывается на расчете энтропии для выбранных форматов и весовых коэффициентов, отражающих важность каждого значения данных. Путем сравнения энтропий можно определить требуемый формат передачи данных. Такой подход учитывает не только компактность данных, но и контекст их использования, а также возможность включения дополнительной метаинформации в сами файлы и поддержку данных, готовых к анализу.

### **Ключевые слова:**

Форматы хранения данных, JSON, CSV, Аналитически готовые данные, Метаданные, Обработка данных, Анализ данных, Интеграция форматов данных, Apache Parquet, Большие данные

### **Введение**

Задача интеграции данных в различных научных областях является чрезвычайно важной. Развитие такого направления, как большие данные, привело к появлению большого числа разрозненных инструментов, используемых при проведении различных научных исследований, и написанных с использованием разных стеков технологий, а самое главное, форматы используемых данных разнятся от инструмента к инструменту, что приводит к необходимости переформатирования данных, что для больших данных может быть затруднительно.

Для решения данной проблемы можно использовать разнообразные программные интерфейсы (API-Application programming interface) [\[1\]](#). Данный подход имеет ряд несомненных преимуществ, но одним из существенных недостатков является высокая нагрузка на сериализацию и десериализацию данных, которая может стать серьезным узким местом для приложений. Кроме того, отсутствует стандартизация представления

данных, что приводит к тому, что разработчики выполняют настраиваемые интеграции с использованием существующих или настраиваемых систем. Для решения обозначенных выше проблем можно применить альтернативный метод обмена данными: обмен данными с использованием общих форматов файлов, оптимизированных для высокой производительности в аналитических системах. Фактически, способ преобразования формата данных является довольно распространённым способом интеграции данных [2]. Наиболее часто данные, без использования узкоспециализированных платформ, накапливаются в форматах CSV, JSON, а в случае таких систем, как Hadoop, может использоваться формат, например, avro.

CSV (Comma-Separated Values) – текстовый формат, предназначенный для представления табличных данных [3]. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми. В первой строке допускается указать наименования столбцов данных, но это не является обязательным требованием. Данный формат не предполагает типизацию данных, все значения данных считаются строками. CSV является наиболее часто используемым форматом для хранения структурированных данных, полученных из реляционной базы данных.

JSON (JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript, при этом формат независим от JS и может использоваться в любом языке программирования [4]. Формат предназначен для неструктурированных данных, обрел популярность в связи с распространением REST и NoSQL-подхода в хранении данных.

Формат Avro – это компактный, быстрый и сериализуемый формат данных, разработанный Apache Software Foundation [5]. Он обеспечивает схему данных в JSON формате, что делает его самоописывающим. Avro поддерживает различные типы данных, сжатие данных и является эффективным для передачи данных между различными системами. Он широко используется в Big Data и распределенных системах для обмена данными и хранения данных.

Основной сложностью использования таких форматов для хранения и представления больших данных является проблема гарантированной доступности метаданных для описания наборов данных с целью их использования в дальнейшем. Метаданные значительно упрощают процесс интеграции разрозненных данных, предоставляя пользователям дополнительную информацию о способах сбора или генерации набора данных, что позволяет оценить качество набора данных. Также в метаданных может заключаться информация об используемых методах при проведении исследований, которые привели к появлению данного датасета. К сожалению, представление метаданных не всегда является общепринятой практикой, что приводит к тому, что метаданные не поставляются или предоставляются в неполном объеме, что усложняет дальнейшее использование набора данных [6]. В ходе проведенного анализа, было также выявлено, что рассмотренные форматы также имеют не очень высокие показатели [7], но их преимуществом является полная платформенезависимость. Так, они могут не поддерживать сложные структуры данных или не предоставлять достаточного контроля над метаданными, или являются неоптимальными для обработки больших объемов данных из-за своей структуры или ограничений в производительности при чтении/записи.

Таким образом, разработка и использование форматов файлов больших данных, ориентированных на оптимизацию хранения больших данных, является важной и актуальной задачей.



## Разработка формата

Определим частный случай данных, при котором ни CSV, ни JSON не будут являться предпочтительным форматом передачи данных. Этот случай представлен ниже, на рисунке 1, как пример фрагмента диаграммы классов.

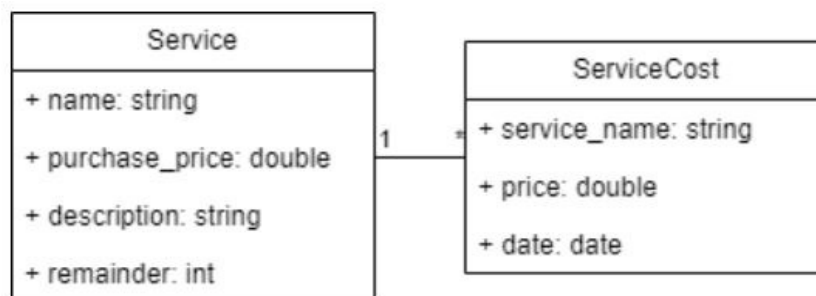


Рисунок 1 – Фрагмент диаграммы классов

На диаграмме представлены два класса: Service (список товаров / услуг) и ServiceCost (история цены товара). Service содержит наименование товара, его закупочную цену, описание характеристик и остаток товаров. ServiceCost содержит наименование товара, его стоимость и дату назначения этой стоимости.

Для Service оптимальным форматом передачи данных является CSV-таблица, а ServiceCost удобнее передавать в JSON, так как если добавить эти данные в тот же файл CSV, все данные товара будет дублироваться для каждой его цены. На данный момент предлагается передавать данные в виде двух заархивированных файлов. В листингах 1 и листинге 2 представлена структура этих файлов.

Листинг 1 – Передача Service в файле service.csv

```

sofa_low,10000.00,description,8
armchair_rocking,8000.00,description,14
bed_double,18000.00,description,6
chair_computer, 12000.00,description,2
bed_children,16000.00,description,5
  
```

Листинг 2 – Передача ServiceCost в файле service\_cost.json

```

{
  "sofa_low": [
    {
      "price": "21000.00",
      "date": "2021.01.16"
    },
    {
      "price": "19000.00",
  
```

```

    "date": "2021.02.21"
  }
],
  "armchair_rocking": [
    {
      "price": "15000.00",
      "date": "2021.01.16"
    }
  ]
}

```

Данный формат все еще будет обладать лишней избыточностью, так как присутствует много лишних символов, но также и обладать недостаточностью информации, так как отсутствует дополнительная информация о типе данных в файлах.

Введем критерий выбора эффективного способа передачи данных для классов Service и ServiceCost, воспользовавшись для этого теорией информации и энтропии. Энтропия представляет собой меру информационной неопределенности данных [8]. Она вычисляется путем суммирования вероятностей появления каждого значения данных и их логарифмов по основанию 2 (формула Шеннона) [9]. Пусть  $H_{CSV}$  и  $H_{JSON}$  будут энтропиями данных в файлах service.csv и service\_cost.json. Введём дополнительные весовые коэффициенты для каждой вероятности появления значения данных. Пусть  $w_i$  и  $v_i$  будут весовыми коэффициентами для вероятностей появления значений данных в CSV и JSON. Эти коэффициенты могут отражать важность каждого значения или контекст, в котором оно используется. Тогда можно использовать энтропию для оценки информационной содержательности и в качестве критерия выбора форматов данных, как показано в (1) и (2).

$$H_{CSV} = - \sum_{i=1}^n w_i \cdot p_i \log_2 p_i, \quad (1)$$

где  $p_i$  - вероятности появления данных в записях CSV,

$n$  - количество различных значений в CSV файле.

$$H_{JSON} = - \sum_{i=1}^m v_i \cdot q_i \log_2 q_i \quad (2)$$

где  $q_i$  - вероятности появления данных в записях JSON,

$m$  - количество различных значений в CSV файле.

Сравним энтропии данных, на примере CSV и JSON форматов, для определения оптимального формата передачи данных. Меньшая энтропия может указывать на более компактное представление данных. Иными словами, в случае, если  $H_{CSV} < H_{JSON}$ , то CSV может быть более оптимальным форматом для передачи данных. В противном случае, если  $H_{CSV} > H_{JSON}$ , то JSON может быть предпочтительнее. Или для определения предпочтительного из двух форматов можно использовать немного иной подход

учитывающий усложненную энтропию и весовые коэффициенты, как показано в (3).

$$K = (\alpha \cdot H_{CSV} + \beta \cdot H_{JSON}), \quad (3)$$

где  $\alpha, \beta$  - коэффициенты, отражающие важность энтропии в каждом формате.

Если  $K < 0$ , то предпочтительным форматом будет CSV. Если же  $K > 0$ , то предпочтительным форматом будет JSON. В противном случае, при  $K = 0$ , форматы хранения данных будут одинаково предпочтительными.

Такой подход позволяет учитывать не только компактность данных, но и возможность включения ключевой информации в самих файлах. Для задачи разработки нового формата такой ключевой информацией может являться дополнительная метainформация, или, например, наличие шума в данных. В случае, если данные содержат много шума или случайных значений, высокая энтропия может указывать на неопределенность и сложность анализа этих данных. Предложенный критерий оценки учитывает контекст использования данных, их ценность, степень сложности и возможных последствий ошибок в анализе данных из-за высокой энтропии.

Исходя из вышеобозначенного можно также сформулировать базовые требования к формату данных для области BigDate. К ним можно отнести:

1. Стандартизация и распространённость. Стандартизация означает установление общих и унифицированных стандартов для представления и обработки данных. Например, к стандартизированным форматам данных можно отнести XML (eXtensible Markup Language), который используется для структурирования и обмена данными в различных областях, включая веб-службы. В контексте форматов хранения больших данных, стандартизация помогает создать общепринятые и унифицированные способы представления информации. Это облегчает обмен данными между различными системами и приложениями, упрощает разработку программного обеспечения и обеспечивает совместимость данных между различными инструментами. Распространенность же формата данных связана с тем, насколько широко данный формат используется в индустрии и среди различных приложений. Чем более распространен формат данных, тем более вероятно, что множество инструментов и систем его поддерживают, что очень важно для области анализа больших данных, где имеется развитая экосистема для работы с этим форматом, включая языки программирования и инструменты анализа данных и системы хранения.
2. Компактность. Формат должен занимать минимальное количество места для хранения данных. Это важно для эффективного использования хранилища и передачи данных по сети, особенно в случае больших объемов информации. Компактные форматы могут экономить ресурсы хранения, ускорить передачу данных и уменьшить нагрузку на сеть.
3. Возможность сжатия. Поддержка механизмов сжатия данных может значительно уменьшить объем хранимых данных и ускорить их передачу по сети.
4. Хранение метаданных. Хранение метаданных в формате данных играет важную роль в обеспечении полной и правильной интерпретации информации, сохраненной в этом формате. Метаданные представляют собой данные об данных и содержат информацию о структуре, типах, формате, и других характеристиках данных.
5. Поддержка данных, готовых к анализу. Analysis-Ready Data (ARD) - это концепция в области анализа больших данных, которая подразумевает предварительную обработку и

подготовку данных перед анализом. Она призвана упростить и ускорить процесс анализа данных, обеспечивая готовые к использованию наборы данных. В настоящее время данная концепция находит широкое развитие в области анализа спутниковых изображений [\[10\]](#).

Предложенный формат хранения данных будет базироваться на синтезе двух распространённых форматов хранения данных, таких как CSV и JSON. Такое решение имеет ряд преимуществ, в силу того, что CSV позволяет хранить данные в табличной структуре с явно определенными столбцами, в то время как JSON обладает более гибкой структурой. Объединение их может обеспечить удобное хранение табличных данных с сохранением иерархии и вложенных объектов. Также использование JSON для хранения данных предоставляет типизацию данных, что облегчает восприятие и работу с данными, а дополнительный метафайл может содержать информацию о типах данных для CSV-части, что обеспечивает полную типизацию.

### **Модификация формата хранения данных с метаданными**

В CSV-файле есть возможность добавить названия столбцов и, так как CSV не может хранить информацию о типе данных, можно добавить отдельным meta-файлом информацию о типах. Модифицированный CSV-файл и метафайл с информацией о типах представлены в листингах 3 и 4.

Листинг 3 – Модификация service.csv.

```
name,purchase_price,description,remainder  
  
sofa_low,10000.00,description,8  
  
armchair_rocking,8000.00,description,14  
  
bed_double,18000.00,description,6  
  
chair_computer, 12000.00,description,2  
  
bed_children,16000.00,description,5
```

Листинг 4 – Мета-файл service.meta.

```
{  
  
  "name": "string",  
  
  "purchase_price": "double",  
  
  "description": "string",  
  
  "remainder": "int"  
  
}
```

Формат JSON при десериализации в различных языках программирования предусматривает типизацию данных, но для унификации формата типы данных будут храниться в отдельном метафайле (листинг 5).

Листинг 5 – Метафайл service\_cost.meta.

```
{
```

```

    "price": "double",

    "date": "date"

}

```

Таким образом, разработанный формат представляет собой архив, состоящий из четырех вышеописанных файлов.

Формат может быть расширен дополнительными данными в формате CSV или JSON с соответствующими метафайлами при необходимости.

В качестве названия формата предлагается «PandasKeepFormat», расширение формата соответственно имеет вид PDKF.

### **Математическое описание формата и применимых функций над структурой данных**

Далее рассмотрим математическое описание для предложенного в данной работе формата данных, который объединяет CSV и JSON, определив, таким образом множество допустимых операций над данными.

Пусть:

- $C$  - множество CSV-файлов, представляющих собой набор строк и столбцов.
- $J$  - множество JSON-файлов, содержащих структурированные данные в формате JSON.
- $M$  - множество метаданных, описывающих информацию о структуре данных в CSV или JSON, такую как типы данных, названия столбцов, и другие характеристики.

Тогда, новый формат данных  $D_{pdkf}$  можно определить следующим образом, как показано в формуле 4.

$$D_{pdkf} = (C \cup J) \times M \quad (4).$$

В вышеобозначенном равенстве  $C \cup J$  является множеством, объединяющее данные из CSV и JSON. Элементы этого множества могут быть как табличными (CSV), так и иерархическими (JSON) данными.  $A^{(C \cup J) \times M}$  является множеством пар, где первый элемент представляет собой данные из  $C \cup J$ , а второй элемент - соответствующие метаданные  $M$ . Таким образом, каждая запись данных из  $C \cup J$  сопровождается своими метаданными.

Данный подход позволит объединить данные и метаданные, однако, он не учитывает наличие более сложных зависимостей в данных. Например, одни данные могут зависеть от других, но для некоторых задач может быть важным ещё и понимание контекста и последовательности использования этих данных. Или, в некоторых аналитических сценариях может быть необходимость в проведении сложных операций, которые включают в себя связанные элементы данных. Например, вычисления, которые требуют комбинации данных из нескольких источников, могут использовать информацию о связях. Особенностью предложенной выше формата хранения данных является то, что он может быть относительно легко расширен, за счёт возможности добавления дополнительных данных, о чем было указано выше. Например, функции, определенные при реализации данного формата, добавляются в словарь функций и могут быть

применены к данным, хранящимся в структуре.

Для расширения данной модели в работе предлагается внедрить элементы из теории графов и теории категорий. Для этого дополнительно введём в модель ориентированный граф  $G$ , где вершины  $V$  представляют элементы из  $C \cup J$  и ребра  $E$  представляют связи или зависимости между этими элементами, иными словами  $G=(V,E)$ .

Для работы с данными, определенными предложенным форматом, необходимо дополнительно определить набор применимых функций над данными. Здесь, в данной работе, под применимыми функциями, в контексте структур данных понимаются функции или операции, которые могут быть корректно применены или выполнены на данных предложенной структуры. То есть, это функции, которые соответствуют типу или структуре данных и могут быть использованы для обработки или изменения этих данных. Соответственно, пусть имеется множество функций  $F$ , которые могут быть применены к данным из  $C \cup J$  с учетом их структуры и метаданных. Предположим, что каждая функция имеет определенные характеристики, такие как агрегация, фильтрация, преобразование, комбинация и т.д. Определим некоторые из этих функций.

Пусть  $D$  - структура данных,  $C$  - множество CSV-файлов,  $J$  - множество JSON-файлов,  $M$  - множество метаданных, а  $f$  - функция из множества применимых функций  $F = \{f_1, f_2, \dots\}$ , где  $f_i$  - функции, которые могут выполнять различные операции, такие как фильтрация, сортировка, агрегация и т.д.

Обозначим элемент данных в  $D$  как  $d_{ij}$ , где  $i$  и  $j$  - координаты элемента в координатной сетке. Иными словами  $D = \{d_{ij}\}$ ,  $d_{ij} \in \mathbb{R}^{n \times m}$ , где  $n$  - количество строк,  $m$  - количество столбцов.

Тогда, с учетом вышеизложенного, **агрегирующая функция** может быть выражена  $f_{\text{агрегация}}(D, M) \rightarrow \text{Result\_Aggregation}$ .

Преобразующая функция  $f_{\text{преобразование}}$  преобразует данные из структуры  $D$  в новую структуру  $f_{\text{преобразование}}(D, M) \rightarrow D'$ .

**Функция извлечения** данных столбца извлекает определенный столбец из CSV или JSON данных (при соответствующей организации данных). Пусть  $D_i$  представляет собой элемент из  $D$ , а  $M_i$  - соответствующий элемент из  $M$ . Тогда функция  $f_{\text{извлечение}}(D_i, M_i, \text{columnName}) \rightarrow \text{ColumnData}$ . Например, если  $D_i$  - CSV файл,  $M_i$  - метаданные, и  $\text{columnName} = "Age"$ , то  $f_{\text{извлечение}}$  вернет столбец возрастов.

**Функция фильтрации** по значению оставляет только те строки данных, где определенный столбец удовлетворяет заданному условию. Аналогично функции извлечения данных, функция  $f_{\text{фильтрация}}$  может быть выражена через  $f_{\text{фильтрация}}(D_i, M_i, \text{columnName}, \text{condition}) \rightarrow \text{FilteredData}$ , где  $\text{condition}$  означает условие, которое определяет, какие строки данных следует оставить после применения функции фильтрации. Это условие проверяется для определенного столбца в данных, и только те строки, которые удовлетворяют этому условию, остаются в результирующем наборе данных. Например, если  $D_i$  - JSON файл,  $M_i$  - метаданные,  $\text{columnName} = "Price"$  и  $\text{condition} = "> \$100"$ , то  $f_{\text{фильтрация}}$  оставит только записи с ценой выше \$100.

С учётом вышесказанного, окончательно, новый формат данных  $D$  можно определить как показано в формуле 5.



$$D = (C \cup J) \times (G, F) \quad (5)$$

### Реализация обработчика формата

В данном разделе представлена реализация обработчика формата PDKF на языке программирования Python, который преобразует данные из файла в список датафреймов Pandas. Фрагмент исходного кода обработчика представлен в листинге 6.

Листинг 6 – Реализация обработчика формата.

```
import pandas

import json

from zipfile import ZipFile

from io import BytesIO

from io import StringIO

def read_file(file_name, file_content):

    if file_name[-3:] == 'csv':

        data = pandas.read_csv(BytesIO(file_content))

        return data

    elif file_name[-4:] == 'json':

        json_list = json.loads(file_content.decode())

        data = pandas.DataFrame()

        for item in json_list:

            if len(data) == 0:

                data = pandas.json_normalize(list(item.values()))[10]

            else:

                data = pandas.concat([data, pandas.json_normalize(list(item.values()))
                [0]], ignore_index = True)

        return data

    else:

        raise Exception(f'Wrong file format: {file_name}')

def read_pdbc(file):

    if file[-4:] != 'pdkf':

        raise Exception(f'Wrong file format')

    dataframe_dict = {}

    file_zip = ZipFile(file, 'r', allowZip64=True)
```

```

try:

for file_name in file_zip.namelist():

if file_name[-4:] != 'meta':

file_content = file_zip.read(file_name)

dataframe_dict[file_name[:file_name.find('.')]]= read_file(file_name,
file_content)

for file_name in file_zip.namelist():

if file_name[-4:] == 'meta':

file_content = file_zip.read(file_name)

data = json.loads(file_content.decode())

for column, type in data.items():

dataframe_dict[file_name[:-5]][column] = dataframe_dict[file_name[:-5]]
[column].astype(type)

finally:

file_zip.close()

return dataframe_dict

```

Для реализации обработчика необходимы следующие зависимости:

- Pandas – популярная библиотека, необходимая для аналитики больших данных, именно в ней содержится тип данных DataFrame.
- Zipfile – встроенная библиотека, необходимая для работы с архивами ZIP.

Представленный выше код позволяет читать файлы в форматах CSV и JSON, а также из ZIP-архива с определенной структурой, и создавать соответствующие pandas DataFrame для дальнейшей обработки данных. Кроме того, он обеспечивает обработку метаданных и преобразование типов данных в DataFrame в соответствии с указанными типами в метаданных. Пример использования вышеописанного модуля и некоторые манипуляции с полученными данными представлены в листинге 7.

Листинг 7 – Пример использования модуля.

```

import pdkf

dataframe_dict = pdkf.read_pdkf('file.pdkf')

for key, dataframe in dataframe_dict.items():

print(key)

print(dataframe.info())

```

В данном случае, выводится информация о каждом полученном из файла датафрейме. В

дальнейшем можно производить с данными любые манипуляции, доступные в модуле Pandas, такие как аналитика, прогнозирование на основе этих данных и т.д.

Разработанный модуль выложен в каталог пакетов Python PYPI [<https://pypi.org/project/pdkf/>] под лицензией MIT, что позволяет установить его консольной командой «`pip install pdkf`». На рисунке 2 показан скриншот размещения разработанного решения в официальной репозитории программного обеспечения для Python - PyPI, аналоге CRAN для языка R.

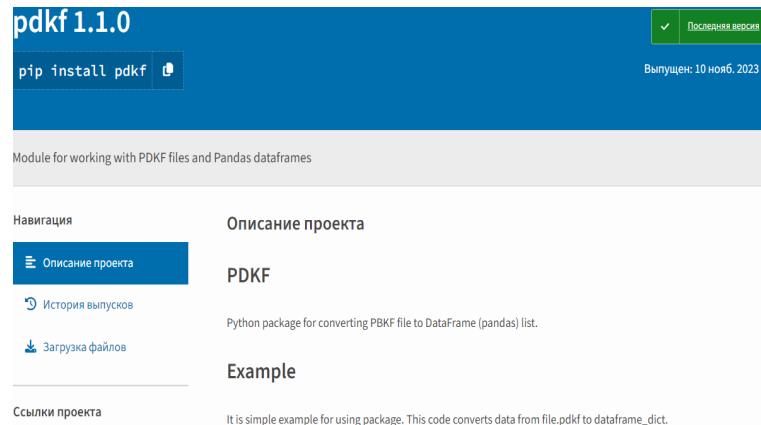


Рисунок 2 – Размещение модуля-обработчика формата pdkf в репозитории PYPI

## Заключение

Предложенное решение предлагает использовать комбинацию форматов CSV и JSON для оптимального хранения и передачи данных. Для удобства восприятия данных и обеспечения полной типизации, предлагается дополнительно использовать метафайлы, которые содержат информацию о типах данных для соответствующих CSV и JSON файлов. Модификация формата CSV включает добавление названий столбцов и хранение информации о типах данных в отдельном метафайле. JSON формат уже обеспечивает типизацию данных при десериализации, но также предлагается хранить информацию о типах в метафайле для унификации формата.

Таким образом, предложенное решение представляет собой архив, состоящий из CSV и JSON файлов, а также соответствующих метафайлов, которые обеспечивают структурированное хранение данных с сохранением типов и метаданных. Формат может быть дополнен дополнительными данными в CSV или JSON формате с соответствующими метафайлами при необходимости.

## Библиография

1. Malcolm R., Morrison C., Grandison T., Thorpe S., Christie K., Wallace A., Green D., Jarrett J., Campbell A. Increasing the accessibility to big data systems via a common services api // IEEE International Conference on Big Data. 2014. Pp. 883–892.
2. Wu T. System of teaching quality analyzing and evaluating based on data warehouse // Computer Engineering and Design. 2009. No. 6(2). Pp. 1545-1547.
3. Vitagliano G. et al. Pollock: A Data Loading Benchmark //Proceedings of the VLDB Endowment. 2023. No. 8(16). Pp. 1870-1882.
4. Xiaojuan L., Yu Z. A data integration tool for the integrated modeling and analysis for east //Fusion Engineering and Design. 2023. No. 195. Pp. 113933. <https://doi.org/10.1016/j.fusengdes.2023.113933>
5. Lemzin A. Streaming Data Processing //Asian Journal of Research in Computer

- Science. 2023. No. 1(15). Pp. 11-21.
6. Hughes LD, Tsueng G, DiGiovanna J, Horvath TD, Rasmussen LV, Savidge TC, Stoeger T, Turkarslan S, Wu Q, Wu C, Su AI, Pache L. Addressing barriers in FAIR data practices for biomedical data //Scientific Data. 2023. No. 1(10). P. 98.  
DOI:https://doi.org/10.1038/s41597-023-01969-8
7. Gohil A., Shroff A., Garg A., Kumar S. A Compendious Research on Big Data File Formats. *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE Press, Madurai, India. 2022. Pp. 905-913. DOI: https://doi.org/10.1109/ICICCS53718.2022.9788141
8. Елсуков П. Ю. Информационная асимметрия и информационная неопределенность //ИТНОУ: Информационные технологии в науке, образовании и управлении. 2017. No. 4 (4). С. 69-76.
9. Bromiley P. A., Thacker N. A., Bouhova-Thacker E. Shannon entropy, Renyi entropy, and information //Statistics and Inf. Series (2004-004). 2004. No. 9. Pp. 2-8.
10. Dwyer, J. L. Roy, D. P., Sauer B., Jenkerson C. B., Zhang H. K., Lymburner L. Analysis ready data: enabling analysis of the Landsat archive //Remote Sensing. 2018. №. 9(10). 1363.

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Статья посвящена актуальной задаче интеграции и оптимизации хранения данных в разнообразных форматах, особенно акцентируя внимание на форматах CSV и JSON. Исследование затрагивает вопросы эффективного представления больших объемов данных, с целью упрощения процесса аналитики и обработки. Авторы предлагают новаторский подход к объединению форматов хранения данных с использованием метаданных и графовых структур для оптимизации и улучшения доступности данных. Исследование включает теоретический анализ, разработку нового формата данных, математическое описание предложенных методов, а также практическую реализацию на языке программирования Python. В эпоху "больших данных", актуальность исследования несомненна. Проблематика интеграции и эффективного хранения данных остается важной задачей, особенно в контексте растущего объема неструктурированных данных и необходимости их анализа. Научная новизна работы заключается в предложении оригинального формата хранения данных "PandasKeepFormat", который синтезирует преимущества CSV и JSON форматов с дополнением метаданных и графов зависимостей. Это позволяет значительно упростить работу с большими объемами данных и обеспечивает новые возможности для аналитики. Статья имеет логичное и последовательное изложение материала, начиная с обоснования актуальности проблемы и заканчивая практической реализацией предложенного решения. Стиль изложения ясный и доступный, материал структурирован, а содержание информативно и насыщено. Выводы статьи подчеркивают значимость разработанного формата для упрощения процессов аналитики и интеграции данных. Работа будет интересна широкому кругу читателей, включая специалистов в области информационных технологий, аналитики данных, а также разработчиков программного обеспечения. Статья представляет значительный вклад в развитие методов работы с большими данными и заслуживает публикации. Было бы полезно рассмотреть возможность дальнейшего развития исследования, например, путем проведения сравнительного

анализа производительности нового формата с существующими решениями на реальных наборах данных. Рекомендую принять к публикации, учитывая актуальность темы, научную новизну и качество представленного материала.

Для дальнейшего развития представленной в статье работы по новому формату хранения данных можно предложить несколько направлений, которые включают проведение экспериментального исследования для сравнения производительности предложенного формата "PandasKeepFormat" с другими существующими форматами данных, такими как Parquet, ORC или Avro. Важно рассмотреть различные аспекты, включая скорость чтения и записи данных, эффективность сжатия данных и потребление ресурсов памяти. Также предлагаю исследовать возможность дополнения формата поддержкой более широкого спектра типов данных, включая сложные структуры данных и пользовательские типы, что может повысить его применимость для различных специализированных областей анализа данных. Одним из важных направлений развития является разработка плагинов или модулей для интеграции с популярными библиотеками и фреймворками для работы с данными, чтобы облегчить использование нового формата в существующих экосистемах. Оптимизация формата и алгоритмов сериализации и десериализации для эффективного использования в распределенных вычислительных системах также является ключевым аспектом, включая возможности эффективного разбиения и параллельной обработки данных. Важно уделить внимание безопасности данных, разрабатывая механизмы шифрования и обеспечения безопасности данных, встроенные в формат, для повышения защиты конфиденциальности информации при хранении и передаче. Улучшение управления метаданными, включая продвинутое управление и версионирование метаданных, обеспечит лучшую совместимость и возможности для эволюции формата данных без потери информации о предыдущих версиях. Создание инструментов для визуализации структуры данных и метаданных, а также для мониторинга производительности работы с данными в новом формате, позволит пользователям более эффективно управлять и анализировать свои данные. Наконец, организация платформы для взаимодействия с пользователями и разработчиками, заинтересованными в использовании и развитии нового формата, а также сбор и анализ обратной связи помогут выявить потребности пользователей и определить направления для дальнейших улучшений. Эти шаги способствуют улучшению эффективности предложенного формата хранения данных, расширению его применимости и популярности в научном и профессиональном сообществе.

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Шарипов Р.Р., Юсупов Б.З., Мартынов А.М., Зарипова Р.С. Разработка методики эффективного размещения охранно-пожарной системы // Программные системы и вычислительные методы. 2024. № 2. DOI:

10.7256/2454-0714.2024.2.41036 EDN: ZNJPJH URL: [https://nbpublish.com/library\\_read\\_article.php?id=41036](https://nbpublish.com/library_read_article.php?id=41036)

## Разработка методики эффективного размещения охранно-пожарной системы

**Шарипов Рифат Рашатович**

ORCID: 0000-0002-4957-8132

кандидат технических наук

доцент кафедры систем информационной безопасности Казанского национального исследовательского технического университета им. А.Н. Туполева-КАИ

420015, Россия, республика Татарстан, г. Казань, ул. Большая Красная, 55

✉ [riphat@mail.ru](mailto:riphat@mail.ru)



**Юсупов Булат Зуфарович**

обучающийся кафедры систем информационной безопасности Казанского национального исследовательского технического университета им. А.Н. Туполева-КАИ (КНИТУ-КАИ)

420015, Россия, республика Татарстан, г. Казань, ул. Большая Красная, 55

✉ [Bulatusupov9@gmail.com](mailto:Bulatusupov9@gmail.com)



**Мартынов Артур Михайлович**

обучающийся кафедры систем информационной безопасности Казанского национального исследовательского технического университета им. А.Н. Туполева-КАИ (КНИТУ-КАИ)

420015, Россия, республика Татарстан, г. Казань, ул. Большая Красная, 55

✉ [Keshka628@gmail.com](mailto:Keshka628@gmail.com)



**Зарипова Римма Солтановна**

ORCID: 0000-0002-3548-1866

кандидат технических наук

доцент кафедры "Цифровые системы и модели" Казанского государственного энергетического университета

420066, Россия, республика Татарстан, г. Казань, ул. Красносельская, 51

✉ [zarim@rambler.ru](mailto:zarim@rambler.ru)



[Статья из рубрики "Модели и методы управления информационной безопасностью"](#)

**DOI:**

10.7256/2454-0714.2024.2.41036

**EDN:**



ZNJPJH

**Дата направления статьи в редакцию:**

18-06-2023

**Дата публикации:**

17-04-2024

**Аннотация:** Статья акцентирует внимание на системах охранно-пожарной сигнализации (ОПС) как средствах обеспечения безопасности объектов, рассматривая их как интегрированные комплексы для оперативного обнаружения потенциальных угроз. Основной фокус сделан на извещателях, включая их классификацию и роль в системе. Рассматриваются различные конфигурации ОПС и способы подключения и обработки сигналов от извещателей, что позволяет оценить, как эти факторы влияют на эффективность системы. Также описывается жизненный цикл ОПС, подчеркивая важность каждого этапа, от проектирования до эксплуатации. Статья предоставляет обзор регламентирующих документов, выделяя важность соблюдения стандартов и требований при реализации систем ОПС. Рекомендовано для специалистов в области безопасности и лиц, заинтересованных в освоении данной тематики. В статье рассматриваются проблемы размещения систем ОПС и их влияние на эффективность системы. Анализируются уязвимости, возникающие из-за нерационального размещения элементов, и представлена методика оптимизации размещения для усиления безопасности. Методика описана пошагово, с учетом входных и выходных процессов на каждом этапе. Авторы проводят практическое тестирование методики в учебной лаборатории с установленной системой ОПС, выявляя ошибки в размещении и формулируя рекомендации по их исправлению. Статья полезна для специалистов в области проектирования и установки ОПС, а также для тех, кто стремится повысить уровень защиты объектов, акцентируя внимание на критической важности правильного размещения компонентов.

**Ключевые слова:**

Системы безопасности, Пожарная сигнализация, Размещение, Методика, Эффективность, Оптимизация, Компоненты, Установка, Защита, Проектирование

**Введение**

Система охранно-пожарной сигнализации занимает центральное место в структуре комплексных мер, направленных на обеспечение физической безопасности объектов. Это неотъемлемый элемент, создающий преграды для возможных злоумышленников и снижающий риски возникновения пожаров, что, в свою очередь, способствует сохранению материальных активов и предотвращает потери среди населения. Пожары могут иметь катастрофические последствия, влекущие за собой значительные материальные потери и, что гораздо хуже, гибель людей. Поэтому крайне важно выработать рациональный подход к обеспечению противопожарной безопасности. Кроме

того, в современном мире, который становится все более и более технологичным, важность информационной безопасности неоспорима. Анализ текущих тенденций показывает, что особое внимание стоит уделять защите инфраструктуры компаний. Это может включать в себя охрану различных элементов, таких как подсистемы, рабочие станции, серверы и т.д. Следует осознавать, что иногда упускается важность физической защиты в контексте информационной безопасности. Например, глобальная сеть интернет, которую мы часто воспринимаем как абстрактное пространство, на самом деле опирается на физические компоненты: коммутационное оборудование, проводные и радиоканальные устройства, устройства обработки данных, хранилища данных и серверы. Все эти элементы имеют конкретное физическое расположение и, следовательно, могут подвергаться угрозам от злоумышленников. В этом контексте, системы охранно-пожарной сигнализации выступают как первая линия защиты, предупреждая несанкционированный доступ к критически важным компонентам инфраструктуры и предотвращая возможные пожары. Таким образом, важность систем ОПС не может быть переоценена. Они играют решающую роль в обеспечении физической и информационной безопасности, содействуя защите жизни и благосостояния людей, а также сохранению ценных ресурсов и активов организаций. При этом, важно подчеркнуть, что эффективность системы ОПС напрямую зависит от качества ее проектирования, монтажа и регулярного обслуживания. Ответственный подход к выбору оборудования, учет особенностей конкретного объекта и строгое следование нормативным требованиям – залог высокой надежности и функциональности системы. Кроме того, необходимо уделять внимание интеграции системы ОПС с другими системами безопасности, такими как системы видеонаблюдения, контроля доступа и антитеррористической защиты. Такая интеграция позволит обеспечить комплексный подход к безопасности, что значительно усилит ее эффективность. В современном мире, где технологии развиваются стремительными темпами, также необходимо учитывать и адаптироваться к новым угрозам и вызовам. Это включает в себя мониторинг и применение новейших технологических решений в области безопасности, обучение персонала и создание планов реагирования на чрезвычайные ситуации.

Следует отметить, что ОПС представляют собой сложные комплексы технических средств, задачей которых является обеспечение безопасности объектов путем мониторинга возможных угроз и оповещения о них. В качестве основных компонентов типовой системы ОПС можно выделить датчики или извещатели, которые, как правило, представляют собой сенсоры, способные регистрировать различные виды событий, такие как движение, открытие дверей, повышение температуры или дым. Эти датчики коммутируются к центральному устройству, называемому приёмно-контрольным прибором (ПКП) <sup>[1]</sup>. Приёмно-контрольный прибор выполняет ключевую роль в системе, поскольку он обрабатывает сигналы, получаемые от датчиков, и в случае обнаружения возможной угрозы активирует соответствующие меры реагирования. К таким мерам могут относиться, например, включение светового и звукового оповещения для предупреждения персонала объекта и/или вызова службы безопасности. К ПКП также могут быть подключены считыватели идентификаторов, что позволяет управлять системой с помощью специальных карт или брелоков. Это особенно удобно для снятия и постановки объекта на охрану. В некоторых случаях, в системе ОПС может быть предусмотрен отдельный пульт управления, который обеспечивает расширенные возможности управления и мониторинга системы. Это может быть полезно на крупных объектах, где необходимо обеспечить централизованный контроль над системой безопасности. Кроме того, в систему могут быть интегрированы дополнительные компоненты, такие как пульта ручного запуска, которые позволяют персоналу объекта

немедленно активировать систему оповещения в чрезвычайных ситуациях, или модули расширения, предоставляющие дополнительные интерфейсы и функции для интеграции с другими системами. В целом, система охранно-пожарной сигнализации представляет собой многофункциональный инструмент, способный адаптироваться к различным условиям и требованиям безопасности. Модульная структура системы позволяет настраивать её в соответствии с особенностями конкретного объекта, учитывая его размеры, количество помещений, особенности инфраструктуры и потенциальные риски. Важным аспектом является также возможность интеграции системы ОПС с другими системами безопасности и управления зданием. Так, современные ОПС могут совмещаться с системами видеонаблюдения, контроля доступа, а также с автоматизированными системами управления зданием для оптимизации энергопотребления и обеспечения комфорта. Качественное проектирование, монтаж и настройка системы охранно-пожарной сигнализации – это ключевые факторы, обеспечивающие её эффективность. При этом следует уделить внимание не только выбору подходящих компонентов, но и их правильной установке, а также обучению персонала основам работы с системой. Регулярное техническое обслуживание и проверка работоспособности всех элементов системы также являются важными составляющими обеспечения надежной работы ОПС.

Существует множество видов охранно-пожарных извещателей и дана их классификация по нескольким параметрам [\[2\]](#). Однако следует учитывать некоторые особенности при их эксплуатации в зависимости от множества факторов, которые существенно зависят от объекта охраны. Например, производственные объекты с разными факторами воздействия среды на извещатели. Среди них: состояние влажности воздуха, вызывающее коррозию, температурные и вибрационные воздействия, высокий электромагнитный фон, вызывающие ложные срабатывания или же выхода из строя устройств охраны в пределах срока эксплуатации. Для коммерческих офисных помещений и образовательных учреждений обычно применяются несколько видов стандартных охранно-пожарных извещателей. Среди них [\[3,4\]](#):

- оптико-электронные извещатели;
- акустические (пассивные) извещатели;
- магнитно-контактные извещатели;
- противопожарные извещатели.

Распространенные типы противопожарных извещателей – это дымовые и тепловые. В больших помещениях с высокими потолками применяются противопожарные линейные системы, состоящие из групп сонаправленных приемников и передатчиков.

Известна так же и классификация извещателей по методам подключения и обработки сигналов с извещателей приборами контроля. Тут выделяют следующие типы извещателей [\[5,6\]](#):

- аналоговые;
- адресные;
- адресно-аналоговые.

Кроме того, в последние годы активно используются радио-канальные извещатели [\[7,8\]](#),

позволяющие упростить развёртывание систем ОПС, поскольку не требуется прокладка проводных шлейфов для подключения извещателей к ПКП.

Существуют несколько этапов жизненного цикла работы систем ОПС, начиная от постановки задачи охраны объекта, контроля доступа на объект, разграничения зон защиты и выставления требования заказчику [\[9,10\]](#). На последнем этапе составляется техническое задание, в котором обычно отражаются следующие моменты [\[11\]](#):

- указываются цели проведения работ,
- указывается перечень нормативных и регламентирующих документов,
- даются краткое описание объектов,
- выставляются требования к исполнителю, к оборудованию и выполнению работ и к порядку выполнения работ.

После согласования технического задания, компания-исполнитель переходит к исследованию объекта защиты, изучения её структуры, конструктивных особенностей и разрабатывает рабочий проект согласно требованиям нормативной документации.

### **Проблемы при размещении компонентов ОПС**

Рассмотрим основные нормативные документы, регламентирующие проектирование и внедрение системы ОПС на объектах защиты.

В основном известны следующие основные нормативные документы [\[12,13,14\]](#):

- Постановление Правительства РФ от 16.02.2008 N 87 "О составе разделов проектной документации и требованиях к их содержанию" [\[15\]](#). Это постановление устанавливает состав разделов проектной документации и требования к содержанию этих разделов при подготовке проектной документации на различные виды объектов капитального строительства и при подготовке проектной документации в отношении отдельных этапов строительства, реконструкции и капитального ремонта объектов капитального строительства;
- ГОСТ Р 50776-95 (МЭК 60839-1- 4:1989) «Системы тревожной сигнализации. Часть 1. Общие требования». Введен в действие постановлением Госстандарта РФ от 22 мая 1995 г. N 256 [\[16\]](#). Данный национальный стандарт устанавливает требования при проектировании, монтаже, наладке, испытаниях, эксплуатации и техническом обслуживании систем тревожной сигнализации, охранной, охранно-пожарной сигнализации, применяемых для обеспечения безопасности людей и имущества;
- СНиП 3.05.05-84, СНиП 3.05.06-85 и СНиП 3.05.07-85. В этих строительных нормах и правилах указаны обязательные требования, для пусконаладочных работ, которые должны выполняться монтажно-наладочной организацией;
- СП 68.13330.2017 "СНиП 3.01.04-87 Приемка в эксплуатацию законченных строительством объектов. Основные положения" [\[17\]](#). В этом своде правил указаны требования и правила приёма в эксплуатацию. После приёма в эксплуатацию компания-исполнитель обеспечивает гарантийное обслуживание и сопровождение системы.

Несмотря на требования нормативной базы на всех этапах жизненного цикла системы ОПС появляется очень много некорректных решений, отраженные как в проектах, так и размещенных на реальных объектах системах ОПС. Одна из распространённых проблем — это неверное размещение извещателей на объектах защиты, что приводит к некорректной логической работе системы ОПС. В последствии это приводит к появлению

инцидентов при котором злоумышленник может проникнуть на критические зоны незамеченным, а также может деактивировать датчики. Иногда бывают другие случаи, когда количество извещателей на объекте могут быть установлены в избыточном количестве и установлены они с нарушением требований по размещению, например, два извещателя могут быть расположены близко друг к другу, что может привести к взаимному влиянию и перекрытию зон обнаружения.

### **Разработка методики**

С целью обеспечения максимальной эффективности систем охранно-пожарной сигнализации на этапе проектирования, а также для коррекции уже существующих установок на объектах защиты, была разработана специальная методика, направленная на оптимизацию размещения элементов ОПС (см. рис. 1). Эта методика основана на комплексном подходе, включающем в себя анализ и учет множества факторов. Основным акцентом в рамках разработанной методики является строгое следование требованиям нормативных документов, регулирующих работу и установку систем ОПС. При этом также учитываются и специфические характеристики самого объекта защиты. К таким характеристикам можно отнести, например, геометрию помещения, расположение окон, дверей, стен и других элементов, что играет существенную роль в обеспечении эффективности системы. Необходимо отметить, что на практике при установке элементов ОПС могут возникать различные сложности, связанные с особенностями конкретного объекта или внешними обстоятельствами. К сожалению, монтажники зачастую допускают отклонения от первоначального проекта по причинам различного характера, таким как стремление сэкономить время на установку, минимизировать расход кабеля или из-за несерьезного отношения к выполнению работы. Такие отклонения могут привести к серьезному снижению эффективности системы и увеличению потенциальных рисков безопасности. Методика, разработанная с учетом всех упомянутых факторов, призвана обеспечить грамотное и обоснованное размещение элементов ОПС, минимизируя возможные ошибки и их негативные последствия. Она станет ценным инструментом как на этапе проектирования, так и при последующей эксплуатации объектов, оснащенных системами охранно-пожарной сигнализации.

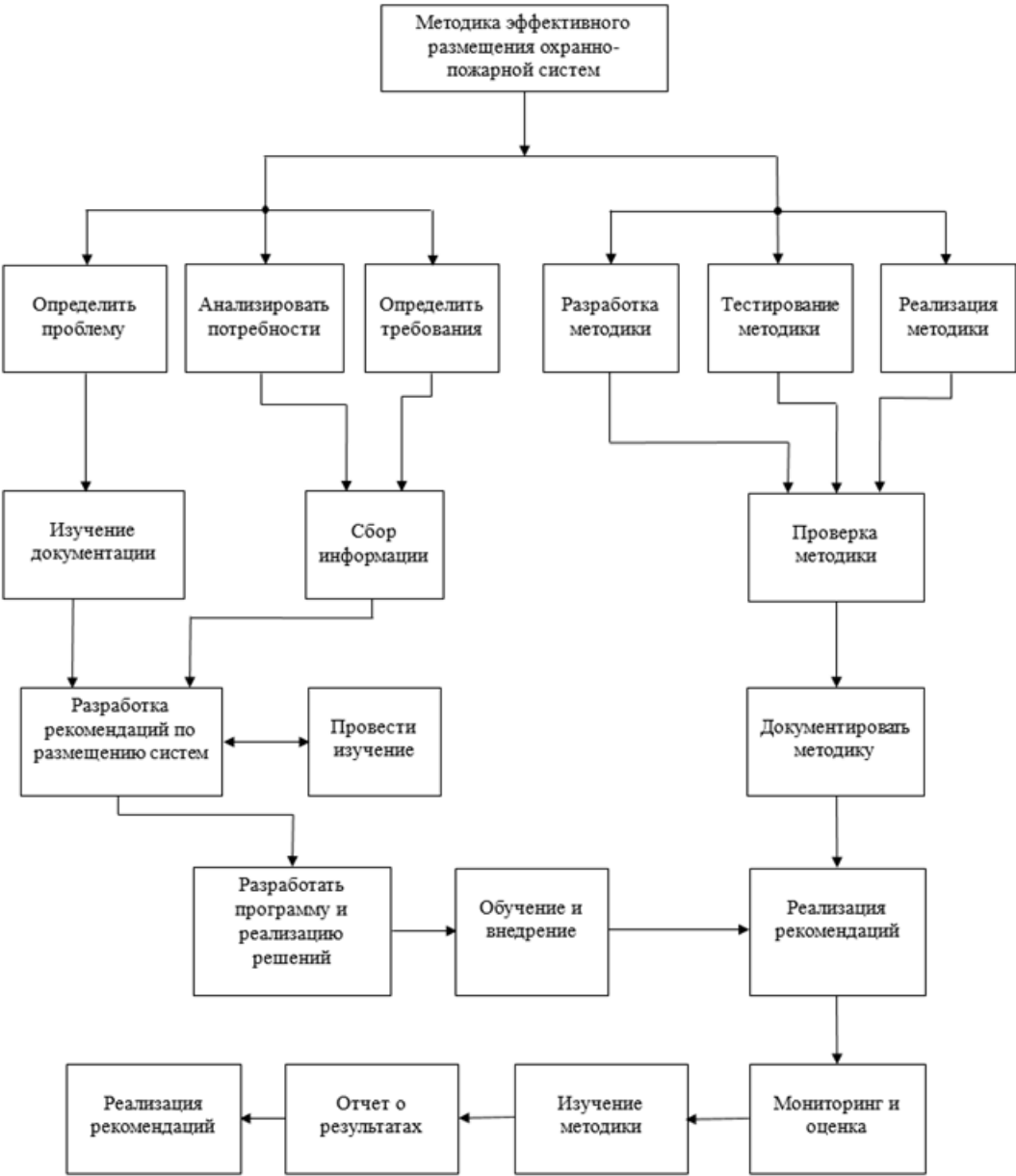


Рисунок 1. Методика эффективного размещения охранно-пожарных систем

Рассмотрим более подробно разработанную методику.

В следующей таблице (таблица 1) показаны входные и выходные зависимости в блоках, описывающие процессы.

Таблица 1. Процесс – Входные данные – Выходные данные

| Процесс              | Входные данные   | Выходные данные   |
|----------------------|--|---|
| Определение проблемы | Информация о текущих системах безопасности и пожарной безопасности в здании, потенциальных опасностях и рисках | Четкое определение проблемы                                       |
| Анализ потребностей  | Планировка здания, занятость, использование и другие соответствующие факторы                                   | Список требований к системам безопасности и пожарной безопасности |
|                      |  | Набор технических   |

|                                  |   |   |
|----------------------------------|---|---|
| Определение требований           | Список требований   | характеристик систем безопасности и пожарной безопасности                                 |
| Разработка методики              | Технические характеристики, лучшие практики, отраслевые стандарты                           | Методика эффективной установки систем безопасности и пожарной безопасности                |
| Тестирование методики            | Тестовые данные, системы безопасности и пожарной безопасности, оборудование для мониторинга | Обратная связь о эффективности методики   |
| Сбор информации                  | Отраслевые стандарты, лучшие практики, кейс-стадии  | Релевантная информация для разработки методики  |
| Разработка руководящих принципов | Технические характеристики, отраслевые стандарты, лучшие практики                           | Руководящие принципы для установки систем безопасности и пожарной безопасности            |
| Разработка программы и решения   | Технические характеристики, руководящие принципы  | Программа и решение для эффективной установки систем безопасности и пожарной безопасности |
| Внедрение программы и решения    | Программа и решение для эффективной установки систем безопасности и пожарной безопасности   | Установленные системы безопасности и пожарной безопасности в здании                       |
| Мониторинг и оценка              | Оборудование для мониторинга, данные обратной связи   | Данные об эффективности систем безопасности и пожарной безопасности                       |
| Изменение методики               | Данные обратной связи, отраслевые стандарты   | Готовый продукт   |

Для проверки разработанной методики эффективного размещения охранно-пожарных систем было проведено тестирования на реальном объекте с целью исследования эффективности разработанной методики.

### Тестирование методики

В качестве объекта исследования была выбрана учебная лаборатория П429 (рис. 2). Учебная лаборатория расположена на 4 этаже учебного здания. Высота аудитории 3 метра 30 сантиметров, ширина 5 метров 10 сантиметров, длина 8 метров, площадь аудитории 40,8 кв.м. В аудитории есть одна дверь и два окна.

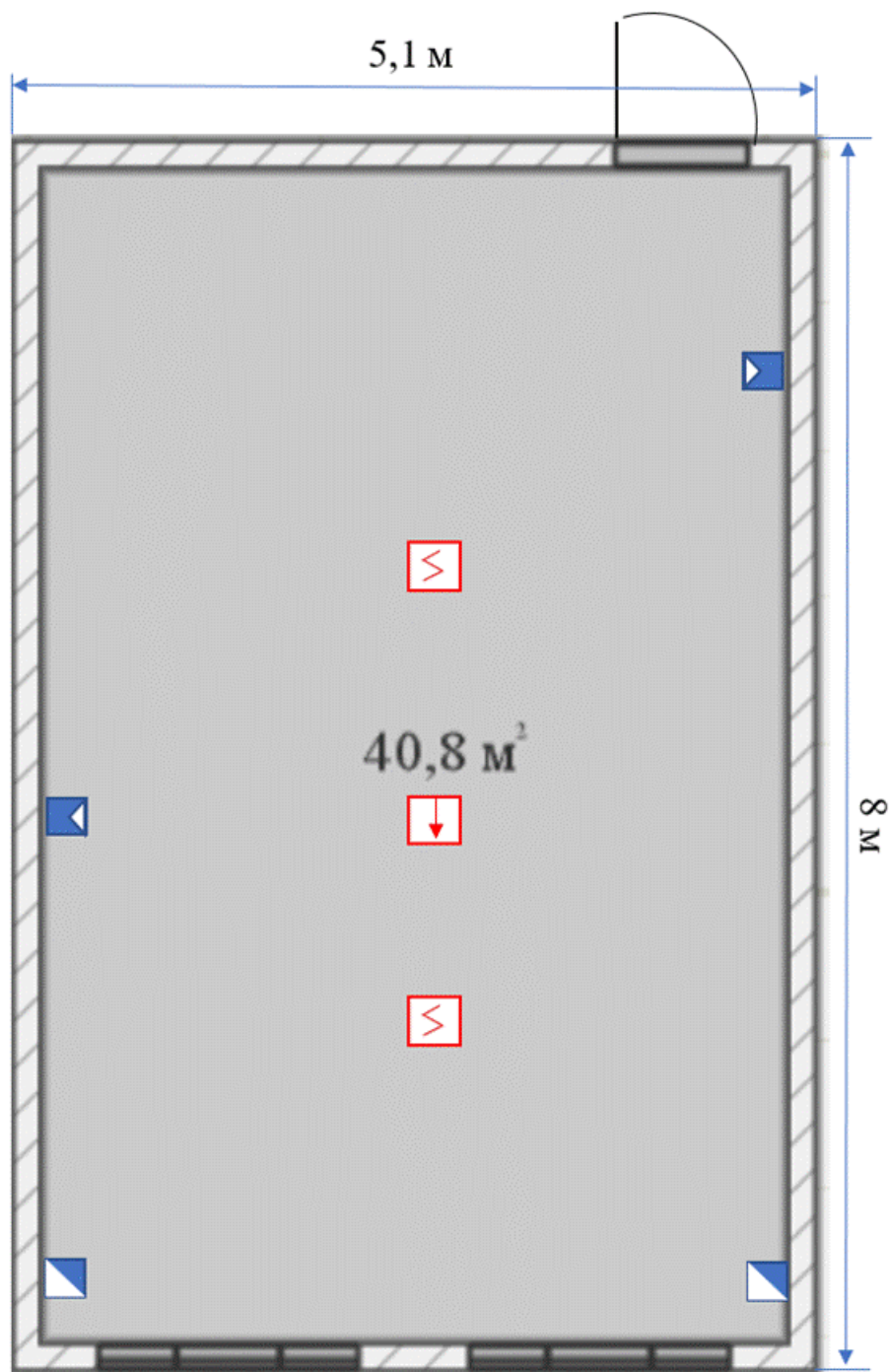






Рисунок 2. Схема учебной лаборатории П429 с размещёнными извещателями

В аудитории размещены следующие извещатели:

-  – извещатель разбития стекла, акустический пассивный,
-  – оптико-электронный извещатель, пассивный,
-  – извещатель противопожарный, точечный дымовой,
-  – извещатель противопожарный, тепловой.

Все эти извещатели подключаются к ПКП, который находится в другом помещении. После анализа помещения и теста по разработанной методике были выработаны рекомендации:

- Разместить: оптикоэлектронный, магнитоконтактные, точечные дымовые и тепловые



противопожарные извещатели, согласно требованиям приказа МЧС России от 31.07.2020 N 582 "Об утверждении свода правил "Системы противопожарной защиты. Системы пожарной сигнализации и автоматизация систем противопожарной защиты. Нормы и правила проектирования", пункта 6 и подпунктов 6.6.15 и 6.6.16.

- Магнитоконтактные извещатели установить на входную дверь и по одному извещателю на каждую открывающуюся створку окна.
- Разместить два противопожарных тепловых извещателей согласно требованиям приказа МЧС России от 31.07.2020 N 582 пункт 6.6.15 таблица 1 и технических характеристик теплового извещателя. Данный извещатель охватывает 39,55 кв.м, площадь лаборатории – 40,8 кв.м.
- Разместить один оптико-электронный извещатель над дверью так как других входов нет, а на окно устанавливать данные извещатели не целесообразно так как аудитория находится на четвёртом этаже и нету возможности злоумышленнику проникнуть в данное помещение через окна.
- Разместить один противопожарный дымовой извещатель по середине помещения согласно требованиям приказа МЧС России от 31.07.2020 N 582, пункта 6.6.15 таблицы 2 и технических характеристикам данного извещателя, который охватывает 128,61 кв.м., что в три раза меньше площади лаборатории.
- Убрать извещатели разбития стекла из лаборатории. Данный извещатель устанавливать не рационально так как аудитория находится на четвёртом этаже.

Следуя данным рекомендациям на рисунке 3 показано рекомендованное размещение следующих извещателей:



– магнитоконтактный извещатель,



– оптико-электронный извещатель, пассивный,



– извещатель противопожарный, точечный дымовой,



– извещатель противопожарный, тепловой.

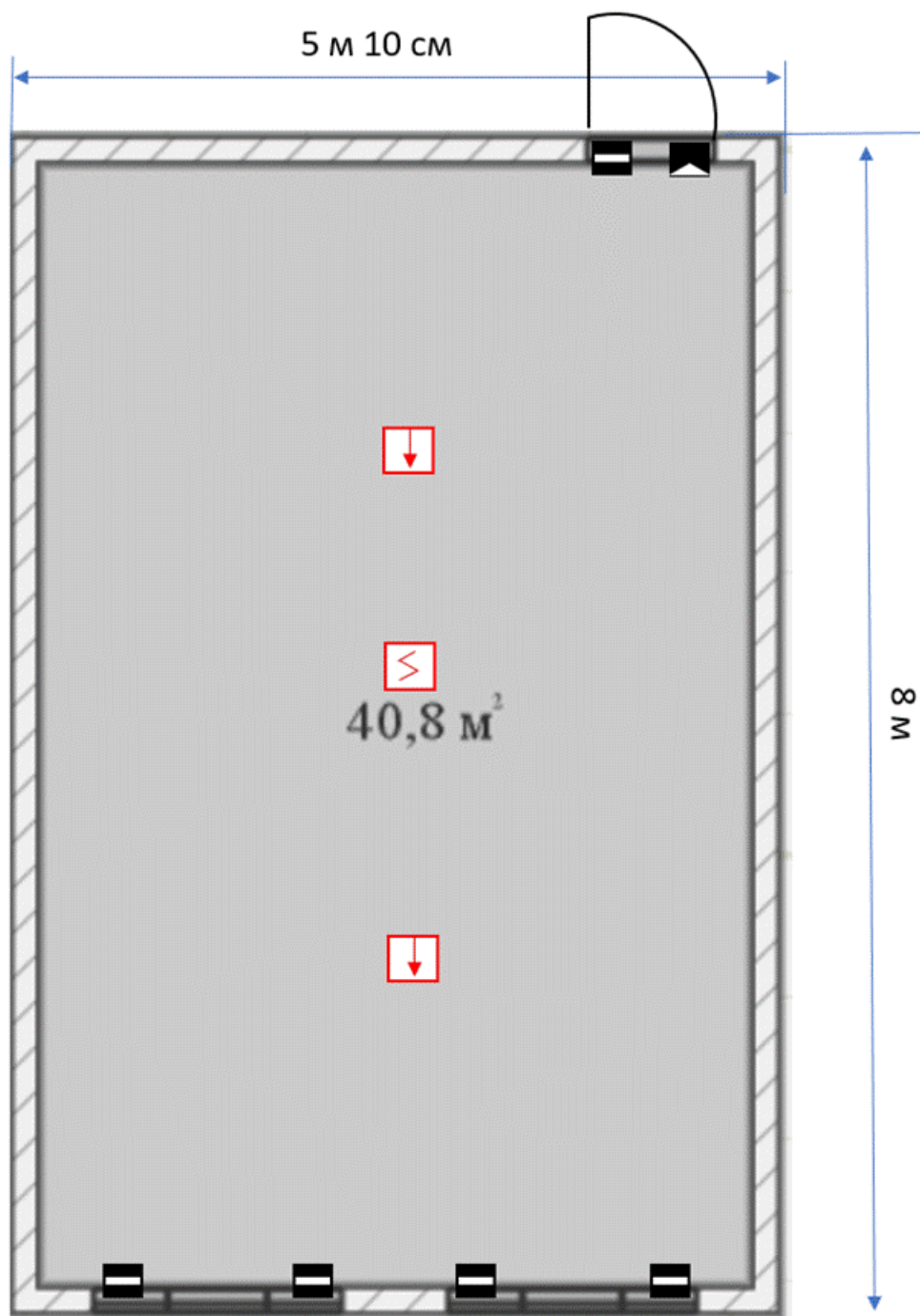


Рисунок 3. Размещение рекомендованных извещателей в помещении П429

В результате мы получили новое размещение охранно-пожарных извещателей, размещение которых позволит извещателям эффективно и своевременно обнаруживать вторжения на охраняемый объект, противопожарную защиту и своевременно информировать службу охраны для быстрой нейтрализации угроз.

### Заключение

В заключение хочется подчеркнуть, что в результате проведенных исследований и разработки была создана методика, способствующая рациональному размещению элементов ОПС. Это касается как начального этапа проектирования, так и уже функционирующих объектов защиты, на которых элементы ОПС установлены.

Особенно актуальным становится применение этой методики в свете анализа, проведенного лабораторией П429. Анализ выявил ряд проблем в размещении

извещателей, которые ограничивают их функциональные возможности. К примеру, было обнаружено, что некоторые извещатели размещены неэффективно из-за близкого расположения друг к другу или неправильного выбора зон защиты. Также были выявлены случаи использования нерациональных извещателей, как, например, размещение датчиков разбития стекла на верхних этажах зданий, где их срабатывание маловероятно.

Стоит отметить, что дополнительную угрозу безопасности представляет размещение пультов управления и коммутационных шлейфов в местах, доступных для несанкционированного доступа. Такое размещение позволяет злоумышленникам манипулировать этими элементами, что, в свою очередь, снижает уровень защиты.

В планах на будущее — автоматизация разработанной методики путем создания программного продукта. Такое решение будет обеспечивать возможность ввода данных с объекта защиты и автоматического получения рекомендаций на основе анализа. Для реализации этой задачи можно использовать современные методы, представленные в литературе [\[18,19,20\]](#), а также языки программирования.

В следующей таблице (таблица 1) показаны входные и выходные зависимости в блоках, описывающие процессы.

## Библиография

1. Гавришев А. А. Оценка скрытности беспроводных охранно-пожарных систем // Достижения и приложения современной информатики, математики и физики : Материалы VIII Всероссийской научно-практической заочной конференции, Нефтекамск, 15 ноября 2019 года. – Нефтекамск: Башкирский государственный университет, 2019. – С. 21-29.
2. Ряполова Е. И. Разработка автоматизированного рабочего места диспетчера охранно-пожарной сигнализации // Новая наука: Современное состояние и пути развития. – 2015. – № 4-1. – С. 81-85.
3. Юсупов Б. З. Разработка лабораторного стенда охранно-пожарной сигнализации по дисциплине технические средства охраны // XXV Туполевские чтения (школа молодых ученых) : Международная молодёжная научная конференция, посвященная 60-летию со дня осуществления Первого полета человека в космическое пространство и 90-летию Казанского национального исследовательского технического университета им. А.Н. Туполева-КАИ. Материалы конференции. Сборник докладов. В 6-ти томах, Казань, 10–11 ноября 2021 года. Том V. – Казань: Индивидуальный предприниматель Сагиева А.Р., 2021. – С. 758-763.
4. Кашпур Е. И. Исследование перспективных технологий цифровой модуляции в системах охранно-пожарной сигнализации // Молодежный научно-технический вестник. – 2015. – № 8. – С. 29.
5. Гизатуллин З. М. Методика и модели для физического моделирования электромагнитных помех на примере анализа помехоустойчивости электронных средств автотранспорта / З. М. Гизатуллин, Р. М. Гизатуллин, М. Г. Нуриев // Радиотехника и электроника. – 2021. – Т. 66. – №6. – С. 609-613.
6. Бонч-Бруевич А. М. Исследование перспективных технологий цифровой модуляции в системах охранно-пожарной сигнализации / А. М. Бонч-Бруевич, Е. И. Кашпур // Спецтехника и связь. – 2015. – № 3. – С. 24-28.
7. Юсупов Б. З. Разработка лабораторного стенда охранно-пожарной сигнализации по дисциплине технические средства охраны / Б. З. Юсупов, А. М. Мартынов // Актуальные проблемы науки и образования в условиях современных вызовов : Сборник материалов XIX Международной научно-практической конференции, Москва, 21 марта 2023 года. –

Москва: Печатный цех, 2023. – С. 80-91.

8. Бурыкин И. А. Организация защищённого канала связи в системе охранно-пожарной сигнализации / И. А. Бурыкин, Д. А. Петров, Д. С. Луппа // Актуальные вопросы в научной работе и образовательной деятельности : сборник научных трудов по материалам Международной научно-практической конференции: в 10 томах, Тамбов, 30 мая 2015 года. Том 6. – Тамбов: ООО "Консалтинговая компания Юком", 2015. – С. 23-25.

9. Берлев С. В. Экспериментальная оценка эффективности технических средств охранно-пожарной сигнализации / С. В. Берлев // Современные технологии обеспечения гражданской обороны и ликвидации последствий чрезвычайных ситуаций. – 2015. – № 1-1(6). – С. 62-63.

10. Юсупов Б. З. Разработка методики проведения лабораторных работы на стенде «ОПС Астра-713» по дисциплине технические средства охраны // XXV Туполевские чтения (школа молодых ученых) : Международная молодёжная научная конференция, посвященная 60-летию со дня осуществления Первого полета человека в космическое пространство и 90-летию Казанского национального исследовательского технического университета им. А.Н. Туполева-КАИ. Материалы конференции. Сборник докладов. В 6-ти томах, Казань, 10–11 ноября 2021 года. Том V. – Казань: Индивидуальный предприниматель Сагиева А.Р., 2021. – С. 764-767.

11. Синилов В. Г. Системы охранной, пожарной и охранно-пожарной сигнализации : учебник для образовательных учреждений, реализующих программы начального профессионального образования / В. Г. Синилов ; В. Г. Синилов. – 6-е изд., стер.. – Москва : Академия, 2011. – (Начальное профессиональное образование. Радиоэлектроника).

12. Вершинин И. С. Ассоциативная защита текстовых характеристик объектов / И. С. Вершинин, С. В. Пыстогов, Р. Ф. Гибадуллин, Д. А. Гашигуллин // Вестник Казанского государственного технического университета им. А.Н. Туполева. – 2020. – Т. 76, № 1. – С. 117-125.

13. Петик Н. С. Проектирование системы охранно-пожарной сигнализации / Н. С. Петик // Молодость. Интеллект. Инициатива : Материалы VI Международной научно-практической конференции студентов и магистрантов, Витебск, 19 апреля 2018 года. – Витебск: Витебский государственный университет им. П.М. Машерова, 2018. – С. 33-34.

14. Орешина Ю. В. Деловая репутация предпринимательских структур на рынке охранно-пожарной сигнализации / Ю. В. Орешина // Актуальные вопросы в научной работе и образовательной деятельности : сборник научных трудов по материалам Международной научно-практической конференции: в 10 томах, Тамбов, 30 мая 2015 года. Том 4. – Тамбов: ООО "Консалтинговая компания Юком", 2015. – С. 112-113.

15. Постановление Правительства РФ от 16 февраля 2008 г. N 87 "О составе разделов проектной документации и требованиях к их содержанию" [Электронный ресурс]. Доступ из системы ГАРАНТ // ЭПС "Система ГАРАНТ" : URL: <https://base.garant.ru/> (дата обращения: 16.06.2023).

16. ГОСТ Р 50776-95 (МЭК 60839-1-4:1989) «Системы тревожной сигнализации. Часть 1. Общие требования». [Электронный ресурс]. Доступ из системы ГАРАНТ // ЭПС "Система ГАРАНТ". URL: <https://base.garant.ru/> (дата обращения: 16.06.2023).

17. СП 68.13330.2017 "СНиП 3.01.04-87 Приемка в эксплуатацию законченных строительством объектов. Основные положения"[Электронный ресурс]. Доступ из системы ГАРАНТ // ЭПС "Система ГАРАНТ". URL: <https://base.garant.ru/> (дата обращения: 16.06.2023).

18. Нуриев М. Г. Конвертер Markdown-файлов в LaTeX-документ / М. Г. Нуриев, Е. С. Белашова, К. А. Барабаш // Программные системы и вычислительные методы. – 2023. –

№ 1. – С. 1-12.

19. Гибадуллин Р. Ф. Потокбезопасные вызовы элементов управления в обогащенных клиентских приложениях / Р. Ф. Гибадуллин // Программные системы и вычислительные методы. – 2022. – № 4. – С. 1-19.

20. Гибадуллин Р.Ф. Неоднозначность результатов при использовании методов класса Parallel в рамках исполняющей среды .NET Framework / Р.Ф. Гибадуллин, И.В. Викторов // Программные системы и вычислительные методы. – 2023. – № 2. – С. 1-14.

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Рецензируемая статья посвящена актуальной практической задаче безопасности помещений и снижению рисков распространения пожаров. Данная задача решается с помощью предлагаемого авторами подхода к размещению тепловых и звуковых датчиков. Авторы отмечают основные проблемы монтажа и проектирования пожарных систем, наиболее распространенные ошибки. Нет четкой формулировки, чем предложенная методика отличается от стандартного алгоритма. Положительной стороной работы является рассмотрения практического примера, однако он весьма формализован и не содержит количественных оценок.

Структура статьи отвечает требованиям к публикации. Статья не содержит экспериментальной части или результатов измерений, и носит методический характер. Подробно рассматривается схема размещения датчиков методика. Результаты измерений отсутствуют, анализ не включает оценку количественных показателей.

Стиль изложения соответствует требованиям. Имеются иллюстрации, схема. Встречаются единичные ошибки.

Библиография содержит 20 источников в отечественных журналах, из них 7 по материалам конференций. Ссылки по тексту имеются.

Замечания.

Введение рекомендуется сократить. Содержатся неоднократные повторы тезисов в разных формулировках, расплывчатость не позволяет четко выделить задачу исследования. Громоздкие абзацы.

Отсутствует анализ публикаций, посвященных проблемам снижения рисков пожара на объектах различного назначения.

Не приводится обзор существующих методик.

При перечислении нормативных документов и наличии ссылки, не требуется в тексте приводить и номер, и название, не рекомендуется перегружать статью справочными данными. В библиографии указать выходные данные (издательство, год), ссылка на интернет-сайт не требуется.

Описываемая методика включает стандартные этапы, нет четкой формулировки в чем именно состоит вклад авторов. Рекомендуется сосредоточиться на примерах, добавив количественные показатели.

В сх.1 какого назначения элемента «изучение методики» на одном из заключительных этапов? Для всех элементов рекомендуется привести в соответствие формулировки с содержанием.

Рис.2 и 3 рекомендуется уменьшить и объединить. Изменить подрисовочную надпись, напр., схема размещения датчиков до и после использования предлагаемого алгоритма. Привести в соответствие общий размер и подписи.

Обоснование размещения датчиков носит формализованный характер и соответствует

регламентирующим документам. Роль предлагаемой авторами методики не ясна. Каким образом рассчитаны расстояния? Какие количественные показатели меняются в результате. Какие датчики использованы (модель, характеристики) и почему. В заключении необходимо сосредоточиться на результатах выполненной авторами работы. Данные по размещению шлейфов, пультов управления, не рассматривались. Расчётные формулы не приводились. Назначение ссылок 18-20 не ясно. Библиографию оформить в соответствии с требованиями Журнала и ГОСТ. Статья будет интересна узкому кругу специалистов. Статья может быть опубликована после внесения правок, повторное рецензирование не требуется.

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Зубов Д.В., Лебедев Д.А. Диагностика отказов технологического оборудования химических производств с помощью искусственного интеллекта // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70729 EDN: XBIJYK URL: [https://nbpublish.com/library\\_read\\_article.php?id=70729](https://nbpublish.com/library_read_article.php?id=70729)

## Диагностика отказов технологического оборудования химических производств с помощью искусственного интеллекта

**Зубов Дмитрий Владимирович**

ORCID: 0000-0002-0703-1577

кандидат технических наук

доцент, кафедра информационных компьютерных технологий; Российский химико-технологический университет имени Д.И. Менделеева

142602, Россия, г. Москва, ул. Героев Панфиловцев, 20, ауд. 127

✉ [dvzubov@gmail.com](mailto:dvzubov@gmail.com)



**Лебедев Данила Александрович**

ORCID: 0009-0007-2873-2341

аспирант, кафедра информационных компьютерных технологий; Российский химико-технологический университет имени Д.И. Менделеева

119331, Россия, г. Москва, ул. Марии Ульяновой, 16, кв. 188

✉ [lebedev.d.a@muctr.ru](mailto:lebedev.d.a@muctr.ru)



---

[Статья из рубрики "Базы знаний, интеллектуальные системы, экспертные системы, системы поддержки принятия решений"](#)

**DOI:**

10.7256/2454-0714.2024.2.70729

**EDN:**

XBIJYK

**Дата направления статьи в редакцию:**

13-05-2024

**Дата публикации:**

24-05-2024

**Аннотация:** В работе рассмотрена задача автоматизированного распознавания одиночных аварийных ситуаций на химических и нефтеперерабатывающих производствах. Современные химико-технологические производства обслуживаются и управляются небольшим количеством персонала, что повышает нагрузку на каждого оператора. Для уменьшения количества ошибок операторов регулярно проводится их обучение на тренажёрах, оснащённых набором как штатных ситуаций (регламентный пуск, останов, нормальное ведение процесса, переход с одного режима на другой), так и аварийных сценариев (разгерметизация колонны, отказ насоса, выход из строя системы энергоснабжения). Тем не менее, предусмотреть все возможные отказы во время обучения операторов невозможно, и даже подготовленный оператор может не заметить первые признаки аварии, в связи с чем необходимо создание системы поддержки решений, помогающей оператору своевременно распознавать отказы технологического оборудования. Для распознавания отказов предложено использовать нейронную сеть, обученную на массиве данных смоделированных аварий. Для моделирования типовых аварий использован промышленный тренажёр на базе платформы RTsim. Новизна исследования заключается в использовании методов искусственного интеллекта для диагностики достояния технологического процесса по данным SCADA-системы и использовании для обучения нейронной сети данных не с реального объекта (которых всегда будет недостаточно), а с модели, точно соответствующей конкретному технологическому участку. Количество смоделированных сценариев, использованных для обучения нейронной сети может быть достаточно велико, что позволяет снизить долю ошибочных срабатываний системы. Разработанная система уверенно справляется с распознаванием отдельных отказов оборудования. Полученные результаты могут быть использованы для помощи операторам-технологам и для усовершенствования систем противоаварийной защиты. Анализ времени, необходимого системе для распознавания аварийной ситуации может быть использован для проектирования новых производств, модификации системы контроля и управления.

**Ключевые слова:**

отказ, авария, компьютерный тренажёр, RTsim, цифровой двойник, имитационное моделирование, система принятия решений, промышленная безопасность, искусственный интеллект, нефтепереработка

Современные информационные технологии значительно облегчают управление сложными химическими и нефтеперерабатывающими производствами: SCADA-системы позволяют операторам видеть текущую и архивную информацию с датчиков на своём рабочем месте, осуществлять удалённое управление исполнительными механизмами и получать подсказки в ходе ведения технологического процесса. Современные АСУТП позволили снизить количество работников, занятых рутинными операциями, однако полностью отказаться от операторов, осуществляющих диспетчерское управление технологическим процессом не удаётся. Значительные сложности для управления представляют процессы пуска и останова технологических линий, перехода на новое сырьё, изменение технологических режимов и т.д. Особую сложность представляют процессы пуска новых технологических линий и пуска после ремонта, но и на установках непрерывного действия, работающих в стационарном режиме тоже могут возникнуть сложности – например, в результате аварий.

Аварийные ситуации проявляются изменением показаний датчиков и важнейшей задачей оператора является своевременная правильная оценка ситуации и принятие



необходимых действий. Значительной помощью операторам может служить ПЛАС – план ликвидации аварийных ситуаций, в котором прописаны признаки аварии, возможные причины её возникновения и рекомендуемые действия. Тем не менее, своевременное распознавание аварий является сложной задачей для операторов, особенно на ранних стадиях, когда ещё не сработали предупредительные и аварийные сигнализаторы технологических величин.

Полезным инструментом для подготовки операторов являются компьютерные тренажёры, в частности – компьютерные тренажёры на базе платформы RTsim [1], совмещающие в себе динамическую модель конкретного производства, имитатор SCADA-системы, обучающий комплекс из набора упражнений разной сложности и систему оценки уровня подготовки оператора. В набор упражнений входят задания от самых простых (ведение стационарного непрерывного технологического процесса на отдельном аппарате) до очень сложных (самостоятельный пуск и останов участка нефтеперерабатывающего производства, распознавание и ликвидация типовых аварийных ситуаций).

Достоинством тренажёрного комплекса является соответствие реальному производству, в том числе и по тем показаниям индикаторов на мнемосхемах, трендов и т.д., которые получает АСУТП и которые видит оператор [2], т.е. есть экран «реального мира» – модели технологического процесса, на котором можно увидеть состояние и управлять местным оборудованием (в частности – ручными задвижками) и при необходимости можно включить режим отображения всех переменных состояния модели, в том числе и тех, которые не контролируются датчиками и поэтому не видны на мнемосхемах SCADA-системы. Фрагмент экрана модели технологического процесса (Модель ТП) представлен на рис. 1.

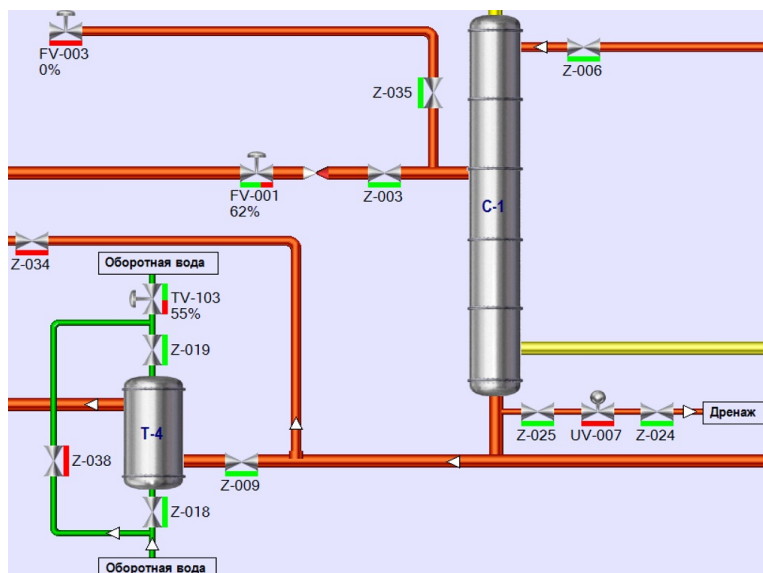


Рис. 1. Фрагмент экрана модели ТП тренажёрного комплекса RTsim

На экране распределённой системы управления (PCY) представлен интерфейс SCADA-системы, причём обучаемый в некоторых случаях может выбрать используемую систему, так как они различаются видом элементов интерфейса и т.д. На рис. 2 представлен тот же самый участок технологической линии, что и на рис. 1, но так, как он виден в SCADA-системе на базе программного комплекса Yokogawa.

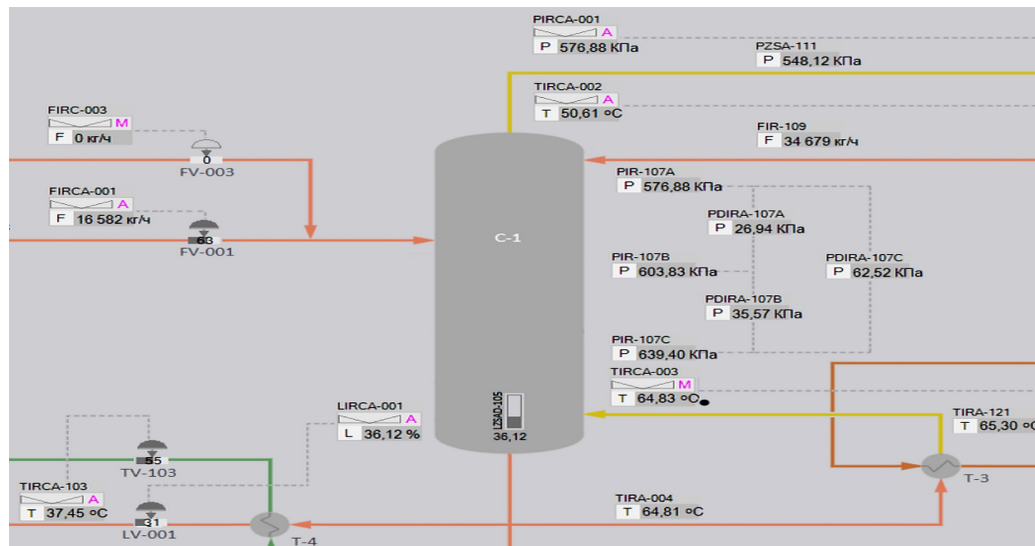


Рис. 2. Фрагмент экрана PCY тренажёрного комплекса RTsim

Пример отображения аварии типа «Полное разрушение колонны C-1» приведён на рис. 3. Оператор по показаниям индикаторов, трендам и предупредительным сообщениям должен идентифицировать тип происшествия и предпринять действия по остановке технологического процесса.

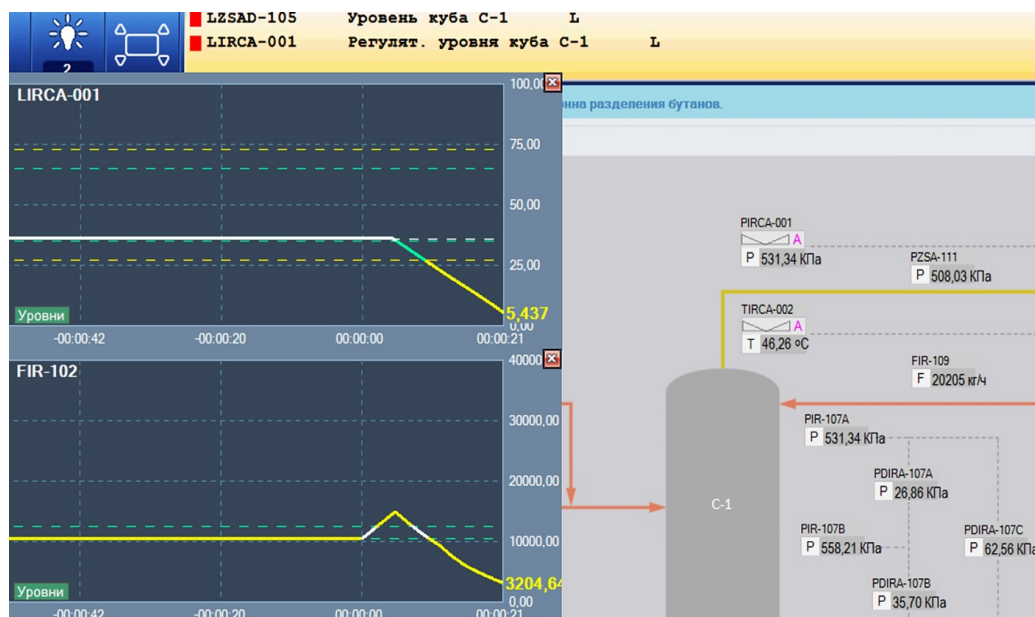


Рис. 3. Фрагмент экрана PCY тренажёрного комплекса RTsim при имитации аварии типа «Полное разрушение колонны C-1»

Подготовка операторов к работе в условиях нештатных ситуаций снижает время их реакции на уже знакомую ситуацию, но набор аварийных ситуаций, которые рассматриваются в процессе обучения ограничен, ведь возможных аварий может быть несколько типов с каждым аппаратом технологической цепочки, в результате операторы будут готовы к уже знакомым аварийным ситуациям, которые они видели в ходе обучения и будут менее готовы к ситуациям, которые им не встречались, что повысит вероятность совершения неоптимальных действий. Логичным шагом представляется создание автоматизированной системы распознавания аварийных ситуаций.

В работе [3] предложено строить дерево отказов технологической линии и с учётом вероятностей возникновения исходных событий и ожидаемого вреда от последствий

аварий выбирать действия оператора, которые сведут к минимуму наиболее вероятный ущерб. К сожалению, проблемой остаётся получение вероятностей исходных событий (течь аппарата, закупорка трубопровода, отказ клапана и т.д.) и поэтому этот подход применим скорее на этапе проектирования.

В работе [4] представлен обзор публикаций по использованию искусственного интеллекта для диагностики аварий в промышленности. Большинство решений ориентированы на использование в транспорте и машиностроении, но есть и публикации по применению искусственного интеллекта в химической промышленности: в работе [5] предложено использовать байесовскую сеть для обнаружения причин аварий на химических производствах, в работе [6] предложено использовать частный случай рекуррентной нейронной сети – LSTM-сеть – для ранней диагностики аварий в системе принудительного окисления при мокрой сероочистке дымовых газов. В работе [7] предложено использовать нейронную сеть для диагностики отказов предохранительного клапана. Во всех рассмотренных случаях для обучения сети использовались данные с реальных объектов, что ограничило объём использованного датасета.

Нами предлагается в процессе работы оператора использовать автоматизированную систему распознавания аварий, основанную на нейронной сети, обученной с помощью данных, полученных с модели конкретного производства.

В работе [8] предложен подход для имитации типовых аварий технологических аппаратов с помощью систем моделирования химико-технологических процессов, таких как Aspen One, UniSim и других. Использование систем моделирования или цифровых двойников [9] значительно облегчит сбор модельных данных, но предлагаемый подход вполне может быть использован и на тренажёрах.

В тренажёрном комплексе RTsim моделируются технологические линии вплоть для ручных и предохранительных клапанов, что позволяет смоделировать широкий спектр аварийных ситуаций. Рассмотрим несколько примеров:

- моделирования стабильной разгерметизации аппарата – открытие отсечного клапана на дренаж (в рассмотренном на рис. 1-3 случае – клапан UV-007), который в нормальном режиме полностью закрыт и полностью открывается в момент начала имитации аварии;
- моделирования развивающейся разгерметизации аппарата – предварительно отсечной клапана на дренаж (UV-007) открыт, но закрыт ручной клапан (Z-024), который в момент начала имитации аварии начинает плавно открываться;
- отказ центробежного насоса имитируется резким закрытием ручного клапана на линии нагнетания, полностью открытого в нормальном режиме.

Использование предложенного подхода позволяет смоделировать большое количество аварий и получить тренды технологических переменных, срабатывания блокировок и т.д., которые можно использовать для обучения нейронной сети.

Для обучения нейронной сети необходимо сформировать датасет, структура которого может быть различной, но очевидно, что в нём целевыми признаками будет код типа аварии, а предикторами – величины технологических переменных. Для определённости, будем считать, что каждый тип аварии кодируется значением 1 в соответствующем столбце, т.е. есть  $N_a$  столбцов целевых признаков,  $N_a$  – количество кодируемых типов аварий, причём 0 означает что авария данного типа не произошла. Допустим, что второй

столбец целевых признаков кодирует тип аварии «отказ насоса Н-1А», а третий столбец – тип аварии «отказ насоса Н-1В», тогда появление в ячейке второго столбца 1 будет означать отказ насоса Н-1А, а 0 – означать его исправное состояние (в том числе и когда он выключен).

Количество столбцов предикторов будет зависеть от количества информационных сигналов в АСУТП, их можно разделить на группы:

- технологические переменные с аналоговым или цифровым сигналом (данные с датчиков температуры, давления, уровня, состава и т.д.). Переменные этой группы нормализуем в диапазоне от 0 до 255, т.е. 0 – будет соответствовать нижней границе диапазона измерения измерительного преобразователя, 255 – верхней границе. При этом происходит некоторое огрубление сигнала, выполняющее функцию цифрового фильтра нулевого порядка, что несколько спасает от случайных флуктуаций технологической величины. Если по каким-то причинам это нежелательно, на этапе разработки системы можно задать диапазон нормировки в соответствии с мнением разработчика.
- технологические переменные с дискретным сигналом (сигнал с сигнализаторов нижнего и верхнего уровней, реле давления и т.д.) – кодируются как 0 (соответствующий сигнализатор не сработал) и 1 (сигнализатор сработал);
- цифровые управляющие переменные (степень открытия регулирующего клапана, мощность обогревателя и т.д.) – нормализуется в диапазоне от 0 (клапан полностью закрыт/обогреватель полностью выключен) до 255 (клапан полностью открыт/мощность обогревателя максимальна);
- дискретные управляющие переменные (отсечные клапаны, включение центробежного насоса и т.д.) – кодируются как 0 (клапан закрыт, насос выключен) и 1 (клапан открыт, насос включен).

В большинстве случаев распределённая система управления в рамках одного технологического участка управляется одним промышленным контроллером, который работает циклическим образом: в начале цикла с блоков ввода в соответствующую область памяти заносятся значения технологических переменных и до окончания этого цикла работы контроллера они считаются неизменными. На их основе контроллер рассчитывает величины управляющих воздействий, заносит их в выходную область данных и они одновременно выдаются на блоки вывода, после чего начинается новый цикл работы контроллера. Отход от данной схемы возможен, но резко усложняет вычислительную сложность системы управления и встречается сравнительно редко. Аналогично, системы моделирования технологических процессов обычно оперируют понятием такта, в рамках которого данные, рассчитанные на предыдущей итерации считаются неизменными и одновременно обновляются по окончании такта. Набор данных о состоянии системы – т.е. все значения технологических величин, положения клапанов и т.д. называют снимком системы.

По одному снимку системы можно было бы оценивать её состояние, если бы у системы не было «памяти», а у каналов управления – времени запаздывания. По факту, для управления инерционными объектами (ёмкостные аппараты, теплообменники, трубопроводы) нужно значить не только их текущее состояние, но и некоторое количество предыдущих. Для определённости предположим, что длительность одного периода опроса датчиков в АСУТП и одного цикла в системе моделирования равна 1 секунде, а величина самого большого из времён запаздывания в рассматриваемом

фрагменте технологической линии равна  $t$ , тогда можно предположить, что для корректного распознавания всех состояний системы достаточно 1.5 – 2  $t$  снимков. Вероятно, точное минимальное количество данных, достаточное для обучения нейронной сети будет зависеть и от топологии потоков внутри рассматриваемого фрагмента технологической линии.

Для иллюстрации предложенного метода рассмотрим предельно упрощённый пример: имеется центробежный насос, перекачивающий охлаждающую жидкость, состояние насоса описывается переменной  $S$ ,  $S=0$  означает, что насос исправен,  $S=1$  означает что насос вышел из строя. Управление насосом описывается переменной  $H$ ,  $H=1$  означает что на насос подано напряжение, он должен быть включён,  $H=0$  означает что на насос не подаётся напряжение, он должен быть выключён. Расход охлаждающей жидкости на линии нагнетания насоса обозначим переменной  $F$ ,  $F=0$  обозначает отсутствие расхода,  $F=255$  – максимально возможный расход. Предположим, что продолжительность такта моделирования в системе 1 секунда, а время запаздывания по каналу управления – 2 секунды. Тогда фрагмент датасета будет иметь вид, представленный в таблице 1.

Таблица 1. Пример фрагмента датасета для обучения нейронной сети

| Номер строки | $H(t)$ | $F(t)$ | $H(t-1)$ | $F(t-1)$ | $H(t-2)$ | $F(t-2)$ | $S(t)$ |
|--------------|--------|--------|----------|----------|----------|----------|--------|
| $i$          | 0      | 0      | 0        | 0        | 0        | 0        | 0      |
| $i+1$        | 1      | 0      | 0        | 0        | 0        | 0        | 0      |
| $i+2$        | 1      | 0      | 1        | 0        | 0        | 0        | 0      |
| $i+3$        | 1      | 10     | 1        | 0        | 1        | 0        | 0      |
| $i+4$        | 1      | 50     | 1        | 10       | 1        | 0        | 0      |
| $i+5$        | 1      | 100    | 1        | 50       | 1        | 10       | 0      |
| $i+6$        | 1      | 10     | 1        | 100      | 1        | 50       | 1      |
| $i+7$        | 1      | 0      | 1        | 10       | 1        | 100      | 1      |

На представленном в таблице 1 фрагменте датасета мы видим следующий сценарий:

строка  $i$  – исходное состояние, команда на включение насоса не подана, насос не работает, расход жидкости отсутствует, считается что насос исправен;

строка  $i+1$  – подана команда на включение насоса, но пока он ещё не успел включиться, расхода жидкости нет, считается что насос исправен;

строка  $i+2$  – подана команда на включение насоса, насос уже работает, но датчик расхода пока его не замечает, считается что насос исправен;

строка  $i+3$  – подана команда на включение насоса, насос работает, датчик расхода измерил расход величиной 10/255 от максимального, считается что насос исправен;

строка  $i+4$  – подана команда на включение насоса, насос работает, датчик расхода измерил расход величиной 50/255 от максимального, считается что насос исправен;

строка  $i+5$  – подана команда на включение насоса, насос работает, датчик расхода измерил расход величиной 100/255 от максимального, считается что насос исправен;

строка  $i+6$  – подана команда на включение насоса, насос работает, датчик расхода измерил расход величиной 10/255 от максимального – меньше чем было, это признак аварии.

Вариант структуры нейронной сети для распознавания аварийных ситуаций изображён на рис. 4.

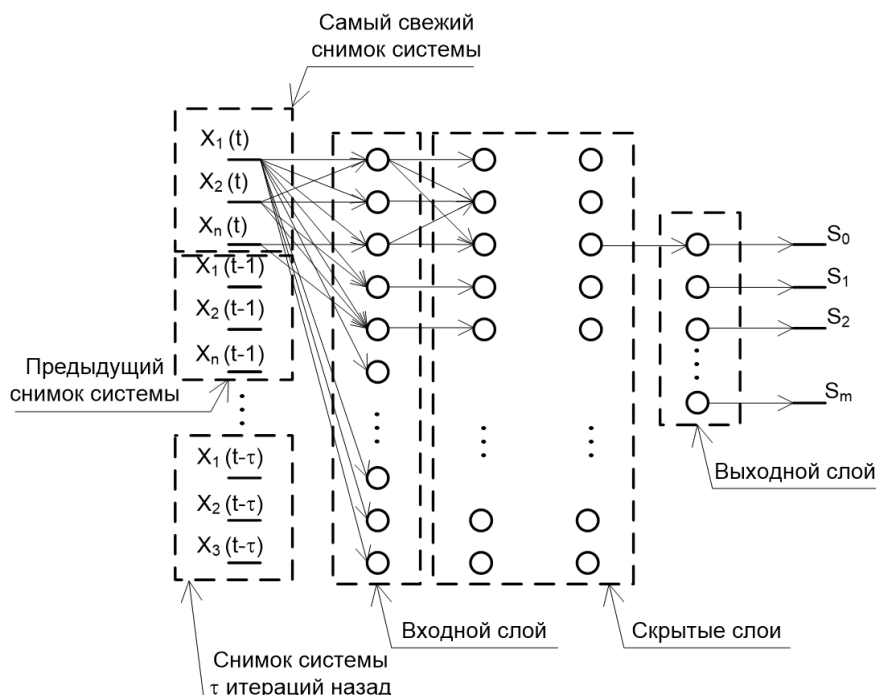


Рис. 4. Структура нейронной сети для распознавания аварийных ситуаций

Количество нейронов во входном слое целесообразно выбрать равным числу входных переменных в одном снимке системы, умноженному на число снимков, т.е. в нашем примере с диагностикой насоса получим 2 переменных в снимке на 3 снимка = 6 нейронов во входном слое. В выходном слое число нейронов выберем равным числу распознаваемых типов аварий ( $m$ ), увеличенному на 1 (состояние  $S_0$  исправной работы), в нашем примере число типов аварий было 1, значит, в выходном слое нужно предусмотреть 2 нейрона. Поскольку не может быть одновременно состояния исправной работы и аварии, то для выходного слоя выберем функцию активации типа Softmax (многопеременная логистическая функция). Для выбора количества скрытых слоёв и функций активаций в них необходимо более детально рассмотреть датасет, но для рассмотренного примера с насосом вполне достаточно двух скрытых слоёв с функцией активации ReLu.

В состав тренажёра RTsim входит модель и ряд типовых сценариев работы цеха разделения бутанов, в частности есть возможность провести пуск, останов колонны, наблюдать за стационарным режимом, переходить с режима на режим. Среди типовых аварий в комплект тренажёра входят сценарии полного разрушения колонны C-1, отказа насоса H-1A и выхода из строя системы подачи силового электропитания. Все комплектные сценарии (пуск, останов, штатная работа, три типа аварий) были использованы для формирования обучающего датасета, дополнительно сгенерированы данные по результатам имитации аварии насоса H-1B.

Полученный датасет использовался для обучения нейронной сети с двумя скрытыми слоями и удалось найти совокупность весов нейронов, обеспечивающую распознавание указанных четырёх видов аварий. При дальнейшем наборе данных моделирования планируется обучить сеть распознавать большее число типов отказов оборудования.

**Заключение.** Предложен метод распознавания отказов аппаратов химико-технологических систем с использованием нейронной сети, обученной на модели

конкретного производства. Выбрана структура нейронной сети, распознающей одиночные аварии. Созданная система в перспективе может быть усовершенствована для распознавания не только одиночных, но множественных и зависимых отказов.

## Библиография

1. Цифровые тренажеры технологических процессов РТСИМ. Карьера для обеспечения безопасности в нефтехимическом комплексе / Р. Р. Габдрахманов, Э. В. Гарифуллина, В. В. Бронская [и др.] // Наука, технологии, общество: Экологический инжиниринг в интересах устойчивого развития территорий : Сборник научных статей, Красноярск, 09–10 ноября 2023 года. – Красноярск: Красноярский краевой Дом науки и техники Российского Союза научных и инженерных общественных объединений, 2023. – С. 131–135. – EDN BLETRM.
2. Титова, Л. М. Цифровые двойники при подготовке инженерных кадров химических производств в высшей школе / Л. М. Титова, Н. П. Мемедейкина // Международный форум KAZAN DIGITAL WEEK-2023 : Сборник материалов, Казань, 20–22 сентября 2023 года / Сост. Р.Ш. Ахмадиева, Р.Н. Минниханов. Под общей редакцией Р.Н. Минниханова. – Казань: Научный центр безопасности жизнедеятельности, 2023. – С. 954–960. – EDN RNNMLO.
3. Имитация процессов управления агрегатом производства метанола на тренажере / В. Л. Перов [и др.].-Текст : непосредственный // Тр. Моск. хим.-технол. ин-та им. Д.И.Менделеева.-1988.-Вып. 152.-С. 12-18.
4. Siguo Bi, Cong Wang, Bochun Wu, Shuyan Hu, Wutao Huang, Wei Ni, Yi Gong, Xin Wang. A comprehensive survey on applications of AI technologies to failure analysis of industrial systems// Engineering Failure Analysis, Volume 148, 2023, 107172, ISSN 1350-6307, <https://doi.org/10.1016/j.engfailanal.2023.107172>.
5. Pallavi Kumari, Bhavana Bhadriraju, Qingsheng Wang, Joseph Sang-II Kwon. A modified Bayesian network to handle cyclic loops in root cause diagnosis of process faults in the chemical process industry//Journal of Process Control. Volume 110, 2022, Pages 84-98, ISSN 0959-1524, <https://doi.org/10.1016/j.jprocont.2021.12.011>.
6. Chunbo Pang, Dawei Duan, Zhiying Zhou, Shangbo Han, Longchao Yao, Chenghang Zheng, Jian Yang, Xiang Gao, An integrated LSTM-AM and SPRT method for fault early detection of forced-oxidation system in wet flue gas desulfurization //Process Safety and Environmental Protection. Volume 160, 2022, Pages 242-254, ISSN 0957-5820, <https://doi.org/10.1016/j.psep.2022.01.062>.
7. Hao Yin, He Xu, Weiwang Fan, Feng Sun. Fault diagnosis of pressure relief valve based on improved deep Residual Shrinking Network// Measurement. Volume 224, 2024, 113752, ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2023.113752>.
8. Имитация аварий оборудования в системах моделирования химико-технологических процессов / Д. В. Зубов, А. В. Лобанов, Е. Б. Филиппова, Д. А. Лебедев // Математические методы в технологиях и технике. – 2022. – № 6. – С. 31-35. – DOI 10.52348/2712-8873\_MMTT\_2022\_6\_31. – EDN GSNFHJ.
9. Разработка программных модулей цифрового двойника химико-технологического производства / Е. Б. Филиппова, Д. В. Зубов, Р. С. Крашенинников, А. В. Лобанов // Автоматизация в промышленности. – 2021. – № 7. – С. 16-20. – DOI 10.25728/avtprom.2021.07.04. – EDN CJBIPX

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

Со списком рецензентов издательства можно ознакомиться [здесь](#).

Статья посвящена применению искусственного интеллекта для диагностики отказов технологического оборудования в химической промышленности. Автор рассматривает возможности использования нейронных сетей для автоматизированного распознавания аварийных ситуаций, что является крайне актуальным направлением в контексте повышения надежности и безопасности промышленных процессов.

В исследовании использована методика моделирования и анализа данных, полученных с помощью системы RTsim. Моделирование включает создание датасетов на основе данных, полученных из технологических процессов, и их последующую обработку с использованием нейронных сетей. Подход основан на применении методов машинного обучения для распознавания различных типов отказов оборудования.

Актуальность исследования обусловлена стремлением к повышению безопасности и эффективности химических производств. Использование искусственного интеллекта для диагностики отказов позволяет значительно сократить время реакции на аварийные ситуации, минимизировать риски и снизить затраты на устранение последствий аварий.

Научная новизна статьи заключается в разработке и применении новой методики распознавания аварийных ситуаций с использованием нейронных сетей. Автор предлагает подход, который учитывает специфику химических процессов и позволяет более точно и оперативно выявлять неисправности на ранних стадиях. Впервые предлагается использование данных, полученных непосредственно с моделей конкретного производства, для обучения нейронных сетей.

Статья написана в научном стиле, грамотно структурирована и логически изложена. Введение четко описывает актуальность проблемы и цели исследования. Основная часть подробно рассматривает методологию, результаты и их анализ, что делает статью понятной и доступной для восприятия. Выводы логично вытекают из представленных данных и подкреплены иллюстрациями и таблицами, что облегчает восприятие информации.

Выводы исследования показывают, что использование нейронных сетей для диагностики отказов технологического оборудования является перспективным направлением. Автор демонстрирует, что предложенный подход позволяет значительно повысить точность и скорость распознавания аварийных ситуаций. Статья содержит конкретные рекомендации по применению разработанной методики в промышленной практике, что делает ее полезной для инженерного и научного сообщества.

Статья представляет значительный интерес для специалистов в области автоматизации и управления технологическими процессами, инженерного персонала химических производств, а также для исследователей, занимающихся применением искусственного интеллекта в промышленности. Она предлагает практические решения и инновационные подходы, которые могут быть использованы для улучшения безопасности и эффективности производственных процессов.

Статья является значимым вкладом в область применения искусственного интеллекта для диагностики отказов технологического оборудования. Она содержит ценные методологические и практические результаты, которые могут быть использованы для дальнейших исследований и практического применения. Рекомендуется к публикации без существенных изменений.

Для дальнейшего развития данной работы можно предложить несколько направлений. В первую очередь, необходимо провести тестирование предложенной методики на реальных промышленных установках. Экспериментальные данные помогут подтвердить точность и надежность модели, а также выявить возможные недостатки и области для улучшения. Кроме того, важно собрать и использовать более обширные и разнообразные данные для обучения нейронных сетей. Включение данных о различных



типах оборудования и аварийных ситуаций повысит универсальность и точность системы диагностики.

Разработка методов интеграции предложенной системы с уже существующими SCADA-системами и другими автоматизированными системами управления производством обеспечит более широкое применение технологии в промышленности. Проведение исследований по анализу динамического поведения технологических процессов в реальном времени позволит улучшить модели прогнозирования и раннего обнаружения аварийных ситуаций. Также необходимо создать удобный и интуитивно понятный интерфейс для операторов, позволяющий быстро и эффективно реагировать на обнаруженные неисправности. Важно обеспечить легкость интерпретации данных, предоставляемых нейронной сетью.

Разработка программ обучения и тренировки для операторов и технического персонала, включая использование виртуальных тренажеров, повысит уровень подготовки сотрудников и их готовность к действиям в аварийных ситуациях. Проведение оценки экономической эффективности внедрения предложенной системы позволит обосновать инвестиции в технологию и продемонстрировать ее преимущества перед традиционными методами. Исследование возможности применения других архитектур нейронных сетей и алгоритмов машинного обучения для повышения точности и производительности системы, а также адаптация моделей под специфические условия различных производств, могут улучшить результаты.

Программные системы и вычислительные методы

Правильная ссылка на статью:

Филиппова К.А., Аюшеев Т.В., Дамдинова Т.Ц., Цыдыпов Ц.Ц. Исследование напряженно-деформированного состояния композитной лопасти в ANSYS WorkBench // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70712 EDN: XDTLCG URL: [https://nbpublish.com/library\\_read\\_article.php?id=70712](https://nbpublish.com/library_read_article.php?id=70712)

## Исследование напряженно-деформированного состояния композитной лопасти в ANSYS WorkBench

**Филиппова Ксения Анатольевна**

старший преподаватель; кафедра "Самолето- вертолетостроение"; Восточно-Сибирский государственный университет технологий и управления

670013, Россия, республика Бурятия, г. Улан-Удэ, ул. Ключевская, 40 В

✉ [ipq84@mail.ru](mailto:ipq84@mail.ru)



**Аюшеев Тумэн Владимирович**

доктор технических наук

доцент; кафедра "Инженерная и компьютерная графика"; Восточно-Сибирский государственный университет технологий и управления

670013, Россия, Бурятия область, г. Улан-Удэ, ул. Ключевская, 40 В

✉ [atv62@bk.ru](mailto:atv62@bk.ru)



**Дамдинова Татьяна Цыбиковна**

ORCID: 0000-0002-3597-3262

кандидат технических наук

доцент; кафедра "Инженерная и компьютерная графика"; Восточно-Сибирский государственный университет технологий и управления

670000, Россия, республика Бурятия, г. Улан-Удэ, ул. Ключевская, 40 В

✉ [dtatyanac@mail.ru](mailto:dtatyanac@mail.ru)



**Цыдыпов Цыбик Цырендоржиевич**

кандидат технических наук

доцент; кафедра "Самолето- вертолетостроение"; Восточно-Сибирский государственный университет технологий и управления

670013, Россия, Бурятия область, г. Улан-Удэ, ул. Ключевская, 40 В

✉ [sssibik@mail.ru](mailto:sssibik@mail.ru)



---

[Статья из рубрики "Математическое моделирование и вычислительный эксперимент"](#)

**DOI:**

10.7256/2454-0714.2024.2.70712

**EDN:**

XDTLCG

**Дата направления статьи в редакцию:**

12-05-2024

**Дата публикации:**

25-05-2024

**Аннотация:** В настоящей работе был выполнен расчет статической прочности лопасти БПЛА из композиционного материала. Композиционные материалы имеют преимущество над традиционными материалами (металлами и сплавами) в области авиации - выигрыш в весе, низкая чувствительность к повреждениям, высокая жесткость, высокие механические характеристики. При этом определение уязвимых мест в слоистой конструкции является сложной задачей и на практике решается с помощью разрушающего контроля. При моделировании были использованы композитные материалы, имеющиеся в библиотеке материалов ANSYS: Epoxy Carbon Woven (230 GPa) Prepreg тканый углепластик в виде препрега – полуфабриката, пропитанной эпоксидной смолой углеткани с модулем Юнга  $E=230$  ГПа и Epoxy Carbon (230 GPa) Prepreg однонаправленный углепластик-препрег, пропитанный эпоксидной смолой с модулем Юнга  $E=230$  ГПа. Комплексно исследовать слоистую конструкцию позволяют современные программные продукты, такие как, ANSYS WorkBench. Были исследованы несколько вариантов конструкций лопасти с разными наполнителями в качестве срединного материала. Был использован прямой и обратный критерий разрушения на основе теории Цая-Хилла. Влияние силы тяжести не учитывалось. Показано, что разработанная конструкция лопасти соответствует предъявляемым требованиям. В качестве срединного материала лопасти были выбраны материалы - древесина бальзы, сосны, осины и пенополиуретан. Древесина сосны и осины были выбраны по критерию их доступности и имеющие наименьшую плотность. В библиотеке материалов используемого программного комплекса ANSYS WorkBench имеются характеристики не на все из них, поэтому характеристики выбранных материалов (сосны и осины) были добавлены вручную. Для моделирования и расчетов в программе ANSYS WorkBench необходимы такие характеристики как плотность, модули упругости по осям, коэффициенты Пуассона, модули сдвига и пределы прочности при растяжении и сжатии.

**Ключевые слова:**

композиционный материал, лопасть, статическая прочность, ANSYS WorkBench, напряжение, критерий разрушения, теории Цая-Хилла, углеткань, стеклопластик, срединные наполнители

*Исследование выполнено при поддержке гранта Минобрнауки Республики Бурятия (соглашение № 413 от 21.12.2023 г.).*

**Введение**

Композиционные материалы имеют преимущество над традиционными материалами (металлами и сплавами) в области авиации - выигрыш в весе, низкая чувствительность к повреждениям, высокая жесткость, высокие механические характеристики. При этом определение уязвимых мест в слоистой конструкции является сложной задачей и на практике решается с помощью разрушающего контроля. Комплексно исследовать слоистую конструкцию позволяют современные программные продукты, такие как, ANSYS WorkBench.

Тема настоящей работы является актуальной, вычислительные мощности современных компьютеров позволяют проводить исследования все более сложных деталей и систем, производить анализ и подбор состава композиционных материалов и конструкций лопасти воздушного винта. В исследовании [1] рассматривается лопасть вертолета, которая имеет отличную от рассматриваемой конструкцию лопасти, в работе [2] рассматривается конструкция лопасти, содержащая в себе лонжерон в виде многослойной стеклопластиковой трубы, и сотовый наполнитель. В исследовании [3] проведены стендовые испытания по определению разрушающей нагрузки, выявлены особенности характера разрушения комлевой части, что подтверждают полученные результаты численного исследования в настоящей работе, приведенные в выводах.

### **Задачи исследования**

1. Провести оценку напряженного-деформированного состояния (НДС) композитной лопасти с классической сборкой: срединный материал – бальза, сверху уложены слои однонаправленный углепластиковый препрег под углом  $0^0$ , углеткань под углом  $-45^0$ , углеткань под углом  $45^0$ .
2. Рассмотреть и рассчитать НДС лопасти с другими типами срединного материала: сосна, осина, а также пенополиуретан.
3. Рассмотреть и рассчитать НДС лопасти с гибридным композиционным материалом: однонаправленный стеклопластиковый препрег под углом  $0^0$ , углеткань под углом  $-45^0$ , углеткань под углом  $45^0$ .
4. Провести оценку запаса прочности по критерию разрушения Цая –Хилла.

### **Материалы и методы**

Исследуемая лопасть БПЛА представляет собой сэндвич конструкцию, внутри которой находится срединный материал - бальза, на верхней части бальзы уложены слои снизу вверх: слой однонаправленного углепластика под углом  $0^0$  и два слоя тканого углепластика под углами  $-45^0$ ,  $45^0$ . На нижней поверхности бальзы укладка выполняется аналогично. Толщина композиционного тканого слоя 0,16 мм, толщина композиционного однонаправленного слоя 0,08. Максимальная длина образца 304,81 мм, ширина образца варьируется от 15,26 мм до 52 мм. Крепление лопасти консольное. На исследуемый образец действуют центробежная сила  $F_{цб}$  и подъемная сила  $F_{под}$ .

Расчет нагрузки проводился следующим образом:

$$F_{цб} = m_l \cdot \omega^2 \cdot R,$$

$$F_{под} = m_{взл} \cdot k,$$

где масса лопасти  $m_l = 0,06$  кг,

угловая скорость лопасти  $\omega = 6000$  об/мин,  
радиус вращения винта  $R = 300$  мм,  
взлетная масса БПЛА  $m_{взл} = 25$  кг,  
коэффициент запаса  $k = 1,5$ .

По итогам расчета получены следующие данные:

$$F_{цб} = 3553,058 \text{ Н};$$
$$F_{под} = 367,5 \text{ Н}.$$

При моделировании были использованы композитные материалы, имеющиеся в библиотеке материалов ANSYS: Epoxy Carbon Woven (230 GPa) Prepreg тканый углепластик в виде препрега – полуфабриката, пропитанной эпоксидной смолой углеткани с модулем Юнга  $E=230$  ГПа и Epoxy Carbon (230 GPa) Prepreg однонаправленный углепластик-препрег, пропитанный эпоксидной смолой с модулем Юнга  $E=230$  ГПа. Характеристики бальзы, сосны, осины и пенополиуретана в библиотеке материалов отсутствуют, поэтому характеристики [4] были добавлены вручную, такие как плотность, модули упругости по осям, коэффициенты Пуассона, модули сдвига и пределы прочности при растяжении и сжатии (табл. 1, 2)

Таблица 1 – Характеристики древесных материалов

| Характеристики<br>срединного<br>материала  | Бальза |                    | Сосна |                    | Осина |                    |
|--|--------|--------------------|-------|--------------------|-------|--------------------|
|  Density                    | 160    | kg m <sup>-3</sup> | 520   | kg m <sup>-3</sup> | 395   | kg m <sup>-3</sup> |
|  Orthotropic Elasticity     |        |                    |       |                    |       |                    |
| Young's Modulus X direction  | 15     | GPa                | 11,9  | GPa                | 12,8  | GPa                |
| Young's Modulus Y direction  | 10     | GPa                | 0,67  | GPa                | 1     | GPa                |
| Young's Modulus Z direction  | 10     | GPa                | 0,55  | GPa                | 1     | GPa                |
| Poisson's Ratio XY   | 0,3    |                    | 0,3   |                    | 0,3   |                    |
| Poisson's Ratio YZ   | 0,3    |                    | 0,3   |                    | 0,3   |                    |
| Poisson's Ratio XZ   | 0,3    |                    | 0,3   |                    | 0,3   |                    |
| Shear Modulus XY   | 1      | GPa                | 1,23  | GPa                | 1     | GPa                |
| Shear Modulus YZ   | 1      | GPa                | 0,76  | GPa                | 0,7   | GPa                |
| Shear Modulus XZ   | 1      | GPa                | 0,5   | GPa                | 0,7   | GPa                |
|  Tensile Yield Strength     | 30     | MPa                | 90    | MPa                | 121   | MPa                |
|  Compressive Yield Strength | 50     | MPa                | 43    | MPa                | 43    | MPa                |

Таблица 2 – Характеристика пенополиуретана

| Характеристики<br>срединного материала   | Пенополиуретан |                    |
|--|----------------|--------------------|
|  Density              | 50             | kg m <sup>-3</sup> |
|  Isotropic Elasticity |                |                    |

|                           |       |     |
|---------------------------|-------|-----|
| Orthotropic Stress Limits |       |     |
| Tensile X direction       | 0,5   | MPa |
| Tensile Y direction       | 0,5   | MPa |
| Tensile Z direction       | 0,8   | MPa |
| Compressive X direction   | -0,05 | MPa |
| Compressive Y direction   | -0,05 | MPa |
| Compressive Z direction   | -0,07 | MPa |
| Shear XY                  | 0,18  | MPa |
| Shear YZ                  | 0,18  | MPa |
| Shear XZ                  | 0,18  | MPa |

В ANSYS Workbench импортируется модель, состоящая из лопасти и бальзы в формате Parasolid, построенная в программном комплексе NX. Импортированная модель представлена на рисунке 1.

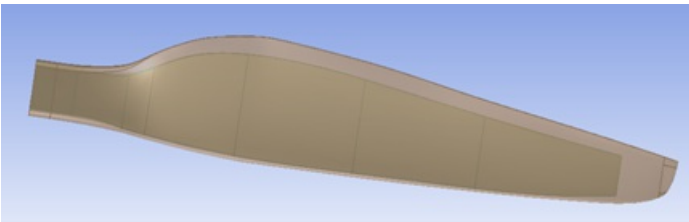


Рисунок 1 – Импортированная модель лопасти и бальзы

Модель разбивается сеточным генератором на сетку конечных элементов, в качестве параметра выбран размер элементов 1 мм.

Задается укладка композиционного материала сверху вниз (Top - Down), слои моделируются по следующим ориентациям по углам укладки: 45<sup>0</sup>, -45<sup>0</sup>, 0<sup>0</sup>(рис. 2).



Рисунок 2 – Схема укладки композитной лопасти

Задаются граничные условия на комлевую часть (из-за симметрии исследуем только половину лопасти) - консольная заделка (запрет перемещений по всем осям). К центру масс прикладывается центробежная сила Fцб. Равномерно задается подъемная сила Fпод по нижней поверхности лопасти по линии ¼ хорд лопасти.

По аналогии смоделированы и исследованы варианты лопасти с другими срединными материалами: сосна, осина, пенополиуретан, и рассмотрен вариант лопасти с однонаправленным стеклопластиковым препрегом вместо однонаправленного углепластикового препрега ориентированного под углом 0<sup>0</sup>.

**Результаты**

Полученные результаты исследования НДС лопасти из углепластика представлены в таблице 3, НДС лопасти из стекло-углепластика в таблице 4, визуализация исследования

НДС лопасти приведена на рисунке 3. Максимальные деформации лопасти приведены в виде гистограммы для сравнительного анализа (рис. 4).

Таблица 3 - Расчет НДС углепластиковой лопасти: полная деформация, деформации по осям X, Y, Z

| Срединный материал | Максимальная деформация, мм | Деформация по оси X, мм | Деформация по оси Y, мм | Деформация по оси Z, мм |
|--------------------|-----------------------------|-------------------------|-------------------------|-------------------------|
| Бальза             | 0,72409                     | 0,18478                 | 0,008113                | 0,087863                |
| Сосна              | 0,83737                     | 0,20401                 | 0,0090682               | 0,096759                |
| Осина              | 0,82147                     | 0,19924                 | 0,0088863               | 0,094065                |
| Пенополиуретан     | 1,7645                      | 0,39324                 | 0,021362                | 0,33716                 |

Таблица 4 - Расчет НДС стекло-углепластиковой лопасти: полная деформация, деформации по осям X, Y, Z

| Срединный материал | Максимальная деформация, мм | Деформация по оси X, мм | Деформация по оси Y, мм | Деформация по оси Z, мм |
|--------------------|-----------------------------|-------------------------|-------------------------|-------------------------|
| Бальза             | 0,81438                     | 0,2139                  | 0,010093                | 0,077528                |
| Сосна              | 0,94891                     | 0,23745                 | 0,01126                 | 0,082537                |
| Осина              | 0,93103                     | 0,23106                 | 0,010965                | 0,080557                |
| Пенополиуретан     | 1,9255                      | 0,47155                 | 0,02847                 | 0,32144                 |

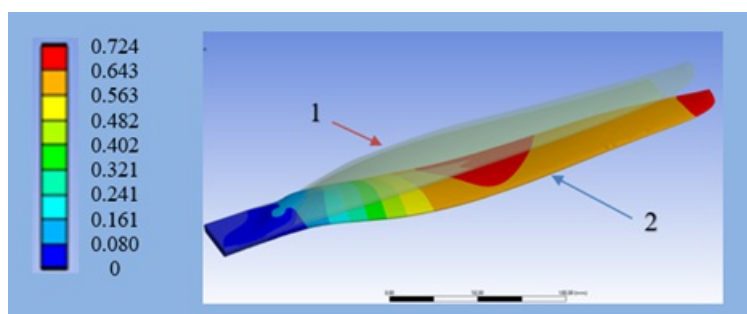


Рисунок 3 - Деформация (мм) углепластиковой лопасти (срединное тело – бальза).

1 - Исходное положение лопасти, 2 - Деформированная лопасть

Существенной особенностью оценки несущей способности композитных конструкций является послойный анализ НДС композитной конструкции, который заключается в анализе НДС каждого слоя. В настоящей работе был произведен анализ послойно: определены максимальные и минимальные напряжения, а также получено отображение уязвимых мест в каждом слое.

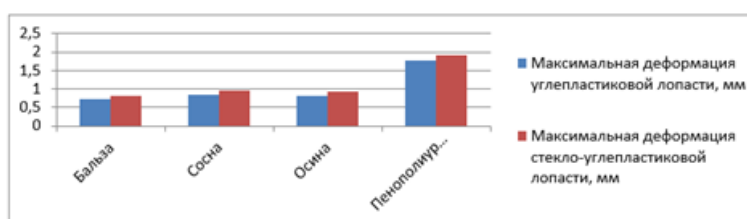


Рисунок 4 - Максимальные деформации лопасти

Полученные результаты послойного исследования для углепластиковой лопасти

представлены в таблице 5 и на рисунках 5-7, для стекло-углепластиковой лопасти в таблице 6. Для удобства использовалась кодировка слоев: слой 1 – слой, ориентированный под углом  $45^0$ , слой 2 – слой, ориентированный под углом  $-45^0$ , слой 3 – слой, ориентированный под углом  $0^0$ .

Таблица 5 – Максимальные и минимальные напряжения послойно для углепластиковой лопасти

| Срединный материал | НДС лопасти, $\sigma$ макс / $\sigma$ мин (Мпа) |                       |                                   |
|--------------------|---|-----------------------|-----------------------------------|
|                    | Слой 1<br>(Углеткань)                           | Слой 2<br>(Углеткань) | Слой 3<br>(Однонапр.углепластик.) |
| Бальза             | 68,88 / -24,365                                 | 67,072 / -61,839      | 266,39 / -28,567                  |
| Сосна              | 71,086 / -26,369                                | 73,13 / -69,699       | 290,04 / -31,634                  |
| Осина              | 70,08 / -26,189                                 | 71,786 / -66,848      | 282,16 / -31,896                  |
| Пенополиуритан     | 123,79 / -43,03                                 | 123,85 / -142,67      | 526,16 / -78,56                   |

Таблица 6 – Максимальные и минимальные напряжения послойно для стекло-углепластиковой лопасти

| Срединный материал | НДС лопасти, $\sigma$ макс / $\sigma$ мин (Мпа) |                       |                                    |
|--------------------|---|-----------------------|------------------------------------|
|                    | Слой 1<br>(Углеткань)                           | Слой 2<br>(Углеткань) | Слой 0<br>(Однонапр.стеклопластик) |
| Бальза             | 81,52 / -25,988                                 | 61,191 / -49,038      | 71,009 / -9,9571                   |
| Сосна              | 85,61 / -28,371                                 | 62,207 / -54,558      | 76,645 / -11,09                    |
| Осина              | 84,009 / -28,149                                | 65,716 / -53,126      | 74,83 / -11,111                    |
| Пенополиуритан     | 145,28 / -45,175                                | 125,4 / -133,25       | 139,31 / -22,375                   |

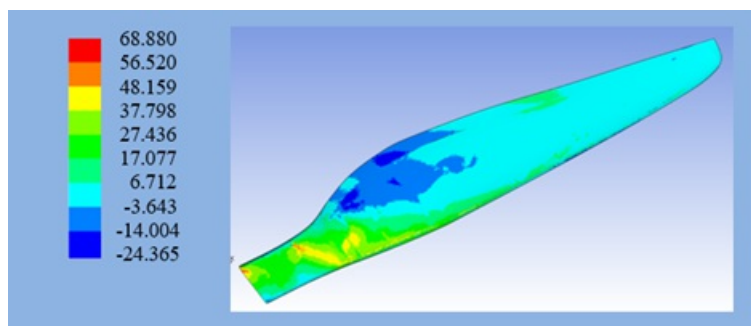


Рисунок 5 – Картина напряжений (МПа) в слое 1 (под углом  $45^0$ ) углепластиковой лопасти (срединное тело – бальза)



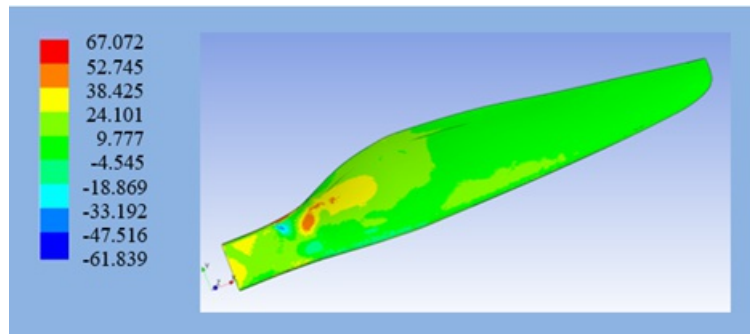


Рисунок 6 – Картина напряжений (МПа) в слое 2 (под углом  $-45^0$ ) углепластиковой лопасти (срединное тело – бальза)

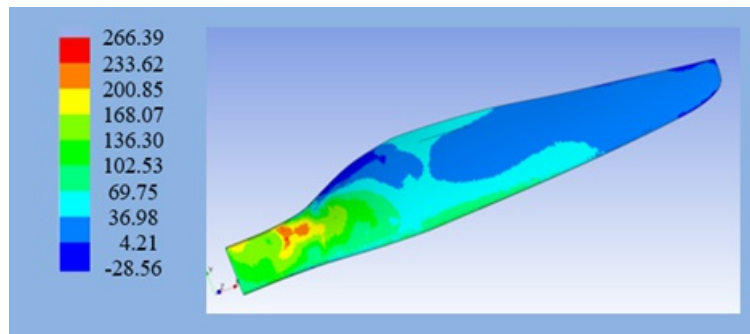


Рисунок 7 – Картина напряжений (МПа) в слое 3 (под углом  $0^0$ ) углепластиковой лопасти (срединное тело – бальза)

Для оценки прочности был использован обратный коэффициент запаса прочности, который показывает во сколько раз действующие напряжения меньше допускаемых [5].

Разрушение возникает при  $IRF > 1$ .

$$IRF = \frac{1}{RF},$$

где  $RF$  - коэффициент запаса прочности, определяется как отношение разрушающей нагрузки к приложенной нагрузке.

Критериев прочности для композитов разработано достаточно много [6,7] и каждый из них отражает те или иные особенности разрушения. Вышеуказанный коэффициент запаса прочности основан на критерии Цая-Хилла, который представляет собой аналитическую аппроксимацию результатов испытаний однонаправленного слоя при различных видах нагружения. Данный критерий имеет следующий вид:

$$\left(\frac{\sigma_1}{\bar{\sigma}_1}\right)^2 - \frac{\sigma_1 \cdot \sigma_2}{\bar{\sigma}_1 \bar{\sigma}_2} + \left(\frac{\sigma_2}{\bar{\sigma}_2}\right)^2 + \left(\frac{\tau_{12}}{\bar{\tau}_{12}}\right)^2 \leq 1$$

где  $\sigma_1$ ,  $\sigma_2$ ,  $\tau_{12}$  – нормальные и касательные напряжения,

$\bar{\sigma}_1$ ,  $\bar{\sigma}_2$ ,  $\bar{\tau}_{12}$  – их предельные значения.

В таблице 7 приведены обратные коэффициенты запаса прочности для углепластиковой лопасти с разными вариантами срединного материала, в таблице 8 приведены обратные коэффициенты запаса прочности для стекло-углепластиковой лопасти с разными вариантами срединного материала (курсивом выделено значение коэффициента,

превышающего допускаяемые напряжения, визуализация этого значения приведена на рисунке 8).

Таблица 7 – Сводная таблица IRF по слоям углепластиковой лопасти с разными вариантами срединных материалов

| Углепластиковая лопасть, срединный материал | Номер самого напряженного слоя | Максимальный коэффициент IRF |
|---|--------------------------------|------------------------------|
| Бальза                                      | 3                              | 0,58                         |
| Сосна                                       | 3                              | 0,65                         |
| Осина                                       | 3                              | 0,63                         |
| Пенополиуретан                              | 3                              | <b>1,38</b>                  |

Таблица 8 – Сводная таблица IRF по слоям стекло-углепластиковой лопасти с разными вариантами срединных материалов

| Стекло-углепластиковая лопасть, срединный материал | Номер самого напряженного слоя | Максимальный коэффициент IRF |
|--|--------------------------------|------------------------------|
| Бальза   | 1                              | 0,13                         |
| Сосна  | 1                              | 0,14                         |
| Осина  | 1                              | 0,15                         |
| Пенополиуретан                                     | 1                              | 0,38                         |

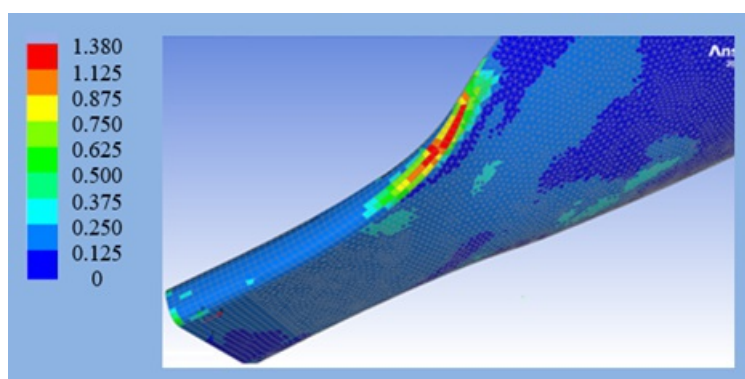


Рисунок 8 – Углепластиковая лопасть (срединное тело – пенополиуретан).

IRF в самом напряженном слое углепластика (слой 3)

### Выводы

1. Проведена оценка НДС композитной лопасти с классической сборкой: срединный материал – бальза, сверху уложены слои однонаправленного углепластикового препрега под углом  $0^0$ , углеткань под углом  $-45^0$ , углеткань под углом  $45^0$ .

Срединный материал – бальза, является оптимальным вариантом по показателям максимальных деформаций, деформаций по осям  $x$ ,  $y$ ,  $z$ .

2. Рассчитано НДС лопасти с другими типами срединного материала: сосна, осина, пенополиуретан.

Среди рассматриваемых срединных материалов наихудшим вариантом является

пенополиуретан. Показатели по максимальной деформации лопасти с пенополиуретаном превышают аналогичный показатель лопасти с бальзой в 2,4 раза.

Разница по деформациям между лопастью с бальзой и лопастью с осиной или сосной в качестве срединного материала, составляет не более 15 %, что является удовлетворительным результатом, учитывая стоимость и труднодоступность материала бальзы, в отличие от распространенных сосны и осины.

3 . Рассчитано НДС лопасти с гибридным композиционным материалом: однонаправленный стеклопластиковый препрег под углом  $0^0$ , углеткань под углом  $-45^0$ , углеткань под углом  $45^0$ .

Композиционный материал послойно работает и на растяжение. и на сжатие. Оптимальным вариантом срединного материала является бальза, однако ухудшение механических характеристик (напряжения при растяжении, сжатии) по слоям для материалов сосна и осина не превышает 20%, что является удовлетворительным результатом, учитывая стоимость и труднодоступность материала бальзы, в отличие от распространенных сосны и осины.

4. Проведена оценка запаса прочности всех вариантов лопасти.

Анализ самых напряженных участков послойно выявил, что прочность лопасти не обеспечивается при варианте: углепластиковая лопасть, срединный материал – пенополиуретан. В слое под углом  $00$  возникает участок с коэффициентом  $IRF = 1,38$ , что превышает допускаемое напряжение на 38%.

Нагрузка, вызвавшая разрушение волокон в каком-либо слое композитной конструкции, считается предельной для всей конструкции. Это позволяет определить максимально возможную несущую способность исследуемой композитной конструкции. Таким образом, оценка НДС конструкции из слоистого КМ базируется на оценке НДС однонаправленного слоя. Необходимо учесть, что полученные результаты моделирования нуждаются в экспериментальной проверке.

Исследование выполнено при поддержке гранта Минобрнауки Республики Бурятия (соглашение № 413 от 21.12.2023 г.).

## **Библиография**

1. Фэн, Г. Исследование статической прочности композитной лопасти вертолета / Г. Фэн, А. М. Думанский, А. Н. Русланцев // Деформирование и разрушение композиционных материалов и конструкций: Труды Второй международной конференции, Москва, 18–20 октября 2016 года. – С. 196-198.
2. Сидоров, И. Н. Расчет напряженно-деформированного и предельного состояний композитной лопасти несущего винта вертолета при различных режимах полета с учетом повреждений в комлевом сечении / И. Н. Сидоров, А. В. Горелов, Е. И. Николаев // XI Всероссийский съезд по фундаментальным проблемам теоретической и прикладной механики: сборник докладов, Казань, 20–24 августа 2015 года. – С. 3453-3457.
3. Бурцев, Б. Н. Особенности напряженно-деформированного состояния комлевой части лопасти несущего винта / Б. Н. Бурцев, Н. П. Тютюнников // Механика композиционных материалов и конструкций. – 2012. – Т. 18, № 4. – С. 552-561.
4. Глебов, И. Т. Физика древесины: учебное пособие / И. Т. Глебов. — Екатеринбург : УГЛУ, 2018. — 80 с.
5. Первушин, Ю. С., Жернаков, В. С. Основы механики, проектирования и технологии изготовления изделий

из слоистых композиционных материалов: учеб. пособие / Ю. С. Первушин, В. С. Жернаков; Уфимск. гос. авиац. техн. ун-т. Уфа. 2008. –303с.

6. Хилл, Р. Теория механических свойств волокнистых композиционных материалов // Механика / Сб. переводов. 1966. № 2. С. 131 - 149.

7. Механика конструкций из композиционных материалов / В. В. Васильев. - Москва : Машиностроение, 1988. - 271 с.

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Статья посвящена исследованию напряженно-деформированного состояния (НДС) композитной лопасти с различными срединными материалами, такими как бальза, сосна, осина и пенополиуретан, с использованием программного продукта ANSYS WorkBench. Это важная и актуальная тема в области материаловедения и авиационной инженерии, поскольку композитные материалы становятся все более востребованными благодаря своим преимуществам перед традиционными материалами.

Авторы провели численное моделирование НДС лопасти с разными типами срединных материалов, используя методы конечных элементов в ANSYS WorkBench. В работе подробно описан процесс моделирования, включая геометрические параметры, материалы и условия нагружения. Это позволяет воспроизвести исследование и проверять его результаты на практике.

Актуальность работы обусловлена возрастающим использованием композитных материалов в авиационной промышленности. Композиты обеспечивают значительное уменьшение веса конструкций при сохранении или улучшении их механических характеристик. В этом контексте исследование НДС композитной лопасти имеет большое значение для разработки и оптимизации авиационных компонентов.

Научная новизна статьи заключается в комплексном подходе к исследованию НДС композитной лопасти с разными типами срединных материалов и гибридными композитами. Авторы впервые проводят детализированный анализ влияния различных срединных материалов на НДС лопасти, что позволяет определить оптимальные сочетания для достижения наилучших характеристик прочности и жесткости.

Статья написана в научном стиле, логично структурирована и последовательно излагает материал. Введение четко определяет цели и задачи исследования. Основная часть содержит подробное описание методологии и результатов, подкрепленное иллюстрациями и таблицами, что облегчает восприятие и анализ данных. Выводы обоснованы и подтверждены результатами численного моделирования.

Исследование показало, что наиболее оптимальным срединным материалом для композитной лопасти является бальза, обеспечивающая наилучшие показатели НДС. Лопасты с пенополиуретаном показали худшие результаты, что делает этот материал менее предпочтительным. Анализ гибридных композитов также выявил значительные преимущества использования бальзы по сравнению с сосной и осиной.

Статья представляет интерес для ученых и инженеров, работающих в области материаловедения, авиационной и машиностроительной промышленности. Исследование содержит ценные данные и методики, которые могут быть использованы для дальнейших исследований и практического применения в разработке и оптимизации композитных конструкций.

Статья является важным вкладом в область исследования композитных материалов и их применения в авиации. Она содержит значительные теоретические и практические результаты, которые могут быть полезны для научного сообщества и индустрии.

Для дальнейшего развития данной работы можно предложить несколько направлений. В первую очередь следует провести лабораторные испытания исследованных композитных лопастей для подтверждения численных данных. Экспериментальные результаты позволят проверить адекватность используемой модели и уточнить ее параметры. Кроме того, важно рассмотреть возможность использования других видов композитных материалов, таких как углеродно-кевларовые и углеродно-базальтовые композиты. Сравнение их характеристик с уже исследованными материалами может выявить дополнительные преимущества.

Анализ динамического поведения композитных лопастей при воздействии различных нагрузок, включая вибрационные и ударные нагрузки, также является важным направлением. Это поможет определить долговечность и надежность лопастей в реальных эксплуатационных условиях. Следует применить методы оптимизации для улучшения конструкции лопасти, что может включать изменение формы, размера и расположения слоев композита для достижения наилучших показателей прочности и жесткости при минимальном весе.

Не менее значимым является изучение поведения композитных лопастей при различных условиях эксплуатации, таких как экстремальные температуры, влажность и воздействие агрессивных сред. Это поможет оценить долговечность и устойчивость материалов в различных климатических условиях. Важно также провести моделирование распространения трещин и повреждений в композитных лопастях. Это позволит разработать методы раннего выявления и предотвращения разрушений, что повысит надежность и безопасность эксплуатации.

Экономический анализ применения различных композитных материалов, учитывающий стоимость материалов, технологии производства и эксплуатационные расходы, также необходим. Это поможет выбрать оптимальное решение с точки зрения затрат и эффективности. На основе полученных данных и анализа следует разработать рекомендации по производству композитных лопастей, включая выбор материалов, методы укладки и технологические параметры. Это поможет улучшить качество и снизить стоимость производства.

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Малахов С.В., Якупов Д.О. Исследование стохастических моделей генерации пакетов в компьютерных сетях //

Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70340 EDN:

EKXYBU URL: [https://nbpublish.com/library\\_read\\_article.php?id=70340](https://nbpublish.com/library_read_article.php?id=70340)

## Исследование стохастических моделей генерации пакетов в компьютерных сетях

**Малахов Сергей Валерьевич**

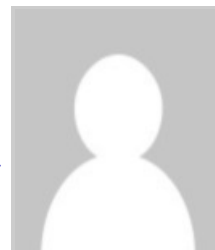
ORCID: 0009-0001-8666-6713

кандидат технических наук

доцент, кафедра управления в технических системах; Поволжский государственный университет телекоммуникаций и информатики

443090, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [s.malakhov@psuti.ru](mailto:s.malakhov@psuti.ru)



**Якупов Денис Олегович**

ORCID: 0009-0003-2371-0822

ассистент, кафедра программной инженерии; Поволжский государственный университет телекоммуникаций и информатики

443090, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [d.yakupov@psuti.ru](mailto:d.yakupov@psuti.ru)



---

[Статья из рубрики "Телекоммуникационные системы и компьютерные сети"](#)

### DOI:

10.7256/2454-0714.2024.2.70340

### EDN:

EKXYBU

### Дата направления статьи в редакцию:

02-04-2024

### Дата публикации:

02-06-2024

**Аннотация:** Стохастические модели генерации пакетов – это модели, которые используются для генерации трафика в компьютерных сетях с определенными

характеристиками. Эти модели могут быть использованы для симуляции сетевой активности и тестирования производительности сети. Стандартная передача данных в сети – это генерация пакетов с задержками, при котором пакеты отправляются через определенные промежутки времени. Для генерации пакетов с задержками могут использоваться различные стохастические модели, включая равномерное распределение, экспоненциальное распределение, распределение Эрланга. В данной работе была собрана экспериментальная установка и разработано клиент-серверное приложение для проведения исследования и анализа производительности канала передачи данных. Был предложен алгоритм, позволяющий восстанавливать моментные характеристики случайной величины интервала между пакетами для дальнейшего использования моделей массового обслуживания. Выполнен анализ законов распределения на производительность экспериментального образца сети и получены оценки эффективности использования канала и среднего времени генерации пакетов в сегментах сети, а также гистограммы задержек по законам распределения. Была создана экспериментальная установка, разработано клиент-серверное приложение для анализа производительности канала передачи данных. Предложен алгоритм восстановления моментных характеристик временных интервалов между пакетами. Проведен анализ законов распределения на производительность сети, получены оценки эффективности использования канала и среднего времени генерации пакетов в сегментах сети, также гистограммы задержек по законам распределения. Генерация пакетов с задержками по стохастическим законам распределения (равномерное, экспоненциальное, Эрланга) имеет большое значение при моделировании и анализе работы сетевых систем. Также генерация пакетов с задержками по вышеупомянутым законам распределения позволяет проводить тестирование и отладку сетевых приложений и устройств в условиях, близких к реальным. Это позволяет выявлять возможные проблемы и улучшать работу сетевых систем. В результате эксперимента был предложен алгоритм, позволяющий восстанавливать моментные характеристики случайной величины интервала между пакетами для дальнейшего использования моделей массового обслуживания. Также, выполнен анализ влияния законов распределения на производительность экспериментального образца сети и получены оценки эффективности использования канала и среднего времени генерации пакетов в сегментах сети, а также гистограммы задержек по законам распределения.

**Ключевые слова:**

равномерное распределение, экспоненциальное распределение, распределение Эрланга, пакетная коммутация, задержки, клиент-серверное приложение, анализ трафика, передача данных, генерации пакетов, Стохастические модели

**Введение**

Генерация пакетов – это процесс создания и отправки пакетов данных через сеть. Это важная функция, которая используется в современных компьютерных сетях для передачи информации между устройствами. Генерация пакетов осуществляется программным обеспечением, которое позволяет создавать и отправлять пакеты данных с заданными параметрами.

Генерация пакетов в компьютерных сетях позволяет решать следующие задачи:

- тестирование сети: генерация пакетов может помочь проверить работоспособность

- сети и выявить проблемы, такие как низкая пропускная способность и задержки;
- • тестирование безопасности: генерация пакетов может помочь проверить защиту сети от вредоносных программ, хакерских атак и несанкционированного доступа к данным;
- • тестирование производительности: генерация пакетов может помочь оценить производительность сети и определить ее пропускную способность;
- • тестирование новых приложений: генерация пакетов может помочь проверить работу новых приложений в сети и выявить возможные проблемы;
- • тестирование настройки сети: генерация пакетов может помочь проверить правильность настройки сети и выявить возможные ошибки в конфигурации.

Назначение генерации пакетов состоит в том, чтобы обеспечить эффективную передачу данных в сети. Это достигается путем разбиения больших объемов данных на меньшие блоки, которые могут быть переданы через сеть более быстро и надежно. Кроме того, генерация пакетов позволяет управлять потоком данных и контролировать качество обслуживания в сети.

Преимущества генерации пакетов включают высокую скорость передачи данных, эффективное использование сетевых ресурсов и возможность обеспечения надежности и безопасности передачи данных. Генерация пакетов также позволяет оптимизировать сетевую инфраструктуру и улучшить производительность сетевых приложений.

Начальный момент и эксцесс – это две основные характеристики, которые могут использоваться для описания распределения трафика в сети [\[1\]](#).

Начальный момент – это числовая характеристика, которая описывает распределение данных относительно их среднего значения. Начальный момент первого порядка равен нулю, а начальный момент второго порядка равен дисперсии случайной величины.

В контексте сетевого трафика начальный момент может использоваться для описания распределения объема или длительности пакетов в трафике. Например, начальный момент второго порядка может использоваться для определения дисперсии размеров пакетов в трафике.

Эксцесс – это числовая характеристика, которая описывает форму распределения данных. Эксцесс может быть положительным, отрицательным или равным нулю. Положительный эксцесс означает, что распределение имеет более тяжелые хвосты и более высокий пик, чем нормальное распределение. Отрицательный эксцесс означает, что распределение имеет более легкие хвосты и более низкий пик, чем нормальное распределение. Эксцесс равный нулю означает, что распределение имеет форму нормального распределения.

В контексте сетевого трафика эксцесс может использоваться для описания формы распределения размеров пакетов в трафике. Например, положительный эксцесс может указывать на наличие тяжелых хвостов в распределении размеров пакетов, что может означать наличие крупных пакетов или необычных событий в трафике. Отрицательный эксцесс может указывать на наличие легких хвостов в распределении размеров пакетов, что может означать отсутствие крупных пакетов и однородность трафика.

Кроме того, начальный момент и эксцесс могут быть использованы для сравнения различных типов трафика или различных сетевых устройств. Например, анализ начального момента и эксцесса может помочь в сравнении трафика, проходящего через различные маршрутизаторы или переключатели. Это может помочь выявить проблемы с производительностью сетевых устройств или определить оптимальную конфигурацию



сети.

#### **Постановка проблемы**

В современном мире компьютерные сети играют ключевую роль в передаче и обработке информации. Одной из важных задач при проектировании и оптимизации таких сетей является моделирование трафика, в частности, генерации пакетов данных. Стохастические модели генерации пакетов являются одними из наиболее часто используемых подходов для решения этой задачи.

Однако, существующие модели не всегда адекватно отражают реальные условия передачи данных в компьютерных сетях, что приводит к неточностям при прогнозировании и оптимизации параметров сети.

Целью данной работы является исследование стохастических моделей генерации пакетов в компьютерных сетях, основанных на равномерном, экспоненциальном и распределении Эрланга. Будут проведены теоретический и экспериментальный анализ этих моделей, а также сравнение их точности и эффективности при моделировании реального трафика.

Равномерное распределение – это распределение вероятности, при котором все значения в заданном диапазоне имеют одинаковую вероятность появления. Другими словами, каждое значение в этом диапазоне имеет одинаковую вероятность быть выбранным.

Например, если генерируется случайное число от 1 до 6 с равномерным распределением, то каждое число от 1 до 6 будет иметь вероятность  $1/6$  быть выбранным.

Равномерное распределение широко используется в различных областях, включая статистику, физику, экономику, инженерию, компьютерные науки и многие другие. Оно является одним из простейших и наиболее понятных видов распределения вероятности, и поэтому часто используется в качестве модели случайного процесса.

Равномерное распределение может быть дискретным или непрерывным. В случае непрерывного равномерного распределения все значения в определенном интервале равновероятны и распределение задается формулой  $f(x) = 1/(b-a)$ , где  $a$  и  $b$  – концы интервала, а  $f(x)$  – плотность вероятности в точке  $x$ .

Равномерное распределение также может использоваться для создания псевдослучайных чисел, которые используются в компьютерных приложениях для генерации случайных последовательностей.

Равномерное распределение имеет несколько свойств, которые могут быть полезны при его анализе или использовании в моделировании. Некоторые из этих свойств включают:

- математическое ожидание равномерного распределения равно среднему значению всех возможных значений в заданном диапазоне. Для непрерывного равномерного распределения это будет равно  $(a+b)/2$ , где  $a$  и  $b$  – концы интервала;
- дисперсия равномерного распределения определяется формулой  $(b-a)^2/12$  для непрерывного равномерного распределения;
- гистограмма равномерного распределения будет иметь прямоугольную форму, если диапазон значений равномерно разделен на несколько интервалов.

Равномерное распределение также может быть использовано для решения различных

задач, таких как определение вероятности того, что случайная величина попадет в определенный диапазон значений, или для нахождения оптимальных значений в задачах оптимизации.

Для генерации пакетов с использованием равномерного распределения необходимо выполнить следующие шаги:

1. Определить интервал, в котором будет генерироваться равномерно распределенное время между последовательными пакетами.
2. Используя генератор случайных чисел, сгенерировать случайные значения с равномерным распределением в заданном интервале. Эти случайные значения будут представлять время между последовательными пакетами, отправляемыми в сеть.
3. Повторять шаг 2, чтобы сгенерировать необходимое количество пакетов.
4. Важно учитывать, что время между отправкой пакетов должно быть больше, чем время передачи пакета, чтобы избежать коллизий и утечек пакетов в сети.

Экспоненциальное распределение – это статистическое распределение вероятностей, которое описывает время между двумя последовательными событиями в процессе Пуассона.

Например, время между получением двух пакетов данных в компьютерных сетях может быть описано экспоненциальным распределением.

Основные свойства экспоненциального распределения:

- оно имеет единственный параметр, который называется параметром масштаба (обычно обозначается как  $\lambda$ );
- функция плотности вероятности экспоненциального распределения имеет форму  $f(x) = \lambda * \exp(-\lambda x)$ , где  $x$  – время между двумя событиями. Эта функция показывает вероятность того, что время между двумя событиями не превысит заданное значение  $x$ ;
- экспоненциальное распределение является распределением без памяти, то есть вероятность наступления события не зависит от того, сколько времени прошло с момента предыдущего события;
- среднее время между двумя событиями в экспоненциальном распределении, т.е. математическое ожидание равно  $1/\lambda$ , а дисперсия равна  $1/\lambda^2$ ;

Экспоненциальное распределение обладает свойством отсутствия памяти. Это означает, что вероятность наступления события в любой момент времени не зависит от того, сколько времени прошло с момента предыдущего события. Например, если среднее время между двумя звонками в call-центре равно 5 минутам, то вероятность того, что следующий звонок произойдет в течение следующей минуты, равна  $1/5$ , независимо от того, сколько времени прошло с момента предыдущего звонка.

Стоит отметить, что экспоненциальное распределение является одним из основных распределений, используемых в статистическом анализе данных, и его свойства могут быть использованы для оценки параметров других распределений и для проверки статистических гипотез.

Для генерации пакетов с использованием экспоненциального распределения необходимо выполнить следующие шаги:

1. Определить параметр  $\lambda$  экспоненциального распределения, который описывает

среднее время между двумя последовательными событиями.

2. Используя генератор случайных чисел [\[2\]](#), сгенерировать случайные значения с экспоненциальным распределением с параметром  $\lambda$ . Эти случайные значения будут представлять время между последовательными пакетами, отправляемыми в сеть.
3. Повторять шаг 2, чтобы сгенерировать необходимое количество пакетов.
4. Важно учитывать, что время между отправкой пакетов должно быть больше, чем время передачи пакета, чтобы избежать коллизий и утечек пакетов в сети.

Распределение Эрланга – это математическая модель, используемая для описания времени между двумя последовательными событиями в процессе Пуассона [\[3\]](#).

Процесс Пуассона – это стохастический процесс, в котором случайные события происходят независимо друг от друга с постоянной интенсивностью.

Распределение Эрланга определяет вероятность того, что количество событий, произошедших в фиксированном временном интервале, будет равно заданному значению. Данное распределение может быть использовано для моделирования времени между вызовами в центре обработки вызовов, времени между прибытием клиентов в магазине или времени между двумя последовательными срабатываниями определенного события в производственном процессе.

В общем случае распределение Эрланга может быть представлено как сумма  $k$  независимых случайных величин, каждая из которых имеет экспоненциальное распределение с параметром  $\lambda$ .

Распределение Эрланга является особой формой гамма-распределения.

Параметр  $k$  в распределении Эрланга называется формой распределения, а параметр  $\lambda$  – интенсивностью.

Математическое ожидание распределения Эрланга равно  $k/\lambda$ , а дисперсия равна  $k/\lambda^2$ .

Существует несколько методов для вычисления вероятностей для распределения Эрланга, включая использование таблиц и специальных программных пакетов.

Если процесс Пуассона является неоднородным, то можно использовать распределение Эрланга с переменной интенсивностью. В этом случае параметр  $\lambda$  будет зависеть от времени.

Если количество событий, которые могут произойти в фиксированном временном интервале, не ограничено, то вместо распределения Эрланга может быть использовано распределение Вейбулла или распределение Парето.

Для генерации пакетов с использованием распределения Эрланга необходимо выполнить следующие шаги:

1. Определить форму и интенсивность распределения Эрланга, то есть параметры  $k$  и  $\lambda$ .
2. Сгенерировать  $k$  независимых случайных величин с экспоненциальным распределением, используя генератор случайных чисел.
3. Сложить сгенерированные значения, чтобы получить время между  $k$  событиями в процессе Пуассона с интенсивностью  $\lambda$ .
4. Это время будет являться временем между отправкой последовательных пакетов в сеть.

5. Повторять шаги 2-4, чтобы сгенерировать необходимое количество пакетов.
6. Важно учитывать, что время между отправкой пакетов должно быть больше, чем время передачи пакета, чтобы избежать коллизий и утечек пакетов в сети.

**Методология и условия исследования**

Была собрана экспериментальная установка, состоящая из клиента и сервера, на котором было установлено программное обеспечение для анализа входящего трафика Wireshark [\[4\]](#) (рис. 1)

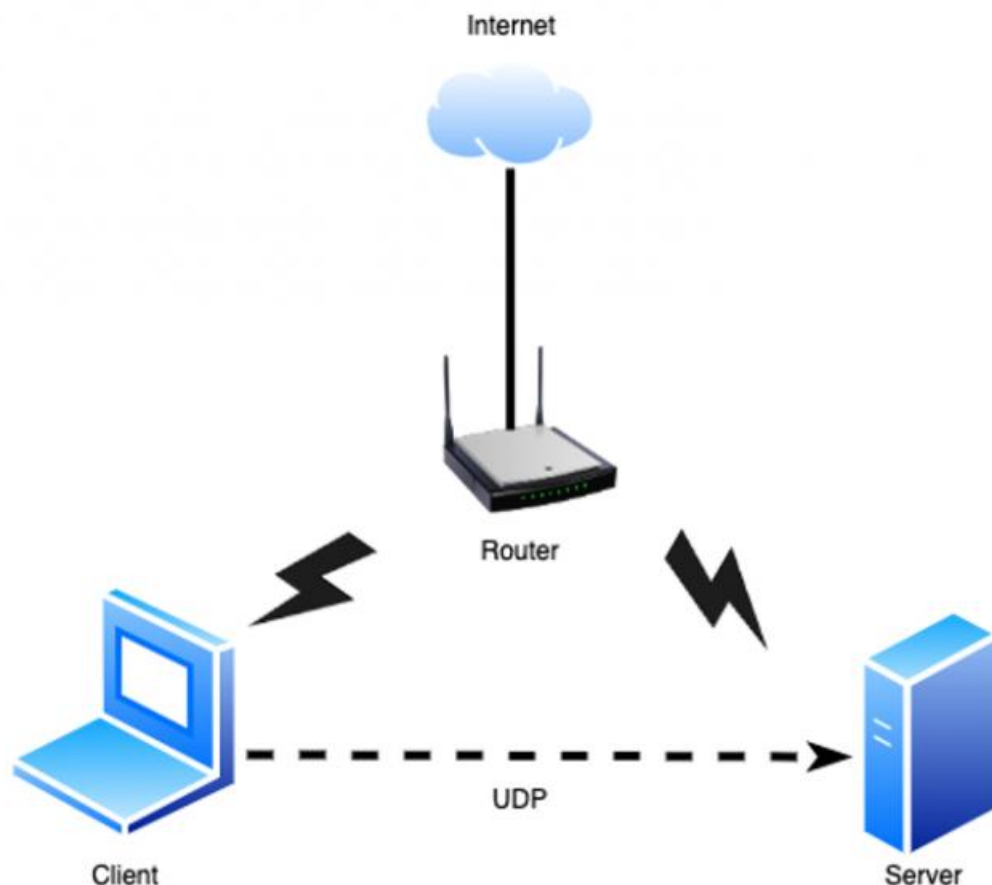


Рис. 1 – Экспериментальная установка

Экспериментальная установка состоит из:

- клиента – Lenovo Ideapad 3;
- сервера – HP Pavilion G6;
- маршрутизатора – ASUS RT-AX53U;
- пропускная способность канала 100 Мбит/с.

Разработанное клиент-серверное приложение написано на языке программирования Java. Объем программы: 774 Кбайт. Установка не требуется [\[5\]](#).

Версия операционной системы для клиентского и серверного приложений: MS Windows 7 – 11, Mac OS X, Linux

Для запуска приложения на клиенте и на сервере требуется установленная на любой операционной системе (Windows 7 – 11, Mac OS X, Linux) реализация спецификации платформы Java, включающая в себя компилятор и библиотеки классов (Jdk-17).

Клиент генерирует пакеты данных на сервер (рис. 2). Сервер получает пакеты (рис. 3) и анализирует получаемый трафик с помощью программы Wireshark.

The screenshot shows a window titled "Traffic Generator" with a standard macOS-style title bar (red, yellow, green buttons). The window contains several input fields and buttons. At the top, there are four buttons: "На..." (disabled), "Дублиров..." (disabled), "Очист..." (disabled), and "Запустить" (highlighted in blue). Below these are the following fields:

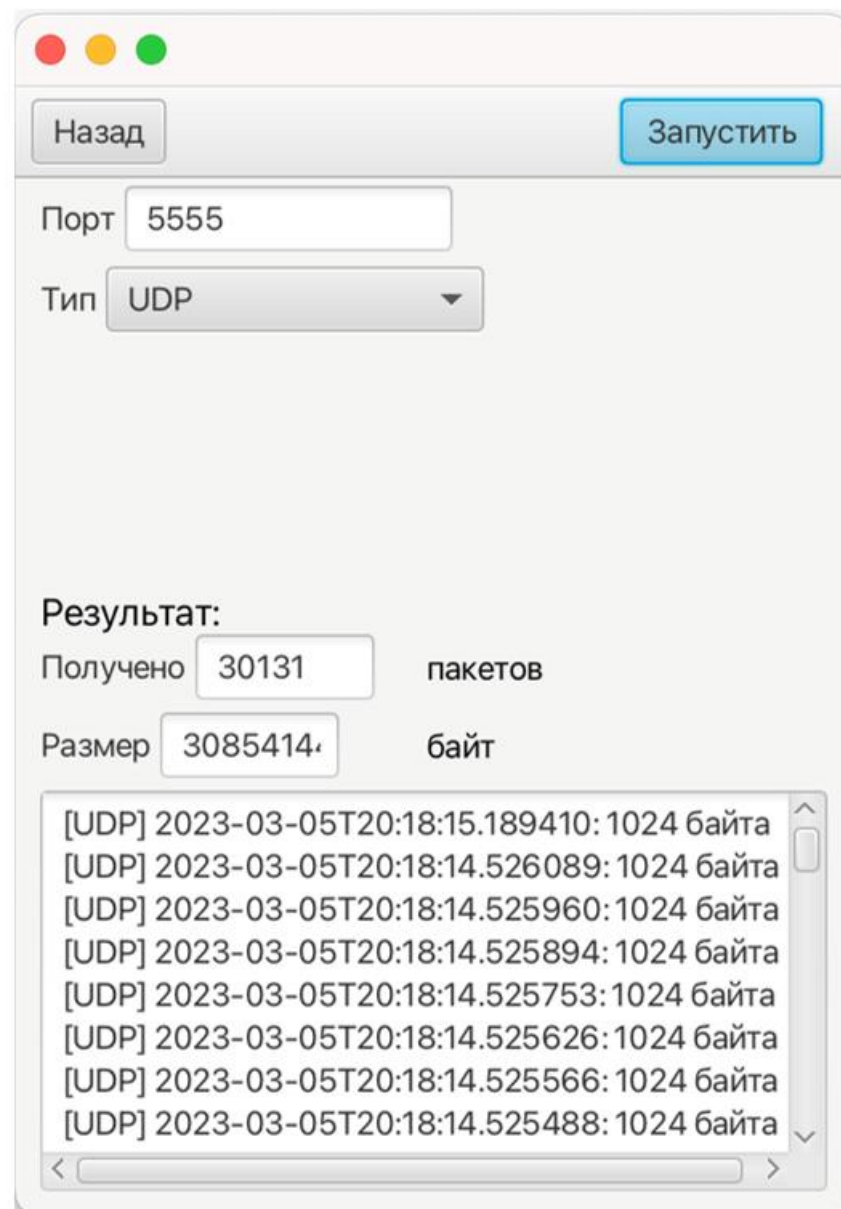
- Адрес: 192.168.0.102:5555
- Размер: 1024 байт
- Время генерации: 85299 мс
- Всего передать: 100 байт
- Закон распредел.: EXPONENTIAL (dropdown menu)
- Значение: 0.00001
- Тип: UDP (dropdown menu)

Below these fields, the text "Результат:" is displayed. Underneath it are two more fields:

- Отправлено: 168783 пакетов
- Объем: 17283379 байт

Рис. 2 – Генерация пакетов клиента с Экспоненциальным распределением

В данном окне указываются следующие параметры генерации: Адрес и порт сервера, Размер пакета, Время генерации пакетов, Размер передаваемого файла, Закон распределения, Параметры закона распределения, Тип трафика.



Назад Запустить

Порт 5555

Тип UDP

Результат:

Получено 30131 пакетов

Размер 3085414 байт

[UDP] 2023-03-05T20:18:15.189410: 1024 байта  
[UDP] 2023-03-05T20:18:14.526089: 1024 байта  
[UDP] 2023-03-05T20:18:14.525960: 1024 байта  
[UDP] 2023-03-05T20:18:14.525894: 1024 байта  
[UDP] 2023-03-05T20:18:14.525753: 1024 байта  
[UDP] 2023-03-05T20:18:14.525626: 1024 байта  
[UDP] 2023-03-05T20:18:14.525566: 1024 байта  
[UDP] 2023-03-05T20:18:14.525488: 1024 байта

Рис. 3 – Принятие пакетов сервера

В данном окне происходит настройка сервера и указываются Порт и Тип трафика.

Эксперимент состоит из 3-основных этапов:

1. Выбрать по очереди закон распределения для каждого эксперимента (Эрланга, экспоненциальный, равномерный).
2. В каждом эксперименте установить размер пакета (несколько разных значений). Было выбрано 3 разных размера пакета, в которых размер полезных данных составляет – S (64 байта), M (512 байт), L (1024 байта).
3. Генерация пакетов происходит 120 секунд (2 минуты)

#### Результаты и анализ исследования

С использованием известных формул математической статистики, определяются моментные характеристики временных интервалов. В работе использованы статистики до 3-го порядка, которые позволяют судить о характере распределения интервалов. [\[5,6\]](#).

Среднее значение интервала между соседними пакетами (1) вычисляется для оценки частотности пакетов в сети. Это позволяет определить, с какой регулярностью пакеты передаются по сети. Оценивая этот параметр, можно управлять потоком данных, что важно для поддержания качества услуг в сети.

$$\bar{\tau} = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k) \quad (1)$$

где – моменты времени поступления пакетов;

N – количество анализируемых интервалов.

Выборочная дисперсия (2) – это статистический параметр, показывающий насколько велико отклонение каждого элемента выборки от её среднего значения. Этот показатель отражает степень разброса данных в выборке. Данный параметр полезен при анализе данных, чтобы понять, насколько данные изменяются или различаются.

$$D_B = \overline{t^2} - \bar{\tau}^2 \quad (2)$$

где – второй начальный момент.

Второй начальный момент (3) своего рода статистической величины – это математическое ожидание квадрата этой величины. С его помощью можно оценить разброс наблюдаемых значений относительно нуля. Это важная характеристика при анализе случайных величин, особенно при использовании статистического моделирования и прогнозирования.

$$\overline{t^2} = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k)^2 \quad (3)$$

Коэффициент вариации (4) – это статистический показатель, который определяет относительное изменение или разброс наблюдений в данной выборке. Чем больше коэффициент вариации, тем больше разброс данных в выборке. Используется для сравнения уровня изменчивости различных данных.

$$c = \sigma_B \sqrt{\tau} \quad (4)$$

где – стандартное отклонение

Асимметрия (5) – это статистический показатель, который характеризует степень отклонения распределения выборки от симметричного. Это позволяет оценить асимметрию распределения данных. Если асимметрия положительна, то распределение смещено вправо, если отрицательна – влево. Асимметрия равна нулю для симметричного распределения. Расчет асимметрии особенно важен при анализе данных для правильного выбора статистического подхода для анализа.

$$A_s = \left(\overline{t^3} - 3\overline{t^2} \cdot \overline{t} + 2\overline{t}^3\right)/\sigma_B^3$$

(5)

где – третий начальный момент

Третий начальный момент (6) – это математическое ожидание куба случайной величины. Он используется для расчета коэффициента асимметрии, который характеризует степень отклонения распределения данных от симметричного. Данный параметр нужен для понимания формы распределения набора данных. Он дает информацию о том, является ли распределение набора данных скошенным и, если да, в какую сторону. Это важный параметр при выборе статистического подхода к анализу данных.

$$\overline{t^3} = \frac{1}{N} \sum_{k=0}^N (t_{k+1} - t_k)^3$$

(6)

Для расчета моментных характеристик данные из лог-файлов из Wireshark были импортированы в MS Excel. Будут учитываться пакеты, направляемые от клиента на сервер. В таблице 1 представлены данные разработанного клиент-серверного приложения, а в таблице 2 – данные программы Netem [\[7\]](#).

Таблица 1. Средние моментные характеристики интервала между пакетами по законам распределения с помощью разработанного клиент-серверного приложения

| Закон<br>распределения | Равномерный | Экспоненциальный | Эрланга |
|------------------------|-------------|------------------|---------|
|                        | 2,1E-04     | 2,0E-04          | 1,4E-04 |
|                        | 1,1E-07     | 2,3E-07          | 2,1E-07 |
|                        | 4,1E-09     | 6,2E-09          | 6,8E-09 |
|                        | 1,3E-07     | 1,9E-07          | 1,8E-07 |
|                        | 2,2005      | 2,2192           | 3,0606  |
|                        | ~0          | >0               | >0      |
| Время(с)               | 120         |                  |         |

Таблица 2. Средние моментные характеристики интервала между пакетами по законам распределения с помощью программы Netem

| Закон<br>распределения | Равномерный | Экспоненциальный | Эрланга |
|------------------------|-------------|------------------|---------|
|                        | 1,52E-03    | 2,19E-03         | 4,5E-02 |
|                        | 0,8E-06     | 2,9E-07          | 2,4E-06 |
|                        | 4,4E-09     | 5,1E-09          | 7,1E-09 |
|                        | 0,9E-06     | 3,3E-07          | 1,9E-06 |
|                        | 1,7955      | 2,4991           | 2,8375  |
|                        | ~0          | >0               | >0      |
| Время(с)               | 120         |                  |         |

Исходя из расчета средних моментных характеристик интервала между пакетами по законам распределения с помощью разработанного клиент-серверного приложения и



программы Netem можно сделать вывод, что разработанное приложение более точно генерирует пакеты с задержками по законам распределения. Разница точности  $\sim 14\%$  в пользу разработанного клиент-серверного приложения.

Коэффициент вариации показывает отличие трафика от пуассоновского потока и совместно с асимметрией позволяет судить о степени весомости хвостов распределений интервалов между пакетами

Второй начальный момент показывает дисперсию случайной величины, то есть меру разброса значений вокруг среднего значения. Чем больше значение второго начального момента, тем больше разброс значений и тем менее предсказуемым является процесс генерации пакетов. Наиболее меньший разброс наблюдается в равномерном распределении.

Третий начальный момент случайной величины показывает среднее значение ее куба. Если 3 начальный момент не равен нулю, то это говорит о том, что вероятности для значений, находящихся на разных расстояниях от среднего значения, не симметричны. Для равномерного распределения третий начальный момент самый низкий.

Асимметрия случайной величины показывает, насколько ее распределение отличается от симметричного. Если асимметрия равна нулю, то распределение симметрично. Если асимметрия больше нуля, то распределение скошено вправо (больше значений находится справа от среднего значения) и распределение имеет более тяжелый правый хвост. Для равномерного распределения асимметрия практически равна 0, для распределения Эрланга и Экспоненциального больше 0.

Если коэффициент вариации больше нуля, то это говорит о том, что разброс значений случайной величины относительно ее среднего значения велик. Таким образом, чем больше значение коэффициента вариации, тем выше уровень изменчивости случайной величины. В равномерном распределении коэффициент вариации самый маленький [8,9].

По полученным задержкам были построены гистограммы распределений.

При распределении Эрланга (рис. 4) значение задержки максимально на интервалах 336-430 мс. С увеличением интервалов задержка уменьшается.

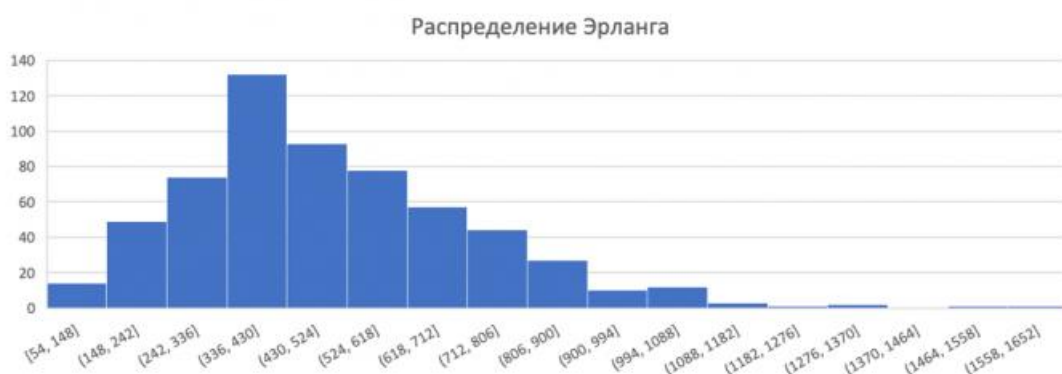


Рис. 4 – Полученная гистограмма распределения Эрланга

При Экспоненциальном распределении (рис. 5) задержки максимальные на коротких интервалах времени.

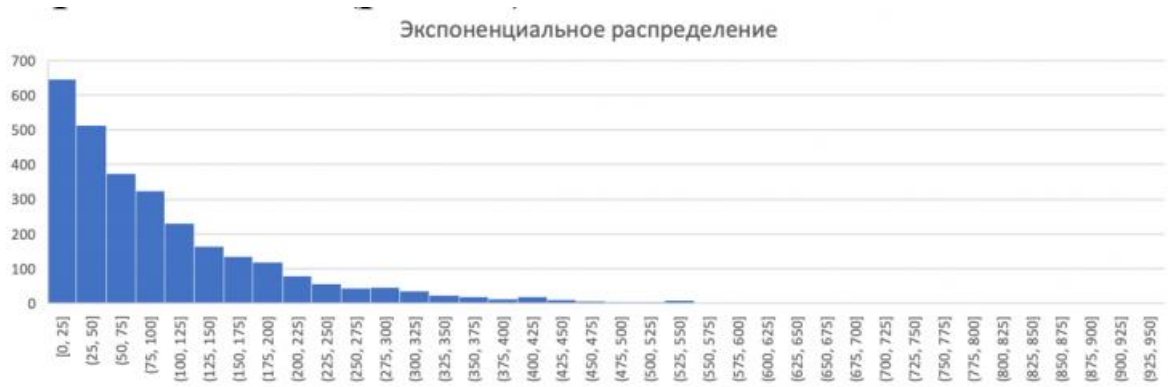


Рис. 5 – Полученная гистограмма Экспоненциального распределения

При Равномерном распределении (рис. 6) задержки для разных интервалов составляют 300 мс. Что значительно хуже из вышеперечисленных.

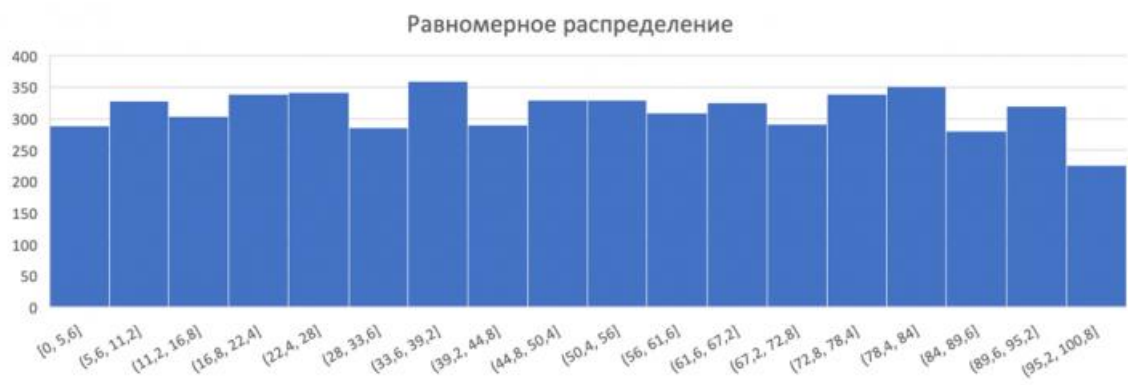


Рис. 6 – Полученная гистограмма Равномерного распределения

Объем, переданный за 2 минуты по каждому закону распределения включал в себя как полезные данные, так и служебные. В зависимости от размера пакета эффективность полезных данных менялась [\[10\]](#) (рис. 7). Максимальная эффективность достигалась при больших размерах пакетов.



Рис. 7 – Эффективность полезных данных (S – 64 байта, M – 512 байт, L – 1024 байта)

В ходе эксперимента при пересылке данных происходили мелкие потери пакетов, в среднем менее 50 пакетов ( $<0.01\%$ ).

В процессе генерации пакетов от клиента, входящий трафик на сервере нагружал канал. В зависимости от размера пакета и закона распределения, нагрузка имела разные значения (рис. 8). При распределении Эрланга загруженность канала оказалась самой большой, что говорит о большей эффективности данной модели.



Рис. 8 – Загруженность канала по законам распределения (S – 64 байта, M – 512 байт, L – 1024 байта)

Считается, что для загруженных систем Ethernet и Fast Ethernet хорошим значением показателя использования сети является 30%. Это значение соответствует отсутствию длительных простоев в работе сети и обеспечивает достаточный запас в случае пикового повышения нагрузки [\[11\]](#). Самым эффективным по загруженности канала законом стал трафик, генерируемый по распределению Эрланга, с размером пакета полезных данных 1024 байта (29,5%).

Этот же результат подтверждает среднее время генерации пакета. В зависимости от размера пакета и закона распределения, время имело разные значения (рис. 9) [\[12\]](#). Модель с распределением Эрланга показала наименьшее среднее время генерации пакетов.



Рис. 9 – Среднее время генерации пакетов по законам распределения (S – 64 байта, M – 512 байт, L – 1024 байта)

## Вывод

Генерация пакетов с задержками по стохастическим законам распределения (равномерное, экспоненциальное, Эрланга) имеет большое значение при моделировании и анализе работы сетевых систем. Также генерация пакетов с задержками по вышеупомянутым законам распределения позволяет проводить тестирование и отладку сетевых приложений и устройств в условиях, близких к реальным. Это позволяет выявлять возможные проблемы и улучшать работу сетевых систем. В результате эксперимента был предложен алгоритм, позволяющий восстанавливать моментные характеристики случайной величины интервала между пакетами для дальнейшего использования моделей массового обслуживания. Также, выполнен анализ влияния законов распределения на производительность экспериментального образца сети и получены оценки эффективности использования канала и среднего времени генерации пакетов в сегментах сети, а также гистограммы задержек по законам распределения. Наибольшая эффективность полезных данных наблюдается при размере пакета 1056 байт (1024 байт полезных данных). Наибольшая производительность канала наблюдается при использовании распределения Эрланга, с размером полезных данных 512 байт и 1024 байт. Также, данный факт подтверждает среднее время генерации одного пакета, которое показывает при распределении Эрланга наибольшую производительность

## Библиография

1. Жукова Г.Н. Карта коэффициентов асимметрии и эксцесса в преподавании теории вероятностей и математической статистики// Концепт 2015. №8 С. 1-4.
2. Дмитриев Е.И., Медведев А.В. Р-генератор случайных чисел, распределенных по экспоненциальному закону// Актуальные проблемы авиации и космонавтики 2011. №7., том 1 С. 316-317.
3. Распределение Эрланга URL:

<http://algolist.ru/maths/matstat/erlang/index.php#:~:text=%D0%A0%BC>. (Дата обращения 06.03.2023).

4. Как пользоваться Wireshark для анализа трафика URL: <https://losst.pro/kak-polzovatsya-wireshark-dlya-analiza-trafika> (Дата обращения 06.03.2023).

5. Приложение для генерации пакетов в компьютерных сетях с помощью стохастических моделей распределения. URL: <https://elibrary.ru/item.asp?id=50133060>

6. Тарасов В.Н., Бахарева Н.Ф., Горелов Г.А., Малахов С.В. Анализ входящего трафика на уровне трех моментов распределений временных интервалов// Информационные технологии 2014. №9 С. 54-59.

7. Эмуляция влияния глобальных сетей URL: <https://habr.com/ru/articles/24046/> (Дата обращения 10.05.2023).

8. Руководство по настройке производительности URL: [http://www.regatta.cs.msu.su/doc/usr/share/man/info/ru\\_RU/a\\_doc\\_lib/aixbman/prftungd/2365c91.htm](http://www.regatta.cs.msu.su/doc/usr/share/man/info/ru_RU/a_doc_lib/aixbman/prftungd/2365c91.htm) (Дата обращения 10.05.2023).

9. Алгоритмы сети Ethernet/Fast Ethernet URL: [https://intuit.ru/studies/professional\\_retraining/943/courses/57/lecture/1690?page=2](https://intuit.ru/studies/professional_retraining/943/courses/57/lecture/1690?page=2) (Дата обращения 10.05.2023).

10. Снабжение пакетов данных точными временными метками в системах сетевого мониторинга URL: <http://www.treatface.ru/solutions/sistemy-setevogo-monitoringa/snabzhenie-paketov-dannykh-tochnymi-vremennymi-metkami-v-sistemakh-setevogo-monitoringa> (Дата обращения 10.05.2023)

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Предмет исследования. Статья, исходя из названия, должна быть посвящена исследованию стохастических моделей генерации пакетов в компьютерных сетях. Содержание статьи соответствует заявленной теме.

Методология исследования включает в себя использование таких методов исследования как анализ и синтез данных. Также ценно, что автор активно применяет графический инструментарий для демонстрации полученных научных результатов.

Актуальность исследования вопросов, связанных с применением стохастических моделей генерации пакетов в компьютерных сетях, не вызывает сомнения. Решение существующих проблем в данном направлении позволит значительно ускорить технологическую модернизацию российской экономики, тем самым обеспечить вклад в достижение национальных целей развития Российской Федерации на период до 2030 года, определённых в Указе Президента России от 07.05.2024 года.

Научная новизна в представленном на рецензирование материале присутствует. В частности, она может быть связана с представленной на рисунке 1 схемой экспериментальной установки (при условии, что это авторская разработка).

Стиль, структура, содержание. Стиль изложения научный. Структура статьи автором не выстроена. Рекомендуется при доработке статьи сформировать следующие структурные элементы: «Введение», «Постановка проблемы», «Методология и условия исследования», «Результаты исследования», «Обсуждение результатов исследования»,

«Выводы и дальнейшие направления исследования». Ознакомление с содержанием позволяет сделать вывод о том, что автор немного сбивчиво излагает полученные результаты (что, вероятнее всего, обусловлено отсутствием чётко выраженной структуры исследования). Также автору рекомендуется добавить анализ каждого графического объекта, т.к. в текущей редакции в статье содержатся только отсылки констатационного формата («на рисунке представлено то-то»). Читателя интересуют выводы, формируемые автором по каждому графическому объекту. Представляется выигрышным добавление в текст статьи таблицы, содержащей список ключевых авторских идей и проблем, решаемых при реализации данных рекомендаций. Более того, автору необходимо внимательно вычитать статью на предмет орфографических и пунктуационных ошибок как в тексте статьи, так и в списке литературы.

Библиография. Сформированный автором библиографический список состоит из 10 наименований, из которых только 2 представляют собой научные публикации. При проведении доработки статьи автору рекомендуется расширить количество изученных источников, в т.ч. добавив иностранные научные публикации.

Апелляция к оппонентам. При проведении доработки статьи и расширении числа источников, являющихся научными публикациями, автору следует сравнить полученные научные результаты с итогами исследований, проведённых другими авторами в России и за рубежом. Это позволит также усилить научную новизну исследования и привлечь дополнительную читательскую аудиторию.

Выводы, интерес читательской аудитории. С учётом всего вышеизложенного, заключаем о том, что автором выбранная тема исследования, научные статьи по которой будут востребованы у читательской аудитории. При этом статья требует доработки по указанным в тексте рецензии замечаниям.

## **Результаты процедуры повторного рецензирования статьи**

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Рецензируемая статья посвящена исследованию стохастических моделей процесса создания и отправки пакетов данных через компьютерные сети.

Методология исследования базируется на использовании собранной авторами публикации экспериментальной установки, состоящей из клиента и сервера, на котором было установлено программное обеспечение языке программирования Java для анализа входящего трафика Wireshark.

Актуальность работы авторы связывают с тем, что генерация пакетов – важная функция, которая используется в современных компьютерных сетях для передачи информации между устройствами путем разбиения больших объемов данных на меньшие блоки, которые могут быть переданы через сеть более быстро и надежно, кроме этого появляется возможность управлять потоком данных и контролировать качество обслуживания в сети.

Научная новизна рецензируемого исследования: предложен алгоритм, позволяющий восстанавливать моментные характеристики случайной величины интервала между пакетами для дальнейшего использования моделей массового обслуживания; получены оценки эффективности использования канала и среднего времени генерации пакетов в сегментах сети, а также гистограммы задержек по различным законам распределения;

выявлено, что наибольшая производительность канала наблюдается при использовании распределения Эрланга.

Структурно в статье выделены следующие разделы: Введение, Постановка проблемы, Методология и условия исследования, Результаты и анализ исследования, Вывод и Библиография.

Авторами выполнен эксперимент по описанной в статье схеме, показаны средние моментные характеристики интервала между пакетами по законам распределения с помощью разработанного клиент-серверного приложения, приведены полученные гистограммы распределения Эрланга, экспоненциального и равномерного распределений, отражена эффективность полезных данных. Отмечено, что наибольшая эффективность полезных данных наблюдается при размере пакета 1056 байт (1024 байт полезных данных), наибольшая производительность канала наблюдается при использовании распределения Эрланга с размером полезных данных 512 байт и 1024 байт.

Библиографический список включает 10 источников – электронные интернет-ресурсы и научные публикации отечественных ученых по рассматриваемой теме. В тексте публикации имеются адресные отсылки к списку литературы, подтверждающие наличие апелляции к оппонентам.

Из недостатков публикации, требующих своего устранения, стоит отметить, что формулы 1 – 6 не отражаются в тексте статьи, размещенной в информационной системе издательства – это делает невозможным проведение оценки их смыслового содержания. Кроме этого отсутствуют расшифровки условных обозначений на рисунках 7, 8 и 9, а также подписи осей на рисунках 4, 5 и 6.

Рецензируемый материал соответствует направлению журнала «Программные системы и вычислительные методы», отражает результаты проведенного авторского исследования, может вызвать интерес у читателей, может быть рекомендован к опубликованию после доработки в соответствии с высказанным замечанием.

### **Результаты процедуры окончательного рецензирования статьи**

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

В статье исследуются стохастические модели генерации пакетов в компьютерных сетях. Основное внимание уделяется сравнению равномерного, экспоненциального распределений и распределения Эрланга в контексте их точности и эффективности при моделировании реального сетевого трафика. Методология исследования включает теоретический анализ выбранных стохастических моделей и их экспериментальную проверку с использованием клиент-серверной установки. Применение программного обеспечения Wireshark для анализа входящего трафика позволяет детально оценить характеристики генерируемых пакетов. Программное обеспечение, используемое для генерации пакетов, написано на языке Java и тестировалось на различных операционных системах, что обеспечивает широкую применимость результатов исследования. Актуальность исследования обоснована значимостью точного моделирования сетевого трафика для обеспечения стабильности и безопасности компьютерных сетей. В условиях быстро растущего объема данных и усложняющихся сетевых структур, точное моделирование и анализ сетевого трафика являются критически важными для поддержания эффективной работы сетевых инфраструктур. Научная новизна работы заключается в сравнении различных стохастических моделей генерации пакетов и в разработке методики для более точного моделирования



реального сетевого трафика. Авторы приводят новые данные о точности и эффективности различных моделей, что позволяет более объективно оценивать их применение в реальных сетевых условиях. Статья написана в ясном и доступном стиле, с четкой структурой, что облегчает восприятие материала. Введение плавно вводит в тему исследования, подробно объясняя необходимость и значимость генерации пакетов. Основная часть статьи посвящена детальному описанию моделей и экспериментальной установки, а также анализу полученных результатов. Заключение подводит итоги исследования и формулирует основные выводы. Выводы статьи четко сформулированы и обоснованы экспериментальными данными. Авторы приходят к заключению, что модель распределения Эрланга показала наибольшую производительность и точность в моделировании реального сетевого трафика, что подтверждается полученными результатами. Также отмечается, что использование данной модели позволяет улучшить параметры сети и повысить эффективность ее работы. Статья представляет интерес для широкой аудитории, включая исследователей в области компьютерных сетей, разработчиков программного обеспечения и специалистов по сетевой безопасности. Работа способствует углублению понимания процессов генерации пакетов и улучшению методов моделирования сетевого трафика, что имеет важное практическое значение. Статью рекомендую принять к публикации, так как она представляет собой ценный вклад в область исследования стохастических моделей генерации пакетов в компьютерных сетях и обладает высокой научной и практической значимостью.

Рекомендация: междисциплинарные исследования могут существенно обогатить данную работу. Следует изучить возможность применения стохастических моделей генерации пакетов в смежных областях, таких как интернет вещей (IoT), облачные вычисления и большие данные. Сотрудничество с исследователями из других областей для проведения междисциплинарных исследований и обмена опытом будет способствовать развитию новых идей и подходов.

Программные системы и вычислительные методы

Правильная ссылка на статью:

Вальков В.А., Столяров Е.П., Корчагин А.А., Ермашин М.В., Якупов Д.О. Сравнение методов оптимизации скорости чтения/записи накопителей // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70900 EDN: DXCLJH URL: [https://nbpublish.com/library\\_read\\_article.php?id=70900](https://nbpublish.com/library_read_article.php?id=70900)

## Сравнение методов оптимизации скорости чтения/записи накопителей

**Вальков Вадим Александрович**

ORCID: 0009-0002-0365-7756

студент; кафедра Информатика и вычислительная техника; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [BonetyFallen@yandex.ru](mailto:BonetyFallen@yandex.ru)



**Столяров Евгений Павлович**

ORCID: 0009-0003-7877-2369

студент; кафедра Информатика и вычислительная техника; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [evgeniystolyarov1@mail.ru](mailto:evgeniystolyarov1@mail.ru)



**Корчагин Антон Александрович**

ORCID: 0009-0008-8519-1796

студент; кафедра Информатика и вычислительная техника; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [anton\\_kor2004@mail.ru](mailto:anton_kor2004@mail.ru)



**Ермашин Максим Владимирович**

ORCID: 0009-0007-1302-5759

студент; кафедра Информатика и вычислительная техника; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [maks.yer.04@mail.ru](mailto:maks.yer.04@mail.ru)



**Якупов Денис Олегович**

ORCID: 0009-0003-2371-0822

ассистент; кафедра Управление в технических системах; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [d.yakupov@psuti.ru](mailto:d.yakupov@psuti.ru)



[Статья из рубрики "Операционные системы"](#)

**DOI:**

10.7256/2454-0714.2024.2.70900

**EDN:**

DXCLJH

**Дата направления статьи в редакцию:**

26-05-2024

**Дата публикации:**

02-06-2024

**Аннотация:** Объектами данного исследования являются накопители данных различных типов и уровней сложности, а также принципы их работы. Они представляют собой сложные технические системы, включающие в себя множество компонентов и отличающиеся высокой степенью интеграции. Предмет исследования заключается в изучении основных характеристик жестких дисков и твердотельных накопителей. Важное значение имеет их структура, функциональные особенности, принципы работы и способы оптимизации. Целью исследования является определение наиболее эффективных методов оптимизации работы этих устройств. Это включает в себя такие аспекты, как управление памятью, балансировка нагрузки, управление энергопотреблением и другие. Результаты этого исследования могут быть использованы для повышения эффективности работы с данными, улучшения производительности систем хранения данных и создания новых технологий в этой области. В этом исследовании рассматривается производительность различных решений для хранения данных на дисках с помощью ряда тестов, направленных на понимание скорости и зависимости от внешних факторов. Основные выводы проведенного исследования отражают значимость комплексного использования подходов оптимизации для улучшения скорости чтения и записи данных. Оптимизация процессов чтения и записи данных является критически важной для современных высокопроизводительных вычислительных систем, а также для приложений, которые требуют быстрого доступа к большим объемам информации. Усовершенствованные методики, применяемые в ходе исследования, способствуют значительному повышению производительности накопителей данных. Они учитывают специфику работы различных типов устройств хранения данных, включая жесткие диски и твердотельные накопители, и предлагают подходы к оптимизации, которые учитывают их уникальные характеристики. В целом, результаты этого исследования предоставляют ценные взгляды на принципы оптимизации хранения данных, и они могут служить основой для разработки новых стратегий и решений в этой важной области информационных технологий. Это исследование представляет собой значимый вклад в научное понимание оптимизации процессов чтения и записи данных, и его выводы могут иметь долгосрочные последствия для развития технологий хранения данных.

**Ключевые слова:**

Жесткие диски, Твердотельные накопители, Оптимизация, Производительность, Фрагментация, Чтение данных, Дефрагментация, Интерфейс, Буфер кэша, Файловая система

**Введение:**

Для пользователя накопителей, таких как HDD или SSD безусловно важно знать основные характеристики подобных устройств хранения информации, чтобы приобрести нужный товар и увеличить производительность его системы. Для этого необходимо понимать какие характеристики влияют на производительность и как их можно оптимизировать.

Основными операциями для накопителей являются запись и чтение, их разделяют на последовательный и случайный доступ. Последовательный доступ подразумевает обращение к данным последовательно, начиная с начала, в то время как случайный доступ позволяет напрямую обращаться к любой части накопителя без необходимости чтения предшествующих данных. Случайный доступ к файлам – один из ключевых аспектов производительности хранения данных, особенно для дисков, и является одним из наиболее трудоемких операций. Скорость случайного доступа к файлам имеет решающее значение для эффективной работы с данными. Случайные операции чтения, особенно небольшие, нелегко буферизировать или каким-либо образом предварительно загрузить. Последовательные операции могут выполняться по цепочке и достигать максимальной скорости, записи могут кэшироваться в памяти для последующей сортировки. Настоящие случайные чтения не имеют ни одного из этих преимуществ.

В этой статье мы рассмотрим одни из основных характеристик, влияющих на скорость работы накопителей, а также способы их оптимизации.

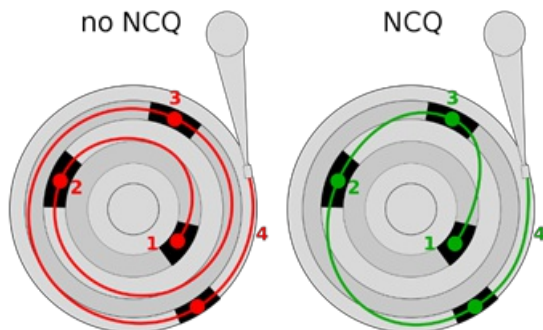
**Факторы, влияющие на скорость чтения и записи данных :**

Существует несколько ключевых факторов, которые влияют на скорость чтения и записи данных с диска. Рассмотрим некоторые из них:

1. Тип накопителя: Существует различные типы накопителей, такие как HDD (жесткие диски) и SSD (твердотельные накопители). SSD обеспечивают более высокую скорость чтения и записи данных по сравнению с HDD из-за отсутствия движущихся частей.
2. Интерфейс подключения: Существуют различные интерфейсы подключения дисков, такие как SATA, NVMe (для SSD) и другие. Более современные интерфейсы, такие как NVMe, обычно обеспечивают более высокую скорость передачи данных.
3. Скорость вращения диска (для HDD): В случае жесткого диска, скорость вращения диска влияет на скорость доступа к данным на диске. Более высокая скорость вращения обычно означает более быстрый доступ к данным.
4. Размер буфера кэша: Буфер кэша на диске используется для временного хранения данных перед их записью на диск или чтением с него. Большой размер буфера кэша может ускорить процессы чтения и записи данных.
5. Файловая система: Выбранная файловая система также может влиять на скорость работы с данными на диске. Некоторые файловые системы более эффективны в управлении файлами и обеспечении быстрого доступа к ним.
6. Фрагментация: Фрагментация диска может замедлить скорость чтения и записи данных, поскольку файлы хранятся в разных фрагментах на диске, что требует дополнительного времени на их объединение при чтении.

Кроме того, физические диски могут использовать встроенную очередь команд (NCQ),

позволяющую жестким дискам внутренне оптимизировать порядок выполнения полученных команд чтения и записи. Это позволяет уменьшить количество ненужных перемещений головки накопителя, что приводит к повышению производительности (и незначительному износу накопителя) при выполнении рабочих нагрузок, требующих выполнения нескольких одновременных запросов на чтение/запись, которые чаще всего возникают в приложениях серверного типа.



NCQ может изменить производительность, но в некоторых случаях плохая реализация может ухудшить ситуацию, особенно при большой нагрузке.

#### **Оптимизация скорости чтения записи:**

Существует множество способов оптимизации скорости чтения и записи данных, можно использовать комплексный подход, включающий сразу несколько из них.

Исходя из факторов, влияющих на производительность носителя, можно выделить некоторые способы оптимизации такие как:

Использование кэша: Кэширование данных может значительно ускорить доступ к часто используемым данным. Это особенно полезно в случае частого доступа к тем же данным, что позволяет избежать повторного чтения с медленного носителя. Параллельное выполнение операций: При наличии нескольких независимых операций чтения/записи можно использовать параллельное выполнение, чтобы уменьшить общее время выполнения. Это может быть достигнуто как с помощью многопоточности, так и с помощью асинхронных операций ввода/вывода. Сюда можно отнести как использование утилит командной строки, так и организацию RAID массивов. Фрагментация данных: Если возможно, данные могут быть фрагментированы и распределены по нескольким носителям, что позволит параллельно выполнять операции чтения/записи с различных устройств, увеличивая общую скорость доступа.

Использование более быстрых носителей: Перенос данных на более быстрые носители, такие как твердотельные накопители (SSD), может значительно ускорить время доступа к данным.

#### **Материалы и задачи**

Перед началом применения методов оптимизации были установлены программы: Defraggler – программа для отслеживания фрагментации данных и выполнения дефрагментации дисков, PrimoCache – программное решение для кэширования данных для ускорения хранения.

Тесты проводились с использованием следующих накопителей:

- HDD WD Blue WD10EZEX
- HDD WD Blue WD10EALX
- SSD WD Blue WDS250G2B0A

Для проведения сравнения методов оптимизации проводились тесты производительности до и после применения методов, затем проводился сравнительный анализ.

Мы поставили для себя следующие задачи: 1)Провести анализ производительности дисков. Для начала нужно посмотреть насколько хорошо накопители выполняют свои функции.

2)Выявить проблемную область. Нужно определить с какими проблемами мы имеем дело и проанализировать примерный план решения.

3)Разработать план по оптимизации работы накопителей. Для улучшения работы существует множество способов. Нам необходимо выявить наиболее эффективные их них.

4)Проверить результативность. Повторно провести анализ производительности и организовать полученный опыт в результаты.

5)Сделать выводы по проведённой работе.

## **Результаты**

Для оценки производительности использовалась программа:

CrystalDiskCrystal (версия 9.0.3) – Программа для тестирования производительности дисков.

### **1. Дефрагментация диска**

Дефрагментация диска - это процесс переупорядочивания файлов на жестком диске с целью улучшения производительности. Когда файл сохраняется на диск, он разбивается на несколько фрагментов и размещается на разных участках диска. Это происходит из-за постоянного чтения и записи файлов, при котором места для новой информации могут располагаться не подряд. Последствиями фрагментации является увеличенное время доступа к файлам из-за необходимости перемещаться по диску для чтения всех фрагментов, что в свою очередь замедляет скорость чтения и записи. Дефрагментация позволяет объединить фрагментированные файлы, переместив их на диск так, чтобы они располагались ближе друг к другу, что сокращает время доступа к данным и повышает производительность.

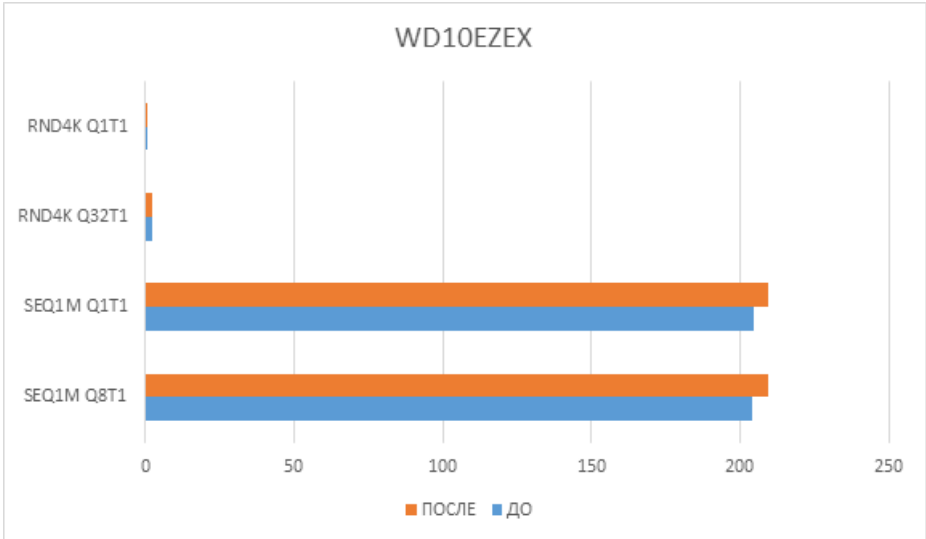


Рис.1 WD10EZEX Reading Test

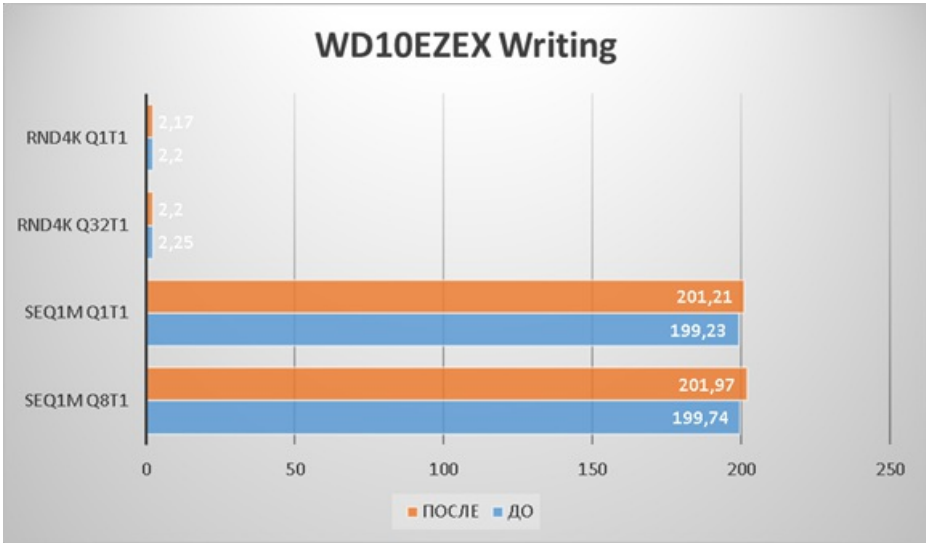


Рис.2 WD10EZEX Writing Test

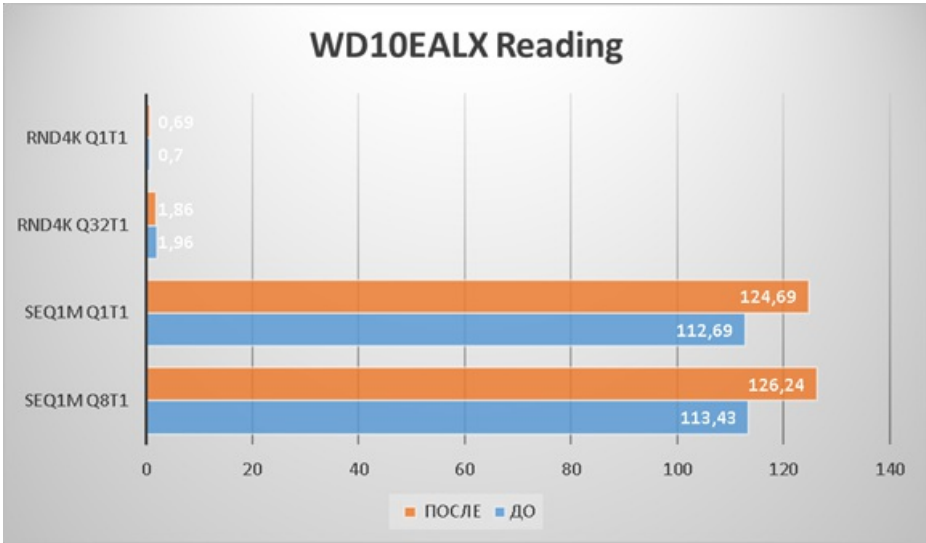


Рис.3 WD10EALX Reading Test

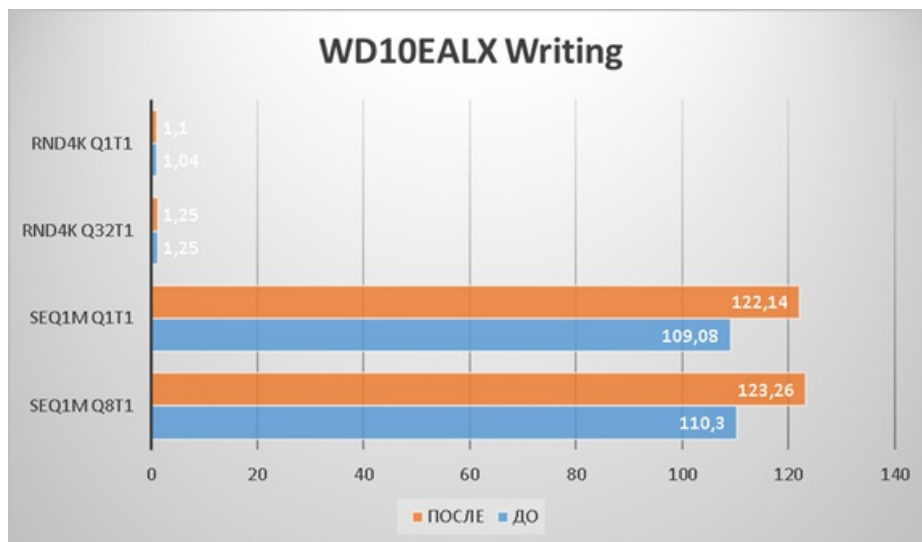


Рис.4 WD10EALX Writing Test

Таким образом, дефрагментация диска способствует улучшению производительности и повышению скорости чтения и записи диска за счет оптимизации расположения файлов на нем.

## 2.Использование твердотельных накопителей (SSD).

Твердотельные накопители SSD (Solid State Drive) обладают рядом преимуществ перед обычными жесткими дисками, что делает их гораздо быстрее в плане скорости чтения и записи данных.

-Отсутствие подвижных частей: в твердотельных накопителях нет движущихся механических элементов, в отличие от жестких дисков, где данные записываются на вращающиеся магнитные пластины. Это позволяет SSD обеспечивать более быстрый доступ к данным без задержек из-за механического движения.

-Быстрый доступ к данным: SSD используют флэш-память NAND для хранения информации, что обеспечивает очень быстрый доступ к данным. Время чтения и записи данных на SSD существенно меньше, чем на жестких дисках.

-Отсутствие фрагментации данных: на SSD данные хранятся в виде блоков в памяти, а сами блоки в свою очередь поделены на страницы. Данные записываются в отдельные страницы блоков, при этом невозможно обновить данные, просто перезаписав старые. Более того, стереть можно только блок целиком. В связи с этим в SSD присутствует сборка мусора - процесс оптимизации, в ходе которого контроллер SSD реорганизовывает данные, чтобы упростить и ускорить процесс записи данных в будущем.

Команда TRIM также способствует оптимизации распределения и использования памяти, позволяющая операционной системе уведомить твердотельный накопитель о том, какие блоки данных (страницы) не несут полезной нагрузки и их можно не хранить физически.

Проверить включен ли TRIM в Windows возможно при помощи команды (0 означает что TRIM включен, 1 - отключен)

```
C:\Users\dedde>fsutil behavior query DisableDeleteNotify
NTFS DisableDeleteNotify = 0 (Отключено)
ReFS DisableDeleteNotify = 0 (Отключено)
```

Рис .5 TRIM



-Поддержка технологий чтения и записи на ячейку: Стандартными типами флэш-памяти NAND являются SLC, MLC, TLC и QLC. Ключевые различия между типами памяти NAND заключаются в стоимости, емкости и сроке службы. Ресурс определяется количеством циклов программирования-стирания (P/E), которые может выдержать ячейка флэш-памяти до износа. Цикл P/E — это процесс стирания и записи ячейки, и чем больше циклов P/E может выдержать технология NAND, тем выше ресурс устройства.

Эти факторы делают твердотельные накопители SSD популярным выбором для повышения производительности и скорости работы компьютеров.

Часто SSD берут в пару к HDD, вместо одного большого SSD можно купить NVMe маленького размера, только для операционной системы, рабочих приложений, а все остальные файлы, дистрибутивы и резервные копии хранить на дешёвом медленном SATA HDD. Хотя так делают скорее для экономии и увеличения объёма хранилища, большим плюсом в подобном подходе является некоторое снижение нагрузки на SSD, увеличение его срока жизни. Кроме того, в более свободных SSD больше размер кэша и выше производительность, чем в заполненных.

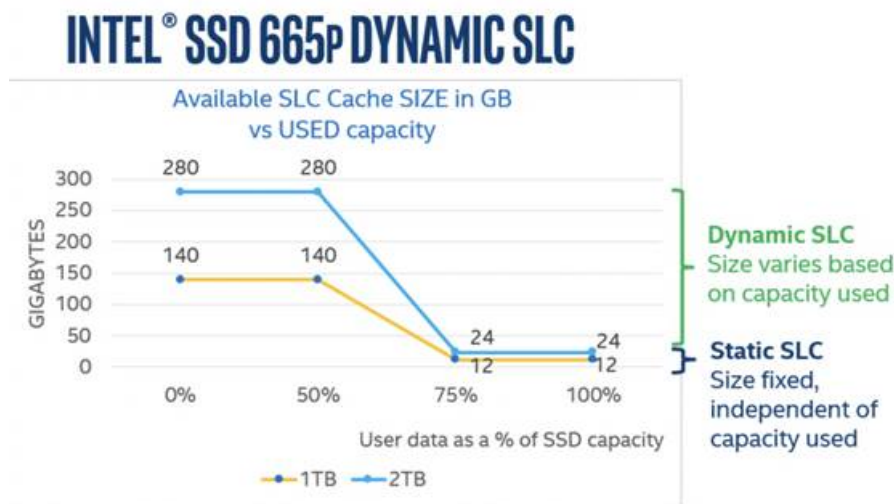


Рис.6 Изменение размера кэша SLC в зависимости от объёма свободного места в Intel SSD 665p

### 3. Настройка кэша

Внешний вид кэша в жестких дисках (HDD) обычно представляет собой небольшой объем быстрой памяти, чаще всего SRAM или DRAM. Кэш используется для временного хранения данных, которые наиболее часто запрашиваются процессором, чтобы ускорить доступ к ним.

В твердотельных накопителях (SSD) кэш обычно представлен как небольшой объем NAND Flash памяти. SSD кэш используется для кэширования данных в целях улучшения производительности и снижения износа NAND ячеек.

Кэш позволяет увеличить скорость доступа. Когда данные часто запрашиваются, кэш позволяет быстрее получить к ним доступ, не прибегая к медленному чтению с основного накопителя. Поскольку кэш хранит данные, наиболее часто запрашиваемые процессором, это способствует более быстрому отклику системы в целом. Также в случае SSD, кэш помогает снизить количество операций записи на основное хранилище, что в свою очередь может продлить срок службы накопителя.

Таким образом кэш напрямую влияет на производительность накопителя

В целях повышения производительности имеет смысл персональная настройка кэша накопителя под нужды пользователя при помощи различных утилит, например PrimoCache.

В качестве теста выставим, использованному ранее WD Blue 10EZEX, «Cache task», выделив 6144 МБ оперативной памяти в качестве L1 кэша, установим размер блока в 512 КБ.

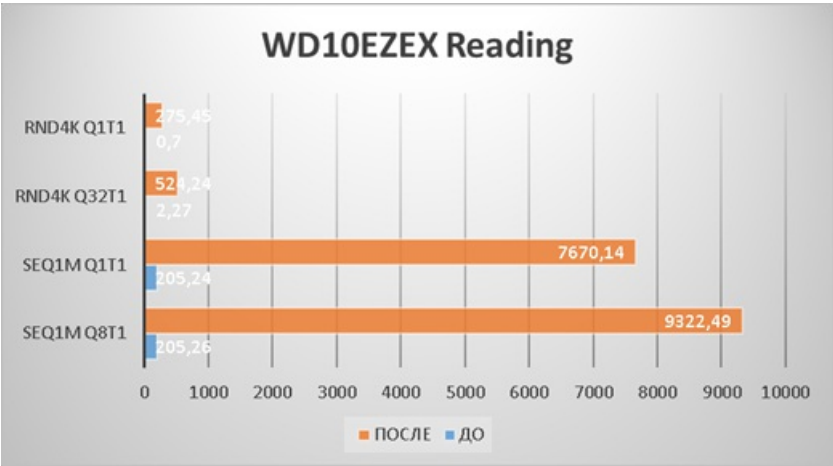


Рис.7 WD10EZEX Reading test PrimoCache

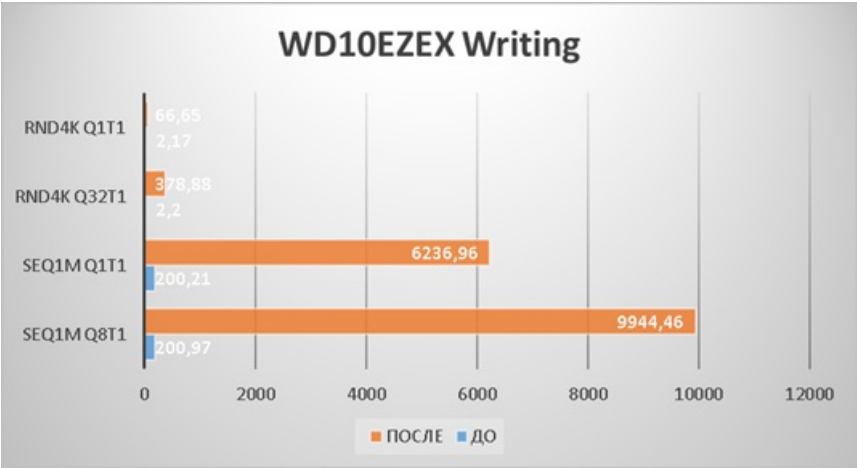


Рис.8 WD10EZEX Writing test PrimoCache

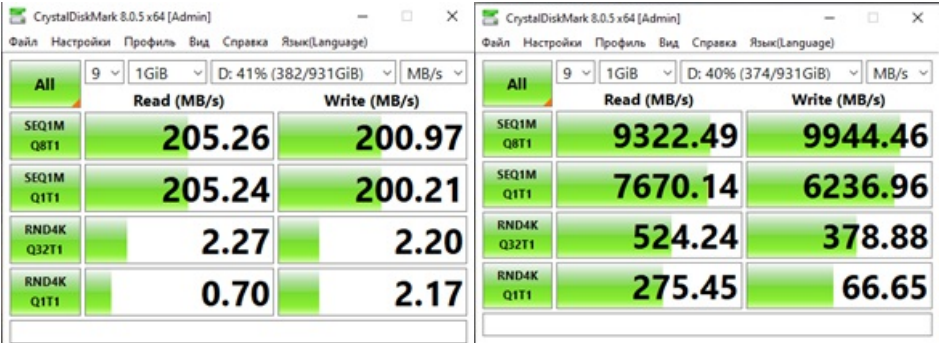


Рис.9 Результат CrystalDiskMark до настройки кэша и после

Из теста видно что скорости последовательного чтения и записи увеличилась примерно в 40 раз, но следует понимать, что такая скорость достигается просто из-за того что данные сначала записываются в выделенную в качестве кэша оперативную память, а затем из оперативной памяти записываются на сам диск со скоростью ограниченной

физическими характеристиками диска. В таком подходе есть свои преимущества как например возможность доступа к данным сразу после записи, копирования, потому что эти данные записаны в оперативной памяти, которая в разы быстрее. Тем не менее, поскольку оперативная память является энергозависимой памятью, если данные ещё не были записаны на диск, а питание было отключено, то данные будут утеряны.

#### 4. Использование утилит командной строки.

Многие пользователи испытывают потребность в копировании, переносе большого объема данных, одним из способов оптимизации работы накопителя в данной ситуации может являться использование утилит командной строки, таких как robocopy в windows или rsync в Linux. Утилиты командной строки предоставляют более гибкие и мощные инструменты для управления и перемещения больших объемов данных. Они обеспечивают более точный контроль над процессом копирования и переноса файлов, что позволяет оптимизировать использование ресурсов и улучшить производительность. Более того, robocopy позволяет копировать и переносить файлы быстрее чем стандартные средства Windows, благодаря более эффективному использованию многопоточности. Количество задействованных потоков можно поменять, воспользовавшись опцией «/mt:».

Ниже приведен лог копирования файла размером 9.19 Гб при помощи robocopy, что заняло 1 минуту 57 секунд.

```
-----
ВсегоСкопировано ПропущеноНесоответствие СБОЙДополнения
Каталогов : 532 532 1 0 0 0
Файлов : 20689 20689 0 0 0 0
Байт : 9.199 g 9.199 g 0 0 0 0
Время : 0:26:51 0:01:57 0:00:00 0:00:10

Скорость : 84040598 Байт/сек.
Скорость : 4808.841 МБ/мин.
Окончание: 15 мая 2024 г. 20:38:48
```

Рис.10 Robocopy

В тоже время копирование того же файла при помощи стандартных средств Windows заняло примерно 2 минуты 26 секунд.

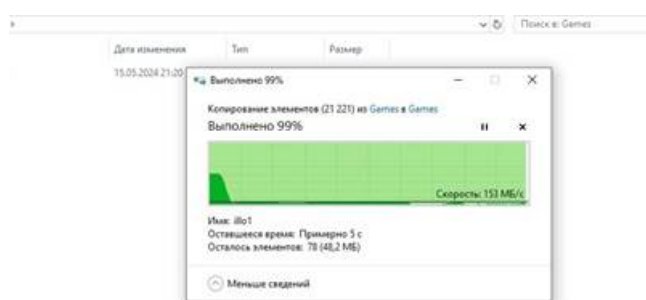


Рис.11 Копирование стандартными средствами Windows

#### 5.Использование RAID

RAID (Redundant Array of Independent Disks) - это технология, которая объединяет несколько физических дисков в единое логическое устройство для повышения производительности, надежности или и то и другое. RAID используется как на серверах, так и на обычных компьютерах для защиты данных и/или увеличения скорости доступа к

ним. Существует множество уровней RAID, каждый из которых обладает своими особенностями, ниже перечислены одни из наиболее популярных: RAID 0 (Стрипирование): Данные разбиваются на блоки и записываются на два или более диска параллельно (стримы). Это увеличивает скорость доступа к данным, так как операции чтения/записи могут выполняться параллельно на всех дисках. Однако в случае выхода из строя одного из дисков, данные на всех остальных дисках теряются, так как информация разбита на части без какой-либо защиты. RAID 1 (Зеркалирование): Данные записываются одновременно на два или более диска. Это обеспечивает резервное копирование данных (зеркалирование), что повышает надежность хранения. Однако объем доступного пространства на дисках будет равен объему одного из них, так как все данные дублируются. RAID 5 (Паритет): Данные записываются на несколько дисков с добавлением контрольной информации (паритетной информации), которая позволяет восстановить данные в случае выхода из строя одного из дисков. RAID 5 обеспечивает баланс между производительностью и надежностью, при этом используется меньше дисков по сравнению с зеркалированием. RAID 6: Похож на RAID 5, но обеспечивает дополнительную защиту от отказа диска путем хранения двух паритетных блоков. Это позволяет системе восстановить данные, даже если два диска выйдут из строя одновременно. RAID 10 (RAID 1+0): Это комбинация RAID 1 и RAID 0. Данные разбиваются на блоки и зеркалируются на другие диски. Таким образом, сочетаются преимущества зеркалирования (высокая надежность) и стрипирования (высокая производительность). Выбор конкретного уровня RAID зависит от требований к производительности, надежности и доступному бюджету. Среди областей применения RAID: системное администрирование; резервное копирование данных; работа с BigData; клиент-серверные модели.

Тем не менее, повышения производительности можно достичь эффективнее всего, используя RAID 10 или RAID 0, нежели RAID 5 или RAID 6. Это связано с тем, что при выполнении RAID 6 или 5 записи на диск — сама рабочая нагрузка хоть и будет представлять собой чистую запись, но отдельные диски тратят на чтение столько же операций, сколько и на запись, что приводит к низкой производительности.

В общем, RAID 0 предпочтительнее использовать в целях повышения производительности, поскольку при его использовании скорости используемых дисков по сути будут суммироваться, однако скорость будет рассчитываться по самому медленному из них.

### **Заключение :**

Таким образом, были рассмотрены, хотя далеко не все существующие методы оптимизации работы накопителей, благодаря рассмотренным подходам к оптимизации и характеристикам, влияющих на производительность, можно добиться улучшения производительности накопителей и системы в целом. Однако следует отметить, что выбор конкретных методов оптимизации будет зависеть от специфики решаемых задач, особенностей используемого оборудования и программного обеспечения.

Наилучшие результаты показывают использование ОЗУ в качестве L1 кэша и применение RAID 0 массивов. Как уже было описано выше, эти методы имеют свои минусы, вероятно главным из них является возможная потеря данных.

Дефрагментация диска, хоть и не даёт такого же прироста, но тоже влияет на производительность диска и системы, автоматическая дефрагментация дисков уже давно стала неотъемлемой частью операционных систем, обеспечивая поддержание

оптимальной производительности хранения данных без необходимости ручного вмешательства пользователя.

Комплексный подход к оптимизации работы накопителей, учитывающий как аппаратные, так и программные решения, позволит добиться значительного повышения производительности современных вычислительных систем и удовлетворить возрастающие требования пользователей к скорости обработки и хранения данных.

## Библиография

1. PMR (Perpendicular Magnetic Recording) and SMR (Shingled Magnetic Recording) hard disk drives [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: [https://kb.synology.com/en-id/DSM/tutorial/PMR\\_SMR\\_hard\\_disk\\_drives](https://kb.synology.com/en-id/DSM/tutorial/PMR_SMR_hard_disk_drives)

2. Черепичная магнитная запись (SMR): что это такое и стоит ли ее избегать?

[Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: [https://interface31.ru/tech\\_it/2022/12/chto-takoe-cherepichnaya-magnitnaya-zapis-smr-i-stoit-li-ee-izbegat.html](https://interface31.ru/tech_it/2022/12/chto-takoe-cherepichnaya-magnitnaya-zapis-smr-i-stoit-li-ee-izbegat.html)

3. Липкин Г. Черепичная магнитная запись SMR: принцип работы, преимущества и недостатки [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: <https://habr.com/ru/articles/494614/>

4. 2021 SSD Benchmark Suite [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: <https://www.anandtech.com/print/16458/2021-ssd-benchmark-suite>

5. CrystalMark [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: <https://crystalmark.info/en/>

6. Defraggler [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: <https://www.ccleaner.com/defraggler?cc-noredirect=>

7. PrimoCache [Электронный ресурс]. Дата обращения: 22 мая 2024 г.

URL: <https://www.romexsoftware.com/en-us/primo-cache/>

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

В данной статье рассматриваются методы оптимизации скорости чтения и записи данных на накопителях различных типов, включая HDD и SSD. Основное внимание уделяется факторам, влияющим на производительность, и способам повышения эффективности работы накопителей. Методология исследования включает анализ основных характеристик накопителей, проведение тестов производительности до и после применения методов оптимизации, и сравнительный анализ полученных результатов. В качестве инструментов использовались программы Defraggler, PrimoCache и CrystalDiskMark. Исследование актуально в свете постоянного увеличения объемов данных и потребности в быстрой и эффективной обработке информации. Оптимизация работы накопителей позволяет существенно улучшить производительность систем, что является важным для различных сфер, от персонального использования до серверных

решений. Научная новизна работы заключается в комплексном подходе к оптимизации накопителей, включающем использование кэша, параллельное выполнение операций, дефрагментацию, применение SSD и RAID-массивов. В статье приводятся практические рекомендации и результаты тестов, подтверждающие эффективность предложенных методов. Статья написана в научном стиле, структура логична и последовательна. Работа состоит из введения, описания факторов, влияющих на скорость работы накопителей, методов оптимизации, результатов тестов и выводов. Каждый раздел четко выделен, что облегчает восприятие информации. Авторы пришли к выводу, что комплексный подход к оптимизации работы накопителей может значительно повысить их производительность. Основные методы, такие как использование кэша, применение SSD и RAID, доказали свою эффективность в рамках проведенных тестов. Статья будет интересна широкому кругу читателей, включая IT-специалистов, системных администраторов, разработчиков и всех, кто заинтересован в улучшении производительности своих компьютерных систем. Практические рекомендации и конкретные результаты тестов делают работу ценной как с теоретической, так и с практической точки зрения.

Рекомендации по доработке

1. В названии статьи допущена грамматическая ошибка. Правильно, полагаю, -- "... оптимизацииИ"
2. Расширить список использованных источников и добавить больше современных исследований по теме.

Статья представляет собой ценный вклад в область оптимизации производительности накопителей. Работа выполнена на высоком уровне. Рекомендую направить статью на доработку с последующим принятием к публикации.

Для дальнейшего развития данной работы рекомендуется расширить круг рассматриваемых методов оптимизации производительности накопителей, включив в исследование такие технологии как использование гибридных систем хранения, сочетающих в себе преимущества HDD и SSD, применение алгоритмов сжатия данных для эффективного использования дискового пространства, а также настройку файловых систем и параметров операционной системы под конкретные задачи. Кроме того, имеет смысл провести более детальный анализ производительности отдельных методов оптимизации в различных сценариях использования, выявить возможные узкие места и ограничения. Для получения более достоверных и воспроизводимых результатов стоит использовать разнообразные тесты производительности и увеличить выборку тестируемых накопителей. Наконец, практическую ценность работы повысит разработка комплексной методологии оптимизации системы хранения данных, учитывающей специфику решаемых задач, особенности имеющегося аппаратного и программного обеспечения, а также требования к производительности, надежности и стоимости решения.

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Мамадаев И.М., Минитаева А.М. Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70658 EDN: LDXXKC URL: [https://nbpublish.com/library\\_read\\_article.php?id=70658](https://nbpublish.com/library_read_article.php?id=70658)

## Оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS

**Мамадаев Ибрагим Магомедович**

Разработчик; ООО "Мэйл.Ру"

-, Россия, г. Москва, бул. Тверской, -

✉ [ibragim.m115@gmail.com](mailto:ibragim.m115@gmail.com)



**Минитаева Алина Мажитовна**

кандидат технических наук

преподаватель; Информатика и Вычислительная Техника (ИУ-6); Московский Государственный Технический Университет им.Н.Э.Баумана

-, Россия, г. Москва, бул. Тверской, -

✉ [aminitaeva@mail.ru](mailto:aminitaeva@mail.ru)



[Статья из рубрики "Операционные системы"](#)

### DOI:

10.7256/2454-0714.2024.2.70658

### EDN:

LDXXKC

### Дата направления статьи в редакцию:

05-05-2024

### Дата публикации:

13-06-2024

**Аннотация:** На сегодняшний день мобильные устройства играют важную роль в

повседневной жизни каждого человека, и одной из ключевых технологий, приводящих к значительным преимуществам для мобильных приложений, является машинное обучение. Оптимизация алгоритмов машинного обучения для мобильных устройств является актуальной и важной задачей, она направлена на разработку и применение методов, которые позволят эффективно использовать ограниченные вычислительные ресурсы мобильных устройств. В работе рассматриваются различные способы оптимизации алгоритмов распознавания изображений на мобильных устройствах, такие как квантизация и сжатие моделей, оптимизация изначальных вычислений. Помимо способов оптимизации самой модели машинного обучения рассматриваются также различные библиотеки и инструменты для применения данной технологии на мобильных устройствах. Каждый из описанных способов имеет свои преимущества и недостатки, в связи с чем в итогах работы предлагается использовать не только комбинацию описанных вариантов, а также дополнительный способ - параллелизацию процессов обработки изображений. В статье рассмотрены примеры конкретных инструментов и фреймворков, доступных для оптимизации производительности машинного обучения на iOS, проведены собственные эксперименты для проверки эффективности различных методов оптимизации. А также приведен анализ полученных результатов и сравнение производительности алгоритмов. Практическая значимость данной статьи заключается в следующем: - Улучшение производительности алгоритмов машинного обучения на мобильных устройствах iOS приведет к более эффективному использованию вычислительных ресурсов и повышению производительности системы, что является весьма важным в контексте ограниченных вычислительных мощностей и энергетических ресурсов мобильных устройств. - Оптимизация производительности машинного обучения на iOS-платформе способствует развитию более быстрых и отзывчивых приложений, что также улучшит пользовательский опыт и позволит разработчикам создавать новые и инновационные функции и возможности. - Расширение применимости машинного обучения на мобильных устройствах iOS открывает новые возможности для разработки приложений в различных областях, таких как распознавание образов, обработка естественного языка, анализ данных и другие.

**Ключевые слова:**

нейронная сеть, машинное обучение, мобильное устройство, iOS, распознавание изображений, оптимизация, ОС Apple, эффективность, производительность, параллелизация

**Введение и актуальность**

На сегодняшний день мобильные устройства играют важную роль в жизни каждого человека, так как предоставляют широкий спектр возможностей и сервисов, без которых многие уже и не могут представить свою повседневную жизнь. Одной из ключевых технологий, приводящих к значительным преимуществам мобильных приложений, является машинное обучение [\[1\]](#), оно уже применяется во многих ведущих приложениях на рынке, а крупные IT-компании конкурируют между собой, пытаясь привлечь на свою сторону большее количество клиентов. Однако, для их эффективного использования на мобильных устройствах, необходимо решить ряд непростых проблем.

Оптимизация алгоритмов машинного обучения для мобильных устройств является актуальной и важной задачей и направлена на разработку и применение методов, которые позволят эффективно использовать ограниченные вычислительные ресурсы



мобильных устройств, минимизировать энергопотребление и достигать высокой производительности при выполнении сложных задач машинного обучения. Проведение оптимизаций открывает новые возможности для развития приложений, таких как умные помощники и голосовые ассистенты, обработка изображений и видео в реальном времени, автоматическая классификация данных.

Вместе с ростом популярности и использования мобильных устройств, компании Apple [\[2\]](#), возникает и потребность в эффективной работе алгоритмов машинного обучения на ограниченных вычислительных ресурсах, малой памяти и недостаточного, ввиду размера, запаса аккумулятора.

Анализируя проблемы, связанные с производительностью машинного обучения на мобильных устройствах под управлением операционной системы iOS, можно выделить следующие аспекты:

- задержки в выполнении алгоритмов из-за их сложности,
- снижение отзывчивости пользовательского интерфейса из-за перегрузки вычислительных мощностей устройства,
- увеличение энергопотребления и как следствие повышение тепловыделения.

Перечисленные аспекты оказывают негативное влияние на пользовательский опыт и ставят перед разработчиками задачи по обеспечению высокой производительности приложения.

Целью данной работы является исследование и оптимизация производительности алгоритмов распознавания изображений на основе машинного обучения на мобильных устройствах iOS. Основная задача состоит в исследовании существующих методов и техник оптимизации; проведении анализа производительности различных алгоритмов машинного обучения; оценке влияния различных факторов на производительность.

В статье рассмотрены примеры конкретных инструментов и фреймворков, доступных для оптимизации производительности машинного обучения на iOS, проведены собственные эксперименты для проверки эффективности различных методов оптимизации. А также приведен анализ полученных результатов и сравнение производительности алгоритмов.

Практическая значимость данной статьи заключается в следующем:

- Улучшение производительности алгоритмов машинного обучения на мобильных устройствах iOS приведет к более эффективному использованию вычислительных ресурсов и повышению производительности системы, что является весьма важным в контексте ограниченных вычислительных мощностей и энергетических ресурсов мобильных устройств.
- Оптимизация производительности машинного обучения на iOS-платформе способствует развитию более быстрых и отзывчивых приложений, что также улучшит пользовательский опыт и позволит разработчикам создавать новые и инновационные функции и возможности.
- Расширение применимости машинного обучения на мобильных устройствах iOS открывает новые возможности для разработки приложений в различных областях, таких как распознавание образов, обработка естественного языка, анализ данных и другие.

На мобильных устройствах iOS довольно часто используются всевозможные вариации алгоритмов машинного обучения для решения разнообразных задач [3]. Некоторые из них включают в себя алгоритмы классификации, регрессии, кластеризации, нейронные сети и глубокое обучение [4].

Яркими примерами таких моделей являются алгоритмы классификации, логистическая регрессия и метод опорных векторов, которые широко применяются для решения задач распознавания образов и классификации данных на мобильных устройствах.

Данные методы обладают относительно низкой сложностью и хорошо масштабируются для работы с большими объемами данных [5]. Алгоритмы регрессии, которые включают в себя линейную регрессию и метод наименьших квадратов, используются для предсказания численных значений на основе исходных данных. Такие алгоритмы широко применяются в задачах прогнозирования и анализа данных на мобильных устройствах. Кластеризация, в свою очередь, является методом группировки схожих объектов на основе их характеристик. Некоторые алгоритмы кластеризации, такие как k-средних и DBSCAN, используются для обработки данных на мобильных устройствах и поиска скрытых структур. Нейронные сети и глубокое обучение также являются одними из самых популярных алгоритмов машинного обучения на сегодняшний день, так как могут обрабатывать сложные данные, изображения, тексты, и при этом достигать высокой точности в задачах классификации, распознавания и генерации контента. Правильное использование методов оптимизации может значительно повысить эффективность работы алгоритмов машинного обучения на мобильных устройствах.

## 2 Квантизация моделей

Одним из ключевых методов оптимизации является квантизация моделей [6], она позволяет уменьшить размер модели и снизить требования к вычислительным ресурсам путем представления весов и активаций с меньшей точностью. Иными словами – квантование это процесс снижения точности весов путем округления, уменьшения точности. Наглядная иллюстрация данного процесса приведена на рисунке 1.



Рисунок 1 – Пример квантования веса 1 нейрона и снижение разрядности в 4 раза

Применение квантизации позволяет ускорить процесс вычислений и снизить использование памяти, несильно влияя на точность модели. Одним из главных преимуществ квантизации является сокращение размера модели, что в свою очередь приводит к уменьшению требований как к памяти устройства, так и непосредственно к самому ускорению вычислений. Кроме того, квантизация позволяет использовать

специализированные аппаратные ускорители, такие как «Neural Engine» [2] в чипах от компании Apple, предназначенных в том числе для эффективного выполнения операций с низкой точностью. Однако у квантизации есть весьма большой недостаток - она может также привести к потере точности модели, особенно при использовании уменьшении точности представления весов и активаций.

Однако этот весомый недостаток перекрывается следующим преимуществом - особенностью данного метода является то, что его можно использовать как во время обучения модели, так и после, что позволяет проводить подобные операции и после поставки приложения пользователям.

### 3 Сжатие моделей

Еще одним методом оптимизации является сжатие моделей, которое позволяет уменьшить размер модели, удаляя ненужные или избыточные параметры. Одним из видов сжатия моделей является «прунинг» [7] (отсечение или прореживание). Графики точности и производительности моделей в зависимости от процента прореживания таким способом приведены на рисунке 2.

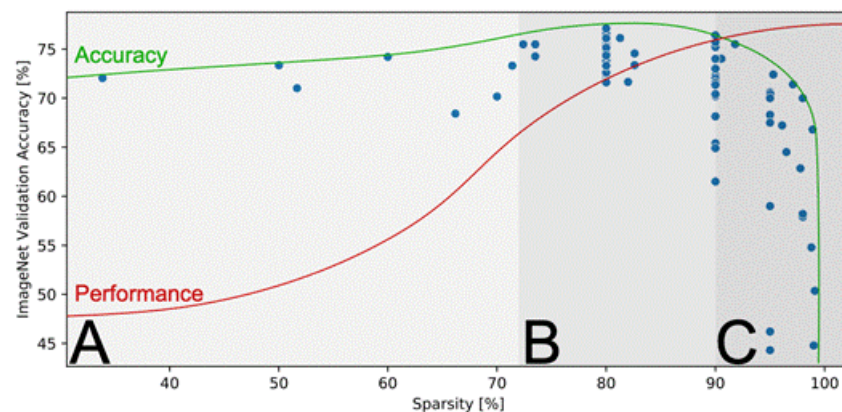


Рисунок 2 – Графики точности и производительности при прунинге (прореживании) моделей

Особенностью прунинга, в отличие от квантизации, является то, что данный процесс возможен только в уже предобученной модели.

Сжатие моделей подобным образом также имеет свои преимущества, оно позволяет уменьшить размер модели, что упрощает развертывание и ускоряет загрузку на устройствах. Оно также может снизить требования к памяти и вычислительным ресурсам. Однако, при сжатии моделей существует риск потери информации и точности модели. Некоторые методы сжатия могут привести к удалению параметров или связей, что может сказаться на производительности и результативности модели.

Тем не менее, несмотря на имеющиеся недостатки, данный способ также рассматривается в статье, так как при его использовании даже при минимальных значениях прореживания в совокупности с другими способами оптимизации может дать приемлемые по точности и производительности результаты.

### 4 Оптимизация вычислений

Оптимизация вычислений также является важным аспектом, она может включать в себя использование более эффективных алгоритмов, оптимизацию вычислительных графов,

распределение вычислений на графический процессор (GPU) или использование специализированного аппаратного ускорителя (например, Tensor Processing Unit) [8]. Особенностью данного чипа является то, что он специально был спроектирован для работы с моделями и обработкой многомерных данных. Упрощенная схема тензорного процессора от компании «Nvidia» приведена на рисунке 3.

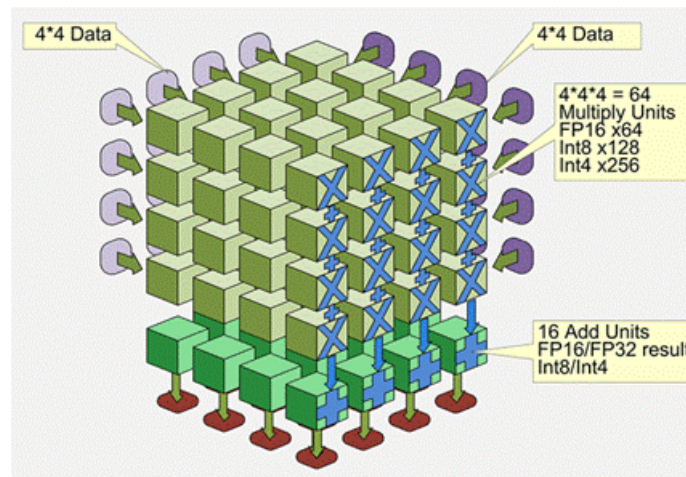


Рисунок 3 – Тензор (TPU) от компании «Nvidia»

Оптимизация вычислений может привести к значительному ускорению работы алгоритмов машинного обучения. Использование более эффективных алгоритмов, оптимизация вычислительных графов и распределение вычислений на специализированные аппаратные ускорители, могут значительно улучшить производительность. Однако эти методы требуют хорошего понимания алгоритмов и вычислительных моделей, а также опыта в их реализации и оптимизации.

## 5 Выбор фреймворков и инструментов машинного обучения

Кроме того, существуют фреймворки и инструменты, специально разработанные для оптимизации производительности машинного обучения на iOS. Некоторые из них включают библиотеки «CoreML» [9], «Metal Performance Shaders», а также фреймворк «Metal API» и другие.

На рисунке 4 приведена схема работы фреймворка «CoreML», суть которого заключается в преобразовании из обычной модели для стационарных вычислительных устройств в специальный оптимизированный формат под мобильные устройства, который обрабатывается непосредственно самой библиотекой и поставляется в разрабатываемое мобильное приложение.

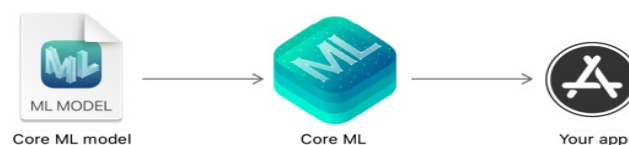


Рисунок 4 – Схема работы «CoreML»

Фреймворк «Metal Performance Shaders» содержит коллекцию высокооптимизированных вычислительных и графических шейдеров, разработанных для легкой и эффективной интеграции в мобильное приложение. Эти параллельные по данным примитивы специально настроены для максимального использования уникальных аппаратных

особенностей каждого семейства графических процессоров (GPU) с целью обеспечения оптимальной производительности. Приложения, использующие фреймворк «Metal Performance Shaders», достигают отличной производительности без необходимости создания и поддержки ручных шейдеров для каждого семейства GPU. «Metal Performance Shaders» могут использоваться вместе с другими существующими ресурсами вашего приложения (такими как объекты `MTLCommandBuffer`, `MTLTexture` и `MTLBuffer`) и шейдерами [\[9\]](#). Фреймворк поддерживает следующий функционал:

- применение высокопроизводительных фильтров к изображениям и извлечение статистических и гистограммных данных из них,
- реализация и запуск нейронных сетей для обучения и вывода машинного обучения,
- решение систем уравнений, факторизация матриц и умножение матриц и векторов [\[10\]\[11\]](#),
- ускорение трассировки лучей с помощью высокопроизводительного тестирования пересечений лучей и геометрии.

В свою очередь библиотека «Metal» — низкоуровневый, малозатратный программный интерфейс для аппаратного ускорения 3D-графики и вычислений, разработанный Apple и дебютировавший в iOS 8. «Metal» объединяет функции, схожие с OpenGL и OpenCL, в одном комплекте. Он предназначен для повышения производительности, предоставляя низкоуровневый доступ к аппаратным возможностям графического процессора (GPU) для приложений на iOS, iPadOS, macOS и tvOS. Его можно сравнить с низкоуровневыми API на других платформах, такими как Vulkan и DirectX 12. Metal является объектно-ориентированным, что позволяет использовать его с такими языками программирования, как Swift, Objective-C или C++17. Согласно промо-материалам Apple: MSL [Metal Shading Language] представляет из себя единый язык, который позволяет более тесно интегрировать графические и вычислительные программы [\[12\]](#).

Существуют также аналоги этих библиотек и для устройств на базе операционной системы Android, однако они не будут рассмотрены ввиду того, что в рамках статьи сделан упор именно на устройства от компании Apple и их чипы с префиксом «А».

Эти инструменты предоставляют оптимизированные функции и API, которые позволяют эффективно использовать аппаратные возможности устройств. Однако использование этих фреймворков требует дополнительных усилий для интеграции существующих моделей и алгоритмов, а также изучения их особенностей и возможностей.

## 6 Проведение экспериментов по совмещению алгоритмов

Для проведения экспериментов по оптимизации алгоритмов машинного обучения на iOS рассмотрим методологию, основанную на систематическом исследовании различных параметров и настроек алгоритмов. Ключевым этапом экспериментов было определение оптимальных значений параметров, таких как «learning rate», «batch size» и количество эпох обучения. Данные параметры были выбраны в связи с тем, что они наибольшим образом влияют на скорость и качество обучения алгоритмов. Так же было уделено особое внимание выбору оптимальной структуры сети и алгоритма оптимизации, адаптированных специально под платформу iOS, что позволило добиться значительного улучшения производительности алгоритмов машинного обучения именно на устройствах iOS.

Для получения наиболее точного результата с наиболее производительными алгоритмами, были проведены следующие эксперименты:

- Парно рассмотрены каждый из перечисленных способов оптимизации нейронных сетей [12] – квантизация [13], сжатие, применение TPU и используемый фреймворк. Были перепробованы различные комбинации в поиске наилучшей эффективности, часть примерных вариантов комбинаций приведена в таблице 1.
- Ввиду того, что квантизация и сжатие довольно критично снижают точность нейронных сетей [14][15] – был проведен отдельный замер без их использования – только применение чипа TPU и две отдельные комбинации с различными фреймворками = CoreML и Metal

Таблица 1 – Примерные варианты совмещения способов оптимизации

| Квантизация | Сжатие | TPU             | Фреймворк |
|-------------|--------|-----------------|-----------|
| Отсутствует | 0%     | Используется    | CoreML    |
| Малая       | 25%    |                 |           |
| Средняя     | 50%    | Не используется | Metal     |
| Сильная     | 75%    |                 |           |

Полученный результат подтвердил гипотезу о том, что применение алгоритмов сжатия и квантизации радикальным образом снижает точность первоначальной нейронной сети – точность алгоритма машинного обучения упала в несколько раз, хотя и скорость работы выросла на порядок.

В то же время, второй эксперимент дал хорошие результаты – совмещение применения чипа TPU и фреймворков CoreML и Metal дали только рост производительности, без снижения точности – за одной лишь особенностью: каждый из фреймворков необходимо использовать только для решения подходящих под них задач, а именно применение алгоритмов машинного обучения с CoreML и обработка 2D/3D изображений и моделей с помощью фреймворка Metal.

В ходе проведения экспериментов так же выявилось еще одно возможное направление для оптимизации – разбиение процесса обработки на 2 составные части, для CPU и GPU с подходящими друг другу классами эквивалентности.

Заключение

В статье были представлены основные способы оптимизации алгоритмов машинного обучения, однако для достижения наибольшего результата необходимо использовать синтез нескольких из описанных подходов.

После проведения экспериментов с описанными алгоритмами оптимизации, была предпринята попытка объединения всех имеющихся способов оптимизация для достижения наилучшего показателя и решения описанной ранее проблемы – недостаточной эффективности отдельно взятых способов оптимизаций [16][17].

Проведенные исследования и необходимые эксперименты выявили, что комбинация сжатия и квантизации радикальным образом уменьшает точность первоначальной нейронной сети. Таким образом, для достижения оптимизации с допустимыми потерями точности рекомендуется использовать только один из способов оптимизации самого алгоритма. Эмпирическим путем было подтверждена рекомендация разработчиков



инструментария, о том, что необходимо комбинировать возможности специального чипа с одним из фреймворков [18].

Еще одним результатом экспериментов стало выявление нового направления для оптимизации - разбиение процесса обработки входных данных на классы эквивалентности таким образом, чтобы обработка происходила параллельно не только с использованием мощностей GPU и TPU, а также CPU. Примерная условная схема разбиения приведена на рисунке 5. Несмотря на то, что центральный процессор не предназначен для выполнения такого рода операций, при правильном разбиении это дало рост скорости выполнения.

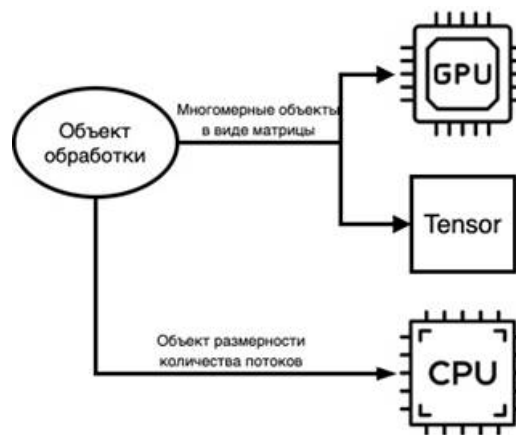


Рисунок 5 – Разбиение объекта обработки по классам эквивалентности

В качестве дальнейших направлений работы планируется исследовать применение предложенного метода к мобильным устройствам под операционной системой Android [19], а также реализовать на практике синтез нескольких алгоритмов оптимизации моделей машинного обучения.

Для практической реализации подобного решения будет проведено отдельное исследование и сравнение производительности различных алгоритмов машинного обучения на мобильных устройствах iOS в рамках диссертации и научной работы. Основная цель же этой статьи – определить эффективность и сравнить производительность различных алгоритмов, чтобы выявить наиболее подходящие для использования на iOS устройствах. Для этого были использованы стандартные наборы данных, такие как «MNIST» для распознавания рукописных цифр и «ImageNet» [20] для классификации изображений, чтобы иметь возможность сравнивать результаты с другими исследованиями. В ходе экспериментов будут учитывались различные факторы, которые могут влиять на производительность алгоритмов, такие как размер набора данных, сложность модели и выбранные параметры. Были проведены эксперименты с различными настройками фреймворков и библиотек, используемых с алгоритмами машинного обучения на мобильных устройствах, и параметрами для оценки их влияния на производительность.

## Библиография

1. Чжан Я., Лю Я., Чэнь Т., Генг У. "Мобильное глубокое обучение для интеллектуальных мобильных приложений: Обзор." // IEEE Access, 8,103,586-103,607.
2. Документация Apple Developer [Электронный ресурс] // "Core ML - Оптимизация производительности на устройствах." // URL: [https://developer.apple.com/documentation/coreml/optimizing\\_for\\_on-device\\_performance](https://developer.apple.com/documentation/coreml/optimizing_for_on-device_performance). Дата обращения: [29.06.2023].

3. Растегари М., Ордонес В., Редмон Дж., Фархади А. "XNOR-Net: Классификация изображений ImageNet с использованием бинарных сверточных нейронных сетей." // Сборник материалов Европейской конференции по компьютерному зрению (ECCV) (стр. 525-542).
4. Сихотан Х., Марк А., Риандари Ф. и Ренделл Л. "Эффективные алгоритмы оптимизации для различных задач машинного обучения, включая классификацию, регрессию и кластеризацию." // IEEE Access, 1. 14-24. 10.35335/idea.v1i1.3.
5. Сандлер М., Ховард А., Чжу М., Жмогинов А., Чен Л. Ц. "MobileNetV2: Инвертированные остаточные блоки и линейные узкие места." // Сборник материалов конференции по компьютерному зрению и обработке изображений // IEEE (стр. 4510-4520).
6. Ховард А. Г., Чжу М., Чен Б., Калениченко Д., Ванг В., Веанд Т. И др. "MobileNets: Эффективные сверточные нейронные сети для мобильных приложений компьютерного зрения." // IEEE 1704.04861.
7. Хан С., Мао Х., Дэлли У. Дж. "Глубокая компрессия: Сжатие глубоких нейронных сетей с помощью отсечения, квантования обучения и кодирования Хаффмана." // IEEE 1510.00149.
8. Документация Google TensorFlow Lite. [Электронный ресурс]. «TensorFlow» // URL: <https://www.tensorflow.org/lite>. Дата обращения: [29.06.2023].
9. Таккар М. "Начало машинного обучения в iOS: CoreML Framework." // IEEE Access, 10.1007/978-1-4842-4297-1. // ISBN: 978-1-4842-4296-4
10. Минитаева, А. М. Принятие решений в условиях интервального задания предпочтений лиц, принимающих решений / А. М. Минитаева // Материалы конференции «Информационные технологии в управлении» (ИТУ-2022) : 15-я МУЛЬТИКОНФЕРЕНЦИЯ ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ, Санкт-Петербург, 04 06 октября 2022 года. – Санкт-Петербург: Концерн; Центральный научно-исследовательский институт ;Электроприбор;;, 2022. – С. 197-200. – EDN RNGSXI.
11. Минитаева, А. М. Многомодельный подход к прогнозированию нелинейных нестационарных процессов в задачах оптимального управления / А. М. Минитаева // Необратимые процессы в природе и технике : Труды Двенадцатой Всероссийской конференции. В 2-х томах, Москва, 31 января – 03 2023 года. – Москва: Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), 2023. – С. 438-447. – EDN FBVVXS.
12. Кочнев А., «Концептуальные основы практического использования нейронных сетей: проблемы и перспективы» // «Общество и инновации» // DOI: 10.47689/2181-1415-vol4-iss1-pp1-10
13. Курбариа М., Бенгио Й., Дэвид Ж. П. "BinaryNet: Обучение глубоких нейронных сетей с ограничениями на веса и активации +1 или -1." // IEEE: 1602.02830.
14. Ли Г., Вэй Гао, Вуэн Г., «Техники квантизации» // DOI: 10.1007/978-981-97-1957-0\_5
15. Самсиана С., Сьямсул А. «Алгоритмы машинного обучения с использованием метода векторного квантования обучения» // DOI: 10.1051/e3sconf/202450003010
16. Адереми А. Атайеро, Сэмюэл Аджани «Обзор машинного обучения на встроенных и мобильных устройствах: оптимизация и приложения» // DOI: DOI:10.3390/s21134412
17. Сандлер М., Ховард А., ЛеКун Й. "Mobilenetv3: Высокоэффективная масштабируемая модель мобильного компьютерного зрения." // Сборник материалов конференции по компьютерному зрению и обработке изображений // IEEE/CVF (стр. 13840-13848).
18. Чен Б., Данда Р. Юан Ч. «На пути к краже глубоких нейронных сетей на мобильных устройствах» // Безопасность и конфиденциальность в сетях связи (стр. 495-508) // DOI:10.1007/978-3-030-90022-9\_27
19. Джармуни Ф., Фавзи А. «Запуск нейронных сетей в Android» // Университет Оттавы.



Введение в глубокое обучение и нейронные сети с Python (снх/ 247-280) //

DOI:10.1016/B978-0-323-90933-4.00001-2

20. Быков К., Мюллер К. «Опасности изображений с водяными знаками в ImageNet» //

Искусственный интеллект. Международные семинары ECAI 2023 (стр. 426–434) //

DOI:10.1007/978-3-031-50396-2\_24

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Статья посвящена исследованию и оптимизации алгоритмов машинного обучения для распознавания изображений на мобильных устройствах под управлением iOS. В работе рассматриваются различные методы оптимизации, такие как квантизация и сжатие моделей, а также использование специализированных фреймворков и инструментов для повышения производительности.

Методология исследования включает анализ существующих решений, проведение экспериментов для проверки эффективности различных методов оптимизации, а также сравнительный анализ производительности алгоритмов. Авторы использовали конкретные инструменты и фреймворки, такие как CoreML и Metal Performance Shaders, для тестирования и оптимизации моделей машинного обучения.

Актуальность работы обусловлена широким распространением мобильных устройств и возрастающим спросом на приложения, использующие машинное обучение. Ограниченные вычислительные ресурсы, энергопотребление и необходимость обеспечения высокой производительности на мобильных устройствах делают данную тему весьма значимой и востребованной.

Научная новизна статьи заключается в предложении комплексного подхода к оптимизации производительности алгоритмов машинного обучения на мобильных устройствах iOS. В статье представлены новые комбинации методов оптимизации, такие как совместное использование квантизации и сжатия моделей, что позволяет достигать более высоких показателей производительности и эффективности.

Стиль изложения статьи научный, текст хорошо структурирован. Статья включает введение, обзор существующих решений, описание методов оптимизации, результаты экспериментов и заключение. Каждая часть логически связана с предыдущей, что облегчает восприятие материала. Содержание статьи соответствует заявленной теме и охватывает все ключевые аспекты исследования.

Библиография содержит актуальные и релевантные источники, включая научные статьи и документацию по используемым фреймворкам и методам оптимизации. Однако рекомендуется добавить больше ссылок на современные исследования и публикации, связанные с мобильными приложениями и машинным обучением.

Авторы подробно рассматривают недостатки и ограничения предложенных методов, что показывает их объективность и стремление к всестороннему анализу проблемы. В статье приведены сравнения с аналогичными решениями, что укрепляет аргументацию и научную значимость работы.

Выводы статьи логичны и обоснованы. Авторы суммируют результаты экспериментов и предлагают направления для дальнейших исследований. Практическая значимость работы заключается в возможности применения предложенных методов оптимизации в реальных мобильных приложениях, что будет интересно разработчикам и исследователям в области машинного обучения и мобильных технологий.

Рекомендации по доработке:

1. Уточнить методологию проведения экспериментов, добавить больше подробностей о параметрах и настройках используемых алгоритмов.
2. Увеличить количество современных источников в библиографии для более полного отражения текущего состояния исследования.
3. Расширить раздел о практическом применении предложенных методов, включив больше примеров и кейсов.
4. Включить обсуждение возможных ограничений и потенциальных рисков при использовании предложенных методов оптимизации в реальных условиях.

Статья представляет собой значимый вклад в область оптимизации алгоритмов машинного обучения на мобильных устройствах. Она обладает научной новизной, актуальностью и практической значимостью. При выполнении вышеуказанных рекомендаций работа может быть рекомендована к публикации.

## **Результаты процедуры повторного рецензирования статьи**

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Предмет исследования. С учётом сформированного заголовка статья должна быть посвящена вопросам оптимизации производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS.

Методология исследования базируется на анализе и синтезе данных. Ценно, что автор применяет графический инструментарий представления полученных результатов. При этом обращает на себя внимание отсутствие указанных источников под таблицами и рисунками. Особое внимание автор уделяет проведению эксперимента, который подтвердил авторскую гипотезу.

Актуальность исследования вопросов, сопряжённых с оптимизацией производительности алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств на базе операционной системы iOS, не вызывает сомнения, т.к. вопросы цифровизации социально-экономических процессов обеспечивает ускорение их осуществления, что в том числе отражается и на экономии финансовых ресурсов. Потенциальную читательскую аудиторию интересуют возможности применения полученных результатов в решении задачи по обеспечению технологического суверенитета Российской Федерации.

Научная новизна в представленном на рецензирование материале присутствует. Например, связана с обоснованием тезиса о том, что «применение алгоритмов сжатия и квантизации радикальным образом снижает точность первоначальной нейронной сети – точность алгоритма машинного обучения упала в несколько раз, хотя и скорость работы выросла на порядок». Также было бы выигрышно в тексте статьи указать потенциальную читательскую аудиторию и конкретные направления использования полученных результатов.

Стиль, структура, содержание. Стиль изложения научный. Структура статьи автором выстроена. Рекомендуется также добавить блок «Обсуждение полученных результатов», а также часть заключения трансформировать в раздел «Дальнейшие направления научных исследований». Ознакомление с содержанием показало логичное изложение

материала в рамках заявленных структурных элементов.

Библиография. Автором сформирован библиографический список из 20 наименований. Ценно, что в нём есть как отечественные, так и зарубежные авторы. Также было бы интересно изучить конкретные статистические данные, описывающие практику применения алгоритмов распознавания изображений на основе машинного обучения для мобильных устройств за последние годы. Это позволило бы автору дополнительно обосновать актуальность исследования с применением конкретного числового обоснования.

Апелляция к оппонентам. Несмотря на сформированный список научных публикаций, какой-либо научной дискуссии в тексте рецензируемой научной статьи не обнаружено. При доработке статьи автору рекомендуется уделить внимание устранению данного замечания. Это позволит автору конкретно показать наличие прироста научного знания, который однозначно автором сделан, но очень многое из текущей редакции не воспринимается настолько выигрышно, как могло бы быть представлено.

Выводы, интерес читательской аудитории. С учётом всего вышеизложенного заключаем о том, что статья подготовлена на высоком уровне, обладает научной новизной и практической значимостью. Доработка статьи по указанным в тексте замечаниям позволит ещё больше расширить потенциальную читательскую аудиторию.

Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Демидов Н.А., Выгоняйло К.В., Маняев А.А., Ефимов Д.А., Баженов А.Э. Сравнительный анализ Wine и PortProton: Перекрёстные платформы в контексте эмуляции Windows приложений // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70773 EDN: MELEFC URL: [https://nbpublish.com/library\\_read\\_article.php?id=70773](https://nbpublish.com/library_read_article.php?id=70773)

## Сравнительный анализ Wine и PortProton: Перекрёстные платформы в контексте эмуляции Windows приложений

**Демидов Никита Александрович**

ORCID: 0009-0006-4903-5724

студент, кафедра информационных систем и технологий; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [danv9471@gmail.com](mailto:danv9471@gmail.com)



**Выгоняйло Клим Владимирович**

ORCID: 0009-0008-0325-8345

студент, информатики и вычислительной техники; ФГБОУ ВО ПГУТИ

443010, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [mjchi023@gmail.com](mailto:mjchi023@gmail.com)



**Маняев Артём Альбертович**

ORCID: 0009-0004-5716-7300

студент, информатики и вычислительной техники; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [arl.mo@mail.ru](mailto:arl.mo@mail.ru)



**Ефимов Дмитрий Александрович**

ORCID: 0009-0008-7567-3775

студент, кафедра информатики и вычислительной техники; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [di.efimoff2015@yandex.ru](mailto:di.efimoff2015@yandex.ru)



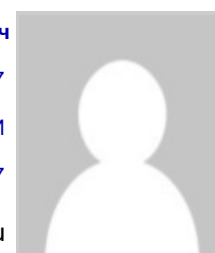
**Баженов Артем Эдуардович**

ORCID: 0009-0001-5887-2077

ассистент, кафедра программной инженерии; ФГБОУ ВО ПГУТИ

443086, Россия, Самарская область, г. Самара, Московское шоссе, 77

✉ [a.bazhenov@psuti.ru](mailto:a.bazhenov@psuti.ru)



[Статья из рубрики "Операционные системы"](#)

**DOI:**

10.7256/2454-0714.2024.2.70773

**EDN:**

MELEFC

**Дата направления статьи в редакцию:**

16-05-2024

**Дата публикации:**

19-06-2024

**Аннотация:** Современное развитие компьютерных технологий и операционных систем сопровождается увеличением потребности в программном обеспечении, способном обеспечивать взаимодействие различных программ и приложений между собой, независимо от их исходной среды. В данном исследовании будет проведен сравнительный анализ двух таких программ - Wine и PortProton. Wine - это программа, способная запустить большинство приложений, разработанных для Windows, на Unix-подобных системах. Это слой совместимости, позволяющий работать с Windows приложениями. Вместе с тем, существует и отечественный вариант - PortProton, который также предлагает запуск приложений Windows. Это исследование призвано сравнить эти две программы, проанализировать их особенности, преимущества и недостатки, а также определить, какая из них является наиболее удобной и функциональной для конечного пользователя в контексте эмуляции Windows приложений. Методология исследования предполагает сравнительный анализ платформ Wine и PortProton через тестирование бенчмарков и проверку работоспособности Windows приложений на Linux. Бенчмарк-тестирование включает оценку производительности, стабильности и скорости работы приложений Windows на каждой платформе. В виду отсутствия научных источников на тему сравнения Wine и PortProton в контексте эмуляции Windows-приложений, данное исследование имеет уникальный характер. Из данного исследования, был сделан вывод, что Wine и PortProton успешно справляются с эмуляцией Windows-приложений, показывая в некоторых моментах наилучшую производительность ввиду оптимизации операционной системы Linux. PortProton лучше всего справляется с задачей эмуляции программ ввиду стабильной работы и удобства использования. Wine, несмотря на незначительное опережение в контексте производительности и возможность запуска нескольких программ одновременно, показал наихудшую эффективность ввиду некорректной работы некоторых программ и отсутствия интуитивно понятного графического интерфейса. Исходя из вышеописанных выводов рекомендуется PortProton для большинства пользователей.

**Ключевые слова:**

Эмуляция, Перекрестные платформы, Wine, Portproton, Совместимость приложений, Производительность, Технические характеристики, Windows, Linux, Unix

**Введение**

В настоящее время дистрибутивы Linux становятся всё более востребованы ввиду экономической эффективности, высокой степени гибкости, обширной языковой поддержки и распространения свободного программного обеспечения (СПО). [\[1\]](#)

Множество пользователей операционных систем Linux и macOS сталкиваются с необходимостью использования Windows-приложений из-за их широкого распространения и функционала. Для этого существует несколько решений, одними из наиболее популярных являются Wine и PortProton. Оба представляют собой программные решения, позволяющие запускать Windows-приложения на Unix-подобных системах. Однако, они имеют разные особенности, преимущества и недостатки, которые следует рассмотреть при выборе подходящего инструмента. В данной статье проведен сравнительный анализ Wine и PortProton, выявлены их ключевые характеристики и сферы применения.

## **1. Описание Wine и PortProton**

### **1.1. Wine: Открытое программное обеспечение для реализации Windows-среды.**

Wine (Wine Is Not an Emulator) представляет собой проект с открытым исходным кодом, который разрабатывается сообществом для запуска приложений, разработанных для операционных системах Windows, на Unix-подобных ОС. [\[2\]](#) Одним из ключевых преимуществ Wine является его доступность для широкой аудитории, благодаря открытой лицензии GNU LGPL. Это означает, что любой желающий может участвовать в разработке проекта, вносить изменения и улучшения, а также использовать Wine бесплатно и без ограничений.

Совместимость приложений с Wine – это одна из его ключевых характеристик, которая продолжает улучшаться с каждым новым выпуском. Wine стремится к обеспечению максимальной совместимости с широким спектром Windows-приложений, включая как старые, так и новые программы. Однако не все приложения могут работать корректно под Wine, особенно те, которые используют специфические функции операционной системы Windows или требуют особых настроек.

Интерфейс Wine предоставляет пользователю гибкость в настройке и управлении запускаемыми приложениями. Пользователи могут использовать различные конфигурационные файлы и параметры командной строки для оптимизации работы приложений под свои нужды. Кроме того, Wine поддерживает использование различных версий Windows API, что позволяет лучше адаптировать запускаемые приложения под конкретные требования.

Для установки пакетов приложений используется терминал, с помощью которого идёт управление всеми файлами, каталогами и приложениями. Wine воспроизводит системные библиотеки Windows с помощью своих собственных библиотек аналогичной функциональности. [\[3\]](#)

### **2.2. PortProton: Реализация Windows приложений для игр и не только.**

PortProton представляет собой отечественную разработку в области интерпритации Windows приложений на Unix-подобных операционных системах. Этот проект, созданный на основе открытого исходного кода [\[4\]](#) Proton (отдельной версии Wine от компании Valve), сфокусирован преимущественно на обеспечении совместимости и оптимизации запуска игровых приложений, но также способен успешно работать с обычными приложениями.

Одним из ключевых преимуществ PortProton является его специализация на игровых приложениях. Команда разработчиков активно работает над оптимизациями, направленными на улучшение производительности и совместимости с широким спектром игр. Это включает в себя оптимизацию работы с графическими API, улучшение поддержки DirectX и Vulkan, а также реализацию дополнительных функций, необходимых для успешного запуска игровых приложений.

Однако, важно отметить, что PortProton также способен запускать обычные приложения, такие как офисные программы, графические редакторы, браузеры и другие. Пользователи могут успешно использовать PortProton для запуска широкого спектра Windows приложений на своих операционных системах Linux.

В целом, PortProton представляет собой гибкий инструмент для интерпритации Windows приложений на Unix-подобных операционных системах, который успешно справляется не только с игровыми приложениями, но и с обычными программами. Его активное сообщество разработчиков и пользователей, постоянные обновления и интеграция с игровыми платформами делают его привлекательным выбором для широкого круга пользователей, желающих получить доступ к богатому миру Windows приложений на альтернативных операционных системах.

## 2. Материалы и методы

Перед началом сравнения были установлены Wine и Portproton на компьютеры со следующими характеристиками:

### Характеристики ноутбука №1:

- Процессор – Ryzen 3 2200u
- Видеокарта – Radeon 535
- Оперативная память – 8 Гб
- Системный накопитель – HDD
- Операционная система – Ubuntu 22.04.4 LTS, Windows 10 Домашняя (22H2)
- Тип ОС – 64-бит
- Версия GNOME – 42.9
- Оконный интерфейс – Wayland

### Характеристики ноутбука №2:

- Процессор – Ryzen 5 5500u
- Встроенная графика – Vega 7
- Оперативная память – 16 Гб
- Системный накопитель – SSD
- Операционная система – Fedora 40
- Тип ОС – 64-бит
- Версия GNOME – 46
- Оконный интерфейс – Wayland

Wine был загружен через стандартный пакетный менеджер. На ubuntu подобных дистрибутивов выполняется командой «`sudo apt install wine`». [\[2\]](#) После этого был установлен Winetricks из стандартного пакетного менеджера, затем было запущено обновление командой «`sudo winetricks --self-update`». [\[5\]](#)

Для проведения сравнения Wine и Portproton использовалась следующая методика: Перед началом работы с новой программой создавался новый префикс в Wine и

Portproton. Для создания префикса и скачивания зависимостей в Wine есть два способа:

### 2.1. Графический способ

Открываем консоль и запускаем Winetricks [\[6\]](#) одноимённой командой. В открывшемся меню выбираем «создать новый префикс Wine» и нажимаем «ок». В новом окне выбираем архитектуру и название префикса. Разработчики Wine рекомендуют использовать 32 битную архитектуру, но не все программы поддерживают такой режим, поэтому его стоит выбирать в зависимости от запускаемой программы. Для примера создадим 32 битный префикс с именем «test». Затем после загрузки откроется предыдущее окно и в нём появится наш созданный префикс (рис. 7). Если появляются информационные окна нажимаем «ок».

### 2.2. Консольный способ

После открытия консоли необходимо задать переменные, для этого следует прописать их, воспользовавшись командой «export WINEARCH=win32» (по умолчанию «win64») и «WINEPREFIX=~/.wine/test» [\[7\]](#) (по умолчанию «~/.wine»). Для создания префикса достаточно прописать любую команду из Wine или Winetricks, но часто используют «wineboot -u».

В обоих случаях, если будет предложено установить дополнительные модули, такие как «wine mono», соглашаемся и ждём загрузки.

Затем выполняется запуск тестируемой программы с помощью обоих средств, при возникновении проблем устанавливаем зависимости и проверяем заново. Оцениваем требуемое количество действий для запуска программ. Производим оценку производительности в программах-бенчмарках и выполняем сравнения.

Для установки дополнительных программ и библиотек используется Winetricks с графическим интерфейсом или прописать названия библиотек, например, «winetricks -q dotnet20 vcrun2005», префикс -q отключает предупреждения.

В Portproton процессы похожи с Wine, однако всё делается через графический интерфейс. В открытом окне во вкладке НАСТРОЙКИ WINE в выпадающий список префиксов можно вписать новый префикс и при нажатии «WINETRICKS» будет предложено установить стандартные библиотеки, после установки будет возможность загрузить требуемые библиотеки.

## 3. Сравнение и анализ работоспособности

Для оценки производительности и простоты запуска использовались следующие программы:

1. Furmark (версия 1.38.1.0) – бенчмарк, основанный на API OpenGL. [\[8\]](#)
2. 3D Benchmark.OK (версия 2.01) – бенчмарк, измеряющий производительность в условиях виртуального пространства. [\[9\]](#)
3. CINEBENCH R23.200 (Build RBBENCHMARK330542) – бенчмарк, основанный на движке Redshift. [\[10\]](#)
4. Mathcad 15.0 (M045 [MC15\_M050\_20171129]) – инженерное и математическое ПО для анализа и документации. [\[11\]](#)



5. FL Studio (версия 21.0.3 build 3517) – цифровая звуковая рабочая станция, ПО для написании музыки. [\[12\]](#)

В Furmark (рис. 1-3) Wine показал лучший результат по количеству фреймов и кадров в секунду, ненамного опережая PortProton. Производительность на Windows при этом оказалась наименьшей по тем же параметрам. Однако, на Wine, Furmark стоит запускать в оконном режиме ввиду наличия вылетов данной программы в полноэкранном режиме.

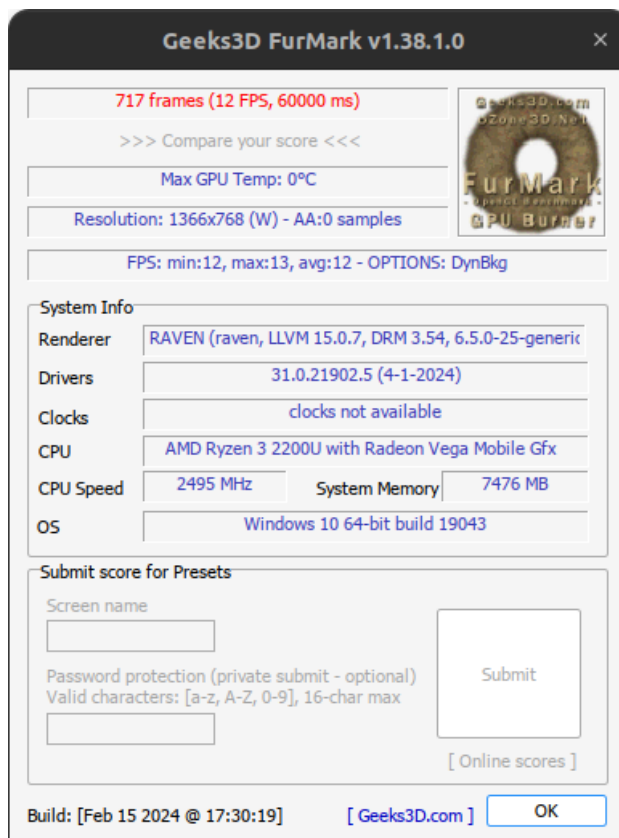


Рис.1 Результаты Furmark на PortProton

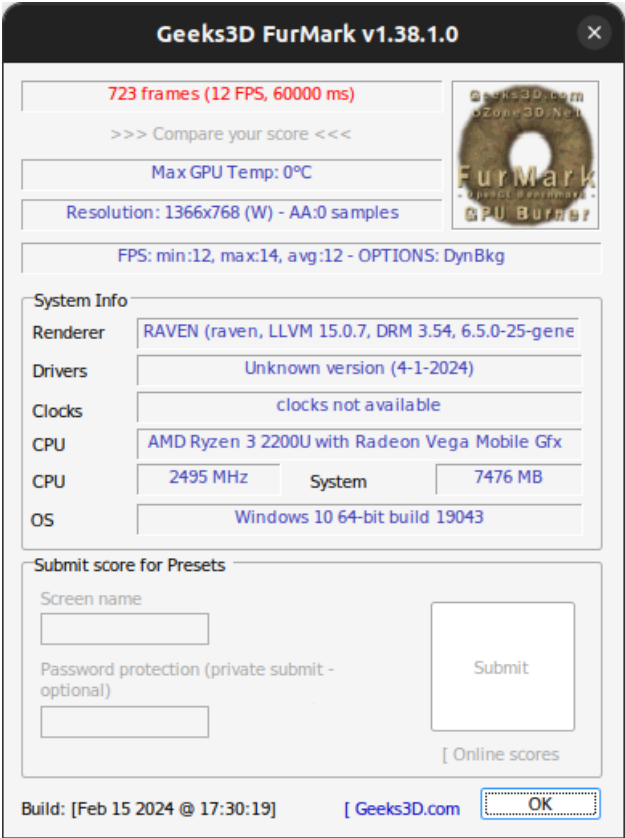


Рис.2 Результаты Furmark на Wine



Рис.3 Результаты Furmark на Windows

В 3D Benchmark (рис. 4-6) результаты оказались аналогичными, при этом производительность Wine и PortProton в бенчмарке значительно опережает Windows как по количеству кадров, так и по времени работы. Стоит отметить, что при выборе сглаживания на Wine наблюдаются проблемы, связанные с появлением графических артефактов при выборе сглаживания, что мешает нормальной работе программы, поэтому стоит его отключить.

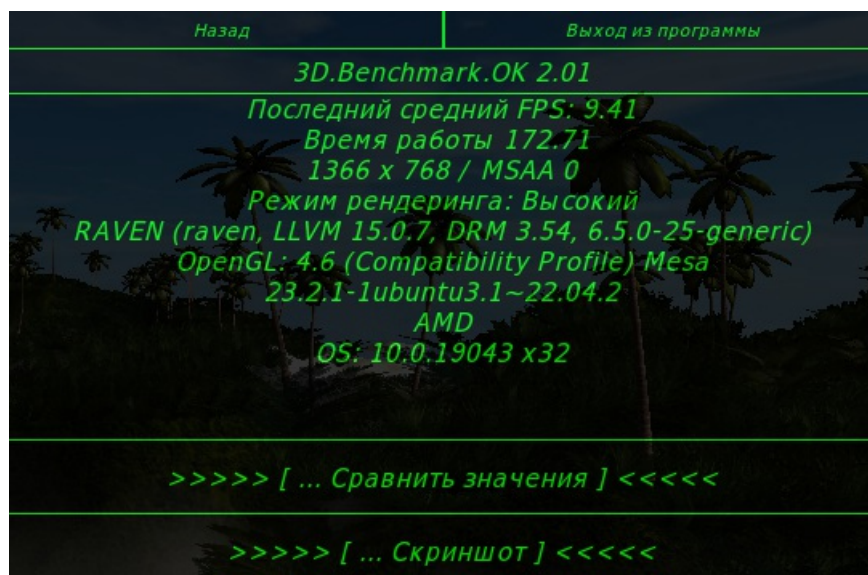


Рис. 4 Результаты 3D Benchmark на PortProton

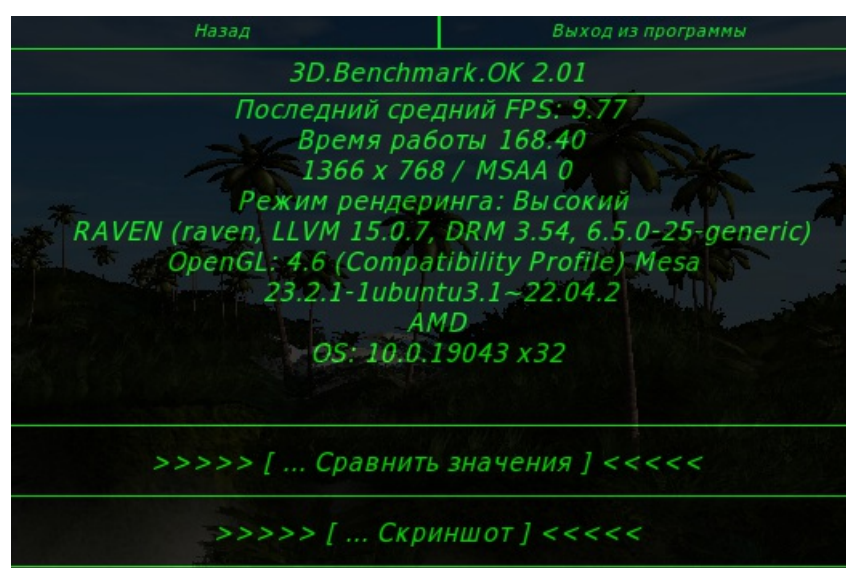


Рис. 5 Результаты 3D Benchmark на Wine



Рис. 6 Результаты 3D Benchmark на Windows

По результатам CineBench (рис.7) также наблюдается значительный отрыв в рендере на Wine и PortProton от Windows в одноядерном и многоядерном тестировании.



Рис.7 Результаты CineBench на PortProton, Wine, Windows

## 4. Запуск прикладных программ

### 4.1. Mathcad

Mathcad 15 показал полную работоспособность на Wine и PortProton, но для его установки рекомендуется придерживаться следующих инструкций:

#### 4.1.1. Wine

Использовалась версия Wine 9.9.

Перед началом создаем префикс и устанавливаем 32bit архитектуру: `$ WINEARCH=win32 WINEPREFIX=~/.wine_mathcad wineboot -u`

Устанавливаем библиотеки: `$ WINEPREFIX=~/.wine_mathcad winetricks -q mdac27 mdac28 msxml3 msxml4 msxml6 vcrun2005 vcrun2010 wsh57`

Ставим .NET (mathcad 15 требует .NET 3.5): `$ WINEPREFIX=~/.wine_mathcad winetricks -f --force dotnet20 dotnet35sp1`

Скачиваем на рабочий стол содержимое архива Mathcad.tar.gz, расположенного по пути.

Запускаем установку Mathcad 15: `$ WINEPREFIX=~/.wine_mathcad wine ~/Рабочий стол/Mathcad/setup.exe`

Процесс установки программы не отличается от установки на Windows системе. При первом запуске есть вероятность зависания программы, в таком случае стоит её перезапустить.

#### 4.1.2 PortProton

Установка Mathcad через PortProton происходит аналогичным способом, но через графический интерфейс. В отличие от Wine, PortProton уже имеет библиотеки vcrun2005 и vcrun2010 в составе рекомендуемых, которые нам в дальнейшем предложат установить (рис.9). Однако в случае с Mathcad они будут конфликтовать с остальными библиотеками, соответственно от их установки стоит отказаться и загрузить vcrun2005 vcrun2010 вручную вместе с остальными библиотеками, описанными в разделе установки Mathcad на Wine.

Для создания нового префикса необходимо перейти на вкладку «НАСТРОЙКИ WINE» (рис. 8 пункт 1). Затем вписать в поле «PREFIX» название префикса (рис. 8 пункт 2). Все символы в названии префикса будут заменены прописными буквами. Затем, необходимо

нажать кнопку «Winetricks» (рис.8 пункт 3) для применения изменений. Далее появится сообщение, хотим ли мы установить стандартные библиотеки (рис. 9) для Mathcad, мы отказываемся по упомянутым выше причинам.

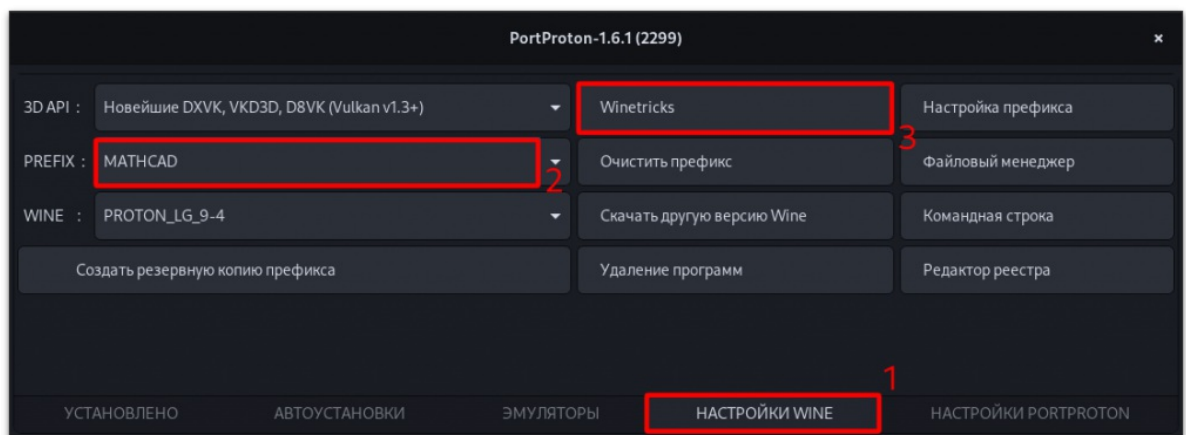


Рис. 8 Создание префикса

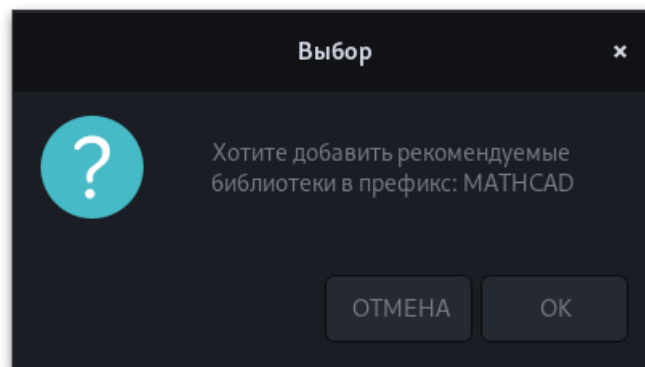


Рис. 9 Окно с предложением установить рекомендуемые библиотеки

После создания префикса откроется меню выбора библиотек (рис. 10), в будущем его можно будет открыть, выбрав соответствующий префикс и нажав на «Winetricks», так же как было описано выше. В списке выбираем те же библиотеки, что при установке в Wine. Шрифты устанавливаются в соседней вкладке «ШРИФТЫ», там выбираем «corefonts». После установки окно откроется повторно, где будут стоять галочки на тех компонентах, что успешно установились. В шрифтах будут стоять галочки напротив многих пунктов, так как в «corefonts» входят базовые шрифты Windows. В случае возникновения проблем при установке, необходимо установить библиотеки в другом порядке. В таком случае пропускаем их, и после скачивания остальных компонентов, запускаем процесс заново, выбрав пропущенные этапы. Winetricks в Wine и PortProton устанавливает всё по очереди в списке.

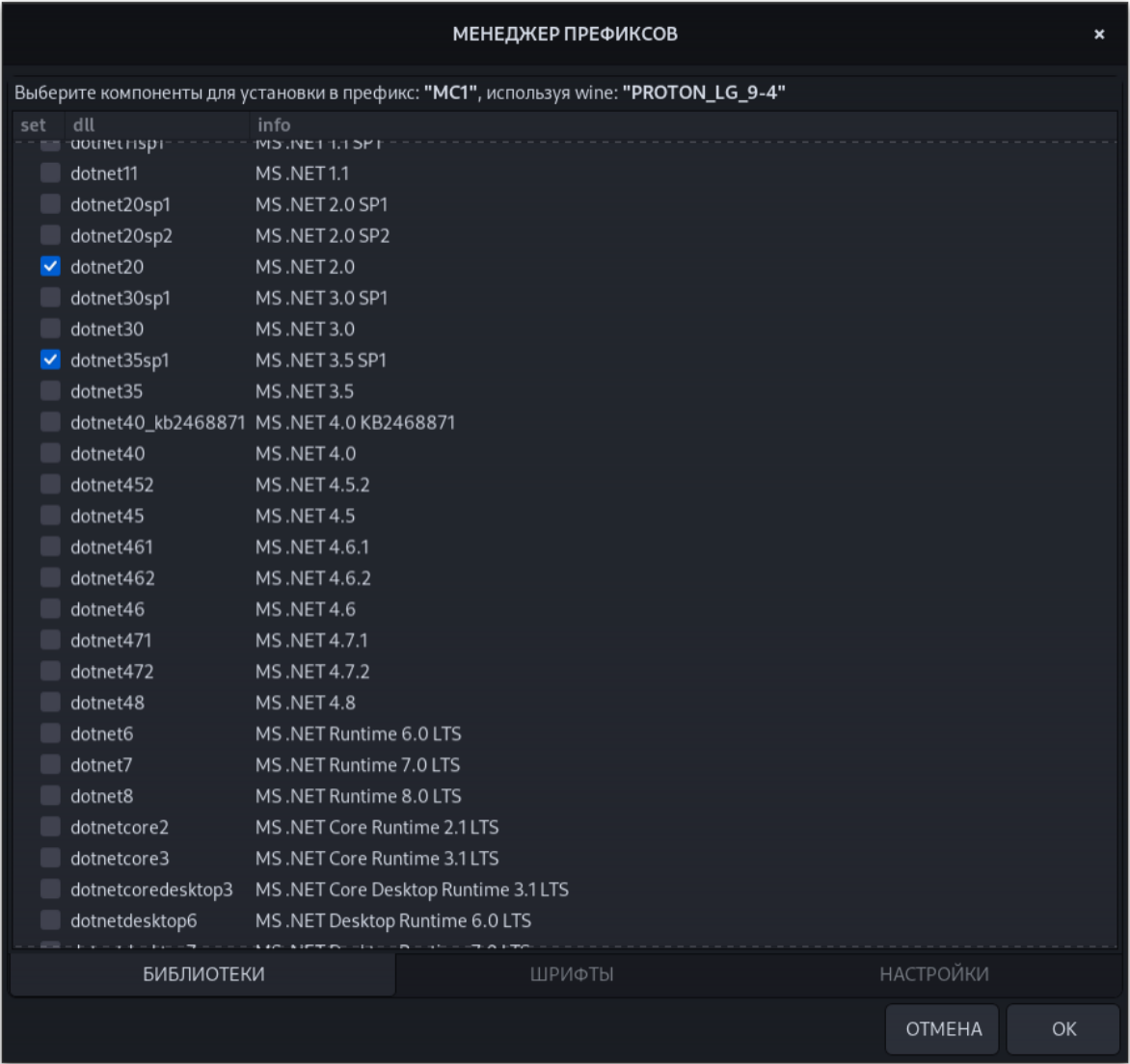


Рис.10 Менеджер префиксов

В отличие от Wine, который создал ссылку на программу автоматически, нам необходимо создать ссылку, для этого достаточно открыть PortProton и выбрать пункт «Создать ярлык...» (рис. 11) и выбрать путь к файлу из предложенных (рис. 12).

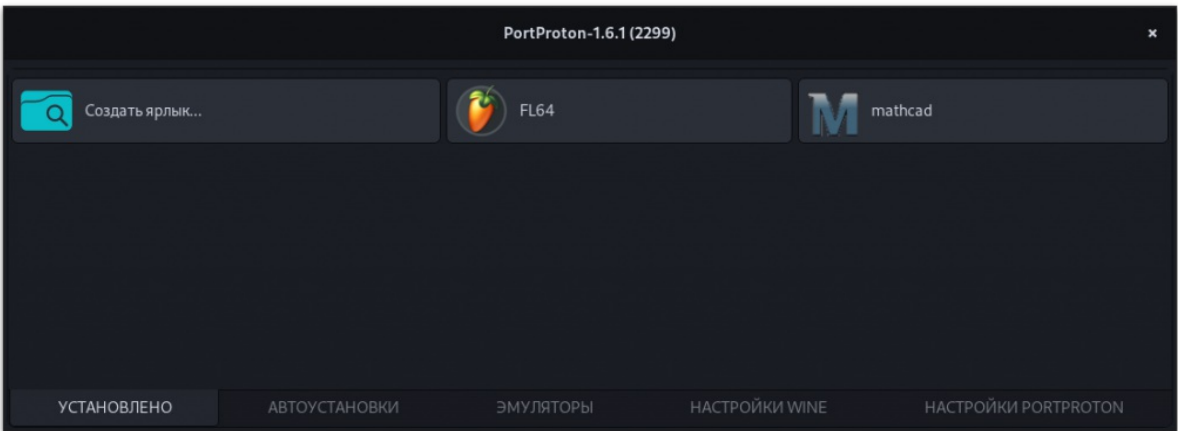


Рис. 11 Основное окно PortProton



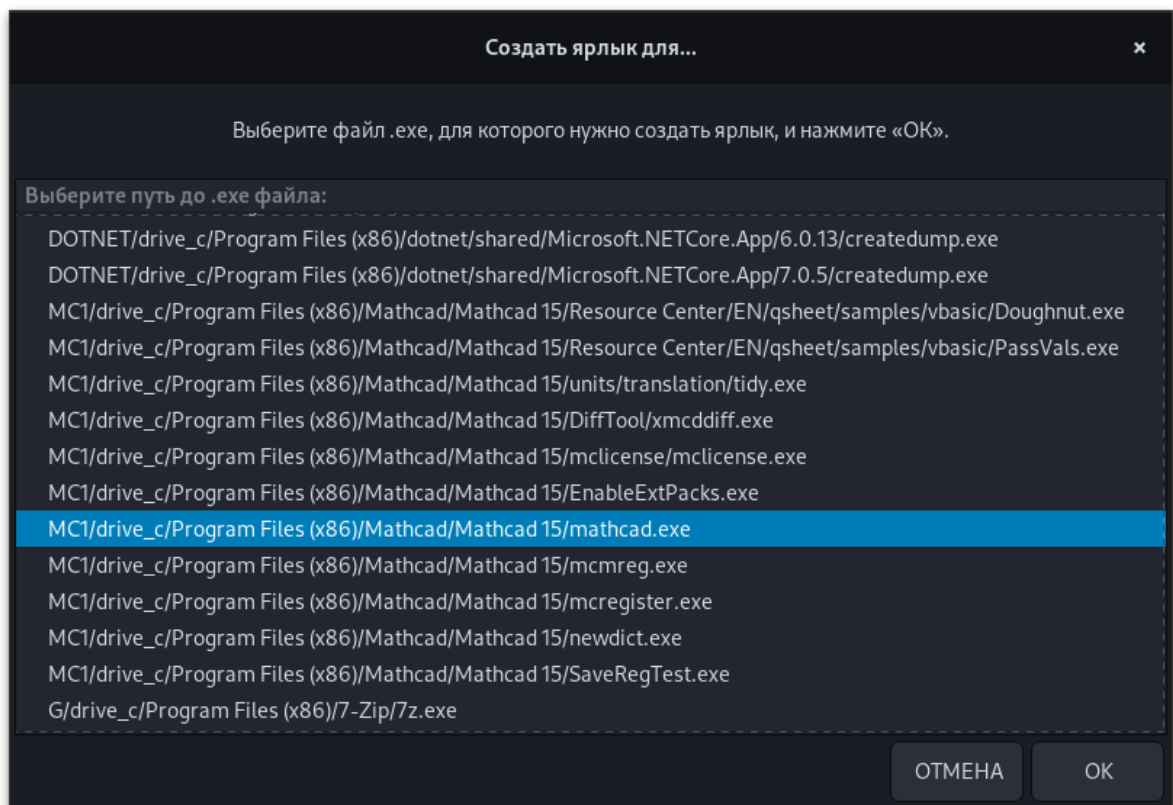


Рис. 12 Окно выбора пути для создания ярлыка

Далее будет возможность запустить Mathcad из той же вкладки или из меню приложений операционной системы. Открывается окно предложения запустить программу (рис. 13), которое появлялось при запуске установщика. После всего этого открывается окно Mathcad идентичное по виду тому, что запускалось через Wine (рис. 14).

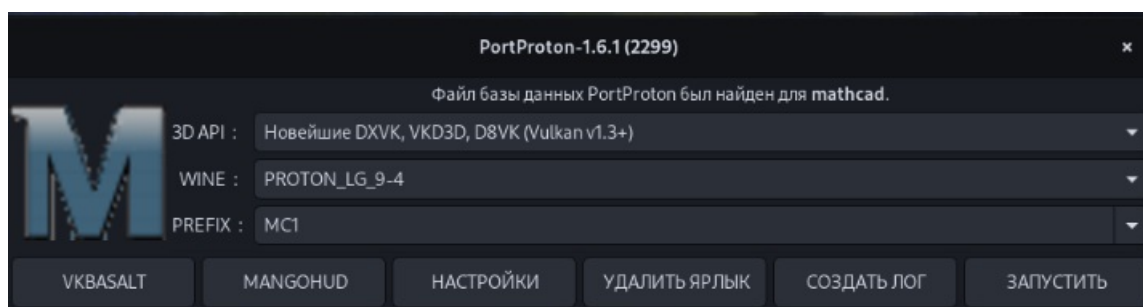


Рис.

13 Окно запуска

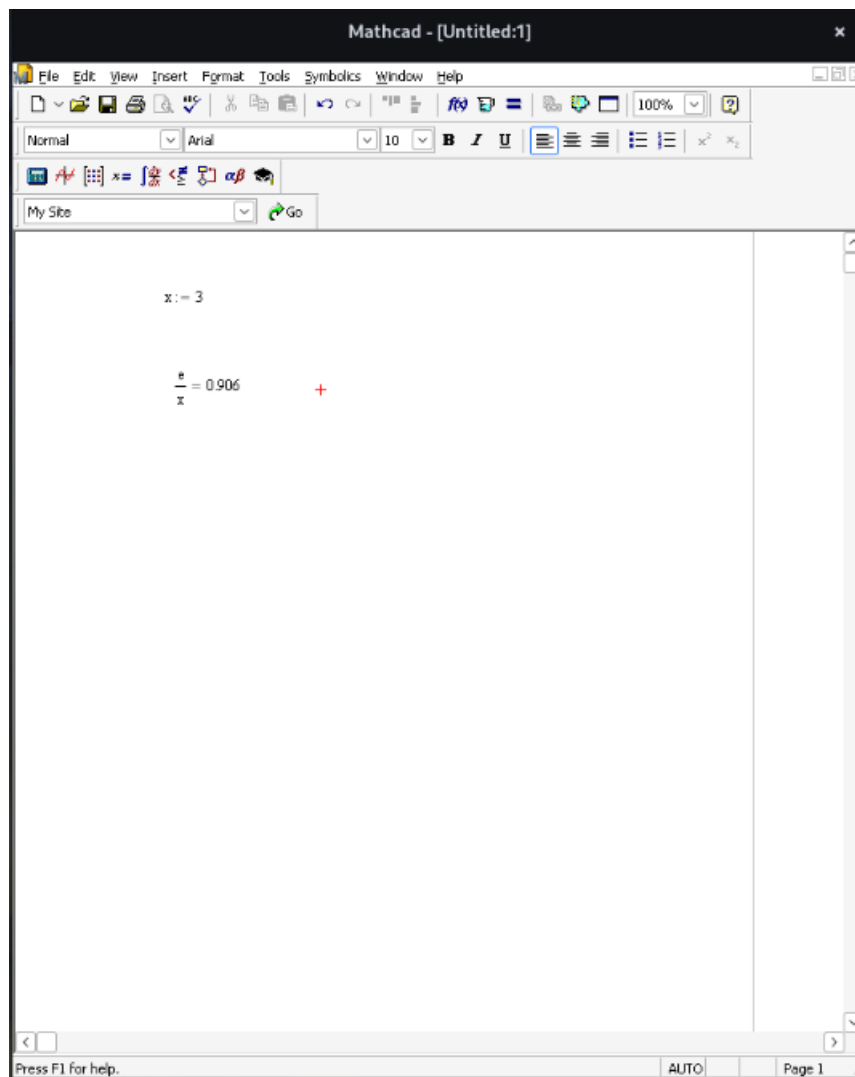


Рис. 14 Mathcad на Fedora

#### 4.2. FL Studio

FL Studio - программа для написания музыки запускается без проблем. В отличие от Mathcad, FL Studio не требует установки дополнительных библиотек, поэтому достаточно запустить установщик и дождаться завершения. Пример работающей FL Studio во вкладке «об авторах» показан на рис. 15.





Рис. 15 Интерфейс FL Studio 21 на Fedora

5. Результаты

Для визуализации результатов сравнения, были добавлены критерии оценивания платформ по 5-ти балльной шкале, где 5 - это максимальный балл, а 1 - минимальный. (Табл.1)

| Критерии                     | Wine | PortProton | Windows |
|------------------------------|------|------------|---------|
| Производительность           | 5    | 5          | 4       |
| Стабильность                 | 3    | 4          | 5       |
| Удобство                     | 3    | 4          | 5       |
| Поддержка windows-приложений | 3    | 3          | 5       |
| Доступность платформ         | 4    | 5          | 2       |
| Итоги                        | 18   | 21         | 21      |

Табл.1 Результаты сравнения Wine, PortProton и Windows

Производительность: В данном пункте Wine и PortProton показали наилучшие результаты ввиду оптимизации операционной системы Linux, поскольку её ядро имеет открытую модель разработки, высокую степень гибкости [\[13\]](#) и целостности. В то время как Windows – закрытая ОС, имеющая ограничения для разработки ПО. [\[14\]](#)

Стабильность: При тестировании бенчмарков наблюдались проблемы с работоспособностью на платформе Wine, описанные в разделе «Сравнение и анализ». В рамках нашего исследования, PortProton показал аналогично стабильную работу наряду с Windows, но с учетом ограниченного количества запускаемых приложений, которые описаны в Wine Application DataBase [\[15\]](#), он получил на один балл меньше.

Удобство: Windows в данном сравнении является лучшим вариантом так, как все рассматриваемые приложения изначально были созданы для данной платформы. PortProton, в свою очередь, удобен в установке и запуске приложений ввиду наличия

графического интерфейса, который будет интуитивно понятен для большинства обычных пользователей. Wine, как платформа, является наименее удобным вариантом для использования, так как не имеет графического интерфейса, и для большинства обычных пользователей будет проблематичным в эксплуатации.

Поддержка Windows-приложений: Не все Windows-приложения запускаются на Wine и PortProton. Проверить возможность запуска программ можно в пользовательской базе данных WineDB [\[15\]](#). Так как Wine и PortProton разрабатывается сообществом, реализовать интерпретацию всех системных вызовов является крайне сложной задачей. Например, в Linux нет поддержки DirectX (API для взаимодействия с видеокартой на платформе Windows), вместо него используется DXVK (библиотека для ретрансляции вызовов DirectX), который преобразует вызовы на Vulkan (кроссплатформенный API). По данной причине, есть вероятность, что некоторые программы могут работать нестабильно или не работать вовсе.

Доступность платформы: Wine и PortProton основаны на открытом исходном коде, при этом PortProton – отечественная разработка, использование которой поддерживает развитие технологического суверенитета. Обе платформы работают на большинстве дистрибутивов, в том числе на российских, таких как ROSA Linux, ALT Linux, Astra Linux, RED OS. Лицензии Wine и PortProton позволяют использовать их в том числе для коммерческих и учебных целей. В связи с возможными региональными ограничениями, использование платформы Windows может вызвать определенные проблемы.

### **Заключение**

Из данного исследования, был сделан вывод, что Wine и PortProton успешно справляются с эмуляцией Windows-приложений, показывая в некоторых моментах наилучшую производительность ввиду оптимизации операционной системы Linux. Но для работы некоторых программ следует устанавливать дополнительные библиотеки, как в нашем случае с Mathcad. Для остальных протестированных программ дополнительные настройки не требовались.

В контексте запуска Windows-приложений, Wine справляется хуже, чем PortProton, ввиду нестабильности работы некоторых программ, например, вылеты в Furmark в полноэкранном режиме и наличие графических артефактов в 3D Benchmark при выборе сглаживания.

PortProton, не смотря на незначительное отставание от Wine в плане производительности, которое может обуславливаться погрешностью, лучше всего справляется с задачей эмуляцией программ, ввиду стабильной работоспособности. Также из преимуществ стоит подчеркнуть наличие графического интерфейса, что обуславливает удобный пользовательский опыт в запуске программ. Из недостатков важно отметить, что через PortProton нельзя запускать несколько приложений одновременно, в таком случае он предложит закрыть уже работающий процесс.

Суммируя всё вышеперечисленное, рекомендуется PortProton для запуска большинства программ, ввиду стабильности и простоты использования. В условиях перехода на отечественное ПО, стоит рекомендовать Wine и PortProton как хорошие альтернативные методы запуска Windows-приложений так, как выгоднее и удобнее использовать данные решения, чем портировать приложения или искать другие аналоги. Данное исследование рассматривает лишь ограниченное количество программ и рассчитано на помощь в дальнейшем развитии сферы запуска Windows-приложений на Unix-подобных системах.

## Библиография

1. Коваленков В.А. Современные тенденции популяризации дистрибутивов Linux // Рязанский государственный университет имени С.А. Есенина. Рязань. 2014 – С. 95
2. WineHQ [Электронный ресурс] URL: <https://www.winehq.org/> Дата обращения: 22/04/2024
3. Мищук, Б. Р. Wine — среда обеспечения исполнения Windows-приложений в ОС Linux // Государственное образовательное учреждение высшего профессионального образования «Томский государственный педагогический университет», кафедра информатики. Томск. 2013 - С. 11
4. GitHub [Электронный ресурс] URL: <https://github.com/ValveSoftware/Proton> Дата обращения: 22/04/2024
5. Debian wine [Электронный ресурс] URL: <https://wiki.debian.org/Wine> Дата обращения: 22/04/2024
6. WineHQ Winetricks [Электронный ресурс] URL: <https://wiki.winehq.org/Winetricks> Дата обращения: 22/04/2024
7. Archlinux wiki Wine [Электронный ресурс] URL: [https://wiki.archlinux.org/title/Wine\\_\(Русский\)](https://wiki.archlinux.org/title/Wine_(Русский)) Дата обращения: 22/04/2024
8. Geeks3D FurMark [Электронный ресурс] URL: <https://www.geeks3d.com/furmark/> Дата обращения: 22/04/2024
9. SoftwareOK.com [Электронный ресурс] URL: <https://www.softwareok.com/?seite=Freeware/3D.Benchmark.OK> Дата обращения: 22/04/2024
10. CineBench [Электронный ресурс] URL: <https://www.maxon.net/en/cinebench> Дата обращения: 22/04/2024
11. Mathcad [Электронный ресурс] URL: <https://www.mathcad.com/en> Дата обращения: 22/04/2024
12. FL Studio [Электронный ресурс]. URL: <https://www.image-line.com/> Дата обращения: 30/05/2024
13. Зайнабидинов Р.Х. Обзор ядра Linux и его роль в современных информационных системах // Ташкентский университет информационных технологий имени Мухаммада аль-Хорезмий. Фергана. 2024 – С. 36
14. Dawid Bis, Kryspin Baran, Oliwia Kulawska. Performance comparison of different versions of Windows and Linux operating systems// Advances in Web Development Journal. 2023 – С.117
15. WineHQ. Wine Application DataBase [Электронный ресурс]. URL: <https://appdb.winehq.org/> Дата обращения: 11/06/2024

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Статья посвящена сравнительному анализу двух программных решений для эмуляции Windows-приложений на Unix-подобных операционных системах: Wine и PortProton. Авторы исследуют их ключевые характеристики, преимущества и недостатки, а также сферы применения. Основное внимание уделено производительности, совместимости и функциональности этих инструментов.

Для проведения сравнительного анализа были установлены обе программы на компьютер с заданными характеристиками. Использовались как графический, так и консольный методы настройки и запуска приложений. Оценка производительности

проводилась с помощью программ-бенчмарков (Furmark, 3D Benchmark.OK, CineBench, Mathcad 15), а также анализа количества действий, необходимых для запуска тестируемых программ.

Актуальность темы исследования очевидна, поскольку многие пользователи Unix-подобных систем нуждаются в запуске Windows-приложений. В условиях возрастающей популярности альтернативных операционных систем, такие инструменты, как Wine и PortProton, становятся незаменимыми для пользователей, которым необходимо использовать специфическое ПО, доступное только на Windows.

Научная новизна статьи заключается в подробном сравнительном анализе двух ключевых инструментов для эмуляции Windows-приложений. Авторы предоставляют детальное описание их возможностей, проводя тестирование производительности и совместимости. Результаты исследования дают объективное представление о функциональности и эффективности Wine и PortProton в различных сценариях использования.

Статья написана в академическом стиле, структура логически последовательна и удобна для восприятия. Введение четко формулирует цель исследования, далее следуют разделы с описанием обоих инструментов, методологии исследования, результатов и заключения. Текст содержит необходимые графики и таблицы для иллюстрации результатов тестирования, что способствует лучшему пониманию материала.

Авторы приходят к выводу, что оба инструмента успешно справляются с задачей эмуляции Windows-приложений. Wine показал лучшую производительность в большинстве тестов, однако требует более тщательной настройки для каждого конкретного приложения. PortProton, в свою очередь, предлагает более удобный и быстрый способ запуска приложений, но уступает в производительности и имеет некоторые ограничения в функциональности.

Статья будет интересна широкой аудитории, включая разработчиков, системных администраторов, а также пользователей Unix-подобных систем, которым необходимо использовать Windows-приложения. Результаты исследования могут быть полезны для принятия решений о выборе наиболее подходящего инструмента для эмуляции в зависимости от конкретных задач.

Рекомендации по доработке:

1. Дополнить описание методологии: уточнить, какие версии программ использовались для тестирования, и подробнее описать процесс настройки окружения.
2. Расширить раздел с результатами: включить дополнительные тесты для других популярных Windows-приложений, чтобы дать более полное представление о совместимости и производительности.
3. Добавить раздел с рекомендациями: включить практические советы по оптимизации использования Wine и PortProton для различных типов приложений.
4. Уточнить выводы: предоставить более детализированные выводы по каждому тесту, чтобы читатели могли лучше понять, в каких сценариях каждая из программ показывает лучшие результаты.
5. Добавить ссылки на дополнительные источники: включите больше статей, исследований и документации, связанных с эмуляцией Windows-приложений, Wine, PortProton и другими подобными инструментами. Если доступны, добавьте ссылки на научные работы и статьи, которые обсуждают эмуляцию, совместимость программного обеспечения и производительность Wine и PortProton.

Статья представляет собой значимый вклад в область исследований по эмуляции Windows-приложений на Unix-подобных системах. Она предоставляет полезную информацию о возможностях и ограничениях Wine и PortProton, что делает её ценным ресурсом для специалистов и пользователей, работающих в этой сфере.

## Результаты процедуры повторного рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Предмет исследования. С учётом сформированного автором заголовка, заключаем о том, что статья должна быть посвящена результатам анализа Wine и PortProton как перекрёстных платформ в контексте эмуляции Windows приложений. Статья не противоречит заявленной теме, но и не раскрывает её. Автором проведена подготовительная работа, на основе результатов которой может быть осуществлена подготовкой заявленной научной статьи.

Методология исследования базируется на изложении общеизвестных фактов и суждений. Заявленные в заголовке методы "сравнение" и "анализ" в тексте не обнаружены. Ценно, что автор сопровождает приводимые тезисы формулами и графическими пояснениями.

Актуальность исследования вопросов, связанных с различными программными решениями, не вызывает сомнения, так как в условиях обеспечения технологического суверенитета Российской Федерации необходимо комплексное изучение существующих программных продуктов. При этом потенциальную читательскую аудиторию интересуют вопросы, связанные с созданием отечественных платформ с функционалом и интерфейсом, не уступающих зарубежным аналогам.

Научная новизна в представленном на рецензирование материале не обнаружена. При этом корректировка по указанным в тексте рецензии замечаниям позволит устранить данную проблему.

Стиль, структура, содержание. Стиль изложения преимущественно научный, но отдельные обороты наполнены разговорным стилем (например, автор использует в тексте статьи слово «можно», что не принято для научных работ). Структура статьи автором выстроена сбивчиво ввиду многоступенчатого подхода к формированию заголовков. Автору рекомендуется обсудить полученные результаты и сформировать конкретные авторские рекомендации по решению существующих проблем, о которых в тексте статьи также ничего не сказано. Автор утверждает, что «Wine и PortProton успешно справляются с эмуляцией Windows-приложений, показывая в некоторых моментах наилучшую работоспособность ввиду оптимизации операционной системы Linux». О каких конкретно моментах идёт речь? Насколько работоспособность лучше? Было бы интересно ответ на этот вопрос представить в виде сравнительной таблицы по 5-7 критериям. Также автор говорит о том, что «для работы некоторых программ следует устанавливать дополнительные библиотеки, как в нашем случае с Mathcad». О каких конкретно программах идёт речь? Какие именно библиотеки нужно устанавливать? Рекомендуются составить соответствующие таблицу, где это будет наглядно показано. Также было бы интересно узнать, как полученные результаты можно использовать при создании отечественных программных решений.

Библиография. Автором сформирован библиографический список из 11 наименований, при этом 10 из них представляет собой электронные ресурсы, а не научные публикации.

Апелляция к оппонентам. Учитывая, что автор не изучил научные публикации по теме статьи, то и какой-либо научной дискуссии в тексте не осуществлено. При доработке статьи и изучении научных трудов отечественных и зарубежных авторов по выбранной теме исследования, рекомендуется обсудить полученные результаты.

Выводы, интерес читательской аудитории. С учётом всего вышеизложенного статья может быть опубликована после проведения серьёзной содержательной доработки, в

т.ч. с учётом интересов потенциальной читательской аудитории. В текущей редакции статья не будет представлять научного интереса для читательской аудитории.

## **Результаты процедуры окончательного рецензирования статьи**

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Рецензируемая работа посвящена сравнительному анализу популярных программных решений, позволяющих запускать Windows-приложения на Unix-подобных системах – Wine и PortProton.

Методология исследования базируется на проведении экспериментов по применению сопоставляемых инструментов на двух ноутбуках с различными характеристиками.

Актуальность работы авторы связывают с тем, что многие пользователи операционных систем Linux и macOS сталкиваются с необходимостью использования Windows-приложений из-за их широкого распространения и функционала, а наиболее популярные программные решения, позволяющие запускать Windows-приложения на Unix-подобных системах – Wine и PortProton – имеют особенности, их преимущества и недостатки следует принимать во внимание при выборе подходящего инструмента.

Научная новизна рецензируемого исследования, по мнению рецензента заключается в результатах сравнительного анализа двух инструментов, обобщении их преимуществ и недостатков, что может быть востребовано пользователями при выборе в пользу одного из программных решений.

В тексте статьи выделены следующие разделы и подразделы: Введение, Описание Wine и PortProton (Wine: Открытое программное обеспечение для реализации Windows-среды; PortProton: Реализация Windows приложений для игр и не только), Материалы и методы (Графический способ; Консольный способ), Сравнение и анализ работоспособности, Запуск прикладных программ (Mathcad; Wine, PortProton, FL Studio), Результаты, Заключение, Библиография.

В статье авторами проведен сравнительный анализ Wine и PortProton, выявлены их ключевые характеристики и сферы применения. Изложены способы загрузки установки сравниваемых пакетов приложений, в ходе сравнения перед началом работы с новой программой создавался новый префикс в Wine и Portproton, для создания префикса и скачивания зависимостей в Wine применялись два способа: графический и консольный. Для оценки производительности и простоты запуска использовались программы самого различного назначения: Furmark, 3D Benchmark.OK, CINEBENCH R23.200, Mathcad 15.0 FL Studio. Статья содержит 15 рисунков и одну таблицу. Заслуживают внимания представленные в таблице результаты сравнения Wine, PortProton и Windows по критериям производительности, стабильности, удобства, поддержки windows-приложений и доступности платформ.

Библиографический список включает 15 источников – научные публикации отечественных и зарубежных авторов, интернет-ресурсы по рассматриваемой теме, на которые в тексте приведены адресные ссылки, что подтверждает наличие апелляции к оппонентам.

В качестве замечания следует отметить чрезмерное количество рисунков – 15 иллюстраций в одной публикации. Рисунок 14 – малоинформативен, поскольку значительная часть его площади не содержит изображения и вряд ли заполнять страницы журнала такими рисунками.

Рецензируемый материал соответствует направлению журнала «Программные системы и

вычислительные методы», отражает результаты проведенной авторами работы, может вызвать интерес у читателей, рекомендуется к опубликованию.



Программные системы и вычислительные методы

*Правильная ссылка на статью:*

Гусенко М.Ю. Создание обобщенной нотации программного интерфейса процессоров x86 для автоматизированного построения дизассемблера // Программные системы и вычислительные методы. 2024. № 2. DOI: 10.7256/2454-0714.2024.2.70951 EDN: EJJSYT URL: [https://nbpublish.com/library\\_read\\_article.php?id=70951](https://nbpublish.com/library_read_article.php?id=70951)

## Создание обобщенной нотации программного интерфейса процессоров x86 для автоматизированного построения дизассемблера

Гусенко Михаил Юрьевич

ORCID: 0009-0007-0524-5604

кандидат технических наук

доцент, кафедра "Прикладная и бизнес-информатика"; Московский технологический университет

119454, Россия, г. Москва, ул. Вернадского, 78, оф. 418

✉ [mikegus@yandex.ru](mailto:mikegus@yandex.ru)



[Статья из рубрики "Математическое и программное обеспечение новых информационных технологий"](#)

### DOI:

10.7256/2454-0714.2024.2.70951

### EDN:

EJJSYT

### Дата направления статьи в редакцию:

04-06-2024

**Аннотация:** Предметом исследования является процесс обратного инжиниринга (обратной разработки) программ с целью получения их исходного кода на языках низкого или высокого уровня для процессоров с архитектурой x86, программный интерфейс которых разрабатывается компаниями Intel и AMD. Объектом исследования являются технические спецификации команд, представленные непосредственно в документации, выпускаемой этими компаниями. Исследована интенсивность обновления документации на процессоры и обоснована необходимость разработки технологических подходов, направленных на автоматизированное построение дизассемблера с учетом регулярно выпускаемых и частых обновлений программного интерфейса процессоров. В статье представлен способ обработки документации с целью получения обобщенной, формализованной и единообразной спецификации команд процессоров для дальнейшей автоматизированной трансляции ее в программный код дизассемблера. Исследована документация производителей процессоров семейства x86 непосредственно в виде,



публикуемом Intel и AMD. На основе встроенных средств автоматизации Microsoft Office написан ряд программ верификации текстов и генерирования выходной спецификации команд процессоров. В статье представлены два основных результата: первый – это разбор различных вариантов описания команд, представленных в документации Intel и AMD, и лаконичное сведение этих описаний к однообразной форме представления; второй – комплексный синтаксический анализ нотаций описания машинного кода и формы представления каждой команды на языке ассемблера. С учетом некоторых дополнительных деталей описания команд (например, допустимого режима работы процессора при исполнении команды), это позволило создавать обобщенное описание команды для трансляции описания в код дизассемблера. К числу результатов исследования можно отнести выявление ряда ошибок в как текстах документации, так и в работе существующих промышленных дизассемблеров, построенных, как показывает анализ их реализации, с применением кодирования вручную. Выявление таких ошибок в существующем инструментарии обратного инжиниринга является косвенным результатом авторского исследования.

### **Ключевые слова:**

дизассемблер, реинжиниринг программ, ассемблер, микропроцессоры, спецификация команд, синтаксис машинного кода, синтаксис ассемблера, документация x86 Intel, документация x86 AMD, режим работы процессора

## **1. Введение**

При обратном инжиниринге (реинжиниринге) исполняемых программ для машины фон Неймана необходимо решать задачу разделения бинарного кода команд и данных, переводя при этом машинный код команд в эквивалентное символическое представление на языке ассемблера и, тем самым, осуществляя дизассемблирование программы.

Построение программы перевода машинного кода в символьную форму для процессоров семейства x86 хотя и имеет столь же длинную историю, как и история самого этого семейства, но на взгляд автора не имеет изящного технологического решения. Доступные тексты программ реализации дизассемблеров показывают [\[1,2\]](#), что его разработчики вручную кодируют процедуры распознавания машинного кода отдельных команд. Т.е. их реализации дизассемблеров технологически сложны как в написании собственно программного кода, так и в его тестировании. С учетом того, что программный интерфейс процессоров непрерывно расширяется и имеет линейную динамику приращения множества команд на протяжении более чем 20 лет, что отражается в довольно частом обновлении документации производителей, то такие программы дизассемблеров морально устаревают довольно быстро. Существующие технологические решения построения дизассемблеров не позволяют производить обновления этих программ синхронно с обновлениями программного интерфейса процессоров, следовательно, необходимо разрабатывать новые подходы к созданию таких программ.

## **2. Тенденция к расширению программного интерфейса процессоров**

Процессоры архитектуры x86 на протяжении многих лет остаются самыми распространёнными устройствами для создания персональных компьютерных систем. Согласно [\[3\]](#) доля компьютерных систем, построенных на базе процессоров этой архитектуры, в настоящий момент составляет около 70% от общего числа применяемых

процессоров. Такая популярность этих устройств у потребителей и, соответственно, у производителей компьютерной техники объясняется как историческими причинами (обеспечивается программная совместимость с ранее разработанными и проданными компьютерами), так и постоянным стремлением производителей процессоров совершенствовать вычислительный потенциал своих продуктов. Результатом этих устремлений являются архитектурные, технологические и интерфейсные обновления процессоров. Архитектурные обновления выражаются, например, в размещении на одном кристалле процессора нескольких вычислительных ядер; технологические - в переходе на более плотную компоновку транзисторов на кристалле процессора; интерфейсные - в расширении множества исполняемых процессором команд.

x86 - это архитектура со сложным набором команд (complex instruction set computing, CISC), которая традиционно предполагает поддержку специализированных для каждого рода вычислений команд, совершающих арифметические и логические действия над данными, содержащимися в регистровой памяти небольшого объема.

Флагманы среди производителей процессоров x86 - компании Intel и AMD - за прошедшие два десятилетия разработали и внедрили целый ряд вычислительных технологий, включающих расширение регистровой памяти и множества команд. Нет оснований сомневаться в том, что подобные усовершенствования прекратятся в будущем, поскольку завоеванные на рынке позиции эти компании вряд ли решаться утратить, а CISC архитектура не предполагает иного решения, кроме как добавление новых специализированных команд к уже существующему их множеству. Единственным заметным конкурентом для Intel и AMD в части производства x86-процессоров является китайская компания Zhaoxin [\[4\]](#), наладившая производство с помощью тайваньской компании Via, но это скорее производитель собственно чипов, нежели разработчик, влияющий на программный интерфейс процессоров.

Предполагая, что тенденция в совершенствовании аппаратной части сохранится, логично ожидать изменений и в программном обеспечении, конкретно в том, что в нем постепенно будут использоваться вновь появившиеся возможности процессоров. Это означает, что и решение задачи по исследованию программ, например, на наличие недокументированных возможностей также потребует разработки новых или модернизации существующих средств анализа.

Традиционно анализ поведения программы включает обратный инжиниринг, составной частью которого является дизассемблирование. Расширение множества поддерживаемых процессорами команд и регистров указывает на необходимость обновления средств дизассемблирования. Очевидным путем в этом направлении является совершенствование технологии создания дизассемблеров при соблюдении следующих требований:

1. охват всего множества команд процессора;
2. возможности тонкой настройки дизассемблера на работу с командами определенного множества или поддерживаемых при определенном режиме работы процессора;
3. минимизация ручного кодирования процедур дизассемблера.

Естественным источником сведений о командах и регистровой памяти является документация производителей процессоров. Динамика обновления документации от Intel [\[5\]](#) указывает на периодическое обновление ее примерно через каждые 106 дней (рис. 1). Если сопоставить этот график с данными, представленными в табл. 1, то можно

заметить корреляцию дат выхода обновленной документации и новой вычислительной технологии, представленной Intel. Реализация вычислительной технологии предполагает реализацию определенного набора команд в процессоре, а значит и уточнений в документации. Однако как показывает изучение версий документации, публикация описаний новых команд происходит не одновременно. Версии документации выходят регулярно, и в них постепенно уточняются или заново публикуются описания команд программного интерфейса процессора.

AMD не следует полностью в фарватере Intel и разрабатывает собственные технологии обработки данных, реализуя поддержку оригинальных наборов команд в своих процессорах. AMD обновляет документацию реже, чем Intel – в среднем каждый 176 дней (рис. 1). И так же, как у Intel, публикация сведений о новых командах происходит постепенно.

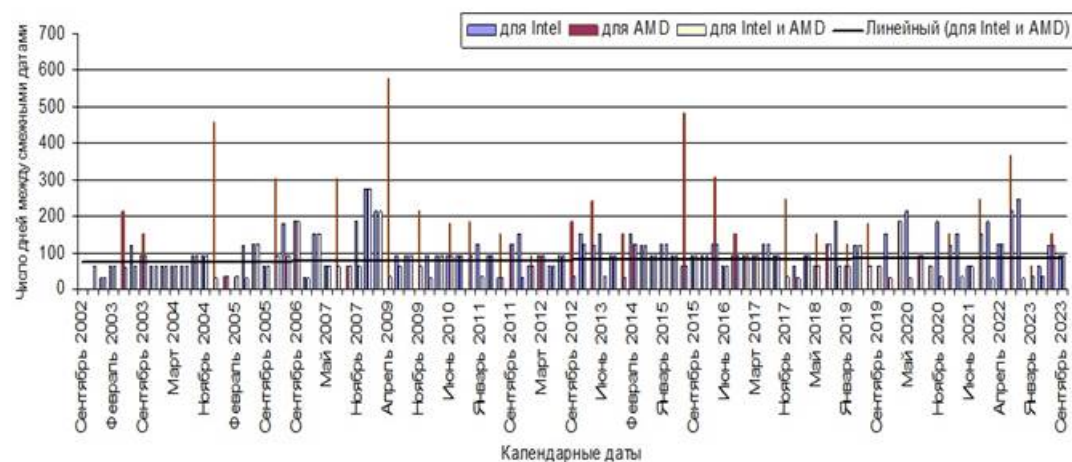


Рис. 1 Динамика выхода обновлений документации на процессоры x86 от Intel и от AMD с линейной интерполяцией среднего периода ее обновлений

Fig. 1 Dynamics of the release of documentation updates for x86 processors from Intel and AMD with linear interpolation of the average period of its updates

Объединенная динамика обновления документации Intel и AMD на рис. 1 показана через линейную интерполяцию (жирная линия). Видно, что в период с 2002 по 2023 годы она поддерживается практически неизменной - в среднем частота выхода обновлений в этом случае составляет 80 дней. Таким образом, если рассматривать официальную документацию от Intel и AMD как основной источник данных о программном интерфейсе процессоров и выпускать обновления средств дизассемблирования, поддерживающих наборы команд от обоих производителей, то это, очевидно, придется делать с периодом в 80 дней.

Отметим, что даты выхода обновлений документации непосредственно связано с публикацией описаний новых команд. Например, версия документации Intel от сентября 2023 содержит описания 4343 команд [\[6\]](#), это на 283 команды больше, чем версия документации от июня 2021. Между ними было выпущено еще пять версий документации. В среднем - это 56 новых команд в очередном выпуске. Следует признать, что это весьма динамичная тенденция расширения программного интерфейса процессоров.

Множества команд от Intel и от AMD пересекаются лишь частично, каждый производитель реализует свои оригинальные подмножества команд, различающиеся как по строению машинного кода, так и по функциональному назначению.

Документация от Intel и от AMD имеет такой объем (4595 команды), что ее автоматизированная обработка для составления спецификаций машинного кода команд представляется наиболее перспективным подходом к автоматизации построения дизассемблера.

Помимо объема текста документации следует отметить и качество исполнения ее текста, в котором регулярно присутствуют опечатки, неточности и, кроме того, Intel может поменять синтаксис и способы описания команд. Прodelать анализ таких особенностей построения текста, выявить и устранить неоднозначности и ошибки в описаниях команд без автоматизированного подхода практически невыполнимо. Такие же претензии можно адресовать и к документации от AMD.

Табл. 1. Обозначения основных технологий обработки данных в процессорах x86 и даты их анонсирования компанией Intel

Table 1. Designations of the main data processing technologies in x86 processors and the date of their announcement by Intel

| Технология | Дата анонсирования |
|------------|--------------------|
| SSE        | 26.02.1999         |
| SSE2       | 20.11.2000         |
| SSE3       | 02.02.2004         |
| SSSE3      | 26.07.2006         |
| SSE4.1     | 27.09.2006         |
| SSE4.2     | 27.09.2006         |
| AESNI      | 01.03.2008         |
| PCLMULQDQ  | 01.03.2008         |
| AVX        | 01.03.2008         |
| AVX2       | 04.06.2013         |
| AVX-512    | 01.07.2013         |
| RDRAND     | 29.04.2012         |
| AMX        | 28.06.2020         |

Единообразный и выверенный список команд вместе с их детализированными нотациями может позволить автоматически генерировать текст программы дизассемблера. Идея подхода, когда на вход обрабатывающей программы подается формальное описание языка программирования и эта программа автоматически генерирует программу распознавания текстов на этом языке, апробирована в таких системах как YACC, Bison и LEXX. В них по грамматическому описанию языка на автоматной или контекстно-свободной грамматике строится часть программы разбора текстов на этом языке.

Эту идею можно применить в отношении автоматной грамматики машинного кода процессоров x86 и, подавая на вход специального транслятора спецификации команд, получить сгенерированный код программы дизассемблера на некотором языке программирования высокого уровня.

Чтобы детально изложить эту идею, необходимо вначале рассмотреть строение документации процессоров с архитектурой x86 от Intel и AMD.

3. Структура документации Intel и AMD

Автору удавалось знакомиться с описанием программного интерфейса процессоров по фирменной документации Intel, начиная с 1995 года. Содержание и объем документации,

естественно, менялись, а вот структура описания отдельных команд процессора остается практически неизменной. В версии, используемой для написания статьи [\[6\]](#), описание отдельной команды имеет следующий типовой вид (рис. 2), где характерными элементами являются:

1) **заголовок**, в котором указаны: мнемоника (иногда мнемоники семейства команд) команды и краткое описание ее семантики (рис. 2(1)): 'ADD' - мнемоника, 'Add' - краткое описание семантики команды;

2) таблица описания мнемоники команд (рис. 2(2)), включая синтаксическую нотацию машинного кода в столбце "Opcode"; синтаксическую нотацию ассемблерного кода команды (или мнемонику) в столбце "Instruction" и прочую информацию. Более детальное описание столбцов таких таблиц представлено в п. 4. Кроме того, на рис. 2(2) представлен только один из вариантов таблицы, которых в документации Intel шесть видов (рис. 5, 6, 7, 8, 9, 10). Отметим, что структура машинного кода не отделена в некоторых таблицах (табл. 6) от синтаксиса команды и такое разделение необходимо будет произвести;

3) таблица кодирования операндов команды (рис. 2(3)), которая всегда следует за таблицей описания команды, и в которой представлены варианты кодирования явных операндов команды и способ (чтение/запись) их использования;

4) развернутое текстовое описание работы команды (рис. 2(4));

5) описание операционной семантики команды на Algol-подобном языке (рис. 2(5)). В документации представлено описание элементов этого языка, однако его фактическое применение для описания команд несколько произвольное, применяется авторами каждого описания команды часто без согласования с исходной нотацией языка и апеллирует скорее к интуитивному восприятию читателя, чем подходит для формальной трансляции;

6) описание воздействия на флаги из регистра [R]EFLAGS (Здесь и далее в статье используется метаязык для описания грамматики в расширенной нотации Бэкуса-Наура (РБНФ) [\[7\]](#)) процессора, которое производит исполнение команды (рис. 2(6));

7) различные дополнительные аспекты поведения процессора в случае распознавания им ошибочного вычислительного контекста исполнения команды (рис. 2(7)).

Из всего этого многообразия информации для построения входных спецификаций дизассемблера существенным является только содержимое таблицы описания мнемоники команд (рис. 2(2)) и таблицы кодирования операндов команды (рис. 2(3)). Из-за разных временных моментов создания и обновления документации описание команд представляется таблицами, отличающимися по строению друг от друга. В рассматриваемой версии документации описание представлено в таблицах шести видов (рис. 5, 6, 7, 8, 9, 10). Для некоторых таблиц показана связь между знаками сносок (используются различные знаки) и текстовым представлением необходимых частных. Большинство таких сносок указывает на ограничения в машинном коде, которые влияют на дизассемблированное представление команды. Эти ограничения необходимо учитывать, поскольку они позволяют отличить корректный машинный код команды от некорректного, который при попытке исполнения заставит процессор сгенерировать исключение #UD (Invalid Opcode/Undefined Opcode). Соответственно, дизассемблер, решая задачу разделения бинарного кода команд и данных в программе для машины фон Неймана, должен отличать корректный код от некорректного. И указанные детали

описания команд могут быть использованы для этого.

Описания команд у Intel сгруппированы в следующих разделах [\[6\]](#):

- Группа глав, описывающих общие команды архитектуры под общим заголовком "Instruction set reference". На момент написания статьи рассматривалось 4289 синтаксических нотаций команд (см. далее) этого вида совместно у Intel и у AMD. Из них оригинальными для Intel являются 2383 команды, оригинальными для AMD являются 252 команды, общие для Intel и AMD - это 1654 команды. И это подсчеты для команд, исключая дублирование, т.е., например, 'CMPSS m8, m8' и 'CMPSSB' считаются одной командой, т.к. машинный код у них одинаковый.
- Команды, относящиеся к расширению SAFER MODE, встречаются только у Intel и, по сути, представлены только одной командой GETSEC, входными параметрами для которой является значение регистров, задающих выполнения частной функциональности (leaf functions).
- Команды, уникальные для процессоров INTEL® XEON PHI™, встречаются только у Intel и это 33 команды.
- Команды поддержки расширения Virtual Machine Extensions (VMX) встречаются только у Intel и это 17 команд.
- Команды поддержки расширения Intel® Software Guard Extensions (Intel® SGX) встречаются только у Intel и это три команды, допускающие на основе входных значений выполнение leaf-функций, как и для GETSEC.

В документации от AMD [\[8\]](#) описание команды представлено на рис. 3 и включает:

- 1) заголовок, который как и у Intel включает мнемонику операции команды и краткое описание ее семантики;
- 2) описание операционной семантики в словесной форме;
- 3) синтаксические формы отдельных команд;
- 4) описание воздействия на флаги из регистра [R|E]FLAGS процессора;
- 5) различные дополнительные аспекты поведения процессора в случае распознавания им ошибочного вычислительного контекста исполнения команды.

Если сравнить эти два варианта описания команд, то предпочтительным вариантом для построения дизассемблера представляется описание от Intel. Во-первых, оно более детализированное и включает подробное описание синтаксиса машинного кода и режимов работы процессора, в которых допустимы те или иные команды. Во-вторых, оно содержит удобное для автоматизированного анализа описание кодирования отдельных операндов команд в систематическом виде, в отличие от словесного описания у AMD. В-третьих, документация от Intel содержит описание большего числа команд, в частности расширение Intel AVX-512, GETSEC, Intel Xeon PHI, VMX, SGX, которые не поддерживаются AMD.

Однако следует отметить, что описание используемых флагов регистра [R|E]FLAGS процессора в документации от AMD удобнее для автоматизированной обработки при составлении схемы информационных связей команд реальной программы, чем вариант от Intel.

Кроме того, у AMD описаны команды, которые отсутствуют в документации Intel (подробности далее). Поэтому спецификации множества описаний команд, создаваемое по документации Intel, можно дополнить описаниями команд от AMD и получить обобщенное множество команд для архитектуры x86.

И у Intel, и у AMD семантическая форма команды может быть как в префиксном представлении, например, ADD DEST, SRC, где ADD - обозначение операции, DEST, SRC - обозначения операндов приемника (**DEST**ination) и источника (**SouR**Ce) значений, так и в инфиксном представлении вида: DEST  $\leftarrow$  DEST + SRC (рис. 2(5)).

У обоих производителей процессоров одной семантической форме описания команды ADD DEST, SRC соответствует, как правило, несколько синтаксических форм вида: ADD AL, imm8; ADD AX, imm16; ...; ADD r64, r/m64. По числу синтаксических форм можно судить о мощности программного интерфейса процессора.

Например, в группе глав, описывающих общие команды архитектуры x86 у Intel [\[6\]](#), число семантических форм 764, а синтаксических форм 3975.

Синтаксические формы команд кодируются различными вариантами машинного кода. И поскольку для многих аспектов статьи будет необходимо ссылаться на конкретные примеры команд, то для этого будет использоваться нотация вида: O(pcode)='машинный код'; I(nstruction)='синтаксис команды'. Например, для команды на рис. 2 эта нотация будет иметь вид: O='04 ib'; I='ADD AL, imm8'. Для команды на рис. 3 эта нотация будет иметь вид: O='14 ib'; I='ADC AL, imm8' и т.д.

В нотации машинного кода числа представлены в шестнадцатеричном виде, т.е. в записи O='37'; I='AAA' число '37' - это число в шестнадцатеричном представлении. В некоторых случаях, когда будут упоминаться биты машинного кода, то они будут записаны суффиксом 'b' (binary), например, '101b' - это три бита '1', '0' и '1'.



1 **ADD—Add**

| Opcode        | Instruction    | Op/En | 64-bit Mode | Compat/Leg Mode | Description       |
|---------------|----------------|-------|-------------|-----------------|-------------------|
| 04 ib         | ADD AL, imm8   | I     | Valid       | Valid           | Add imm8 to AL.   |
| 05 iw         | ADD AX, imm16  | I     | Valid       | Valid           | Add imm16 to AX.  |
| 03 /r         | ADD r32, r/m32 | RM    | Valid       | Valid           | Add r/m32 to r32. |
| REX.W + 03 /r | ADD r64, r/m64 | RM    | Valid       | N.E.            | Add r/m64 to r64. |

2

NOTES:  
\*In 64-bit mode, r/m8 can not be encoded to access the following byte registers if a REX prefix is used: AH, BH, CH, DH.

3

| Op/En | Operand 1        | Operand 2     | Operand 3 | Operand 4 |
|-------|------------------|---------------|-----------|-----------|
| RM    | ModRM:reg (r, w) | ModRM:m (r)   | NA        | NA        |
| MR    | ModRM:m (r, w)   | ModRM:reg (r) | NA        | NA        |
| MI    | ModRM:m (r, w)   | imm8/16/32    | NA        | NA        |
| I     | AL/AX/EAX/RAX    | imm8/16/32    | NA        | NA        |

4

Description

Adds the destination operand (first operand) and the source operand (second operand) and then stores the result in the destination operand. The destination operand can be a register or a memory location; the source operand can be an immediate, a register, or a memory location. (However, two memory operands cannot be used in one instruction.) When an immediate value is used as an operand, it is sign-extended to the length of the destination operand format.

5

Operation

DEST ← DEST + SRC;

6

Flags Affected

The OF, SF, ZF, AF, CF, and PF flags are set according to the result.

7

Protected Mode Exceptions

#GP(0) If the destination is located in a non-writable segment.

Real-Address Mode Exceptions

#GP If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

Virtual-8086 Mode Exceptions

#GP(0) If a memory operand effective address is outside the CS, DS, ES, FS, or GS segment limit.

Рис. 2. Типовой вид описания команды в документации Intel

Fig. 2. A typical view of the instruction description in the Intel documentation



1

ADC

Add with Carry

Adds the carry flag (CF), the value in a register or memory location (first operand), and an immediate value or the value in a register or memory location (second operand), and stores the result in the first operand location.

2

The instruction has two operands:

ADC *dest, src*

The instruction cannot add two memory operands. The CF flag indicates a pending carry from a previous addition operation. The instruction sign-extends an immediate value to the length of the destination register or memory location.

This instruction evaluates the result for both signed and unsigned data types and sets the OF and CF flags to indicate a carry in a signed or unsigned result, respectively. It sets the SF flag to indicate the sign of a signed result.

Use the ADC instruction after an ADD instruction as part of a multibyte or multiword addition.

The forms of the ADC instruction that write to memory support the LOCK prefix. For details about the LOCK prefix, see “Lock Prefix” on page 11.

3

| Mnemonic                            | Opcode          | Description  |
|-------------------------------------|-----------------|--|
| ADC AL, <i>imm8</i>                 | 14 <i>ib</i>    | Add <i>imm8</i> to AL + CF.                              |
| ADC AX, <i>imm16</i>                | 15 <i>iw</i>    | Add <i>imm16</i> to AX + CF.                             |
| ADC EAX, <i>imm32</i>               | 15 <i>id</i>    | Add <i>imm32</i> to EAX + CF.                            |
| ADC RAX, <i>imm32</i>               | 15 <i>id</i>    | Add sign-extended <i>imm32</i> to RAX + CF.              |
| ADC <i>reg/mem8</i> , <i>imm8</i>   | 80 /2 <i>ib</i> | Add <i>imm8</i> to <i>reg/mem8</i> + CF.                 |
| ADC <i>reg/mem16</i> , <i>imm16</i> | 81 /2 <i>iw</i> | Add <i>imm16</i> to <i>reg/mem16</i> + CF.               |
| ADC <i>reg/mem32</i> , <i>imm32</i> | 81 /2 <i>id</i> | Add <i>imm32</i> to <i>reg/mem32</i> + CF.               |
| ADC <i>reg/mem64</i> , <i>imm32</i> | 81 /2 <i>id</i> | Add sign-extended <i>imm32</i> to <i>reg/mem64</i> + CF. |
| ADC <i>reg/mem16</i> , <i>imm8</i>  | 83 /2 <i>ib</i> | Add sign-extended <i>imm8</i> to <i>reg/mem16</i> + CF.  |
| ADC <i>reg/mem32</i> , <i>imm8</i>  | 83 /2 <i>ib</i> | Add sign-extended <i>imm8</i> to <i>reg/mem32</i> + CF.  |
| ADC <i>reg/mem64</i> , <i>imm8</i>  | 83 /2 <i>ib</i> | Add sign-extended <i>imm8</i> to <i>reg/mem64</i> + CF.  |
| ADC <i>reg/mem8</i> , <i>reg8</i>   | 10 / <i>r</i>   | Add <i>reg8</i> to <i>reg/mem8</i> + CF.                 |
| ADC <i>reg/mem16</i> , <i>reg16</i> | 11 / <i>r</i>   | Add <i>reg16</i> to <i>reg/mem16</i> + CF.               |
| ADC <i>reg/mem32</i> , <i>reg32</i> | 11 / <i>r</i>   | Add <i>reg32</i> to <i>reg/mem32</i> + CF.               |
| ADC <i>reg/mem64</i> , <i>reg64</i> | 11 / <i>r</i>   | Add <i>reg64</i> to <i>reg/mem64</i> + CF.               |
| ADC <i>reg8</i> , <i>reg/mem8</i>   | 12 / <i>r</i>   | Add <i>reg/mem8</i> to <i>reg8</i> + CF.                 |
| ADC <i>reg16</i> , <i>reg/mem16</i> | 13 / <i>r</i>   | Add <i>reg/mem16</i> to <i>reg16</i> + CF.               |

| Mnemonic                            | Opcode        | Description                                |
|-------------------------------------|---------------|--|
| ADC <i>reg32</i> , <i>reg/mem32</i> | 13 / <i>r</i> | Add <i>reg/mem32</i> to <i>reg32</i> + CF. |
| ADC <i>reg64</i> , <i>reg/mem64</i> | 13 / <i>r</i> | Add <i>reg/mem64</i> to <i>reg64</i> + CF. |

4

Related Instructions

ADD, SBB, SUB

rFLAGS Affected

| ID | VIP | VIF | AC | VM | RF | NT | IOPL  | OF | DF | IF | TF | SF | ZF | AF | PF | CF |
|----|-----|-----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|
|    |     |     |    |    |    |    |       | M  |    |    |    | M  | M  | M  | M  | M  |
| 21 | 20  | 19  | 18 | 17 | 16 | 14 | 13:12 | 11 | 10 | 9  | 8  | 7  | 6  | 4  | 2  | 0  |

Note: Bits 31:22, 15, 5, 3, and 1 are reserved. A flag set to 1 or cleared to 0 is M (modified). Unaffected flags are blank. Undefined flags are U.

5

Exceptions

| Exception               | Real | Virtual 8086 | Protected | Cause of Exception  |
|-------------------------|------|--------------|-----------|---|
| Stack, #SS              | X    | X            | X         | A memory address exceeded the stack segment limit or was non-canonical.           |
| General protection, #GP | X    | X            | X         | A memory address exceeded a data segment limit or was non-canonical.              |
|                         |      |              | X         | The destination operand was in a non-writable segment.                            |
|                         |      |              | X         | A null data segment was used to reference memory.                                 |
| Page fault, #PF         |      | X            | X         | A page fault resulted from the execution of the instruction.                      |
| Alignment check, #AC    |      | X            | X         | An unaligned memory reference was performed while alignment checking was enabled. |

Рис. 3 Типовой вид описания legacy-команды в документации AMD

Fig. 3. A typical description of the legacy instruction in the AMD documentation

1

ADDSS

VADDSS

Scalar Single-Precision Floating-Point

Adds the single-precision floating-point value in the low-order doubleword of the first source operand to the corresponding value in the low-order doubleword of the second source operand and writes the result into the low-order doubleword of the destination.

There are legacy and extended forms of the instruction:

**ADDSS**

The first source operand is an XMM register and the second source operand is either an XMM register or a 32-bit memory location. The first source register is also the destination. Bits [127:32] of the destination register and bits [255:128] of the corresponding YMM register are not affected.

**VADDSS**

The extended form of the instruction has a 128-bit encoding only.

The first source operand is an XMM register and the second source operand is either an XMM register or a 32-bit memory location. The destination is a third XMM register. Bits [127:32] of the first source register are copied to bits [127:32] of the of the destination. Bits [255:128] of the YMM register that corresponds to the destination are cleared.

**Instruction Support**

| Form   | Subset | Feature Flag                        |
|--------|--------|-------------------------------------|
| ADDSS  | SSE1   | CPUID Fn0000_0001_EDX[SSE] (bit 25) |
| VADDSS | AVX    | CPUID Fn0000_0001_ECX[AVX] (bit 28) |

For more on using the CPUID instruction to obtain processor feature support information, see Appendix E of Volume 3.

**Instruction Encoding**

3

Mnemonic

ADDSS *xmm1, xmm2/mem32*

Opcode

F3 0F 58 /r

Description

Adds a single-precision floating-point value in the low-order doubleword of *xmm1* to a corresponding value in *xmm2* or *mem32*. Writes results to *xmm1*.

3

Mnemonic

VADDSS *xmm1, xmm2, xmm3/mem32*

Encoding

VEX

C4

RXB.map\_select

RXB.00001

W.vvvv.L.pp

X.src.X.10

Opcode

58 /r

**Related Instructions**

(V)ADDPD, (V)ADDPS, (V)ADDSD

4

rFLAGS Affected

None

## MXCSR Flags Affected

| MM | FZ | RC | PM | UM | OM | ZM | DM | IM | DAZ | PE | UE | OE | ZE | DE | IE |   |
|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|---|
|    |    |    |    |    |    |    |    |    |     | M  | M  | M  |    | M  | M  |   |
| 17 | 15 | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7   | 6  | 5  | 4  | 3  | 2  | 1  | 0 |

Note: M indicates a flag that may be modified (set or cleared). Blanks indicate flags that are not affected.

## Exceptions

| Exception                             | Mode |      |      | Cause of Exception  |
|---------------------------------------|------|------|------|---|
|                                       | Real | Virt | Prot |   |
| Invalid opcode, #UD                   | X    | X    | X    | Instruction not supported, as indicated by CPUID feature identifier.  |
|                                       | A    | A    |      | AVX instructions are only recognized in protected mode.   |
|                                       | S    | S    | S    | CR0.EM = 1.   |
|                                       | S    | S    | S    | CR4.OSFXSR = 0.   |
|                                       |      |      | A    | CR4.OSXSAVE = 0, indicated by CPUID Fn0000_0001_ECX[OSXSAVE].   |
|                                       |      |      | A    | XFEATURE_ENABLED_MASK[2:1] != 11b.  |
|                                       |      |      | A    | REX, F2, F3, or 66 prefix preceding VEX prefix.   |
|                                       | S    | S    | X    | Lock prefix (F0h) preceding opcode.   |
| Device not available, #NM             | S    | S    | X    | Unmasked SIMD floating-point exception while CR4.OSXMMEXCPT = 0, see <i>SIMD Floating-Point Exceptions</i> below for details. |
| Stack, #SS                            | S    | S    | X    | CR0.TS = 1.   |
| General protection, #GP               | S    | S    | X    | Memory address exceeding stack segment limit or non-canonical.  |
|                                       |      |      | X    | Memory address exceeding data segment limit or non-canonical.   |
| Page fault, #PF                       |      | S    | X    | Null data segment used to reference memory.   |
| Alignment check, #AC                  |      | S    | X    | Instruction execution caused a page fault.  |
| SIMD floating-point, #XF              | S    | S    | X    | Unaligned memory reference when alignment checking enabled.   |
| <b>SIMD Floating-Point Exceptions</b> |      |      |      |   |
| Invalid operation, IE                 | S    | S    | X    | Unmasked SIMD floating-point exception while CR4.OSXMMEXCPT = 1, see <i>SIMD Floating-Point Exceptions</i> below for details. |
|                                       | S    | S    | X    | A source operand was an SNaN value.   |
| Denormalized operand, DE              | S    | S    | X    | Undefined operation.  |
| Overflow, OE                          | S    | S    | X    | A source operand was a denormal value.  |
| Underflow, UE                         | S    | S    | X    | Rounded result too large to fit into the format of the destination operand.   |
| Precision, PE                         | S    | S    | X    | Rounded result too small to fit into the format of the destination operand.   |
|                                       |      |      |      | A result could not be represented exactly in the destination format.  |

X — AVX and SSE exception

A — AVX exception

S — SSE exception

Рис. 4 Типовой вид описания VEX- или XOP-команд в документации AMD

Fig. 4. A typical description of VEX or XOP instructions in AMD documentation

**ADD—Add**

| Opcode        | Instruction    | Op/En | 64-bit Mode | Compat/Leg Mode | Description       |
|---------------|----------------|-------|-------------|-----------------|-------------------|
| 04 ib         | ADD AL, imm8   | I     | Valid       | Valid           | Add imm8 to AL.   |
| REX + 00 /r   | ADD r/m8, r8   | MR    | Valid       | N.E.            | Add r8 to r/m8.   |
| REX.W + 03 /r | ADD r64, r/m64 | RM    | Valid       | N.E.            | Add r/m64 to r64. |

**NOTES:**  
\*In 64-bit mode, r/m8 can not be encoded to access the following byte registers if a REX prefix is used: AH, BH, CH, DH.

Рис. 5. Пример таблицы описания команд вида 1. Формат используется для описания команд ранних версий процессоров x86 (legacy-команд)

Fig. 5. An example of a instruction description table of the form 1. The format is used to describe instructions from early versions of x86 processors (legacy instructions)

**ADCX—Unsigned Integer Addition of Two Operands With Carry Flag**

| Opcode/ Instruction               | Op/En | 64/32bit Mode Support | CPUID Feature Flag | Description  |
|-----------------------------------|-------|-----------------------|--------------------|--|
| 66 0F 38 F6 /r<br>ADCX r32, r/m32 | RM    | V/V                   | ADX                | Unsigned addition of r32 with CF, r/m32 to r32, writes CF. |

Рис. 6. Пример таблицы описания команд вида 2. Формат используется для различных команд

Fig. 6. An example of a instruction description table of type 2. The format is used for various instructions



F2XM1—Compute 2<sup>x</sup>-1

| Opcode | Instruction | 64-Bit Mode | Compat/Leg Mode | Description                                  |
|--------|-------------|-------------|-----------------|--|
| 09 F0  | F2XM1       | Valid       | Valid           | Replace ST(0) with (2 <sup>ST(0)</sup> - 1). |

Рис. 7. Пример таблицы описания команд вида 3. Формат используется для команд математического сопроцессора (Numeric Data Processor, NDP)

Fig. 7. An example of a instruction description table of the form 3. The format is used for instructions of a mathematical coprocessor (Numeric Data Processor, NDP)

XACQUIRE/XRELEASE—Hardware Lock Elision Prefix Hints

| Opcode/Instruction | 64/32bit Mode Support | CPUID Feature Flag | Description   |
|--------------------|-----------------------|--------------------|---|
| F2 XACQUIRE        | V/V                   | HLE <sup>1</sup>   | A hint used with an "XACQUIRE-enabled" instruction to start lock elision on the instruction memory operand address. |

Рис. 8. Пример таблицы описания команд вида 4. Формат используется для различных команд

Fig. 8. An example of a instruction description table of the form 4. The format is used for various instructions

GETSEC[CAPABILITIES] - Report the SMX Capabilities

| Opcode             | Instruction          | Description   |
|--------------------|----------------------|---|
| NP OF 37 (EAX = 0) | GETSEC[CAPABILITIES] | Report the SMX capabilities.<br>The capabilities index is input in EBX with the result returned in EAX. |

Рис. 9. Пример таблицы описания команд вида 5. Формат используется для команд семейства GETSEC

Fig. 9. An example of a instruction description table of the form 5. The format is used for instructions of the GETSEC family

INVEPT— Invalidate Translations Derived from EPT

| Opcode/Instruction              | Op/En | Description  |
|---------------------------------|-------|--|
| 66 OF 38 80<br>INVEPT r64, m128 | RM    | Invalidates EPT-derived entries in the TLBs and paging-structure caches (in 64-bit mode).      |
| 66 OF 38 80<br>INVEPT r32, m128 | RM    | Invalidates EPT-derived entries in the TLBs and paging-structure caches (outside 64-bit mode). |

Рис. 10. Пример таблицы описания команд вида 6. Формат используется для команд VMX

Fig. 10. An example of a instruction description table of the form 6. The format is used for VMX instructions

| a | Mnemonic                              | Encoding |                |             | Opcode |
|---|---------------------------------------|----------|----------------|-------------|--------|
|   |                                       | VEX      | RXB.map_select | W.vvvv.L.pp |        |
|   | VADDPD <i>xmm1, xmm2, xmm3/mem128</i> | C4       | RXB.00001      | X.arc.0.01  | 58 /r  |
|   | VADDPD <i>ymm1, ymm2, ymm3/mem256</i> | C4       | RXB.00001      | X.arc.1.01  | 58 /r  |

| 6 | Mnemonic                         | Encoding |                |             | Opcode |
|---|----------------------------------|----------|----------------|-------------|--------|
|   |                                  | XOP      | RXB.map_select | W.vvvv.L.pp |        |
|   | VFRCZPD <i>xmm1, xmm2/mem128</i> | 8F       | RXB.09         | 0.1111.0.00 | 81 /r  |
|   | VFRCZPD <i>ymm1, ymm2/mem256</i> | 8F       | RXB.09         | 0.1111.1.00 | 81 /r  |

Рис. 11. Пример таблицы описаний (а) VEX-, (б) XOP-команд в документации AMD

Fig. 11. An example of a table of descriptions of (a) VEX, (b) XOP instructions in the AMD

documentation

По строению машинного кода и по нотации его описания команды Intel можно разделить на варианты:

- EVEX-команды. В нотации машинного кода присутствует префикс 'EVEX': O='EVEX.512.66.0F38.WIG DF /r; I=VAESDECLAST zmm1, zmm2, zmm3/m512';
- VEX-команды. В нотации машинного кода присутствует префикс 'VEX': O='VEX.256.66.0F38.WIG DF /r; I=VAESDECLAST ymm1, ymm2, ymm3/m256';
- REX-команды. В нотации машинного кода присутствует префикс 'REX': O='F2 REX 0F 38 F0 /r; I=CRC32 r32, r/m8';
- Legacy-команды. В нотации машинного кода не присутствует на один из указанных префиксов: O='0F 38 F0 /r; I=MOVBE r16, m16'.

Еще одной особенностью описания команд от Intel является указание на некоторые ограничения/исключения в строении машинного кода команд. Такие указания, как правило, передаются сноской из элемента таблицы мнемоник (табл. 5), в которой ссылка может быть указана как на столбец таблицы целиком, так и на отдельные значения в ее ячейках. Учесть такие ограничения довольно легко в постобработке дизассемблированной команды (п. 9).

В части legacy-команд документация от AMD придерживается представления, схожего с Intel'овским, но в отношении остальных команд, например, VEX-команд, AMD использует для описания табличное представление (рис. 11,а), что осложняет сопоставление с Intel'овским эквивалентом. По нотации машинного кода от AMD, представленной в виде таблицы (рис. 11), можно сгенерировать представление, похожее на то, какое использует Intel. Но это будет лишь примерный вариант, поскольку Intel не придерживается строгих правил для обозначения машинного кода команд. Так, O='VEX.128.0F38.W0 F5 /r'; I='BZHI reg32, reg/mem32, reg32' в варианте от AMD будет соответствовать O='VEX.LZ.0F38.W0 F5 /r'; I='BZHI r32a, r/m32, r32b' в варианте от Intel.

Кроме того, в документации от AMD присутствует описание команд, которые не встречаются у Intel. К числу их относятся:

- команды поддержки технологии 3DNow! Машинный код их похож на legacy-строение. Например, O='0F 0F /r 0C'; I='PI2FW mmx1, mmx2/mem64'. Отличие касается байтов кода операции - они не размещаются на смежных позициях;
- XOP-команды, включающие префикс XOP, имеющие строение, похожее на VEX-команды (рис. 11,б), но все же различающиеся в деталях (здесь Intel-эквивалента нет вообще, поэтому команды представлены в авторской интерпретации). Например, O='XOP.128.MAP9.W0 81 /r'; I='VFRCPD xmm1, xmm2/mem128';
- некоторые VEX-команды, не описанные у Intel. Например, O='VEX.128.66.0F3A.W0 78 /r /is4'; I='VFNMADDPS xmm1, xmm2, xmm3/mem128, xmm4'. В этой же категории присутствуют команды с пятью операндами. Такого вида команды вообще не встречаются у Intel. Это, например, O='VPERMIL2PD xmm1, xmm2, xmm3/mem128, xmm4, m2z'; I='VEX.128.66.0F3A.W0 49 /r is5';
- некоторые команды legacy-формата, не описанные у Intel. Например, O='0F 77'; I='EMMS'.

#### 4. Некоторые элементы таблиц описания команд

Таблицы синтаксического описания команды (таблицы мнемоник) в документации Intel (рис. 5, 6, 7, 8, 9, 10) имеют следующие заголовки столбцов:

- "Opcode" - заголовок столбца, содержащего нотацию машинного кода команды. Чаще всего является отдельным столбцом (рис. 5, 6, 7), но может быть совмещен с описанием синтаксиса команды - со столбцом "Instruction" (рис. 8).
- "Instruction" - заголовок столбца, содержащего синтаксическую нотацию команды. Чаще всего является отдельным столбцом (рис. 5, 6, 7), но может быть совмещен с описанием синтаксиса команды - со столбцом "Opcode" (рис. 8).
- "Op/En" - заголовок столбца, содержащего отсылку к строкам таблицы "Instruction Operation Encoding", следующей за таблицей описания синтаксиса команды (рис. 2(3)).
- "64-bit Mode" или "64/32bit Mode Support" - заголовок столбца указывает на поддержку команды в 64-битном и/или 32-битном режимах работы процессора. Значения, представленные в таких столбцах, описаны в п. 6.
- "Compatibility/Legacy Mode" - заголовок столбца указывает на поддержку команды в режиме совместимости и/или указание на вхождение в семейство legacy-команд IA-32. Значения, представленные в таких столбцах, описаны в п. 6.
- "CPUID Feature Flag" - заголовок столбца указывает на флаг в значениях, выдаваемых командой `0F A2`; `I='CPUID'` процессора, и указывает на поддержку в процессоре определенных технологий обработки данных. Значения, представленные в таких столбцах, разбираются в п. 7.
- Столбец "Description" содержит краткое словесное описание семантики вычислений, выполняемых командой. Примеры такого описания представлены в рис. 5, 6, 7, 8, 9, 10.

#### 5. Кодирование операндов в описании команд

Кодировка операндов каждой команды специальным образом описывается в отдельной таблице, следующей за таблицей описания команд данного вида (рис. 12). Кодировка операндов в исходной таблице представлена в столбце "Op/En" и включает специальные буквенные обозначения, адресующие к методу кодирования операндов в отдельной команде. Значения в столбце "Op/En" (рис. 12(1)) применяются только для соотнесения с описаниями в следующей за ней таблице кодирования операндов (рис. 12(2)). И в этой таблице (Согласно п. "3.1.1.4 Operand Encoding Column in the Instruction Summary Table" [6]) обозначение "r" в паре круглых скобок указывает на чтение операнда, обозначение "w" в паре круглых скобок указывает на запись в этот операнд при совершении вычислений процессором.

Примеры многообразных описаний кодирования операндов в документации Intel представлены в столбце 1 (табл. 2), а полный набор обозначений типов кодирования, которые используются для построения спецификации команд в авторской обобщенной нотации, представлен в столбце 2 (табл. 2). Таким образом, многообразие текстовых описаний от Intel может быть сведено к восьми типам декодирования в дизассемблере.

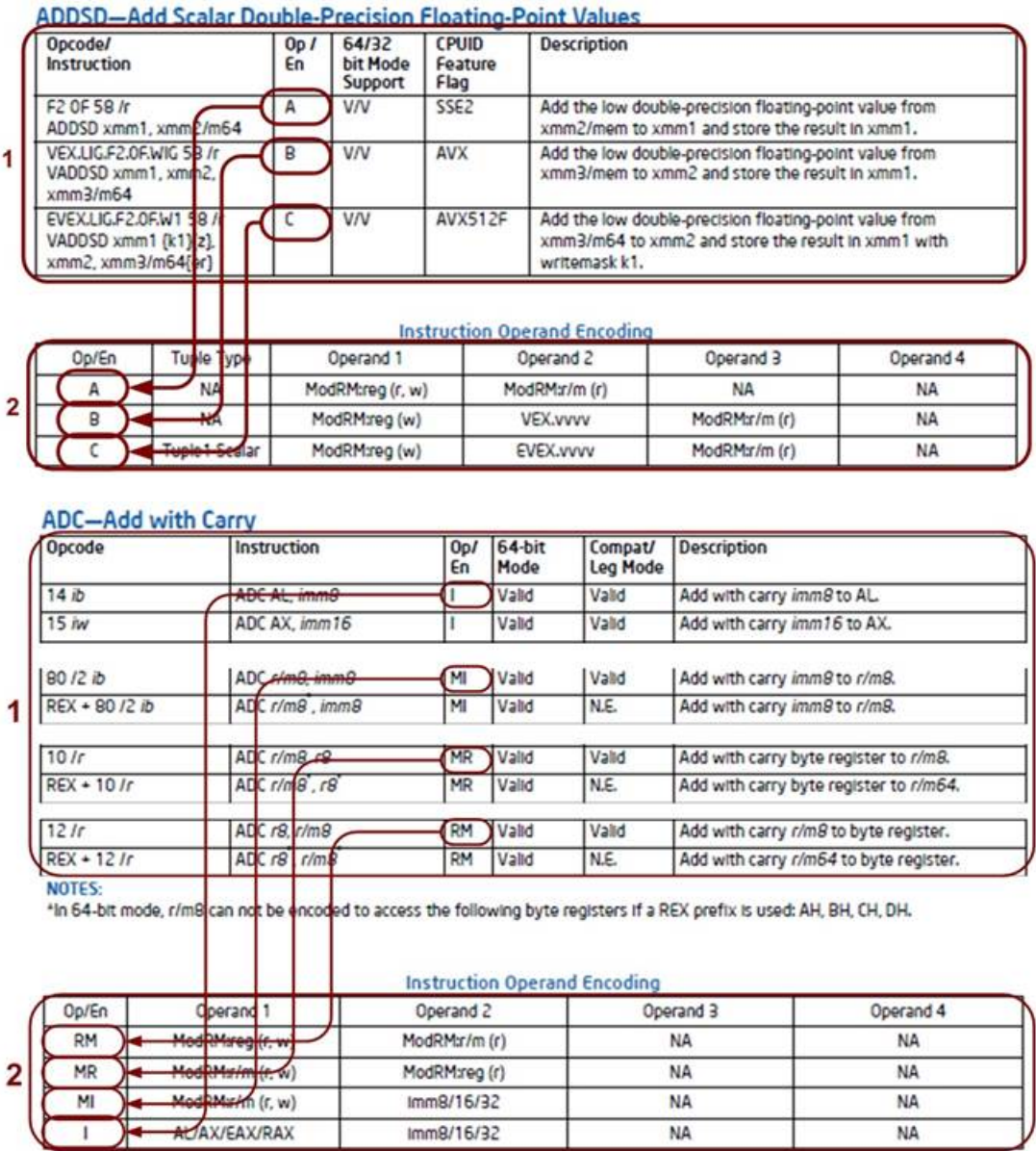


Рис. 12. Основные таблицы (1) описания команды с семантическими мнемониками ADDSD и ADC, и таблицы (2) "Instruction Operand Encoding" кодирования операндов в документации Intel. Показаны связи значений из столбца "Op/En" основных таблиц (1) и таблиц (2)

Fig. 12. The main tables (1) describe instructions with semantic mnemonics ADD and ADC, and tables (2) "Instruction Operand Encoding" encoding operands in Intel documentation. The relationships of the values from the "Op/En" column of the main tables (1) and tables (2) are shown

Табл. 2. Представление вариантов кодирования операндов и эквивалентные типы декодеров операндов

Tabl. 2. Representation of operand encoding options and equivalent types of operand decoders

| Операнд в описании Intel | Тип декодера операнда в дизассемблере | Пример команды            |
|--------------------------|---------------------------------------|---------------------------|
| 1                        | 2                                     | 3                         |
| AI /AX/EAX/RAX           |                                       | O='RFX.W + 15 id'; I='ADC |

|  |        |   |
|--|--------|---|
| ST(0)  | Value  | RAX, imm32'<br><br>O='DC C0+i'; I='FADD ST(i), ST(0)'   |
| ModRM:r/m;<br>ModRM:rm<br><br>BaseReg(R): VSIB<br><br>ST(i)  | ModRM  | O='NP 0F AE /2'; I='LDMXCSR m32'<br><br>O='NP 0F 1A /r'; I='BNDLDX bnd, mib'<br><br>O='D8 C0+i '; I='FADD ST(0), ST(i)'           |
| ModRM:reg  | ModReg | O='VEX.L0.F2.0F.W0 93 /r'; I='KMOVD r32, k1'  |
| VEX.vvvv<br><br>VEX.1vvv<br><br>EVEX.vvvv                    | Vvvv   | O='VEX.L1.66.0F.W1 4A /r'; I='KADDD k1, k2, k3'<br><br>O='EVEX.128.66.0F3A.W0 21 /r ib'; I='VINSERTPS xmm1, xmm2, xmm3/m32, imm8' |
| imm/imm8/16/32<br><br>offset<br><br>Segment+Absolute Address | Imm    | O='REX.W + 15 id'; I='ADC RAX, imm32'<br><br>O='0F 84 cd'; I='JZ rel32'<br><br>O='EA cp'; I='JMP ptr16:32'                        |
| opcode + rd  | Opcode | O='40+ rd'; I='INC r32'   |
| Moffs  | Moffs  | O='A1'; I='MOV EAX,moffs32'   |
| NA<br><br>implicit   | None   | O='NP 0F 38 CB /r'; I='SHA256RND\$2 xmm1,xmm2/m128, '   |

Отметим, что Intel в документации не всегда придерживается единообразия в описании кодировки операндов. Таблица кодирования операндов вместо типового представления (рис. 12(2)) может иметь иной вид (рис.13), в нем операнды 2-9 (Operands 2-9) являются неявными и потому в дизассемблированном представлении команды не указываются. Поскольку в рассматриваемой версии документации только шесть таких случаев описания команд, то они правкой вручную могут быть сведены к типовому описанию (рис.14).

| AESDECWIDE128KL—Perform Ten Rounds of AES Decryption Flow With Key Locker on 8 Blocks Using 128-Bit Key |        |                |                    |   |
|---|--------|----------------|--------------------|---|
| Opcode/ Instruction   | Op/ En | 64/32-bit Mode | CPUID Feature Flag | Description   |
| F3 0F 38 D8 {11}:001:bbb<br>AESDECWIDE128KL m384, <XMM0-7>  | A      | V/V            | AESKLEWIDE_KL      | Decrypt XMM0-7 using 128-bit AES key indicated by handle at m384 and store each resultant block back to its corresponding register. |

| Instruction Operand Encoding |       |               |                        |
|------------------------------|-------|---------------|------------------------|
| Op/En                        | Tuple | Operand 1     | Operands 2—9           |
| A                            | N/A   | ModRM:r/m (r) | Implicit XMM0-7 (r, w) |

Рис. 13. Нетипичное описание операндов команды

Fig. 13. An atypical description of the instruction operands



| AESDECWIDE128KL - Perform Ten Rounds of AES Decryption Flow With Key Locker on 8 Blocks Using 128-Bit Key |                      |               |           |                |                    |   |
|---|----------------------|---------------|-----------|----------------|--------------------|---|
| Opcode  | Instruction          |               | Op/En     | 64/32-bit Mode | CPUID Feature Flag | Description   |
| F3 0F 38 D8<br>1{11}001bbb  | AESDECWIDE128KL m384 |               | A         | V/V            | AESKLWIDE_KL       | Decrypt XMM0-7 using 128-bit AES key indicated by handle at m384 and store each resultant block back to its corresponding register. |
| Instruction Operand Encoding  |                      |               |           |                |                    |   |
| Op/En   | Tuple                | Operand 1     | Operand 2 | Operand 3      | Operand 4          |   |
| A   | N/A                  | ModRM.r/m (r) | N/A       | N/A            | N/A                |   |

Рис. 14. Отредактированное описание нетипичного описания команды и ее операндов для исходной версии на рис. 13

Fig. 14. An edited description of the atypical description of the instruction and its operands for the original version in Fig. 13

6. Совместимость команд с режимами работы процессора

Столбец "64/32-bit Mode" указывает на поддержку команды в 64-битном режиме (64-bit mode), в режиме совместимости (Compatibility mode) и других режимах IA-32, которые применяются в сочетании с флагом, возвращаемым командой O='0F A2'; I='CPUID', связанным с конкретными расширениями функциональности команд. Значения в столбце (табл. 3) показаны согласно п. "3.1.1.5 64/32-bit Mode Column in the Instruction Summary Table" [6].

Табл. 3. Значения столбца "64/32-bit Mode" на рис. 2(2)

Tabl. 3. The values of the column "64/32-bit Mode" in Fig. 2(2)

| Обозначение         | Описание   |
|---------------------|--|
| V или Valid         | Команда поддерживается (Valid) в данном режиме   |
| I, Invalid или Inv. | Команда не поддерживается (Invalid) в данном режиме  |
| NE или N.E.         | Указывает на то, что команда не поддерживается (Not encodable) в 64-битном режиме, но может быть частью поддерживаемых команд в других режимах   |
| N.S.                | Указывает на синтаксис команды, которой требуется префикс замены адреса в 64-битном режиме. Использование префикса замены адреса в 64-битном режиме может привести к специфическим результатам, зависящим от конкретной модели процессора. |

Возможные значения столбца "Compatibility/Legacy Mode" представлены в табл. 4.

Табл. 4. Значения столбца "Compatibility/Legacy Mode" на рис. 2(2)

Fig. 4. Values of the "Compatibility/Legacy Mode" column in Fig. 2(2)

| Обозначение | Описание  |
|-------------|---|
| V           | Команда поддерживается (Valid) в данном режиме  |
| I           | Команда не поддерживается (Invalid) в данном режиме   |
| NE или N.E. | Указывает на то, что команда не поддерживается (Not encodable) в 64-битном режиме, машинный |

|  |   |
|--|---|
|  | код команды не допускается в режиме совместимости или в режиме IA-32. Машинный код может представлять допустимую команду семейства IA-32 legacy команд. |
|--|---|

7. Нотация вычислительных опций процессора

Согласно п. "3.1.1.6 CPUID Support Column in the Instruction Summary Table"<sup>[6]</sup> значения в столбце "CPUID Feature Flag" (рис. 15) передают поддержку процессором соответствующей технологии. Значение в столбце эквивалентно названию флага, который устанавливается в данных, выдаваемых процессором при исполнении команды O='0F A2'; I='CPUID'. Иначе говоря, если процессор поддерживает некоторую технологию вычислений, то соответствующий флаг/бит установлен.

Значения этого столбца, например, 'ADX' на рис. 15, могут быть отражены на множество значений, обозначающих технологии обработки данных, реализованных в процессорах (табл. 1). Соответственно, задавая фильтрацию команд по технологиям, отмеченным в столбце "CPUID Feature Flag", в дизассемблере можно реализовать проверку кода анализируемой программы на поддержку соответствующей технологии обработки данных.

ADCX — Unsigned Integer Addition of Two Operands with Carry Flag

| Opcode/<br>Instruction                  | Op/<br>En | 64/32bit<br>Mode<br>Support | CPUID<br>Feature<br>Flag | Description  |
|---|-----------|-----------------------------|--------------------------|--|
| 66 0F 38 F6 /r<br>ADCX r32, r/m32       | RM        | V/V                         | ADX                      | Unsigned addition of r32 with CF, r/m32 to r32, writes CF. |
| 66 REX.w 0F 38 F6 /r<br>ADCX r64, r/m64 | RM        | V/NE                        | ADX                      | Unsigned addition of r64 with CF, r/m64 to r64, writes CF. |

Рис. 15. Пример значений в столбце "CPUID Feature Flag"

Fig. 15. Example of values in the "CPUID Feature Flag" column

8. Некоторые типовые ошибки, выявляемые в документации Intel и AMD

Детальный анализ показал, что ни одну из версий документации любого производителя невозможно рассматривать как эталон. И у Intel, и у AMD в описаниях команд выявляются синтаксические и семантические ошибки. Некоторые из них удастся выявлять и исправлять при перекрестном сопоставлении описаний команд обоих производителей.

Далее представлены некоторые типовые ошибки, которые приходится исправлять в тексте документации при составлении входной спецификации для построения дизассемблера.

8.1 Ошибочное кодирование операнда в документации Intel

В описании команды O='VEX.128.66.0F.WIG 12 /r'; I='VMOVLPD xmm2, xmm1, m64' кодировки операндов описаны неверно: одновременно указано кодирование через ModRM:r/m и для Operand 1, и для Operand 3 в "Instruction Operand Encoding".

| MOVLPD—Move Low Packed Double Precision Floating-Point Value |            |                              |                          |   |           |
|--|------------|------------------------------|--------------------------|---|-----------|
| Opcode/<br>Instruction                                       | Op / En    | 64/32 bit<br>Mode<br>Support | CPUID<br>Feature<br>Flag | Description   |           |
| VEX.128.66.0F.WIG 12 /r<br>VMOVLPD xmm2, xmm1, m64           | B          | V/V                          | AVX                      | Merge double precision floating-point value from m64 and the high quadword of xmm1. |           |
| Instruction Operand Encoding                                 |            |                              |                          |   |           |
| Op/En  | Tuple Type | Operand 1                    | Operand 2                | Operand 3   | Operand 4 |
| B  | N/A        | ModRM:rr/m (r)               | VEX.vvvv (r)             | ModRM:rr/m (r)  | N/A       |

8.2 Отсутствие способа кодирования операнда в документации Intel

В описании команды O='EVEX.LLIG.66.0F3A.W0 0A /r ib'; I='VRNDSCALESS xmm1 {k1}{z} xmm2, xmm3/m32{sae}, imm8' кодировка операнда Operand 4 в таблице "Instruction Operand Encoding" не указана вообще, тогда как в синтаксисе команды указан операнд imm8. Это также касается команд VSHUFF64, VSHUFF64X2, VSHUFF32, VSHUFF32X4.

| VRNDSCALESS—Round Scalar Float32 Value to Include a Given Number of Fraction Bits       |               |                              |                          |  |           |
|---|---------------|------------------------------|--------------------------|--|-----------|
| Opcode/<br>Instruction  | Op/<br>En     | 64/32<br>bit Mode<br>Support | CPUID<br>Feature<br>Flag | Description  |           |
| EVEX.LLIG.66.0F3A.W0 0A /r ib<br>VRNDSCALESS xmm1 {k1}{z}, xmm2,<br>xmm3/m32{sae}, imm8 | A             | V/V                          | AVX512F                  | Rounds scalar single-precision floating-point value in<br>xmm3/m32 to a number of fraction bits specified by the<br>imm8 field. Stores the result in xmm1 register under<br>writemask. |           |
| Instruction Operand Encoding  |               |                              |                          |  |           |
| Op/En   | Tuple Type    | Operand 1                    | Operand 2                | Operand 3  | Operand 4 |
| A   | Tuple1 Scalar | ModRM:reg (w)                | EVEX.vvvv (r)            | ModRM:rr/m (r)   | N/A       |

8.3 Неверная нотация машинного кода команд в документации AMD

Для некоторых команд в документации AMD машинный код указан неверно. Например, для команды I='VBLENDVPD xmm1, xmm2, xmm3/mem128, xmm4' у AMD представлен код в нотации O='VEX.128.66.0F3A.WIG 4B /r', тогда как по документации Intel верный код: O='VEX.128.66.0F3A.W0 4B /r /is4'. Отсутствует операнд '/is4'.

8.4 Неверная нотация мнемоники команд в документации AMD

Для некоторых команд в документации AMD указана неверно мнемоника. Например, для команды I='VPINSRB xmm, reg/mem8, xmm, imm8'; O=' VEX.128.66.0F3A.WIG 20 /r ib' мнемоника указана неверно. У Intel эта команда представлена в правильной нотации I='VPINSRB xmm1, xmm2, r32/m8, imm8'. Операнд 'r32/m8' должен быть третьим по порядку операндом.

В команде I=**VMOVMSKB** reg64, xmm1'; O='VEX.128.66.0F.WIG D7 /r' неверно указана мнемоника кода операции. У Intel представлен правильный вариант I=**VPMOVMSKB** reg64,xmm1'.

Для команды с O='0F 38 CD /r' мнемоника у AMD представлена в виде I=**SHA256MSG1** xmm1, xmm2/m128' тогда как правильный вариант у Intel I=**SHA256MSG2** xmm1, xmm2/m128'.

9. Обобщенное описание команд и его использование для автоматизированной обработки

Технологию автоматизированного порождения кода дизассемблера иллюстрирует рис. 16. Текстовое описание синтаксиса команд создается непосредственно по тексту документации Intel и AMD. Оно подается на вход специальному транслятору, который подвергает это описание анализу и, в случае его корректности, порождает код дизассемблера на некотором императивном языке программирования высокого уровня.

Затем порожденные файлы компилируются соответствующим компилятором в исполняемый модуль программы дизассемблера.

В качестве декларативного языка программирования оказалось удобным использовать JSON- или XML-описание. Описание синтаксиса команд в этом случае выражается в составлении массива описаний данных, выполненных в любом из этих вариантов.



Рис. 16. Иллюстрация автоматизированного порождения кода дизассемблера

Fig. 16. An illustration of automated generation of disassembler code

При создании обобщенного описания синтаксиса команд вручную и/или с применением средств автоматизации обработки текста, поддерживаемых современными текстовыми редакторами, могут быть удалены ошибки, опечатки и прочие дефекты, которыми изобилует текст документации от Intel и от AMD. Этими же средствами может генерироваться текст исходных спецификаций для транслятора (рис. 16).

Совокупность и последовательность шагов для построения обобщенной спецификации может быть следующей:

- экспорт текста из PDF-представления, например, в DOC-файл формата Microsoft Word;
- приведение текста к единому стилевому представлению;
- преобразование таблиц синтаксиса с разделением представления нотаций машинного кода и мнемоники команд;
- проверка синтаксического представления текста и кодирование всех ограничений, применимых к командам в документации;
- сопоставление описания одних и тех же команд в нотациях Intel и AMD;
- генерирование обобщенного описания синтаксиса команд (обобщенной спецификации).

Пример JSON-спецификации команды O='REX+80 /2 ib'; I='ADC r/m8, imm8' представлен в табл. 5. Фрагменты этой спецификации согласуются с соответствующими частями описаний команд в документации Intel.

Табл. 5. Пример фрагмента JSON-файла с описанием команды O='REX + 80 /2 ib'; I='ADC r/m8, imm8'

Tabl. 5. Example of a fragment of a JSON file with a description of the instruction O='REX + 80 /2 ib'; I='ADC r/m8, imm8'

| Фрагмент JSON-спецификации  | Пояснения  |
|---|--|
| {   | Преамбула  |
| "_comment": "This file contains descriptions of INTEL and AMD x86 instructions" | описания команд, содержащая информацию, в частности, о |
| "_comment": "It based on: Intel release   | версиях Intel и AMD документации,                      |

|   |  |
|---|--|
| 325462-081US September 2023; AMD<br>release 40332-4.07-June 2023"   | на которой она<br>построена  |
| "Instructions": [<br>...  | Начало массива<br>описания<br>синтаксиса команд  |
| Непосредственно описание команды  |  |
| "OrderNum": "12",   | Порядковый номер<br>спецификации. Его удобно<br>использовать для выдачи<br>сообщений транслятора о<br>распознанных ошибках                                     |
| "Opcode": "REX + 80 /2 ib",   | Синтаксис машинного кода<br>команды. Например, из "Opcode"<br>(рис. 2)   |
| "Mnemonic": "ADC r/m8,<br>imm8"   | Синтаксис кода команды на<br>ассемблере. Например, из<br>"Instruction" (рис. 2)  |
| "Operands": [<br>{ "Encoding": "ModRM:r/m (r,<br>w)", "Decoder": "ModRM" },<br><br>{ "Encoding": "imm8/16/32",<br>"Decoder": "Imm" },<br><br>{ "Encoding": "NA",<br>"Decoder": "None" },<br><br>{ "Encoding": "NA",<br>"Decoder": "None" },<br><br>{ "Encoding": "NA",<br>"Decoder": "None" }<br>], | Описание операндов, включая их<br>кодировку в документации рис.<br>2(2) и типы декодеров операндов<br>(табл. 2)  |
| "WidthMode": "Valid",   | Поддерживаемые режимы работы<br>процессора. Например, из<br>столбца вида "64-bit Mode" (рис.<br>5 и п. 6) или вида "64/32-bit<br>Mode Support" (рис. 8 и п. 6) |
| "Cpu": "",  | Значение из столбца "CPUID<br>Feature Flag" (рис. 6 и п. 7)  |
| "CompatLegMode": "N.E.",  | Значение из столбца<br>"Compat/Leg Mode" (рис. 5 и п. 6)   |
| "Description": "Add with<br>carry imm8 to r/m8.",   | Значение из столбца<br>"Description" (рис. 2)  |
| "Notes": "In 64-bit mode,<br>r/m8 can not be encoded to<br>access the following byte<br>registers if a REX prefix is<br>used: AH, BH, CH, DH.",   | Примечания, указывающие для<br>некоторых команд на<br>ограничения их реализации (рис.<br>5)<br><br>Эта строка в виде комментария                               |

|                     |  |
|---------------------|--|
|                     | позволит корректно вручную закодировать процедуру постобработки в дизассемблере.   |
| "Features": "",     | Некоторые особенности команды, например, ее применимость только для 32-разрядного режима адресации. Например, для команды O='A5'; I='MOVSD' по сравнению с O='A5'; I='MOVSW' для 16-разрядного режима адресации. |
| "OEM": "INTEL",     | Указание на источник описания команды  |
| },<br>...<br>]<br>} | Эпилоги описаний команды, массива команд и всего файла   |

В авторском макете транслятора JSON-файл спецификаций команд разбирается и генерируется таблица автомата, ячейки которого построены следующим образом (рис. 18).

Для понимания структуры, показанной на рис. 18, можно рассмотреть обобщенную структуру команды, которая в общем виде представляется как последовательность, включающая (рис. 17,а): факультативную последовательность { Prefix- $i$  }, где  $i \in [0..3]$ ; последовательность { Opcode- $j$  }, где  $j \in [0..2]$ ; последовательность прочих байтов, кодирующих операнды. Такая унифицированная структура получается в результате обобщения структур команд в EVEX-, VEX/XOP-, REX- и Legacy-вариантах. Если еще более упростить структуру, упраздняя разделения на байты, то получим вариант, представленный на рис. 17,б.

Prefix и Opcode в совокупности определяют уникальную комбинацию байтов, позволяющую однозначно распознать данную команду, что полностью согласуется с определением автоматной грамматики, в которой каждое правило вывода имеет вид:  $\alpha @ T \beta$ , где  $\alpha$  – нетерминал;  $T$  – некоторый терминал;  $\beta$  – последовательность нетерминалов и терминалов. Фактически Prefix и Opcode в совокупности позволяют вычислить значение  $T$ , которое в конечном счете позволяет: 1) адресовать конкретную ячейку автомата дизассемблера (рис. 18); 2) выбрать элементы декодирования мнемоники и операндов в ней, и произвести (если необходимо) постобработку.

Значение терминала  $T$  в данном случае имеет двоякую роль, выполняя роль индекса (маршрутизатора) по таблице автомата и определяя контекстные условия выбора декодеров в ячейке автомата. В реальном дизассемблере таблица автомата устроена иерархически, на каждом уровне иерархии ячейки в ней адресуются соответствующим байтом Opcode- $j$ , но для общей схемы, предающей идею построения дизассемблера, это не существенно.

Каждый декодер операнда является булевой функцией, распознающей только отдельный вид операнда и вычисляющий значение операнда на основе факультативной части машинного кода, следующей за Opcode. Постобработка применяется только в

отдельных случаях для обработки исключений из общих правил строения машинного кода.

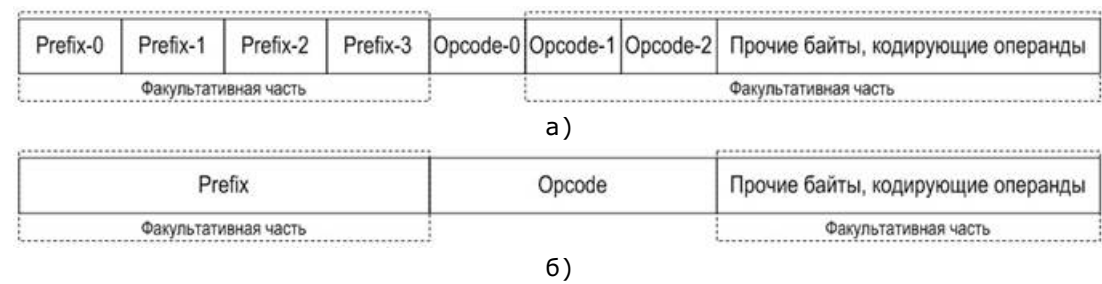


Рис. 17. Обобщенная структура команды: а) с разделением на отдельные байты; б) с разделением только на структурные элементы

Fig. 17. The generalized structure of the instruction: a) divided into individual bytes; b) divided only into structural elements

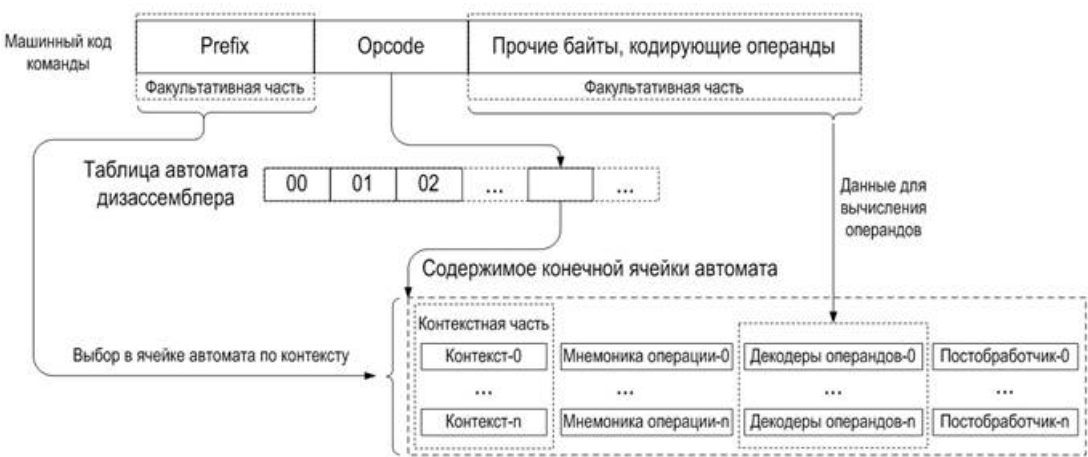


Рис. 18. Обобщенное устройство дизассемблера

Fig. 18. Generalized disassembler construction

В авторском макете дизассемблера таблица автомата (с учетом ее иерархической структуры - таблица переходов автомата), каждая ячейка этой таблицы генерируется автоматически.

Альтернативные проекты дизассемблеров [\[1,2\]](#) реализуются иначе, используя путь объемного (в силу большого числа команд архитектуры x86) и весьма трудоемкого программирования вручную, что делает и обновление кода в этих проектах тоже трудоемким занятием. А значит и поддержка актуальности версий дизассемблера в таких вариантах реализации может происходить не оперативно с обновлениями документации вообще, тем более не синхронно с темпами ее обновления как было представлено выше.

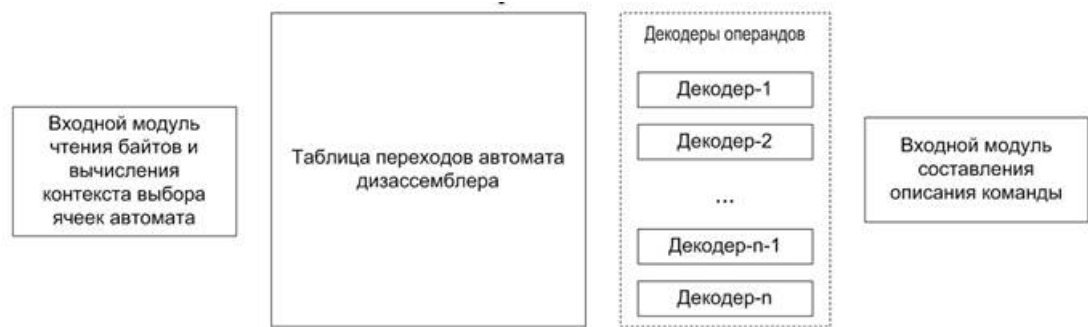




Рис. 19. Модули, составляющие дизассемблер

Fig. 19. Modules that make up the disassembler

На рис. 19 показана структура программных элементов, которые составляют макет дизассемблера, построенный по предлагаемой технологии на языке C. Он включает:

- входной модуль, который написан вручную и остается неизменным при обновлении версии документации (расширения множества команд процессора);
- таблица переходов автомата дизассемблера - наиболее сложный и объемный элемент дизассемблера, который как раз генерируется автоматизировано транслятором (рис. 16) на основе входных спецификаций, получаемых из документации;
- декодеры операндов создаются полуавтоматизированно: каркас создается автоматически, тело пишется вручную. Однажды написанный декодер далее не нуждается в обновлении. Например, для команды `O='13 /r'; I='ADC r32, r/m32'` однажды написанный декодер для операнда `'r32'` будет использоваться для дизассемблирования аналогичных операндов во всех командах с таким же обозначением, вида `O='03 /r'; I='ADD r32, r/m32'`, `O='3B /r'; I='CMP r32, r/m32'`, ..., `O='33 /r'; I='XOR r32, r/m32'`;
- выходной модуль написан вручную и не изменяется.

В идеальной ситуации, когда в новой версии документации не появилось нового формата команд и новых вариантов представления операндов все модули дизассемблера генерируются автоматически. В реальности, например, при реализации дизассемблера для версии документации Intel от сентября 2023 по сравнению с версией для документации от июня 2021 необходимо было немного изменить входной модуль для работы с командами вида `O='EVEX.128.NP.MAP5.W0 58 /r'; I='VADDPH xmm1{k1}{z}, xmm2, xmm3/m128/m16bcst'` и `O='EVEX.128.66.MAP6.W0 13 /r'; I='VCVTPH2PS: xmm1{k1}{z}, xmm2/m64/m16bcst'`, в машинном коде которых появились новые элементы `'MAP5'` и `'MAP6'`.

## 10. Заключение

Доля применения процессоров архитектуры x86 на мировом рынке персональных компьютеров хотя и снизилась примерно на тринадцать процентов за шесть лет с 2016 по 2022, но x86 остается все еще самой массово используемой микропроцессорной архитектурой [\[3\]](#). Основные ее дизайнеры - компании Intel и AMD - непрерывно совершенствуют выпускаемые процессоры, в том числе, и расширяя их программный интерфейс. Эти усовершенствования направлены на повышение производительности при исполнении приложений.

Современная технология разработки программ в целом основывается на использовании языков высокого уровня, что заставляет совершенствовать процедуры генерирования исполняемого кода в компиляторах, в которых задействуются новые команды. Вместе с этим относительная легкость создания приложений на языках высокого уровня дает возможность также легко реализовать в них недокументированные возможности, создающие риск непредсказуемого поведения приложения на некотором массиве входных данных. Вероятность такого поведения приложений ведет к необходимости проверять, по крайней мере, критически важный код программ на корректность выполняемых им вычислений. Сделать это в ситуации недоступности исходного кода можно методами реинжиниринга программ, где одним из действий является дизассемблирование исполняемого кода.



Естественно, что дизассемблеры должны обновляться с учетом обновлений программного интерфейса процессоров. Объем документации на процессоры, в частности, архитектуры x86, таков, что обработать ее текст и собрать описания команд в целостный и единообразный массив данных можно только автоматизированными методами. Исследованию возможности создавать такие спецификации и посвящена данная статья.

Целью же построения таких спецификаций является последующее автоматизированное генерирование кода дизассемблера для поддержания актуальной его версии по отношению к версиям документации.

## Библиография

1. Версия исходного кода операционной системы Windows 2000. Доступно по ссылке: <https://github.com/pustladi/Windows-2000/> (дата обращения: 23.02.2022)
2. Проект Intel X86 Encoder Decoder (Intel XED). Доступно по ссылке: <https://github.com/intelxed/xed> (дата обращения: 23.08.2022)
3. Distribution of Intel and AMD x86 computer central processing units (CPUs) worldwide from 2012 to 2022, by quarter. Доступно по ссылке: <https://www.statista.com/statistics/735904/worldwide-x86-intel-amd-market-share/> (дата обращения: 23.09.2022)
4. Китай научился легально клонировать процессоры Intel. Помог старинный соперник. Доступно по ссылке: [https://www.cnews.ru/news/top/2023-06-09\\_kitajtsy\\_nauchilis\\_legalno](https://www.cnews.ru/news/top/2023-06-09_kitajtsy_nauchilis_legalno) (дата обращения: 14.12.2023)
5. Intel® 64 and IA-32 Architectures. Software Developer's Manual. Documentation Changes. September 2023. Document Number: 252046-073 Доступно по ссылке: <https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-documentation-changes.html> (дата обращения: 21.10.2023)
6. Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. Order Number: 325462-081US September 2023. Доступно по ссылке: <https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html> (дата обращения: 20.10.2023)
7. Расширенная форма Бэкуса-Наура. Доступно по ссылке: [https://ru.wikipedia.org/wiki/Расширенная\\_форма\\_Бэкуса-Наура](https://ru.wikipedia.org/wiki/Расширенная_форма_Бэкуса-Наура) (дата обращения: 23.09.2022)
8. AMD64 Technology. AMD64 Architecture Programmer's Manual. Volumes 1–5. Publication No. 40332 Revision 4.07 Date June 2023. Доступно по ссылке: <https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/40332.pdf> (дата обращения: 20.11.2023)

## Результаты процедуры рецензирования статьи

*В связи с политикой двойного слепого рецензирования личность рецензента не раскрывается.*

*Со списком рецензентов издательства можно ознакомиться [здесь](#).*

Статья посвящена разработке обобщенной нотации программного интерфейса процессоров x86 для автоматизированного построения дизассемблера. Основная цель исследования – создание системы, которая позволит автоматизировать процесс создания дизассемблеров, что значительно упростит и ускорит работу с программным

кодом различных процессоров семейства x86. Автор использует метод анализа и синтеза существующих подходов к разработке дизассемблеров, основанных на документации процессоров от Intel и AMD. В статье приводится детальный обзор структуры документации, описываются типовые ошибки и неточности, а также предлагаются способы их автоматизированного исправления. Актуальность исследования обусловлена постоянным развитием процессоров архитектуры x86 и увеличением множества команд, поддерживаемых этими процессорами. Существующие подходы к разработке дизассемблеров не всегда могут оперативно адаптироваться к изменениям в программном интерфейсе процессоров, что делает предложенное автором решение крайне востребованным. Новизна работы заключается в разработке обобщенной нотации, которая позволяет автоматизировать процесс создания дизассемблеров. Впервые предложен подход, при котором на основе формальных спецификаций команд процессоров создаются автоматизированные трансляторы, генерирующие исходный код дизассемблера. Статья написана в научном стиле, отличается логической последовательностью и четкостью изложения материала. Структура статьи включает введение, обзор текущих решений, описание методологии, анализ документации процессоров, предложение обобщенной нотации и выводы. Содержание статьи соответствует заявленной тематике, каждая часть логически связана с предыдущей, что делает работу целостной и последовательной. В статье сделан вывод о возможности и необходимости автоматизации процесса создания дизассемблеров для процессоров x86. Автор показывает, что предложенная нотация и методология позволяют существенно сократить время и усилия, необходимые для обновления и разработки дизассемблеров в соответствии с изменениями в документации процессоров. Статья будет интересна широкому кругу специалистов в области разработки программного обеспечения, занимающихся вопросами обратного инжиниринга и анализа исполняемого кода. Работа также представляет интерес для разработчиков компиляторов и специалистов по информационной безопасности, занимающихся анализом программного обеспечения на наличие уязвимостей. Для дальнейшего развития работы автору рекомендуется сосредоточиться на нескольких ключевых направлениях. Во-первых, следует провести более детальное тестирование предложенной методологии на различных версиях процессоров от Intel и AMD, чтобы подтвердить универсальность и надежность созданной обобщенной нотации. Во-вторых, стоит разработать и реализовать инструменты для автоматизированного обновления спецификаций дизассемблера на основе новых версий документации процессоров. Это позволит обеспечить синхронизацию с последними изменениями в программном интерфейсе процессоров и повысит актуальность и оперативность обновлений дизассемблеров. Также необходимо рассмотреть возможность интеграции предложенного подхода с существующими системами разработки и анализа программного обеспечения. Это позволит значительно расширить область применения предложенной методологии и сделает ее доступной для широкого круга пользователей. Важно также уделить внимание созданию подробной документации и обучающих материалов, которые помогут другим разработчикам быстро освоить и эффективно использовать предложенные инструменты и методики. Наконец, перспективным направлением является проведение сравнительного анализа эффективности предложенного подхода с другими современными методами разработки дизассемблеров. Это позволит объективно оценить преимущества и недостатки разработанной методологии, а также выявить возможности для ее дальнейшего совершенствования и адаптации к новым вызовам и требованиям рынка. Рекомендуется принять статью к публикации, так как она представляет собой значительный вклад в область автоматизации разработки инструментов для анализа программного обеспечения, обладает высокой актуальностью и новизной, а также интересна для

широкой научной и профессиональной аудитории.

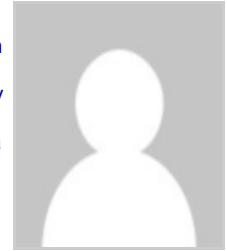
## Англоязычные метаданные

**Data storage format for analytical systems based on metadata and dependency graphs between CSV and JSON****Alpatov Aleksey Nikolaevich**

Associate Professor; IIPPO Department; MREA- Russian Technological University

78 Vernadsky Ave., Moscow, 119454, Russia

✉ aleksej01-91@mail.ru

**Bogatireva Anna Alekseevna**

Student, MREA– Russian Technological University

111033, Russia, Moscow, 4A Tankov ave., 24 sq.

✉ pecherni@gmail.com



**Abstract.** In the modern information society, the volume of data is constantly growing, and its effective processing is becoming key for enterprises. The transmission and storage of this data also plays a critical role. Big data used in analytics systems is most often transmitted in one of two popular formats: CSV for structured data and JSON for unstructured data. However, existing file formats may not be effective or flexible enough for certain data analysis tasks. For example, they may not support complex data structures or provide sufficient control over metadata. Alternatively, analytical tasks may require additional information about the data, such as metadata, data schema, etc. Based on the above, the subject of this study is a data format based on the combined use of CSV and JSON for processing and analyzing large amounts of information. The option of sharing the designated data types for the implementation of a new data format is proposed. For this purpose, designations have been introduced for the data structure, which includes CSV files, JSON files, metadata and a dependency graph. Various types of functions are described, such as aggregating, transforming, filtering, etc. Examples of the application of these functions to data are given. The proposed approach is a technique that can significantly facilitate the processes of information analysis and processing. It is based on a formalized approach that allows you to establish clear rules and procedures for working with data, which contributes to their more efficient processing. Another aspect of the proposed approach is to determine the criteria for choosing the most appropriate data storage format. This criterion is based on the mathematical principles of information theory and entropy. The introduction of a criterion for choosing a data format based on entropy makes it possible to evaluate the information content and compactness of the data. This approach is based on the calculation of entropy for selected formats and weights reflecting the importance of each data value. By comparing entropies, you can determine the required data transmission format. This approach takes into account not only the compactness of the data, but also the context of their use, as well as the possibility of including additional meta-information in the files themselves and supporting data ready for analysis.

**Keywords:** Apache Parquet, Integration of data formats, Data analysis, Analysis Ready Data, Metadata, Data processing, CSV, JSON, Data storage formats, Big Data

## References (transliterated)

1. Malcolm R., Morrison C., Grandison T., Thorpe S., Christie K., Wallace A., Green D., Jarrett J., Campbell A. Increasing the accessibility to big data systems via a common services api // IEEE International Conference on Big Data. 2014. Pp. 883-892.
2. Wu T. System of teaching quality analyzing and evaluating based on data warehouse // Computer Engineering and Design. 2009. No. 6(2). Pp. 1545-1547.
3. Vitagliano G. et al. Pollock: A Data Loading Benchmark // Proceedings of the VLDB Endowment. 2023. No. 8(16). Pp. 1870-1882.
4. Xiaojuan L., Yu Z. A data integration tool for the integrated modeling and analysis for east // Fusion Engineering and Design. 2023. No. 195. Pp. 113933. URL: <https://doi.org/10.1016/j.fusengdes.2023.113933>
5. Lemzin A. Streaming Data Processing // Asian Journal of Research in Computer Science. 2023. No. 1(15). Pp. 11-21.
6. Hughes LD, Tsueng G, DiGiovanna J, Horvath TD, Rasmussen LV, Savidge TC, Stoeger T, Turkarslan S, Wu Q, Wu C, Su AI, Pache L. Addressing barriers in FAIR data practices for biomedical data // Scientific Data. 2023. No. 1(10). P. 98. DOI: <https://doi.org/10.1038/s41597-023-01969-8>
7. Gohil A., Shroff A., Garg A., Kumar S. A Compendious Research on Big Data File Formats. *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE Press, Madurai, India. 2022. Pp. 905-913. DOI: <https://doi.org/10.1109/ICICCS53718.2022.9788141>
8. Elsukov P. Yu. Informatsionnaya asimmetriya i informatsionnaya neopredelennost' // ITNOU: Informatsionnye tekhnologii v nauke, obrazovanii i upravlenii. 2017. No. 4 (4). S. 69-76.
9. Bromiley P. A., Thacker N. A., Bouhova-Thacker E. Shannon entropy, Renyi entropy, and information // Statistics and Inf. Series (2004-004). 2004. No. 9. Pp. 2-8.
10. Dwyer, J. L. Roy, D. P., Sauer B., Jenkerson C. B., Zhang H. K., Lymburner L. Analysis ready data: enabling analysis of the Landsat archive // Remote Sensing. 2018. №. 9(10). 1363.

## Developing the Methodology for the Effective Placement of Security and Fire Alarm Systems

**Sharipov Rifat Rashatovich**

PhD in Technical Science

Associate Professor of the Information Security Systems Department of Kazan National Research Technical University named after A.N. Tupolev-KAI

55 Bolshaya Krasnaya str., Kazan, 420015, Russia, Republic of Tatarstan

✉ [riphat@mail.ru](mailto:riphat@mail.ru)



**Yusupov Bulat Zufarovich**

Student of the Information Security Systems Department of Kazan National Research Technical University named after A.N. Tupolev-KAI (KNRTU-KAI)

420015, Russia, Republic of Tatarstan, Kazan, Bolshaya Krasnaya str., 55

✉ [Bulatusupov9@gmail.com](mailto:Bulatusupov9@gmail.com)



**Martynov Artur Mikhailovich**

Student of the Information Security Systems Department of Kazan National Research Technical University named after A.N. Tupolev-KAI (KNRTU-KAI)

420015, Russia, Republic of Tatarstan, Kazan, Bolshaya Krasnaya str., 55

✉ Keshka628@gmail.com



**Zaripova Rimma Soltanovna**

PhD in Technical Science

Associate Professor of the Department of Digital Systems and Models of Kazan State Power Engineering University

51 Krasnoselskaya str., Kazan, 420066, Russia, Republic of Tatarstan

✉ zarim@rambler.ru



**Abstract.** The article focuses on security and fire alarm systems (SFAS) as means of ensuring the safety of facilities, viewing them as integrated complexes for promptly detecting potential threats. The main emphasis is on detectors, including their classification and role within the system. The article examines various configurations of SFAS and ways of connecting and processing signals from detectors, allowing for an evaluation of how these factors affect the system's effectiveness. The lifecycle of SFAS is described, highlighting the importance of each stage from design to operation. The article provides an overview of regulatory documents, emphasizing the importance of compliance with standards and requirements when implementing SFAS. Recommended for security professionals and individuals interested in delving into this topic. Additionally, the article addresses issues related to the placement of SFAS and their impact on system effectiveness. It analyzes vulnerabilities arising from irrational placement of components and presents a methodology for optimizing placement to enhance security. The methodology is described step by step, considering input and output processes at each stage. The authors conduct practical testing of the methodology in an educational laboratory with an installed SFAS, identifying placement errors and formulating recommendations for correction. The article is beneficial for professionals in the design and installation of SFAS, as well as for those seeking to improve the level of protection of facilities, accentuating the critical importance of proper component placement.

**Keywords:** Protection, Installation, Components, Optimization, Effectiveness, Methodology, Placement, Fire alarm, Security systems, Design

## References (transliterated)

1. Gavishev A. A. Otsenka skrytnosti besprovodnykh okhranno-pozharnykh sistem // Dostizheniya i prilozheniya sovremennoi informatiki, matematiki i fiziki : Materialy VIII Vserossiiskoi nauchno-prakticheskoi zaachnoi konferentsii, Neftekamsk, 15 noyabrya 2019 goda. – Neftekamsk: Bashkirskii gosudarstvennyi universitet, 2019. – S. 21-29.
2. Ryapolova E. I. Razrabotka avtomatizirovannogo rabocheho mesta dispetchera okhranno-pozharnoi signalizatsii // Novaya nauka: Sovremennoe sostoyanie i puti razvitiya. – 2015. – № 4-1. – S. 81-85.
3. Yusupov B. Z. Razrabotka laboratornogo stenda okhranno-pozharnoi signalizatsii po distsipline tekhnicheskie sredstva okhrany // XXV Tupolevskie chteniya (shkola molodykh uchenykh) : Mezhdunarodnaya molodezhnaya nauchnaya konferentsiya, posvyashchennaya 60-letiyu so dnya osushchestvleniya Pervogo poleta cheloveka v kosmicheskoe prostranstvo i 90-letiyu Kazanskogo natsional'nogo issledovatel'skogo

- tekhnicheskogo universiteta im. A.N. Tupoleva-KAI. Materialy konferentsii. Sbornik dokladov. V 6-ti tomakh, Kazan', 10–11 noyabrya 2021 goda. Tom V. – Kazan': Individual'nyi predprinimatel' Sagieva A.R., 2021. – S. 758-763.
4. Kashpur E. I. Issledovanie perspektivnykh tekhnologii tsifrovoy modulyatsii v sistemakh okhranno-pozharnoi signalizatsii // Molodezhnyi nauchno-tekhnicheskii vestnik. – 2015. – № 8. – S. 29.
  5. Gizatullin Z. M. Metodika i modeli dlya fizicheskogo modelirovaniya elektromagnitnykh pomekh na primere analiza pomekhoustoichivosti elektronnykh sredstv avtotransporta / Z. M. Gizatullin, R. M. Gizatullin, M. G. Nuriev // Radiotekhnika i elektronika. – 2021. – T. 66. – №6. – S. 609-613.
  6. Bonch-Bruevich A. M. Issledovanie perspektivnykh tekhnologii tsifrovoy modulyatsii v sistemakh okhranno-pozharnoi signalizatsii / A. M. Bonch-Bruevich, E. I. Kashpur // Spetstekhnika i svyaz'. – 2015. – № 3. – S. 24-28.
  7. Yusupo B. Z. Razrabotka laboratornogo stenda okhranno-pozharnoi signalizatsii po distsipline tekhnicheskie sredstva okhrany / B. Z. Yusupov, A. M. Martynov // Aktual'nye problemy nauki i obrazovaniya v usloviyakh sovremennykh vyzovov : Sbornik materialov XIX Mezhdunarodnoi nauchno-prakticheskoi konferentsii, Moskva, 21 marta 2023 goda. – Moskva: Pechatnyi tsekh, 2023. – S. 80-91.
  8. Burykin I. A. Organizatsiya zashchishchennogo kanala svyazi v sisteme okhranno-pozharnoi signalizatsii / I. A. Burykin, D. A. Petrov, D. S. Luppa // Aktual'nye voprosy v nauchnoi rabote i obrazovatel'noi deyatel'nosti : sbornik nauchnykh trudov po materialam Mezhdunarodnoi nauchno-prakticheskoi konferentsii: v 10 tomakh, Tambov, 30 maya 2015 goda. Tom 6. – Tambov: OOO "Konsaltingovaya kompaniya Yukom", 2015. – S. 23-25.
  9. Berlev S. V. Eksperimental'naya otsenka effektivnosti tekhnicheskikh sredstv okhranno-pozharnoi signalizatsii / S. V. Berlev // Sovremennye tekhnologii obespecheniya grazhdanskoi oborony i likvidatsii posledstviy chrezvychainykh situatsii. – 2015. – № 1-1(6). – S. 62-63.
  10. Yusupov B. Z. Razrabotka metodiki provedeniya laboratornykh raboty na stende «OPS Astra-713» po distsipline tekhnicheskie sredstva okhrany // XXV Tupolevskie chteniya (shkola molodykh uchenykh) : Mezhdunarodnaya molodezhnaya nauchnaya konferentsiya, posvyashchennaya 60-letiyu so dnya osushchestvleniya Pervogo poleta cheloveka v kosmicheskoe prostranstvo i 90-letiyu Kazanskogo natsional'nogo issledovatel'skogo tekhnicheskogo universiteta im. A.N. Tupoleva-KAI. Materialy konferentsii. Sbornik dokladov. V 6-ti tomakh, Kazan', 10–11 noyabrya 2021 goda. Tom V. – Kazan': Individual'nyi predprinimatel' Sagieva A.R., 2021. – S. 764-767.
  11. Sinilov V. G. Sistemy okhrannoi, pozharnoi i okhranno-pozharnoi signalizatsii : uchebnik dlya obrazovatel'nykh uchrezhdenii, realizuyushchikh programmy nachal'nogo professional'nogo obrazovaniya / V. G. Sinilov; V. G. Sinilov. – 6-e izd., ster. – Moskva: Akademiya, 2011. – (Nachal'noe professional'noe obrazovanie. Radioelektronika).
  12. Vershinin I. S. Assotsiativnaya zashchita tekstovykh kharakteristik ob"ektov / I. S. Vershinin, S. V. Pystogov, R. F. Gibadullin, D. A. Gashigullin // Vestnik Kazanskogo gosudarstvennogo tekhnicheskogo universiteta im. A.N. Tupoleva. – 2020. – T. 76, № 1. – S. 117-125.
  13. Petik N. S. Proektirovanie sistemy okhranno-pozharnoi signalizatsii / N. S. Petik // Molodost'. Intellect. Initsiativa : Materialy VI Mezhdunarodnoi nauchno-prakticheskoi konferentsii studentov i magistrantov, Vitebsk, 19 aprelya 2018 goda. – Vitebsk: Vitebskii gosudarstvennyi universitet im. P.M. Masherova, 2018. – S. 33-34.

14. Oreshina Yu. V. Delovaya reputatsiya predprinimatel'skikh struktur na rynke okhranno-pozharnoi signalizatsii / Yu. V. Oreshina // Aktual'nye voprosy v nauchnoi rabote i obrazovatel'noi deyatel'nosti : sbornik nauchnykh trudov po materialam Mezhdunarodnoi nauchno-prakticheskoi konferentsii: v 10 tomakh, Tambov, 30 maya 2015 goda. Tom 4. – Tambov: OOO "Konsaltingovaya kompaniya Yukom", 2015. – S. 112-113.
15. Postanovlenie Pravitel'stva RF ot 16 fevralya 2008 g. N 87 "O sostave razdelov proektnoi dokumentatsii i trebovaniyakh k ikh sodержaniyu" [Elektronnyi resurs]. Dostup iz sistemy GARANT // EPS "Sistema GARANT" : URL: <https://base.garant.ru/> (data obrashcheniya: 16.06.2023).
16. GOST R 50776-95 (MEK 60839-1-4:1989) «Sistemy trevozhnoi signalizatsii. Chast' 1. Obshchie trebovaniya». [Elektronnyi resurs]. Dostup iz sistemy GARANT // EPS "Sistema GARANT". URL: <https://base.garant.ru/> (data obrashcheniya: 16.06.2023).
17. SP 68.13330.2017 "SNiP 3.01.04-87 Priemka v ekspluatatsiyu zakonchennykh stroitel'stvom ob'ektov. Osnovnye polozheniya"[Elektronnyi resurs]. Dostup iz sistemy GARANT // EPS "Sistema GARANT". URL: <https://base.garant.ru/> (data obrashcheniya: 16.06.2023).
18. Nuriev M. G. Konverter Markdown-failov v LaTeX-dokument / M. G. Nuriev, E. S. Belashova, K. A. Barabash // Programmnye sistemy i vychislitel'nye metody. – 2023. – № 1. – S. 1-12.
19. Gibadullin R. F. Potokobezopasnye vyzovy elementov upravleniya v obogashchennykh klientskikh prilozheniyakh / R. F. Gibadullin // Programmnye sistemy i vychislitel'nye metody. – 2022. – № 4. – S. 1-19.
20. Gibadullin R.F. Neodnoznachnost' rezul'tatov pri ispol'zovanii metodov klassa Parallel v ramkakh ispolnyayushchei sredy .NET Framework / R.F. Gibadullin, I.V. Viktorov // Programmnye sistemy i vychislitel'nye metody. – 2023. – № 2. – S. 1-14.

## Diagnostics of failures of technological equipment of chemical industries using artificial intelligence

Zubov Dmitrii Vladimirovich 

PhD in Technical Science

Associate Professor; Department of Information Computer Technologies; D.I. Mendeleev Russian University of Chemical Technology

142602, Russia, Moscow, Geroev Panfilov str., 20, room 127

✉ [dvzubov@gmail.com](mailto:dvzubov@gmail.com)

Lebedev Danila Aleksandrovich

Postgraduate student; Department of Information Computer Technologies; D.I. Mendeleev Russian University of Chemical Technology

119331, Russia, Moscow, Maria Ulyanova str., 16, sq. 188

✉ [lebedev.d.a@muctr.ru](mailto:lebedev.d.a@muctr.ru)



**Abstract.** The paper considers the problem of automated recognition of single emergencies in chemical and oil refining industries. Modern chemical and technological production facilities are maintained and managed by a small number of personnel, which increases the burden on each operator. To reduce the number of operator errors, their training is regularly conducted on simulators equipped with a set of both standard situations (routine start-up, shutdown,



normal process management, switching from one mode to another) and emergency scenarios (column depressurization, pump failure, failure of the power supply system). Nevertheless, it is impossible to foresee all possible failures during operator training, and even a trained operator may not notice the first signs of an accident, and therefore it is necessary to create a decision support system that helps the operator to recognize failures of technological equipment in a timely manner. To recognize failures, it is proposed to use a neural network trained on an array of simulated accident data. An industrial simulator based on the RTsim platform was used to simulate typical accidents. The novelty of the research lies in the use of artificial intelligence methods to diagnose the property of the technological process according to the SCADA system and the use of data for training a neural network not from a real object (which will always be insufficient), but from a model that exactly corresponds to a specific technological site. The number of simulated scenarios used to train a neural network can be quite large, which reduces the proportion of erroneous system responses. The developed system confidently copes with the recognition of individual equipment failures. The results obtained can be used to help process operators and to improve emergency protection systems. The analysis of the time required by the system to recognize an emergency situation can be used to design new production facilities, modify the control and management system.

**Keywords:** industrial safety, decision-support system, simulation modeling, Digital Twin, RTsim, computer trainer, artificial intelligence, accident, failure, oil refining

## References (transliterated)

1. Tsifrovye trenazhery tekhnologicheskikh protsessov RTSIM. Kar'era dlya obespecheniya bezopasnosti v neftekhimicheskom komplekse / R. R. Gabdrakhmanov, E. V. Garifullina, V. V. Bronskaya [i dr.] // Nauka, tekhnologii, obshchestvo: Ekologicheskii inzhiniring v interesakh ustoichivogo razvitiya territorii : Sbornik nauchnykh statei, Krasnoyarsk, 09–10 noyabrya 2023 goda. – Krasnoyarsk: Krasnoyarskii kraevoi Dom nauki i tekhniki Rossiiskogo Soyuza nauchnykh i inzhenernykh obshchestvennykh ob"edinenii, 2023. – S. 131-135. – EDN BLETRM.
2. Titova, L. M. Tsifrovye dvoyniki pri podgotovke inzhenernykh kadrov khimicheskikh proizvodstv v vysshei shkole / L. M. Titova, N. P. Memedeikina // Mezhdunarodnyi forum KAZAN DIGITAL WEEK-2023 : Sbornik materialov, Kazan', 20–22 sentyabrya 2023 goda / Sost. R.Sh. Akhmedieva, R.N. Minnikhanov. Pod obshchei redaktsiei R.N. Minnikhanova. – Kazan': Nauchnyi tsentr bezopasnosti zhiznedeyatel'nosti, 2023. – S. 954-960. – EDN RNNMLO.
3. Imitatsiya protsessov upravleniya agregatom proizvodstva metanola na trenazhere / V. L. Perov [i dr.]. – Tekst : neposredstvennyi // Tr. Mosk. khim. – tekhnol. in-ta im. D.I.Mendeleeva. – 1988. – Vyp. 152. – S. 12-18.
4. Siguo Bi, Cong Wang, Bochun Wu, Shuyan Hu, Wutao Huang, Wei Ni, Yi Gong, Xin Wang. A comprehensive survey on applications of AI technologies to failure analysis of industrial systems// Engineering Failure Analysis, Volume 148, 2023, 107172, ISSN 1350-6307. URL: <https://doi.org/10.1016/j.engfailanal.2023.107172>.
5. Pallavi Kumari, Bhavana Bhadriraju, Qingsheng Wang, Joseph Sang-Il Kwon. A modified Bayesian network to handle cyclic loops in root cause diagnosis of process faults in the chemical process industry // Journal of Process Control. Volume 110, 2022, Pages 84-98. URL: <https://doi.org/10.1016/j.jprocont.2021.12.011>
6. Chunbo Pang, Dawei Duan, Zhiying Zhou, Shangbo Han, Longchao Yao, Chenghang Zheng, Jian Yang, Xiang Gao, An integrated LSTM-AM and SPRT method for fault early detection of forced-oxidation system in wet flue gas desulfurization // Process Safety

and Environmental Protection. Volume 160, 2022, Pages 242-254.

URL: <https://doi.org/10.1016/j.psep.2022.01.062>

7. Hao Yin, He Xu, Weiwang Fan, Feng Sun. Fault diagnosis of pressure relief valve based on improved deep Residual Shrinking Network // Measurement. Volume 224, 2024, 113752. URL: <https://doi.org/10.1016/j.measurement.2023.113752>
8. Imitatsiya avarii oborudovaniya v sistemakh modelirovaniya khimiko-tekhnologicheskikh protsessov / D. V. Zubov, A. V. Lobanov, E. B. Filippova, D. A. Lebedev // Matematicheskie metody v tekhnologiyakh i tekhnike. – 2022. – № 6. – S. 31-35. – DOI: 10.52348/2712-8873\_MMTT\_2022\_6\_31. – EDN GSNFJH.
9. Razrabotka programmnykh modulei tsifrovogo dvoynika khimiko-tekhnologicheskogo proizvodstva / E. B. Filippova, D. V. Zubov, R. S. Krashenninnikov, A. V. Lobanov // Avtomatizatsiya v promyshlennosti. – 2021. – № 7. – S. 16-20. – DOI: 10.25728/avtprom.2021.07.04. – EDN CJBIPX

## Investigation of the stress–strain state of a composite blade in ANSYS WorkBench

**Filippova Kseniya Anatol'evna**

Senior Lecturer; Department of Aircraft and Helicopter Engineering; East Siberian State University of Technology and Management

670013, Russia, Republic of Buryatia, Ulan-Ude, Klyuchevskaya str., 40 V

✉ [ipq84@mail.ru](mailto:ipq84@mail.ru)



**Ayusheev Tumen Vladimirovich**

Doctor of Technical Science

Associate Professor; Department of Engineering and Computer Graphics; East Siberian State University of Technology and Management

670013, Russia, Buryatia region, Ulan-Ude, Klyuchevskaya str., 40 V

✉ [atv62@bk.ru](mailto:atv62@bk.ru)



**Damdinova Tatiana Tsybikovna**

PhD in Technical Science

Associate Professor, East-Siberian State University of Technology and Management

670000, Russia, Republic of Buryatia, Ulan-Ude, Klyuchevskaya str., 40 V

✉ [dtatyanac@mail.ru](mailto:dtatyanac@mail.ru)



**Tsidipov Tsibik Tsirendorzhievich**

PhD in Technical Science

Associate Professor; Department of Aircraft and Helicopter Engineering; East Siberian State University of Technology and Management

670013, Russia, Buryatia region, Ulan-Ude, Klyuchevskaya str., 40 V

✉ [sssibik@mail.ru](mailto:sssibik@mail.ru)



**Abstract.** In this paper, the static strength of a UAV blade made of composite material was calculated. Composite materials have an advantage over traditional materials (metals and alloys) in the field of aviation – gain in weight, low sensitivity to damage, high rigidity, high mechanical characteristics. At the same time, the identification of vulnerabilities in a layered structure is a difficult task and in practice is solved with the help of destructive control.

Composite materials available in the ANSYS materials library were used in the modeling: Epoxy Carbon Woven (230 GPa) Prepreg woven carbon fiber in the form of a semi-finished prepreg impregnated with epoxy resin carbon fiber with Young's modulus  $E=230$  GPa and Epoxy Carbon (230 GPa) Prepreg is a unidirectional carbon fiber prepreg impregnated with epoxy resin with a Young's modulus  $E=230$  GPa. Modern software products, such as ANSYS WorkBench, allow comprehensive investigation of the layered structure. Several variants of blade designs with different fillers as the median material were investigated. The forward and reverse destruction criteria based on the Tsai-Hill theory were used. The influence of gravity was not taken into account. It is shown that the developed blade design meets the requirements. Balsa wood, pine, aspen and polyurethane foam were chosen as the middle material of the blade. Pine and aspen wood were selected according to the criteria of their availability and having the lowest density. The materials library of the ANSYS WorkBench software package used does not have characteristics for all of them, so the characteristics of the selected materials (pines and aspens) were added manually. For modeling and calculations in the ANSYS WorkBench program, such characteristics as density, axial elastic modulus, Poisson's coefficients, shear modulus and tensile and compressive strength limits are required.

**Keywords:** middle fillers, fiberglass, carbon fabrics, Tsai-Hill theory, failure criterion, stress, ANSYS WorkBench, static strength, blade, composite material

## References (transliterated)

1. Fen, G. Issledovanie staticheskoi prochnosti kompozitnoi lopasti vertoleta / G. Fen, A. M. Dumanskii, A. N. Ruslantsev // Deformirovanie i razrushenie kompozitsionnykh materialov i konstruktсии: Trudy Vtoroi mezhdunarodnoi konferentsii, Moskva, 18–20 oktyabrya 2016 goda. – S. 196-198.
2. Sidorov, I. N. Raschet napryazhenno-deformirovannogo i predel'nogo sostoyanii kompozitnoi lopasti nesushchego vinta vertoleta pri razlichnykh rezhimakh poleta s uchetom povrezhdenii v komlevom sechenii / I. N. Sidorov, A. V. Gorelov, E. I. Nikolaev // KhI Vserossiiskii s"ezd po fundamental'nykh problemam teoreticheskoi i prikladnoi mekhaniki: sbornik dokladov, Kazan', 20–24 avgusta 2015 goda. – S. 3453-3457.
3. Burtsev, B. N. Osobennosti napryazhenno-deformirovannogo sostoyaniya komlevoi chasti lopasti nesushchego vinta / B. N. Burtsev, N. P. Tyutyunnikov // Mekhanika kompozitsionnykh materialov i konstruktсии. – 2012. – T. 18, № 4. – S. 552-561.
4. Glebov, I. T. Fizika drevesiny: uchebnoe posobie. – Ekaterinburg: UGLTU, 2018. – 80 s.
5. Pervushin, Yu. S., Zhernakov, V. S. Osnovy mekhaniki, proektirovaniya i tekhnologii izgotovleniya izdelii iz sloistnykh kompozitsionnykh materialov: ucheb. posobie / Yu. S. Pervushin, V. S. Zhernakov; Ufimsk. gos. aviats. tekhn. un-t. Ufa. 2008. – 303 s.
6. Khill, R. Teoriya mekhanicheskikh svoistv voloknistykh kompozitsionnykh materialov // Mekhanika / Sb. perevodov. 1966. № 2. S. 131-149.
7. Mekhanika konstruktсии iz kompozitsionnykh materialov. – Moskva: Mashinostroyeniye, 1988. – 271 s.

## Investigation of stochastic models of packet generation in computer networks

PhD in Technical Science

Assistant, Associate professor, Povolzhskiy State University of Telecommunications and Informatics

443090, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ s.malakhov@psuti.ru



Yakupov Denis Olegovich

Assistant, Povolzhskiy State University of Telecommunications and Informatics

443090, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ d.yakupov@psuti.ru



**Abstract.** Stochastic packet generation models are models that are used to generate traffic in computer networks with certain characteristics. These models can be used to simulate network activity and test network performance. Standard data transmission on the network is packet generation with delays, in which packets are sent at certain intervals. Various stochastic models can be used to generate delayed packets, including uniform distribution, exponential distribution, and Erlang distribution. In this work, an experimental setup was assembled and a client-server application was developed to conduct research and analyze the performance of the data transmission channel. An algorithm has been proposed that allows to restore the moment characteristics of a random value of the interval between packets for further use of queuing models. The analysis of the distribution laws on the performance of the experimental network sample was performed and estimates of the efficiency of channel use and the average packet generation time in network segments, as well as histograms of delays according to the distribution laws, were obtained. An experimental setup was created, and a client-server application was developed to analyze the performance of the data transmission channel. An algorithm for restoring the moment characteristics of the time intervals between packets is proposed. The analysis of the distribution laws on network performance was carried out, estimates of the efficiency of channel use and the average packet generation time in network segments were obtained, as well as histograms of delays according to the distribution laws. The generation of packets with delays according to stochastic distribution laws (uniform, exponential, Erlang) is of great importance in modeling and analyzing the operation of network systems. Also, the generation of packets with delays according to the above-mentioned distribution laws allows testing and debugging of network applications and devices in conditions close to real ones. This allows to identify possible problems and improve the operation of network systems. As a result of the experiment, an algorithm was proposed that allows to restore the moment characteristics of a random value of the interval between packets for further use of queuing models. Also, the analysis of the influence of distribution laws on the performance of the experimental network sample was performed and estimates of the efficiency of channel use and the average packet generation time in network segments, as well as histograms of delays according to distribution laws, were obtained.

**Keywords:** generating packages, data transmission, traffic analysis, client-server application, delays, packet switching, Erlang distribution, exponential distribution, uniform distribution, Stochastic models

## References (transliterated)

1. Zhukova G.N. Karta koeffitsientov asimmetrii i ekstsessa v prepodavanii teorii

- veroyatnostei i matematicheskoi statistiki// Kontsept 2015. №8. S. 1-4.
2. Dmitriev E.I., Medvedev A.V. P-generator sluchainykh chisel, raspredelennykh po eksponentsial'nomu zakonu// Aktual'nye problemy aviatsii i kosmonavтики 2011. №7. Tom 1. S. 316-317.
  3. Raspredelenie Erlanga URL:  
<http://algolist.ru/maths/matstat/erlang/index.php#:~:text=%D0%A0%BC>. (Data obrashcheniya 06.03.2023).
  4. Kak pol'zovat'sya Wireshark dlya analiza trafika. URL: <https://losst.pro/kak-polzovatsya-wireshark-dlya-analiza-trafika> (Data obrashcheniya 06.03.2023).
  5. Prilozhenie dlya generatsii paketov v komp'yuternykh setyakh s pomoshch'yu stokhasticheskikh modelei raspredeleniya. URL: <https://elibrary.ru/item.asp?id=50133060>
  6. Tarasov V.N., Bakhareva N.F., Gorelov G.A., Malakhov S.V. Analiz vkhodyashchego trafika na urovne trekh momentov raspredelenii vremennykh intervalov// Informatsionnye tekhnologii 2014. №9. S. 54-59.
  7. Emulyatsiya vliyaniya global'nykh setei. URL: <https://habr.com/ru/articles/24046/> (Data obrashcheniya 10.05.2023).
  8. Rukovodstvo po nastroiike proizvoditel'nosti. URL: [http://www.regatta.cs.msu.su/doc/usr/share/man/info/ru\\_RU/a\\_doc\\_lib/aixbman/prftungd/2365c91.htm](http://www.regatta.cs.msu.su/doc/usr/share/man/info/ru_RU/a_doc_lib/aixbman/prftungd/2365c91.htm) (Data obrashcheniya 10.05.2023).
  9. Algoritmy seti Ethernet/Fast Ethernet. URL:  
[https://intuit.ru/studies/professional\\_retraining/943/courses/57/lecture/1690?page=2](https://intuit.ru/studies/professional_retraining/943/courses/57/lecture/1690?page=2) (Data obrashcheniya 10.05.2023).
  10. Snabzhenie paketov dannykh tochnymi vremennymi metkami v sistemakh setevogo monitoringa. URL: <http://www.treatface.ru/solutions/sistemy-setevogo-monitoringa/snabzhenie-paketov-dannykh-tochnymi-vremennymi-metkami-v-sistemakh-setevogo-monitoringa> (Data obrashcheniya 10.05.2023)

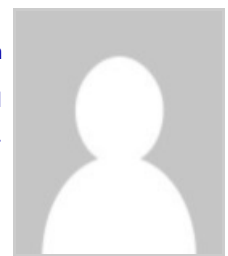
## Comparison of methods for optimizing the speed of reading/writing drives

**Val'kov Vadim Aleksandrovich**

Student, Department of Computer Science and Computer Engineering; FSUE VO PGUTI

443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ [BonetyFallen@yandex.ru](mailto:BonetyFallen@yandex.ru)

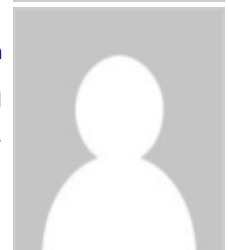


**Stolyarov Evgeniy Pavlovich**

Student, Department of Computer Science and Computer Engineering; FSUE in PGUTI

443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ [evgeniystolyarov1@mail.ru](mailto:evgeniystolyarov1@mail.ru)



**Korchagin Anton Aleksandrovich**

Student; Department of Computer Science and Computer Engineering; FSUE VO PGUTI

443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ anton\_kor2004@mail.ru

**Ermishin Maksim Vladimirovich**

Student; Department of Computer Science and Computer Engineering; FSUE in PGUTI

443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ maks.yer.04@mail.ru

**Yakupov Denis Olegovich**

Assistant; Department of Management in Technical Systems; FSUE in PGUTI

443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ d.yakupov@psuti.ru



**Abstract.** The objects of this study are data storage devices of various types and levels of complexity, as well as the principles of their operation. They are complex technical systems that include many components and are characterized by a high degree of integration. The subject of the research is to study the main characteristics of hard drives and solid-state drives. Their structure, functional features, principles of operation and ways of optimization are important. The purpose of the study is to determine the most effective methods for optimizing the operation of these devices. This includes aspects such as memory management, load balancing, power management, and others.

The results of this research can be used to improve data efficiency, improve the performance of data storage systems and create new technologies in this area. This study examines the performance of various disk storage solutions through a series of tests aimed at understanding speed and dependence on external factors. The main conclusions of the study reflect the importance of the integrated use of optimization approaches to improve the speed of reading and writing data. Optimizing the processes of reading and writing data is critically important for modern high-performance computing systems, as well as for applications that require quick access to large amounts of information.

The improved techniques used in the course of the study contribute to a significant increase in the performance of data storage devices. They take into account the specifics of various types of storage devices, including hard drives and solid-state drives, and offer optimization approaches that take into account their unique characteristics.

Overall, the results of this study provide valuable insights into the principles of optimizing data storage, and they can serve as a basis for developing new strategies and solutions in this important area of information technology. This study represents a significant contribution to the scientific understanding of optimizing data reading and writing processes, and its findings may have long-term implications for the development of data storage technologies.

**Keywords:** Cache Buffer, Interface, Defragmentation, Reading data, Fragmentation, Efficiency, Optimization, Solid state drives, Hard drives, The file system

## References (transliterated)

1. PMR (Perpendicular Magnetic Recording) and SMR (Shingled Magnetic Recording) hard disk drives [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: [https://kb.synology.com/en-id/DSM/tutorial/PMR\\_SMR\\_hard\\_disk\\_drives](https://kb.synology.com/en-id/DSM/tutorial/PMR_SMR_hard_disk_drives)
2. Cherepichnaya magnitnaya zapis' (SMR): chto eto takoe i stoit li ee izbegat'? [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: [https://interface31.ru/tech\\_it/2022/12/chto-takoe-cherepichnaya-magnitnaya-zapis-smr-i-stoit-li-ee-izbegat.html](https://interface31.ru/tech_it/2022/12/chto-takoe-cherepichnaya-magnitnaya-zapis-smr-i-stoit-li-ee-izbegat.html)
3. Lipkin G. Cherepichnaya magnitnaya zapis' SMR: printsip raboty, preimushchestva i nedostatki [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: <https://habr.com/ru/articles/494614/>
4. 2021 SSD Benchmark Suite [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: <https://www.anandtech.com/print/16458/2021-ssd-benchmark-suite>
5. CrystalMark [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: <https://crystalmark.info/en/>
6. Defraggler [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: <https://www.ccleaner.com/defraggler?cc-noredirect=>
7. PrimoCache [Elektronnyi resurs]. Data obrashcheniya: 22 maya 2024 g. URL: <https://www.romexsoftware.com/en-us/primo-cache/>

## Performance optimization of machine learning-based image recognition algorithms for mobile devices based on the iOS operating system

**Mamadaev Ibragim Magomedovich**

iOS-Developer, Mail.ru LLC.

-, Russia, Moscow, blvd. Tverskoy, -

✉ [ibragim.m115@gmail.com](mailto:ibragim.m115@gmail.com)



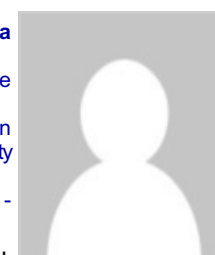
**Minitaeva Alina Mazhitovna**

PhD in Technical Science

Candidate of Technical Sciences, Department of Computer Science and Computer Engineering (IU-6), Bauman Moscow State Technical University

-, Russia, Moscow, blvd. Tverskoy, -

✉ [aminitaeva@mail.ru](mailto:aminitaeva@mail.ru)



**Abstract.** Today, mobile devices play an important role in everyone's daily life, and one of the key technologies leading to significant benefits for mobile applications is machine learning. Optimization of machine learning algorithms for mobile devices is an urgent and important task, it is aimed at developing and applying methods that will effectively use the limited computing resources of mobile devices. The paper discusses various ways to optimize image recognition algorithms on mobile devices, such as quantization and compression of models, optimization of initial calculations. In addition to ways to optimize the machine learning model itself, various libraries and tools for using this technology on mobile devices are also being considered. Each of the described methods has its advantages and disadvantages, and therefore, in the results of the work, it is proposed to use not only a combination of the described options, but also an additional method of parallelization of image processing



processes. The article discusses examples of specific tools and frameworks available for optimizing machine learning performance on iOS, and conducted its own experiments to test the effectiveness of various optimization methods. An analysis of the results obtained and a comparison of the performance of the algorithms are also provided. The practical significance of this article is as follows:

Improving the performance of machine learning algorithms on iOS mobile devices will lead to more efficient use of computing resources and increase system performance, which is very important in the context of limited computing power and energy resources of mobile devices. Optimization of machine learning performance on the iOS platform contributes to the development of faster and more responsive applications, which will also improve the user experience and allow developers to create new and innovative features and capabilities. Expanding the applicability of machine learning on iOS mobile devices opens up new opportunities for application development in various fields such as pattern recognition, natural language processing, data analysis, and others.

**Keywords:** parallelization, performance, efficiency, OS Apple, optimization, image recognition, iOS, machine learning, mobile device, neural network

## References (transliterated)

1. Chzhan Ya., Lyu Ya., Chen' T., Geng U. "Mobil'noe glubokoe obuchenie dlya intellektual'nykh mobil'nykh prilozhenii: Obzor." // *IEEE Access*, 8, 103, 586-103, 607.
2. Dokumentatsiya Apple Developer [Elektronnyi resurs] // "Core ML – Optimizatsiya proizvoditel'nosti na ustroistvakh." // URL: [https://developer.apple.com/documentation/coreml/optimizing\\_for\\_on-device\\_performance](https://developer.apple.com/documentation/coreml/optimizing_for_on-device_performance). Data obrashcheniya: [29.06.2023].
3. Rastegari M., Ordonez V., Redmon Dzh., Farkhadi A. "XNOR-Net: Klassifikatsiya izobrazhenii ImageNet s ispol'zovaniem binarnykh svertochnykh neironnykh setei." // *Sbornik materialov Evropeiskoi konferentsii po komp'yuternomu zreniyu (ECCV)* (str. 525-542).
4. Sikhota Kh., Mark A., Riandari F. i Rendell L. "Effektivnye algoritmy optimizatsii dlya razlichnykh zadach mashinnogo obucheniya, vkluchaya klassifikatsiyu, regressiyu i klasterizatsiyu." // *IEEE Access*, 1, 14-24. doi:10.35335/idea.v1i1.3
5. Sandler M., Khovard A., Chzhu M., Zhmoginov A., Chen L. Ts. "MobileNetV2: Invertirovannye ostatochnye bloki i lineinye uzkie mesta." // *Sbornik materialov konferentsii po komp'yuternomu zreniyu i obrabotke izobrazhenii* // *IEEE* (str. 4510-4520).
6. Khovard A. G., Chzhu M., Chen B., Kalenichenko D., Vang V., Veand T. I dr. "MobileNets: Effektivnye svertochnye neironnye seti dlya mobil'nykh prilozhenii komp'yuternogo zreniya." // *IEEE* 1704.04861.
7. Khan S., Mao Kh., Delli U. Dzh. "Glubokaya kompressiya: Szhatie glubokikh neironnykh setei s pomoshch'yu otsecheniya, kvantovaniya obucheniya i kodirovaniya Khaffmana." // *IEEE* 1510.00149.
8. Dokumentatsiya Google TensorFlow Lite. [Elektronnyi resurs]. «TensorFlow» // URL: <https://www.tensorflow.org/lite>. Data obrashcheniya: [29.06.2023].
9. Takkar M. "Nachalo mashinnogo obucheniya v iOS: CoreML Framework." // *IEEE Access*. doi:10.1007/978-1-4842-4297-1. ISBN: 978-1-4842-4296-4
10. Minitaeva, A. M. Prinyatie reshenii v usloviyakh interval'nogo zadaniya predpochtenii lits, primayushchikh reshenii // *Materialy konferentsii «Informatsionnye tekhnologii v*



- upravlenii» (ITU-2022): 15-ya MUL"TIKONFERENTsIYa PO PROBLEMAM UPRAVLENIYa, Sankt-Peterburg, 04 06 oktyabrya 2022 goda. – Sankt-Peterburg: Kontsern; Tsentral'nyi nauchno-issledovatel'skii institut ;Elektropribor;;, 2022. – S. 197-200. – EDN RNGSXI.
11. Minitaeva, A. M. Mnogomodel'nyi podkhod k prognozirovaniyu nelineinykh nestatsionarnykh protsessov v zadachakh optimal'nogo upravleniya // Neobratimye protsessy v prirode i tekhnike : Trudy Dvenadtsatoi Vserossiiskoi konferentsii. V 2-kh tomakh, Moskva, 31 yanvarya – 03 2023 goda. – Moskva: Moskovskii gosudarstvennyi tekhnicheskii universitet imeni N.E. Baumana (natsional'nyi issledovatel'skii universitet), 2023. – S. 438-447. – EDN FBVVXS.
  12. Kochnev A., «Kontseptual'nye osnovy prakticheskogo ispol'zovaniya neironnykh setei: problemy i perspektivy» // «Obshchestvo i innovatsii» // DOI: 10.47689/2181-1415-vol4-iss1-pp1-10
  13. Kurbaria M., Bengio I., Devid Zh. P. "BinaryNet: Obuchenie glubokikh neironnykh setei s ogranicheniyami na vesa i aktivatsii +1 ili -1." // IEEE: 1602.02830.
  14. Li G., Vei Gao, Vuen G., «Tekhniki kvantizatsii». DOI: 10.1007/978-981-97-1957-0\_5
  15. Samsiana S., S'yamsul A. «Algoritmy mashinnogo obucheniya s ispol'zovaniem metoda vektornogo kvantovaniya obucheniya». DOI: 10.1051/e3sconf/202450003010
  16. Aderemi A. Ataiero, Semyuel Adzhani «Obzor mashinnogo obucheniya na vstroennykh i mobil'nykh ustroistvakh: optimizatsiya i prilozheniya». DOI: 10.3390/s21134412
  17. Sandler M., Khovard A., LeKun I. "Mobilenetv3: Vysokoeffektivnaya masshtabiruemaya model' mobil'nogo komp'yuternogo zreniya." // Sbornik materialov konferentsii po komp'yuternomu zreniyu i obrabotke izobrazhenii // IEEE/CVF (str. 13840-13848).
  18. Chen B., Danda R. Yuan Ch. «Na puti k krazhe glubokikh neironnykh setei na mobil'nykh ustroistvakh» // Bezopasnost' i konfidentsial'nost' v setyakh svyazi (str. 495-508). DOI: 10.1007/978-3-030-90022-9\_27
  19. Dzharmoni F., Favzi A. «Zapusk neironnykh setei v Android» // Universitet Ottavy. Vvedenie v glubokoe obuchenie i neironnye seti s Python (cnh/ 247-280). DOI: 10.1016/B978-0-323-90933-4.00001-2
  20. Bykov K., Myuller K. «Opasnosti izobrazhenii s vodyanymi znakami v ImageNet» // Iskusstvennyi intellekt. Mezhdunarodnye seminary ECAI 2023 (str. 426–434). DOI: 10.1007/978-3-031-50396-2\_24

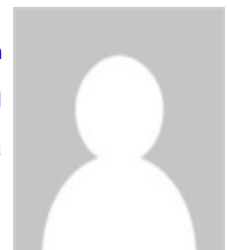
## Comparative analysis of Wine and PortProton: Cross platforms in the context of Windows application emulation

**Demidov Nikita Aleksandrovich**

Student; Department of Information Systems and Technologies; FSUE VO PGUTI

77 Moskovskoe shosse, Samara, Samara region, 443086, Russia

✉ danv9471@gmail.com

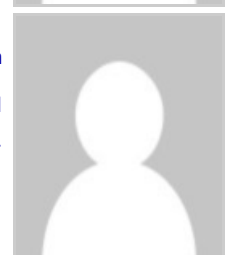


**Vygonyailo Klim Vladimirovich**

Student; Department of Computer Science and Computer Engineering; FSUE in PGUTI

443010, Russia, Samara region, Samara, Moskovskoe shosse, 77

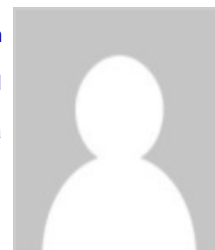
✉ mjchi023@gmail.com



**Manyayev Artem Al'bertovich**

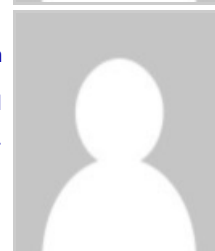
Student; Department of Computer Science and Computer Engineering; FSUE in PGUTI  
 77 Moskovskoe shosse, Samara, Samara region, 443086, Russia

✉ arl.mo@mail.ru

**Efimov Dmitrii Aleksandrovich**

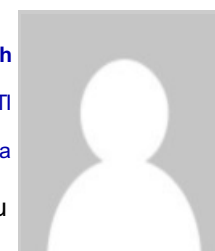
Student; Department of Computer Science and Computer Engineering; FSUE VO PGUTI  
 443086, Russia, Samara region, Samara, Moskovskoe shosse, 77

✉ di.efimoff2015@yandex.ru

**Bazhenov Artem Eduardovich**

Assistant; Department of Software Engineering; FSUE in PGUTI  
 77 Moskovskoe shosse, Samara, Samara region, 443086, Russia

✉ a.bazhenov@psuti.ru



**Abstract.** The modern development of computer technologies and operating systems is accompanied by an increase in the need for software capable of ensuring the interaction of various programs and applications with each other, regardless of their source environment. In this study, a comparative analysis of two such programs will be conducted – Wine and PortProton.

Wine is a program capable of running most applications developed for Windows on Unix-like systems. This is a compatibility layer that allows you to work with Windows applications. There is also a domestic version – PortProton, that offers the launch of Windows applications. This study aims to compare these two programs, analyze their features, advantages and disadvantages, determine which of them is the most convenient and functional for the end user in the context of Windows application emulation. The research methodology involves a comparative analysis of the Wine and PortProton platforms through benchmark testing and checking the performance of Windows applications on Linux. Benchmark testing includes evaluating the performance, stability, and speed of Windows applications on each platform. Due to the lack of scientific sources on the topic of comparing Wine and PortProton in the context of Windows application emulation, this study has a unique character. In conclusion Wine and PortProton successfully cope with the emulation of Windows applications, showing in some moments the best performance due to the optimization of the Linux operating system. PortProton copes best with the task of emulating programs due to stable operation and ease of use. Wine, despite a slight advance in the context of performance and the ability to run several programs at the same time, showed the worst efficiency due to the incorrect operation of some programs and the lack of an intuitive graphical interface. Based on the above conclusions, PortProton can be recommended for most users.

**Keywords:** Unix, Linux, Technical specifications, Windows, Performance, Wine, Portproton, Application compatibility, Emulation, cross platforms

## References (transliterated)

1. Kovalenkov V.A. Sovremennyye tendentsii populyarizatsii distributivovov Linux //

- Ryazanskii gosudarstvennyi universitet imeni S.A. Esenina. Ryazan'. 2014 – S. 95.
2. WineHQ [Elektronnyi resurs] URL: <https://www.winehq.org/> Data obrashcheniya: 22/04/2024
  3. Mishchuk, B. R. Wine – sreda obespecheniya ispolneniya Windows-prilozhenii v OS Linux // Gosudarstvennoe obrazovatel'noe uchrezhdenie vysshego professional'nogo obrazovaniya «Tomskii gosudarstvennyi pedagogicheskii universitet», kafedra informatiki. Tomsk. 2013. – S. 11.
  4. GitHub [Elektronnyi resurs] URL: <https://github.com/ValveSoftware/Proton> Data obrashcheniya: 22/04/2024
  5. Debian wine [Elektronnyi resurs] URL: <https://wiki.debian.org/Wine> Data obrashcheniya: 22/04/2024
  6. WineHQ Winetricks [Elektronnyi resurs] URL: <https://wiki.winehq.org/Winetricks> Data obrashcheniya: 22/04/2024
  7. Archlinux wiki Wine [Elektronnyi resurs] URL: [https://wiki.archlinux.org/title/Wine\\_\(Russkii\)](https://wiki.archlinux.org/title/Wine_(Russkii)) Data obrashcheniya: 22/04/2024
  8. Geeks3D FurMark [Elektronnyi resurs] URL: <https://www.geeks3d.com/furmark/> Data obrashcheniya: 22/04/2024
  9. SoftwareOK.com [Elektronnyi resurs] URL: <https://www.softwareok.com/?seite=Freeware/3D> .Benchmark.OK Data obrashcheniya: 22/04/2024
  10. CineBench [Elektronnyi resurs] URL: <https://www.maxon.net/en/cinebench> Data obrashcheniya: 22/04/2024
  11. Mathcad [Elektronnyi resurs] URL: <https://www.mathcad.com/en> Data obrashcheniya: 22/04/2024
  12. FL Studio [Elektronnyi resurs]. URL: <https://www.image-line.com/> Data obrashcheniya: 30/05/2024
  13. Zainabidinov R.Kh. Obzor yadra Linux i ego rol' v sovremennykh informatsionnykh sistemakh // Tashkentskii universitet informatsionnykh tekhnologii imeni Mukhammada al'-Khorezmii. Fergana. 2024 – S. 36.
  14. Dawid Bis, Kryspin Baran, Oliwia Kulawska. Perfomance comparison of different versions of Windows and Linux operating systems// Advances in Web Development Journal. 2023 – C. 117.
  15. WineHQ. Wine Application DataBase [Elektronnyi resurs]. URL: <https://appdb.winehq.org/> Data obrashcheniya: 11/06/2024

## Creating a common notation of the x86 processor software interface for automated disassembler construction

Gusenko Mikhail YU'evich

PhD in Technical Science

Associate Professor; Department of Applied and Business Informatics; Moscow Technological University

78 Vernadskystr., office 418, Moscow, 119454, Russia

✉ [mikegus@yandex.ru](mailto:mikegus@yandex.ru)



**Abstract.** The subject of the study is the process of reverse engineering of programs in order to obtain their source code in low- or high-level languages for processors with x86 architecture, the software interface of which is developed by Intel and AMD. The object of the study is the technical specifications in the documentation produced by these companies. The

intensity of updating documentation for processors is investigated and the need to develop technological approaches aimed at automated disassembler construction, taking into account regularly released and frequent updates of the processor software interface, is justified. The article presents a method for processing documentation in order to obtain a generalized, formalized and uniform specification of processor commands for further automated translation into the disassembler program code.

The article presents two main results: the first is an analysis of the various options for describing commands presented in the Intel and AMD documentation, and a concise reduction of these descriptions to a monotonous form of representation; the second is a comprehensive syntactic analysis of machine code description notations and the form of representation of each command in assembly language. This, taking into account some additional details of the description of the commands, for example, the permissible operating mode of the processor when executing the command, made it possible to create a generalized description of the command for translating the description into the disassembler code. The results of the study include the identification of a number of errors in both the documentation texts and in the operation of existing industrial disassemblers, built, as shown by the analysis of their implementation, using manual coding. The identification of such errors in the existing reverse engineering tools is an indirect result of the author's research.

**Keywords:** CPU mode, x86 AMD documentation, x86 Intel documentation, assembler, microprocessor, assembler syntax, machine code syntax, command specification, software reengineering, disassembler

## References (transliterated)

1. Versiya iskhodnogo koda operatsionnoi sistemy Windows 2000. Dostupno po ssylke: <https://github.com/pustladi/Windows-2000/> (data obrashcheniya: 23.02.2022)
2. Proekt Intel X86 Encoder Decoder (Intel XED). Dostupno po ssylke: <https://github.com/intelxed/xed> (data obrashcheniya: 23.08.2022)
3. Distribution of Intel and AMD x86 computer central processing units (CPUs) worldwide from 2012 to 2022, by quarter. Dostupno po ssylke: <https://www.statista.com/statistics/735904/worldwide-x86-intel-amd-market-share/> (data obrashcheniya: 23.09.2022)
4. Kitai nauchilsya legal'no klonirovat' protsessory Intel. Pomog starinnyi sopernik. Dostupno po ssylke: [https://www.cnews.ru/news/top/2023-06-09\\_kitajtsy\\_nauchilis\\_legalno](https://www.cnews.ru/news/top/2023-06-09_kitajtsy_nauchilis_legalno) (data obrashcheniya: 14.12.2023)
5. Intel® 64 and IA-32 Architectures. Software Developer's Manual. Documentation Changes. September 2023. Document Number: 252046-073 Dostupno po ssylke: <https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-documentation-changes.html> (data obrashcheniya: 21.10.2023)
6. Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4. Order Number: 325462-081US September 2023. Dostupno po ssylke: <https://www.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html> (data obrashcheniya: 20.10.2023)
7. Rasshirennaya forma Bekusa-Naura. Dostupno po ssylke: [https://ru.wikipedia.org/wiki/Rasshirennaya\\_forma\\_Bekusa-Naura](https://ru.wikipedia.org/wiki/Rasshirennaya_forma_Bekusa-Naura) (data obrashcheniya: 23.09.2022)

8. AMD64 Technology. AMD64 Architecture Programmer's Manual. Volumes 1–5.  
Publication No. 40332 Revision 4.07 Date June 2023. Dostupno po ssylke: [https://  
https://www.amd.com/content/dam/amd/en/documents/processor-tech-  
docs/programmer-references/40332.pdf](https://www.amd.com/content/dam/amd/en/documents/processor-tech-docs/programmer-references/40332.pdf) (data obrashcheniya: 20.11.2023)