

ООО «Базальт СПО»  
Институт Программных Систем РАН

**Восемнадцатая конференция  
«Свободное программное обеспечение  
в высшей школе»**

Переславль-Залесский, 27–29 января 2023 года

Сборник материалов конференции

Москва,  
МАКС Пресс,  
2023

УДК 004.91  
ББК 32.97  
В76



<https://elibrary.ru/evwukn>

Программный комитет:

*А.А. Савченко*, – председатель,  
*Г.В. Курячий*,  
*А.А. Маркина*,  
*А.В. Бондарев*,  
*М.О. Петрова*

**В76** **Восемнадцатая конференция. Свободное программное обеспечение в высшей школе** : материалы конференции / Переславль-Залесский, 27–29 января 2023 г. / Отв. ред. В.Л. Чёрный. – Москва : МАКС Пресс, 2023. – 132 с.

ISBN 978-5-317-06921-6

<https://doi.org/10.29003/m3136.978-5-317-06921-6>

В книге собраны материалы конференции, одобренные Программным комитетом восемнадцатой конференции. Свободное программное обеспечение в высшей школе.

*Ключевые слова*: свободное программное обеспечение, СПО, образование, образовательные программы, практика применения, профессиональное образование, вуз, информационные технологии в образовании.

УДК 004.91  
ББК 32.97

Program Committee:

A.A. Savchenko, – chairman,  
G.V. Kouryachy,  
A.A. Markina,  
A.V. Bondarev,  
M.O. Petrova

**The XVIII-th Conference “Free Software in Higher Education”** : materials of the conference / Pereslavl-Zaleski, 27–29 January 2023 / Chief editor V. L. Chyorny. – Moscow : MAKS Press, 2023. – 132 p.

ISBN 978-5-317-06921-6

<https://doi.org/10.29003/m3136.978-5-317-06921-6>

The book contains materials of the Conference which were approved by the Program Committee of The XVIII-th Conference “Free Software in Higher Education.”

*Keywords*: free software, educational programs, practical applications, education using free software.

ISBN 978-5-317-06921-6

© Коллектив авторов, 2023

© Оформление. ООО «МАКС Пресс», 2023

# Программа конференции

**27 января, пятница**

11.30-14.00 Заселение, обед, регистрация участников

13:20 Автобус от гостиницы «Переславль»

## Дневное заседание 14.00–19.00

14.00 Приветственное слово организаторов

14.20–14.50 А. А. Якушин

Свободные лицензии в эпоху автархий. Нужна ли нам  
русская GPL? ..... 9

14.50–15.20 Н. Н. Непейвода

Низкоквалифицированные задачи и ещё несколько замечаний об  
обучении ..... 10

15.20–15.40 А. С. Казанцев

Педагогические ВУЗы и СПО: проблемы подготовки  
будущих учителей к применению свободного и  
открытого ПО в школах ..... 14

15.40–16.00 И. А. Хахаев

Фестиваль установки российских операционных систем .... 19

16.00–16.20 С. А. Трусов и др.

Инфраструктура разработки отечественного программного  
обеспечения ..... 23

16.20–16.40 Кофе-пауза

|             |  |    |
|-------------|--|----|
| 16.40–17.10 | М. С. Пожидаев и др.   |    |
|             | Опыт инклюзивного преподавания: разработка,<br>распространение и чтение учебных материалов . . . . .                                   | 25 |
| 17.10–17.30 | Д. В. Исакевич   |    |
|             | Рефал: опыт формализации в преподавании информатики .  | 27 |
| 17.30–17.50 | А. А. Смирновский  |    |
|             | Опыт использования СПО OpenFOAM для обучения<br>основам вычислительной гидродинамики в СПбПУ . . . . .                                 | 31 |
| 17.50–18.10 | Ю. В. Шевчук и др.   |    |
|             | Малые компьютерные системы со свойствами<br>масштабируемости и высокой доступности . . . . .   | 33 |
| 18.10–18.30 | А. Г. Уймин  |    |
|             | Опыт развёртывания специализированных лабораторий<br>СПО и ВО на базе отечественного ПО. Проблемы<br>масштабирования решения . . . . . | 40 |
| 18.30–18.50 | А. А. Савченко   |    |
|             | Подготовка дистрибутивостроителей . . . . .  | 43 |
| 19:30       | Автобус в гостиницу «Переславль»   |    |

## 28 января, суббота

09:30 Автобус от гостиницы «Переславль»

### Утреннее заседание 10.00–13.00

|             |  |    |
|-------------|--|----|
| 10.00–10.20 | В. Л. Симонов и др.  |    |
|             | Разработка учебных проектов моделей подвижных<br>автоматических систем наземного и воздушного<br>размещения с использованием СПО . . . . . | 45 |
| 10.20–10.40 | В. Л. Симонов и др.  |    |
|             | Использование СПО в разработке учебных моделей систем<br>определения местоположения объектов различных типов                               | 49 |

|   |  |    |
|---|--|----|
| 10.40–11.00                                     | И. В. Воронин и др.<br>Классификация опухолевых клеток с использованием<br>моделей машинного обучения в среде Альт Линукс....  | 52 |
| 11.00–11.20                                     | Л. Б. Чашкин и др.<br>Использование инструментов и модулей open-source на<br>базе ОС Альт в организации гео-мониторинга .....  | 55 |
| 11.20–11.40                                     | Кофе-пауза   |    |
| 11.40–12.00                                     | Г. В. Курячий, В. В. Балашов<br>Как мы добывали огонь. Организация аудиторного<br>практикума по курсу «ЯП Python» на останках<br>дистанционного обучения .....           | 60 |
| 12.00–12.20                                     | С. А. Фомин<br>Современные «интерактивные среды» и «живые<br>лаборатории» — эффективное дистанционное<br>образование по алгоритмам и математическим<br>дисциплинам ..... | 63 |
| 12.20–12.40                                     | В. П. Кулагин и др.<br>Использование среды виртуализации в подготовке<br>системных программистов .....   | 64 |
| 12.40–13.00                                     | П. Б. Жданович и др.<br>Распределённое образовательное IaaS-облако для изучения<br>информационных технологий .....   | 67 |
| 13:00   | Автобус в гостиницу «Переславль»   |    |
| 13.00–15.00                                     | Перерыв на обед  |    |
| 14:40   | Автобус от гостиницы «Переславль»  |    |
| <b>Вечернее заседание</b><br><b>15.00–19.00</b> |  |    |
| 15.00–15.20                                     | А. А. Шульган и др.<br>Использование свободного программного обеспечения для<br>ЭЭГ-мониторинга психического состояния<br>пользователя в исследовании UX/UI .....        | 69 |

|             |  |    |
|-------------|--|----|
| 15.20–15.40 | В. А. Зенкин   |    |
|             | Применение языка разметки Markdown для написания отчётов о НИР и других сложноструктурированных документов с регламентированными требованиями к оформлению ..... | 73 |
| 15.40–16.00 | С. В Карпеш  |    |
|             | Организация записи и вещания научно-технических конференций и роль открытого ПО в данном процессе .  | 76 |
| 16.00–16.20 | А. Д. Дельман, А. Н. Непейвода и др.   |    |
|             | Интерактивный конвертер Chipollino для наглядного изучения теории автоматов .....  | 79 |
| 16.20–16.40 | М. В. Маркушевич, А. Н. Краснов  |    |
|             | О решении проблемы дефицита учебно-методической литературы, посвящённой методике эффективного использования СПО в учебном процессе российской школы .....        | 82 |
| 16.40–17.00 | Кофе-пауза   |    |
| 17.00–17.20 | А. В. Драгунов и др.   |    |
|             | Развитие региональной информационной образовательной среды, основанной на свободном программном обеспечении .....  | 85 |
| 17.20–17.40 | Я. Б. Шпунт  |    |
|             | Особенности российских операционных систем, используемых в ходе проектов по импортозамещению .   | 92 |
| 17.40–18.00 | А. И. Федосеев   |    |
|             | Всероссийский конкурс open source проектов: опыт проведения и перспективы .....  | 94 |
| 18.00–19.00 | М. О. Петрова, ведущий. Круглый стол: СПО в СПО и высшей школе.  |    |
| 19:30       | Автобус в гостиницу «Переславль»   |    |

**29 января, воскресенье**

09:30 Автобус от гостиницы «Переславль»

**Утреннее заседание****10.00–13.00**

10.00–10.20 Е. Е. Ковалёв

О необходимости изучения студентами методических аспектов внедрения свободного программного обеспечения и механизмов импортозамещения ..... 99

10.20–10.40 Е. Р. Алексеев и др.

Использование дистрибутива Альт Образование при информационной подготовке будущих учителей математики и информатики в Кубанском Государственном Университете ..... 102

10.40–11.00 А. И. Федосеев

Совместные образовательные программы по основам open source разработки Кругового движения и ОС Аврора 105

11.00–11.20 К. В. Чувиллин

Обучение школьников кроссплатформенной разработке на основе Qt и Qt Quick ..... 110

11.20–11.40 Кофе-пауза

11.40–12.00 В. М. Баканов

Практикум по изучению скрытого в алгоритмах параллелизма и его рационального использования в вычислениях ..... 115

12.00–12.20 В. В. Павловский

Взаимодействие высшей и средней школы при внедрении СПО в общеобразовательные учреждения ..... 118

12.20–12.40 Д. В. Диденко

Использование шаблона текстового процессора LibreOffice Writer для подготовки ВКР в колледже ..... 120

12.40–13.00 И. В. Воронин и др.

Настройка домашнего помощника Home Assistant в среде

Альт Линукс ..... 126

14:30 Автобус в Москву

## **Вне программы**

С. А. Мартишин, М. В. Храпченко

Реализация агрегированного графического запроса

конфиденциальных данных из облачной базы данных

с использованием языков Go и R ..... 128



---

Анатолий Якушин  
Москва

## Свободные лицензии в эпоху автаркий. Нужна ли нам российская GPL?

### Аннотация

Выбор лицензии для программного обеспечения, разработанного в процессе деятельности высшего учебного заведения, всегда являлся большой проблемой. Сегодня, когда происходит разрыв мирового экономического и правового пространства, подобный выбор становится всё более сложным. Доклад посвящён современным аспектам лицензирования программных продуктов в новых условиях.

Многие последователи движения свободного программного обеспечения (СПО) традиционно противопоставляют свободные и проприетарные лицензии. Однако следует понимать, что любая лицензия, вне зависимости от предоставляемых ей пользователю прав, может функционировать только в определённом правовом поле, где кроме воли автора программного продукта существует и механизм юридического принуждения к исполнению этой воли.

За те сорок лет, что существует движение СПО, неоднократно была юридически доказана состоятельность свободных лицензий, свидетельством чему являются тысячи выигранных судебных процессов по всему миру. Однако последние геополитические события показывают, что мы постепенно движемся к ситуации автаркий, причём не только и не столько экономически, сколько политически и юридически. Разрыв единого до недавнего времени юридического пространства ведёт к необходимости изменения отношения к лицензированию программных продуктов, в том числе и к свободному лицензированию.

При выборе программных продуктов для нужд любого учреждения или организации следует учитывать, что само наличие свободной лицензии ещё не гарантирует реальной свободы использования. Необходимо тщательно анализировать различные проприетарные отягощения в виде технологических или инфраструктурных ограничений, которые могут повлечь за собой невозможность реального использования продуктов даже в недалёкой перспективе. Весьма иллюстративен в этом плане пример Android или GitHub.

С другой стороны, использование свободного ПО в проприетарных продуктах отечественного производства, к чему сегодня призывают некоторые сторонники «параллельного импорта», является явно негативной тенденцией и не может приветствоваться сообществом. Трансграничный характер свободных лицензий необходимо соблюдать столько, сколько это будет возможно.

Что касается разработки российской свободной лицензии, то данный вопрос на протяжении последних пятнадцати лет возникал неоднократно в острые моменты, но откладывался по мере нормализации ситуации. Сегодня несомненно ситуация созрела окончательно. В постановлении Правительства Российской Федерации от 10.10.2022 № 1804 о проведении эксперимента по предоставлению права использования программ есть пример подобной лицензии. И хотя данная лицензия не лишена целого ряда недостатков, она вполне может быть использована в качестве отправной точки для дальнейшего развития.

Непейвода Н. Н.

Переславль-Залесский, ИПС РАН

## **Низкоэкспертные задачи и ещё несколько замечаний об обучении**

Низкоэкспертные задачи получили такое имя, потому что для них требуется совместная работа нескольких специалистов разных направлений (например, виртуальная реальность, юрист, экономист, астроном, врач, психолог). Специалистов, владеющих на уровне хотя бы экспертов для оценки, всеми направлениями нет. А имеющиеся подобны флюсу (К. Прутков), видят только часть проекта и автоматически дают низкие оценки либо отказываются от экспертизы. Это подтвердила экспертиза первого варианта данного доклада, где эксперт подменил задачу СПОВШ: СПО в обучении, на обучение СПО. Обучение, ставящее задачу создать у студента систему знаний и умений её пополнять, модифицировать и применять на практике, логически подобно низкоэкспертным задачам в программировании, находясь на стыке многих направлений знаний. А когда требуется совместная работа специалистов разных направлений, наступает ад взаимного непонимания и неприятия; даже слова одного раздражают другого вплоть

до идиосинкразии. Единственный работающий практический способ, известный в фирмах сейчас: вывезти людей в глушь и там заставить по крайней мере неделю пытаться работать и успешно совместно выпивать. В результате не менее половины отсеивается, но примерно четверть или треть наконец-то находят подобие общего языка и могут начинать совместную работу.

При обсуждении первого варианта тезисов с А. Непейвода нашлось следующее соображение. Согласно закону Паркинсона [1] решение комитета не выше минимума уровня минимума его членов. Алкоголь способствовал понижению уровня специалистов до момента, когда они наконец-то все поняли друг друга, но при этом выдать они могут лишь низкоуровневое решение.

Ещё с рубежа тысячелетий известен способ обучения, при котором примерно десятая часть студентов (если набор был нормальный и преподаватели не стеснены стандартами) становится способными выполнять роль коммуникаторов более успешно и быстро, не снижая до минимума уровень, снимающими взаимные непонятки у специалистов даже из таких областей, которые они встречают впервые в жизни. При этом и остальные поднимают свой уровень.

Способ обучения создавался в связи с рыночной ситуацией 90-х — первой половины нулевых годов. Тогда русские программисты считались чуть ли не сильнейшими в мире, хотя в отличие от индусов капризными, высокомерными и дорогими. А задачи на программистских фирмах ещё не сводились к потоку индустриального программирования. Из-за высокой репутации русских сложные задачи, причём из неожиданных областей, прилетали, и требовалось быстро и качественно их решить. Поэтому понадобились умения быстро и без *изучения* освоить новую область, с которой чаще всего дальше опять не придётся сталкиваться. Так что решалась задача взаимопонимания с людьми из другого интеллектуального мира. А умение быть коммуникатором при этом автоматически совершенствовалось.

Некоторые приёмы такого обучения.

1. Обязательная работа в СПО. СПО заставляет понимать задачи, решённые другими, и модифицировать их. В частности, студент, нашедший ошибки или недоработки в найденной или рекомендованной системе и частично исправивший их, на экзамене получал автоматом четвёрку, и пятёрку, если ответит на теоретический вопрос. Написание документации при попытке заком-

митить свою программу на репозитории тренировало в умении объяснять свои мысли другим. И заметим, что примерно половина студентов из-за неумения этого не могла пройти процесс интеграции программы в репозиторий, но зато к последнему (пятому) курсу они уже умели объяснять сделанное.

2. На первом же курсе знакомство с высокоуровневым (функциональным, логическим и сентенциальным) программированием и полный отказ от задов стандартных языков. Здесь использовалось лишь СПО, поскольку проприетарные системы были неадекватны задачам обучения.
3. Отказ от изучения языков и систем программирования, студент должен сам при помощи товарищей и преподавателя осваивать их в кратчайшие сроки. Это отрабатывалось в том числе и на СПО, предоставляющем богатый материал.
4. Тренировка в чтении научных и технических текстов на незнакомом языке с минимальным обращением к словарю. А это уже системное дополнение.
5. Работа на устаревших и плохо работающих машинах без нормальной документации. Тоже дополнение, необходимое для выработки системы знаний и умений.
6. На последнем курсе тренировка в решении и постановке междисциплинарных задач. Здесь полезно привлечение материала ТРИЗ [3]. СПО страдает одним недостатком, который стандартно принимается как данное в обучении программированию. Задача считается уже поставленной. А интердисциплинарная коммуникация практически всегда требует перестановки задачи, выданной партнёром. Так что это — необходимое дополнение, не охватывавшееся чистым СПО и не разработанное в его сообществе.

Ещё одно метазамечание. Отказ от научного языка. Как показал Альтшуллер, он *мешает* решению творческих междисциплинарных задач, поскольку направлен на изоляцию предметных областей путём уничтожения неформальных комментариев и создания системы терминов. В ТРИЗ от терминов рекомендуют избавляться, а задачу излагать лучше вульгарно, чем прилично. Здесь мы следуем традициям Платона [2], а не создателя научного языка Аристотеля (полностью признавая его великое достижение).

В библиографии далее приведены книги [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], вдохновившие автора в своё время. Классика не стареет, а новое часто продукт скоропортящийся.

## Литература

- [1] Паркинсон С. Н. *Закон Паркинсона и другие памфлеты*. М.: Прогресс, 1976. — 448 с.
- [2] Платон. *Собрание сочинений в 4-х томах*. СПб: 2006.
- [3] Г. С. Альтшуллер. *Творчество как точная наука*. М: Советское Радио, 1979.
- [4] В. И. Арнольд. «Жёсткие» и «мягкие» математические модели. М.: МЦНМО, 2004. 32 с.
- [5] К. Бек. *Экстремальное программирование*. СПб.: Питер, 2002, ISBN 5-94723-032-1.
- [6] Иммануил Кант. *Критика чистого разума* / Пер. с нем. Н. Лосского сверен и отредактирован Ц. Г. Арзаканяном и М. И. Иткиным; Примеч. Ц. Г. Арзаканяна. — М.: Эксмо, 2007. — 736 с.
- [7] Н. И. Конрад. *У-цзы. Трактат о военном искусстве*.
- [8] Лиддел Гарт Б. Х. *Стратегия непрямых действий*. — М.: ИЛ, 1957. Перевод и исследование. в кн.: Н. И. Конрад. *Избранные труды. Синология*. М., 1977.
- [9] М. Коэн, Э. Нагель. *Введение в логику и научный метод* М.: Социум, 2010. 654 с.
- [10] А. Гейтинг. *Интуиционизм. Введение*. М: 2010, 163 с.
- [11] *Омар Хайям в кругу мудрости*. Симферополь: Реноме, 1999.
- [12] Д. Пойа. *Как решать задачу?* М.: Наука, 1961.
- [13] Д. Пойа. *Математическое открытие*. М.: Наука, 1970.
- [14] Д. Пойа. *Математика и правдоподобные рассуждения*. М.: Наука, 1975
- [15] Р. Дж. Ханк. *Как пасти котов. Наставление для программистов, руководящих другими программистами*. СПб, Питер, 2002
- [16] Э. Йордон. *Путь камикадзе*. Лори: 2004.
- [17] Ло Гуань-чжун. *Троецарствие*. В 2-х томах. М.: 1954.

- [18] Н. Макиавелли. *Рассуждения о первой декаде Тита Ливия*. Ростов-на-Дону, Феникс: 1998
- [19] Евгений Токарь. *Записки о гениях*. Изд. 2-е, М: Аграф, 2006.

**Александр Казанцев**

Глазов, ФГБОУ ВО «Глазовский государственный педагогический институт им. В.Г. Короленко»

## **Педагогические вузы и СПО: проблемы подготовки будущих учителей к применению свободного и открытого ПО в школах**

### **Аннотация**

В статье рассматриваются проблемы подготовки будущих учителей к применению СПО в школах России: от наличия предубеждений к СПО и проблем с методическим обеспечением до проблем, исходящих от самих разработчиков. Также будут предложены варианты решения данной проблемы на примере российского свободного программного обеспечения.

Не секрет, что внедрение свободного программного обеспечения (далее СПО) в образовательных учреждениях идёт уже более 10 лет, но сам процесс банально «буксует» и даже саботируется как руководством, так и преподавателями. И причина этого лежит в том числе и в плоскости подготовки кадров в педагогических вузах России. Ниже я попытался собрать в некоторый «ТОП-10» основные проблемы, которые влияют на этот процесс и попытался выработать рекомендации, как сдвинуть ситуацию с «мертвой точки». Проблематика исследована как на личном опыте, так и на опыте коллег по педагогическому вузу.

### **1. Отсутствие мотивации**

Основная проблема, из-за которой многие образовательные учреждения не переходят на СПО. Хотя государство с 2012 года и после регулярно заявляет о необходимости технологической независимости от зарубежных софтверных компаний и об использовании в

государственных учреждениях, коими являются и школы в том числе, только «безопасного» программного обеспечения, но формально всё это носит «рекомендательный» характер (для образовательных учреждений). Менять рабочие программы, переустанавливать парк программного обеспечения, переобучать как технический персонал в вузах, так и преподавателей на использование СПО руководство вузов не видит смысла, а министерства, отвечающие за образование как на федеральном, так и на региональных уровнях не настаивают на этом. Даже ситуация 2022 года с массовой «отменой» лицензий и невозможностью их продления для образовательных учреждений не помогла процессу внедрения СПО.

## 2. Предубеждение и скепсис

Многие преподаватели относятся к свободному программному обеспечению как к ПО «второго сорта», которое не дотягивает по своим характеристикам и удобству до «стандартов» рынка от Microsoft или Adobe. Массовый скепсис, что та же ОС Linux может занять большую часть рынка в стране, как и другой свободный софт, приводит к нежеланию внедрять в практики обучения будущих учителей свободное программное обеспечение. То же касается и отношения к «бесплатности» программного обеспечения, когда считается что качественный софт и поддержка может быть только у коммерческих продуктов. Исключением стали только некоторые виды ПО, которые получили мировую известность (например Scratch или GIMP). Многие преподаватели «живут» в реалиях десятилетней давности и не понимают, как далеко продвинулось свободное ПО по своим возможностям.

## 3. Отсутствие методической поддержки

Для преподавателей важна методическая поддержка как на этапе подготовки будущих учителей, так и на этапе непосредственной работы в школах. То есть должна быть проработанная методика подготовки/переподготовки, учебные планы и методические указания для проведения занятий на СПО, а самое главное УМК для проведения занятий в образовательных учреждениях. Но на дворе 2023 год, мы имеем по факту «единую» линейку учебников, где СПО хоть и не запрещается, но и упоминается вскользь. Та же утверждённая базо-

вая линейка учебников по дисциплине «Информатика» [2] остаётся в своей массе построенной на Windows-совместимом ПО, как примере операционной системы, как в части учебников, так и в части практикумов. Книги по свободному программному обеспечению за последние года также издаются в минимальном количестве или отсутствуют.

#### **4. Отсутствие информации**

Лично помогая [2] в 2022 году по созданию учебных курсов и материалов по свободному и открытому программному обеспечению разработчикам ПО и движению НТИ, в беседах с преподавателями школ и вузов я столкнулся с тем, что многие до сих пор имеют поверхностные или отсутствующие знания как в части программного обеспечения, так и в части, где можно найти информацию или пройти курсы по организации учебного процесса с помощью СПО. Педагогические вузы, которые должны быть первичной средой для воспитания будущих кадров для продвижения СПО, по большей части, за исключением некоторых пилотных проектов, этого не делают, а существующие курсы больше направлены на администраторов или пользователей АРМ.

#### **5. Неразбериха в классификации, лицензиях и «разрешениях»**

Проблема также заключается в том, что всё-таки можно и нужно использовать при подготовке преподавателей, и что они потом «понесут» в школы. Число лицензий огромное, разницу между свободным, открытым и бесплатным ПО часто могут отличить только юристы, для большей же части «простых» людей эти три вида имеют тождественное понятие. В едином реестре российских программ находятся как проприетарные аналоги, так свободные, при этом часть СПО является зарубежным. Иногда программное обеспечение, которое по виду является свободным, требуют денежных средств за лицензии, или наоборот часть ПО, не являющаяся свободным, предоставляется бесплатно. Что может быть использовано в образовательных учреждениях, какое ПО не «уведут» из страны, отозвав его из «реестра», какое ПО не будет запрещено, так как оно не будет «российским» — все эти вопросы мешают нормальному внедрению не только СПО, но и российского ПО в частности.



## 6. Непостоянство и несовершенство свободного программного обеспечения

Не секрет, что свободное программное обеспечение создаётся силами комьюнити, что с одной стороны, есть хорошо, с другой приносит проблемы в части пользовательского опыта или вектора его развития. Нормальным для такого вида ПО будет потеря интереса со стороны основных разработчиков, появления ответвлений-форков, резкая смена парадигмы развития, вплоть до полной переработки основных функций и интерфейса, отсутствие внятного UI/UX и масс-теста на пользователях. Подготовка же преподавателей и выпуск методических комплексов и учебников требует какого-никакого, но постоянства в функционале используемого программного обеспечения, его преемственности от версии к версии и предсказуемого его развития на протяжении ближайших 5–10 лет. С другой стороны это развитие должно быть эволюционным и следовать современным трендам по функционалу ПО. Этому требованию ещё как-то удовлетворяют крупные проекты прикладного ПО (хотя тот же GIMP или LibreOffice уже достаточно сильно отстают по тем же AI-функциям от своих проприетарных аналогов), но в части самих операционных систем имеем или арахайку в интерфейсе, или неуёмный эксперимент в новых DE. При этом многие виды СПО имеют вид, как будто интерфейс делался или «инопланетянами» или «школьниками» и пригодны только для утилитарного применения.

## 7. «Сектантство» и отсутствие единого движения

Разработчики СПО со стороны часто выглядят «сектой», которая пытается заманить к себе последователей и после насадить свои строгие правила. В крайних проявлениях доходить до уровня, что нельзя использовать никакое ПО, кроме свободного, открытого ПО это ересь, а проприетарное — вселенское зло. И пропагандируют это в массы, в том числе и преподавателям, вызывая отторжение у последних. Также среди российских разработчиков программного обеспечения нет сплочённости и единого понимания, в какую сторону должна двигаться отрасль, что опять же мешает синергии. При этом СПО разработчики умудряются «переругаться» даже между собой, а процесс перевода школ и вузов на СПО требует «единого фронта».

## 8. Смена парадигмы потребления ПО

Современный потребитель софта живёт в «облаках» и мобильных системах и «де-факто» уже не нуждается в операционных системах или настольном ПО. Подготовка же что учителей, что ими учеников «застряла» на уровне 10-15 летней давности и не учитывает современные реалии. Использование СПО упирается в отсутствие необходимых «связей» с гаджетами учащихся (формальный запрет на их использование на уроках является лишь попыткой закрыть глаза на реалии) или в непонимании, для чего тот или иной софт может пригодиться в дальнейшем.

Массовый переход к браузерным технологиям по идее наоборот должен был подстегнуть переход на те же операционные системы на основе Linux, но проблемой теперь (и облачных технологий в целом) становится наличия Интернет в образовательных учреждениях. И игнорирование российскими разработчиками возможностей создания кросс-платформенного софта, работающего вне зависимости от операционной системы или аппаратных характеристик устройства.

## 9. Отсутствие «евангелистов» СПО

Любую идею надо «нести в массы». Для продвижения свободного программного обеспечения (а моё личное мнение, что надо продвигать синергию ПО, включающую свободное, открытое и российское) необходимо задуматься о наличии «евангелистов», то есть людей профессионально занимающихся продвижением технологий и продукты путём написания статей, ведения блогов, проведения семинаров и вебинаров, демонстраций и презентаций, ведения переговоров и т. п. Для образовательных учреждений это должны быть специалисты, разбирающиеся как в ИТ-области, так и в образовательном процессе.

## 10. Грубость разработчиков

Проблема, с которой сталкиваются многие преподаватели. Пытаясь разрешить возникающие у них проблемы или ища ответы на вопросы, они попадают в мир ИТ, где часто вместо помощи получают оскорбления. Интеллигентный человек не будет пытаться прорваться через множество советов «читай инструкцию» или «это уже было на форуме, потрудись поискать», а просто забросит программный про-

дукт. Это также связано и с отсутствием сообществ именно преподавателей, где они могли бы получить помощь и совет от коллег, из-за чего им приходится искать её у аудитории с другими принципами общения.

По итогу приходим к следующим выводам: внедрению СПО в образовательных мешает как отсутствие «воли» государства, так и проблемы с самими разработчиками. Необходимо выстраивание сотрудничества между разработчиками программного обеспечения и образовательными учреждениями, создание и продвижение единой повестки по СПО в законодательной и методической базах, закрепление СПО в УМК для образовательных учреждений если не основным, но обязательным, а не рекомендательным, а также развитие центров обучения СПО на базе педагогических вузов и помощь им в этом.

## Литература

- [1] УМК «Информатика». [электронный ресурс] <https://lbz.ru/metodist/authors/informatika/3/>
- [2] Open Source для образования — взгляд через 10 лет. [электронный ресурс] <https://habr.com/ru/post/660173/>

И.А.Хахаев

Санкт-Петербург, СПбГЭТУ «ЛЭТИ»

<https://etu.ru/>

## Фестиваль установки российских операционных систем

### Аннотация

Фестиваль установки российских операционных систем проводится с целью популяризации российских операционных систем и получения студентами практических навыков создания программных конфигураций для решения поставленных задач с использованием свободного программного обеспечения. В ходе фестиваля используются только дистрибутивы ОС от компаний, чья продукция представлена в «Едином реестре российских программ для электронных вычислительных машин и баз данных» (<https://reestr.digital.gov.ru/reestr/>),

дистрибутивы и дополнительные источники приложений (репозитории) должны быть в свободном доступе (не требовать покупки лицензий, регистрации кодов и т.п.). На Фестиваль приглашаются представители компаний-разработчиков российских операционных систем и компаний-разработчиков прикладных программных средств для работы в среде российских операционных систем.

Первый раз Фестиваль проводился в СПбГЭТУ «ЛЭТИ» в конце 2019 года [1] и планировался как ежегодный, но вследствие пандемийных ограничений второй раз удалось его провести только весной 2022 года [2].

Изменения в 2022 году по сравнению с 2019 годом были следующие

- 2019: Студенты одного факультета, только студенты СПбГЭТУ «ЛЭТИ», три дистрибутива (Simply Linux, Astra Linux Common Edition, Rosa Fresh), количество участников (студентов) – 27
- 2022: Студенты с разных факультетов СПбГЭТУ «ЛЭТИ», студенты СПб колледжа информационных технологий, четыре дистрибутива (Simply Linux, Astra Linux Common Edition, Rosa Fresh, RedOS), количество участников – 38.

В Фестивале участвуют команды в составе от 3 до 5 человек, каждая команда выбирает три или четыре (в 2022 году) дистрибутива из указанных. Участникам команд предоставляются ссылки на сайты разработчиков и соответствующую документацию. В порядке подготовки к фестивалю участники команд создают себе загрузочные варианты выбранных дистрибутивов на носителях USB-флэш. USB-носители для записи установочных образов предоставляются организаторами.

Для проведения Фестиваля требуется компьютерный класс с возможностью установки Linux как второй операционной системы.

После завершения установки предлагается доработка программной конфигурации для решения заданных классов задач с использованием пакетной базы (репозитория) дистрибутивов.

Задания по доработке программной конфигурации адаптированы к профилю инженерной подготовки в области электроники, цифровых коммуникационных и информационных технологий.

Список заданий (кейсов) следующий:

1. Рабочая станция для инженерных и математических расчётов (численные методы, символьная математика)
2. Рабочая станция САПР электроники

| Дистрибутив   | Количество команд |
|---------------|-------------------|
| AstraLinux CE | 4                 |
| RedOS         | 7                 |
| Rosa Fresh    | 9                 |
| Simply Linux  | 8                 |

Таблица 1: Распределение первоначального выбора дистрибутивов

3. Рабочая станция для моделирования аналоговых и цифровых схем
4. Рабочая станция для подготовки научно-технических публикаций.

На всю работу по установке и дополнительной настройке выделяется 6 часов. После завершения работы каждая команда должна презентовать результаты своей работы перед жюри Фестиваля с указанием всех выявленных достоинств и недостатков каждого решения. В презентацию включается как сведения о выбранных дистрибутивах, так и оценка пригодности дистрибутивов для обеспечения вышеуказанных кейсов. Также оценивались понятность процедуры установки и возможность обеспечить выбор варианта загрузки ОС после установки дистрибутива Linux.

Распределение первоначального выбора участников команд по дистрибутивам в 2022 году приведено в таблице 1.

В таблице 2 приведены критерии оценки дистрибутивов и сами оценки, сделанные участниками фестиваля. «0» означает полное отсутствие возможности, «1» – безусловное наличие возможности, промежуточные значения означают ситуацию «есть, но не вполне точно».

Таблица 2: Результаты оценки отечественных ОС участниками Фестиваля

| Параметр для оценки                         | Astra Linux CE | ROSA Fresh | Simply Linux | RedOS |
|---|----------------|------------|--------------|-------|
| Наличие Live-режима на загрузочном носителе | 1              | 1          | 1            | 0     |
| Двойная загрузка с Windows                  | 0              | 1          | 1            | 1     |
| Доступность репозитория                     | 1              | 1          | 1            | 1     |
| Полнота документации                        | 1              | 1          | 1            | 1     |
| ПО для инженерных и математических расчётов | 1              | 1          | 1            | 0,33  |
| САПР электроники                            | 0              | 0,5        | 1            | 0     |

Таблица 2 – Продолжение

| Параметр для оценки                             | Astra Linux CE | ROSA Fresh  | Simply Linux | RedOS       |
|---|----------------|-------------|--------------|-------------|
| ПО для моделирования аналоговых и цифровых схем | 0              | 0,66        | 1            | 0           |
| ПО для подготовки научно-технических публикаций | 1              | 1           | 1            | 1           |
| Понятность установщика                          | 0,6            | 1           | 0,8          | 0,6         |
| Минимальное время установки                     | 0,6            | 0,8         | 1            | 0,4         |
| <b>Итого баллов</b>                             | <b>6,2</b>     | <b>8,96</b> | <b>9,8</b>   | <b>5,33</b> |

По итогам работы команды выбрали предпочтительный дистрибутив. Результаты выбора показаны в таблице 3.

| Дистрибутив   | Количество команд |
|---------------|-------------------|
| AstraLinux CE | 1                 |
| RedOS         | 1                 |
| Rosa Fresh    | 3                 |
| Simply Linux  | 5                 |

Таблица 3: Предпочтения по итогам Фестиваля

## Литература

- [1] Будущие драйверы отечественной промышленности: фестиваль по установке российских операционных систем в ЛЭТИ. Электронный ресурс. Режим доступа: <https://etu.ru/ru/fakultety/fkti/novosti/budushhie-drajvery-otechestvennoj-promyshlennosti-festival-po-ustanovke-rossijskih-operacionnyh-sistem-v-leti>
- [2] Свободный софт и новые возможности: в ЛЭТИ прошёл фестиваль установки российских ОС. Электронный ресурс. Режим доступа: <https://etu.ru/ru/fakultety/fkti/novosti/svobodnyj-soft-i-novye-vozmozhnosti-v-leti-proshel-festival-ustanovki-rossijskih-os>.

Трусов Сергей, Клоков Андрей, Андреев Анна, Кирсанов Виктор, Черствов Тимофей

Москва, Общественно-государственное объединение «Ассоциация документальной электросвязи (АДЭ), базовая кафедра «Технологии электронного обмена данными» (ТЭОД) в Московском Техническом Университете Связи и Информатики (МТУСИ)

## **Инфраструктура разработки отечественного программного обеспечения**

### Аннотация

В докладе обсуждаются проблемы соблюдения свободных лицензий в ПО для «Эльбрус», а также создание и основные разделы новой учебной дисциплины «Инфраструктура разработки отечественного программного обеспечения», разрабатываемой в рамках Академии Программирования для архитектуры «Эльбрус».

В рамках Академии Программирования заключено трёхстороннее соглашение между АО «МЦСТ» и АО «ИВК», которые готовят проект материалов дисциплины по теме «Разработка и оптимизация программного обеспечения для архитектуры «Эльбрус».

На основе материалов, будет разработана новая учебная дисциплина «Инфраструктура разработки отечественного программного обеспечения», предназначенная для студентов бакалавриата 3 курса 2-го семестра и 4 курса 1-го семестра.

Новая учебная дисциплина включает в себя следующие основные разделы:

- Программирование, оптимизация, отладка, портирование на отечественные и зарубежные архитектуры микропроцессоров;
- Сборка и сопровождение совместимого программного продукта в отечественных репозиториях, включая репозиторий Сизиф;
- Тестирование программного продукта в отечественной доверенной программно-аппаратной среде;
- Приложения для ИКТ-инфраструктуры на отечественных программно-аппаратных средствах;
- Обучение C++.

Преподавание всех разделов новой учебной дисциплины будет включать лабораторные и практические занятия в оснащённых отечественными программно-аппаратными средствами учебно-исследовательских и учебно-производственных лабораториях.

С помощью разработанной дисциплины будет осуществляться подготовка кадров, способных использовать, разрабатывать и продвигать отечественные программные решения, работающие на базе процессоров «Эльбрус».

Также в докладе обсуждаются проблемы соблюдения свободных лицензий в ПО для «Эльбрус». Для полноценного использования в учебном процессе важно обеспечить соблюдение условий свободных лицензий на ПО для «Эльбрус». Без этого невозможно легально предоставлять такое программное обеспечение студентам, теряются все преимущества использования в обучении СПО с доступными исходными кодами, и главное, воспитывается правовой нигилизм.

Доработка ключевых компонентов, распространяющихся на условиях GNU GPL, в рамках госконтракта не должна приводить к нарушению лицензии и закрытию кода. Тут важен эксперимент Минцифры по публикации разработанного по госзаказу ПО под свободной лицензией.

## Литература

- [1] Мурад Нейман-заде, Королёв С., *Руководство по эффективному программированию на платформе Эльбрус*, Учебное пособие. — АО МЦСТ, 2020 г.
- [2] Ким А. К., Перекатов В. И., Ермаков С. Г., *Микропроцессоры и вычислительные комплексы семейства Эльбрус*, Учебное пособие, 2013 г.
- [3] Иванюк А. В. *Типовой отечественный модуль изучения интернет-технологий*. — М.: Документальная электросвязь, №28, 2018.
- [4] Кирсанов В. А., Лопухов Р. С. *Апробация типового отечественного модуля изучения интернет-технологий* /Пятнадцатая конференция «Свободное программное обеспечение в высшей школе: Материалы конференции» / Переславль, 7–9 февраля 2020 года. — М.: МАКС Пресс, 2020. — 180 с.



Михаил Пожидаев, Елена Теплых

Томск, ТГУ, Директ-Медиа

Проект: LUWRAIN <https://luwrain.org>

## Опыт инклюзивного преподавания: разработка, распространение и чтение учебных материалов

### Аннотация

Доклад рассматривает проблематику работы с материалами, которые можно использовать в инклюзивном образовании людей с нарушениями зрения. В основе находится личный опыт автора, успешно преподающего ряд дисциплин на кафедре теоретических основ информатики Томского государственного университета. В своей работе автор не использует какое-либо проприетарное программное обеспечение.

Материалы для инклюзивного преподавания и для инклюзивного обучения имеют много общего, хотя и могут различаться в отдельных чертах. Рассмотрим основные особенности работы с ними, преследуя цель организации обучения школьников и студентов с нарушениями зрения. Если обобщить имеющийся опыт, то можно прийти к заключению, что основные задачи могут успешно решаться, но с использованием инструментов, требующих длительное обучение для работы с ними. В таких условиях главной проблемой становится не организация доступной работы с учебными материалами, а выработка решений, пригодных для массового и тиражируемого использования.

Одним из ключевых инструментов для работы с учебными материалами была и остаётся издательская система Tex. Поскольку разработка материалов производится в текстовом виде, их создание хорошо подходит для людей с нарушениями зрения. При этом резко обостряется проблема удобной невизуальной интерактивной среды, способной сделать этот процесс комфортным. Требуется обеспечить проверку правописания, и возможность дополнения системы пользовательскими скриптами для автоматизации наиболее частых действий. Подобный функционал внедряется в платформу невизуальных приложений LUWRAIN, являющейся комплектом компонентов на языке Java для построения интерфейсов, ориентированных на людей с нарушениями зрения. Проверка правописания основана на библиотеке LanguageTool, а поддержка скриптов — на GraalVM.

Для учащихся школы работы с TeX является излишне трудоёмкой. Для организации работы над документами следует рассмотреть возможность интеграции с облачными сервисами, предлагающих функционал офисных приложений. Полноценная интеграция офисных пакетов на Linux с системами экранного доступа затруднена из-за архитектурных недостатков системы передачи данных для вспомогательных технологий.

TeX в той или иной мере позволяет решать задачи подготовки материалов как преподавателями, так и учащимися (хотя неприятной остаётся трудность невизуального чтения выходных материалов TeX). Он же существенно помогает в чтении физико-математических материалов, если они доступны в этом формате (в формате TeX сохранены математические записи в Википедии). В случае чтения материалов для гуманитарных дисциплин наиболее полезными являются форматы, аудиокниг в которых текстовая и аудиоданные синхронизированы друг с другом. Примером такого формата может являться международный формат DAISY, который удобен для чтения, но весьма трудоёмок для создания.

Создание аудиокниг в формате DAISY в нынешних условиях оправдано с использованием коммерческих синтезаторов речи, качество работы которых всё больше и больше приближается к качеству чтения диктором. При этом развитие подобных технологий требует дополнительных исследований в области обработки естественного языка, потому что необходимо решение технических задач по предварительной очистке входных материалов, преобразованию письменной речи в устную и т. д. Издательство «Директ-Медиа» ведёт исследования в этой сфере и разрабатывает облачный сервис, позволяющий автоматизировать изготовление аудиокниг в формате DAISY. Проблемой создания аудиокниг в формате DAISY обычно являются правовые вопросы, потому что адаптацию учебных материалов может произвести только их правообладатель.

Рассматривая работу с учебными материалами, не следует забывать о роли, которую в этом процессе могут играть встраиваемые компьютеры. Платформа LUWRain может быть запущена на одноплатном бюджетном компьютере Raspberry Pi, чей потенциал в сфере образования в России определённо недооценён. От доступных на первый взгляд мобильных телефонов Raspberry Pi отличает возможность запуска полноценной версии Linux с поддержкой TeX и других прило-

жений. Предполагая, что пользователь не будет подключать монитор, цена комплекта варьируется от 10 до 15 тысяч рублей.

Развитие способов работы с учебными материалами в сфере инклюзивного образования, как ни странно, связано не столько с технологическими проблемами (мы видим, что решения в том или ином виде существуют), сколько с правовыми, экономическими и организационными барьерами. Образовательные коллективы и коллективы, ведущие разработку программного обеспечения — это обычно разные коллективы. Внедрение первыми продуктов, полученными последними, предполагает процесс обучения педагогов, разработку документации и иллюстративных материалов и пр., что по затратам времени и усилий сравнимо с самим процессом разработки. Привлечение дополнительного персонала неоправданно в силу не очень большой аудитории людей, испытывающих потребность в инклюзивном образовании. Опыт показывает, что преодолеть этот барьер на практике крайне трудно. Усилий как со стороны педагогов, так и со стороны программистов хватает только для прототипирования основного функционала. Следует также отметить, что инклюзивное образование не может формировать коммерческий рынок, что дополнительно тормозит развитие в этой области.

**Даниил Исакевич**

Владимир, Владимирский строительный колледж

## **Рефал: опыт формализации в преподавании информатики**

### Аннотация

Представлен опыт использования связки Рефала, вычислителя программ в структурном стиле и  $\LaTeX$  в работе преподавателя информатики строительного колледжа. На примере задачи вычисления таблицы истинности выражения в логике высказываний подробно описаны: опыт автоматической массовой генерации примеров и ответов, вынуждающий преподавателя учиться, и устройство генератора. Показаны некоторые возможности, которые отказ от стандартной реализации арифметики может дать преподавателю. Отмечены преимущества написания Рефал-программ с русскими именами переменных и функций.

Цель настоящего доклада — представить годичный (2022 г.) опыт использования Рефала и свободного программного обеспечения внештатным преподавателем информатики (первый курс обучения, фактически повторяющий восьмой класс школы).

## Автоматическая массовая генерация примеров и ответов

Первая из решавшихся задач — автоматическая массовая генерация примеров (выражений логики высказываний, содержащих заданное число переменных и связей) и ответов (таблиц истинности) в рамках изучения «математической логики». Использование автоматической массовой генерации:

- вынуждает самого преподавателя учиться в ходе формализации примеров;
- обеспечивает максимально возможное разнообразие примеров в рамках заданного формализма.

Имеющиеся в виде бумажных распечаток у преподавателя и свободно раздаваемые студентам примеры (к которым у преподавателя имеются ответы) используются как материал для объяснения, а также как массив тестовых заданий.

Генерация выражений и таблиц истинности производится под управлением Bash-скрипта, запускающего последовательное выполнение следующих действий:

1. Вычисляется массив достаточно больших (от 1 до  $10!$ ) псевдослучайных целых чисел  $S_k$ . Выбор вычислителя продиктован личными предпочтениями автора (GNU Octave, но мог бы быть и Scilab).
2. Программа на Рефале строит последовательность структурных выражений Рефала, представляющих выражения логики высказываний, каждое из которых содержит заданное число переменных и связей. Выражение строится как дерево, выбор разбиения дерева на поддеревья, а также переменных и связей (включая 2-, 3- и 4-местные варианты  $\wedge$ ,  $\vee$ ) зависит от  $S_k \bmod N_k$ , где  $N_k$  — число вариантов  $k$ -ого выбора, делаемого программой.
3. Программа на Рефале преобразует последовательность сгенерированных выражений в последовательность команд GNU Octave для вычисления таблицы истинности, выводимую в M-файл.

4. GNU Octave выполняет созданный M-файл, выводя таблицы истинности.
5. Программа на Рефале преобразует последовательность сгенерированных выражений в размеченный для L<sup>A</sup>T<sub>E</sub>X документ примеров. Другая программа на Рефале преобразует вычисленные таблицы истинности в размеченный для L<sup>A</sup>T<sub>E</sub>X документ ответов.
6. Документы примеров и ответов выводятся pdflatex в PDF-файлы и открываются для просмотра.

Передача данных между программами осуществляется через файлы, содержащие текст и доступные для анализа по окончании генерации. Целостность данных на всех этапах (от автоматической генерации до свободного оборота бумажных распечаток) обеспечивается:

- сквозной (по всем этапам обработки) нумерацией сгенерированных выражений и таблиц истинности, передаваемой в составе обрабатываемых данных;
- указанием даты и времени вёрстки в нижнем колонтитуле каждого PDF-файла.

Связка Рефала для преобразования структурированных текстов, вычислителя программ в структурном стиле и L<sup>A</sup>T<sub>E</sub>X для вёрстки документов с формулами была использована автором для генерации примеров и ответов по другим темам; успех не был достигнут там, где отсутствовала приемлемая формализация (в частности, при работе с блок-схемами и «цифровыми схемами»).

## Отказ от стандартной реализации арифметики

Второе направление, для продвижения в котором был применён Рефал, возникло при освоении тем «измерение количества информации» и «позиционные системы счисления». Выяснилось, что отказ от стандартной реализации арифметики и представление числа цепочкой символов текста позволяют в ряде случаев использовать Рефал как вычислитель «абсолютной точности». К таким случаям относятся:

- перевод «из килобитов в гигабайты» — все действия сводятся к умножению и делению на степени 2; получающиеся числа и дроби могут быть очень длинными, но всегда конечные;

- перевод конечной десятичной дроби в позиционную систему с другим основанием может дать периодическую бесконечную дробь, запись которой с использованием скобок тем не менее конечна.

«Абсолютно точные» вычисления позволяют выявлять ошибки, совершаемые студентами при счёте, в том числе использование калькуляторов. Отказ же от стандартной реализации арифметики позволяет ставить студентам нестандартные задачи, решений (и онлайн-калькуляторов для которых) нет в Интернете.

Опыт 2022 года показал, что массовое формальное тестирование имеет ограниченную применимость; от студентов нужно требовать решения задач, сочетающих (и требующих понимать) разные темы.

## Использованный инструментарий

Использованы ALT Linux p9 Starterkit (Hypericum), GNU Octave и TeXLive из репозитория пакетов. Для Рефала использованы Refal-5 Version PZ (<http://www.botik.ru/pub/local/scp/refal5/>) и интерпретатор Рефал-М (<https://github.com/STrusov/refal-machine>).

На Рефал-М возможно писать программы, читаемые как связный текст с русскими словами и словосочетаниями. Пропадает необходимость помнить или реконструировать громоздкие отождествления русских понятий с англоязычными (часто кривыми). Мысль можно выразить в программе по-разному, но с богатством выразительных возможностей русского языка альтернативы могут стать более явными.

Александр Смирновский

Санкт-Петербург, Санкт-Петербургский политехнический университет Петра Великого

## Опыт использования СПО OpenFOAM для обучения основам вычислительной гидродинамики в СПбПУ

### Аннотация

В докладе представляются основные принципы построения программы обучения студентов бакалавриата Физико-механического института СПб политехнического университета по направлению «Прикладные математика и физика» основам вычислительной гидродинамики с использованием СПО, в частности, пакета OpenFOAM. Рассказывается о достоинствах и выявленных недостатках использования СПО в обучении применительно к описываемому курсу, разрабатываемому в течение более пяти лет.

В течение нескольких лет в Высшей школе прикладной математики и вычислительной физики Физико-механического института СПбПУ в рамках курса «Исследовательские проекты в области механики сплошных сред» для последнего года бакалавриата проводятся практические занятия по основам вычислительной гидродинамики с использованием свободного программного обеспечения OpenFOAM [1]. Пакет OpenFOAM широко используется в мире как для проведения научно-исследовательских работ с использованием методов вычислительной гидродинамики (Computational Fluid Dynamics, CFD), так и в промышленности при решении практических задач, связанных с движением жидкости и газа.

OpenFOAM — это открытая интегрируемая платформа для численного решения уравнений в частных производных методом конечного объёма. Изначально OpenFOAM создавался как инструмент для решения задач вычислительной гидродинамики, однако с течением времени благодаря гибкости используемой платформы и открытости исходного кода в нём появилась возможность решать множество других физических задач.

Основное преимущество OpenFOAM перед коммерческими программными продуктами, предназначенными для решения задач CFD, является его открытость и доступность: пакет OpenFOAM распространяется под лицензией GPLv3. Одним из недостатков пакета является отсутствие пользовательской графической оболочки, вся работа

с пакетом осуществляется при помощи редактирования управляющих файлов, а также при помощи сторонних программ (например, для визуализации результатов расчёта используется пакет ParaView [2], распространяющимся под лицензией BSD); следует отметить, что также имеются коммерческие графические оболочки для взаимодействия с OpenFOAM. В настоящее время имеется две независимые ветки кода OpenFOAM, которые развиваются разными организациями: ветка, условно называемая OpenFOAM.org, развивающаяся под эгидой некоммерческой организации OpenFOAM Foundation, и ветка, условно называемая OpenFOAM.com, которая развивается коммерческой компанией OpenCFD Ltd.

Для студентов бакалавриата, обучающихся вычислительной гидродинамике, курс с использованием пакета OpenFOAM является дополнительным; они также изучают другие программные пакеты, в т.ч. коммерческие пакет ANSYS Fluent [3]. Поэтому в курсе, посвящённому изучению OpenFOAM, даются только базовые аспекты использования пакета. В связи с тем, что, к сожалению, в Политехническом университете в основном используется ОС Windows, для удобства использования пакета OpenFOAM, который зачастую применяется в ОС на базе ядра Linux, для студентов подготовлена виртуальная машина VirtualBox с предустановленным дистрибутивом Lubuntu 19.04, OpenFOAM версии 7 и пакетом ParaView. Как показала практика, у студентов в основном не возникает проблем с использованием виртуальной машины, а единая заранее подготовленная среда рабочего окружения позволяет существенно сократить время выявления и решения возникающих в процессе работы с OpenFOAM проблем и затруднений.

В рамках курса по OpenFOAM студенты изучают основные возможности пакета и выполняют несколько компьютерных лабораторных работ, посвящённых простейшим задачам вычислительной гидродинамики, таким как: плоское течение Пуазейля, конвективно-диффузионный перенос скаляра, свободноконвективное течение в квадратной камере с разнонагретыми стенками и др. Поскольку в пакете OpenFOAM имеется возможность распараллеливания вычисления для расчёта трудоёмких задач, одна из лабораторных работ посвящена вопросу подготовки, запуска и тестирования работы OpenFOAM в параллельном режиме. Для этих целей на имеющемся в распоряжении Высшей школы кластере под управлением ОС CentOS



выделена виртуальная машина для студентов, позволяющая проводить расчёты в параллельном режиме.

Ознакомиться со списком лабораторных работ и структурой курса, проведённого в 2021-2022 уч.году, можно по ссылке [4].

## Литература

- [1] Weller H.G., Tabor G., Jasak H., Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques // Computer in Fluids, 1998, V.12, N.6, P.620-631.
- [2] Пакет визуализации ParaView, <https://www.paraview.org>
- [3] Коммерческий CFD пакет ANSYS Fluent, <https://www.ansys.com/products/fluids/ansys-fluent>
- [4] Учебные материалы курса по OpenFOAM, <https://drive.google.com/drive/folders/1hQd8GHN564AQhEfCx1BEpBBW0cmomS1>

Ю. В. Шевчук, А. В. Елистратов, А. Ю. Пономарев  
Переславль-Залесский, Институт программных систем имени  
А.К. Айламазяна РАН

## Малые компьютерные системы со свойствами масштабируемости и высокой доступности

### Аннотация

Рассматривается подход к организации отказоустойчивых информационно-вычислительных систем в виде распределённых программных систем, работающих на сети одноплатных компьютеров. Описывается разработанный авторами одноплатный компьютер, специализированный для использования в таких мультикомпьютерных системах. Приводятся примеры свободного программного обеспечения, применимого для таких систем.

## Введение

С внедрением информационных технологий во все сферы общественной жизни разработчикам информационных систем приходится решать задачи обеспечения высокой доступности (high availability)

и постепенной масштабируемости (incremental scalability). Недоступность критических информационных сервисов парализует работу любого цифровизованного предприятия. Невозможность наращивания характеристик системы (производительности, объёма памяти) без замены серверной платформы на более мощную влечёт неоптимальные финансовые вложения (установка серверов с избытком ресурсов – «на вырост»), и риск временной нестабильности сервиса при каждой замене серверной платформы.

Современным решением обеих проблем является переход предприятий с собственной серверной инфраструктуры на облачные сервисы. Ведущие облачные провайдеры (AWS, Azure, Yandex, ...) поддерживают доступность серверной инфраструктуры на уровне 99.95...99.99. Арендванный в облаке сервер легко масштабируется путём выбора более высокого тарифного плана. И есть ещё много преимуществ в использовании арендованных облачных серверов, но есть и аргументы против:

- если пользователи системы не рассредоточены по Интернет, а находятся на территории предприятия (пример: кассы супермаркета), доступность сервера нужно домножить на коэффициент доступности канала от предприятия до облачного провайдера, который может быть существенно ниже единицы;
- масштабируемость облачного сервера ограничена размером физического сервера, используемого облачным провайдером;
- тарифная сетка облачных провайдеров может не соответствовать потребностям реализуемой системы; особенно это касается систем, требующих больших объёмов дискового пространства при сравнительно небольших требованиях к ресурсам процессора и оперативной памяти;
- лимитированный объём трафика и/или пропускная способность;
- риск изменения (ухудшения) условий предоставления облачных сервисов после того как система создана, отлажена и от неё зависит работа предприятия;
- обратной стороной удобства сервисов PaaS являются трудности перехода к другому провайдеру (vendor lock-in);
- если есть требования конфиденциальности данных, ресурсы публичных облачных провайдеров использовать нельзя.

Альтернативный способ получения информационной системы с высокой доступностью и масштабируемостью—реализация системы на кластере компьютеров в локальной сети предприятия в виде распределённой программной системы. Коэффициент снижения производительности системы при отказе одного компьютера  $K_d = 1 - 1/N$ , где  $N$ —число компьютеров в кластере, т.е. влияние отказов на производительность системы тем меньше, чем больше компьютеров в кластере. Но стоимость кластера  $C = \sum_{i=1}^N C_i$ , и экономически неэффективно увеличивать  $N$  так, чтобы суммарный объём ресурсов кластера  $P = \sum_{i=1}^N P_i$  значительно превышал необходимый для работы системы объём ресурсов  $P_{\min}$ . Таким образом, для предприятий с небольшим  $P_{\min}$  единственный способ получения экономически эффективной системы с высокой доступностью это использование компьютеров с небольшой стоимостью  $C_i$  и объёмом ресурсов  $P_i$ .

В настоящее время существует и бурно развивается класс недорогих одноплатных компьютеров (SBC—Single Board Computer), наиболее известным представителем которого является компьютер Raspberry Pi (2012—н.вр.). Сейчас рынке SBC работает значительное число производителей, выпускающих SBC на базе микросхем SoC (System-on-Chip—система на кристалле) фирм Broadcom, Allwinner, Amlogic, NXP, Samsung, Mediatek, Rockchip с архитектурами armhf и arm64, работающие под управлением ОС Linux. Их производительность уже достаточна для использования в качестве персонального компьютера или небольшого сервера, и их можно было бы использовать для построения кластера для информационно-вычислительных систем с высокой доступностью. Но несмотря на многообразие представленных на рынке SBC, не удаётся найти моделей, сочетающих необходимые характеристики:

- достаточный объём оперативной памяти (не менее 4ГБ);
- интерфейс SATA для подключения внешних накопителей массового класса;
- два или более коммуникационных интерфейса Gigabit Ethernet для организации отказоустойчивой локальной сети;
- схема питания с резервированием для исключения одновременного выхода из строя нескольких узлов системы при отказе внешних компонентов системы электропитания, таких как UPS и конверторы AC-DC.



Рис. 1: Одноплатный компьютер с подключённым HDD 3.5 дюйма

В то же время, конструкция коммерческих SBC в значительной степени открыта: используется свободное ПО, доступны электрические принципиальные схемы, хотя и недоступны топологии печатных плат. Это даёт возможность при сравнительно небольших трудозатратах разработать специализированный SBC для распределённых информационных систем с высокой доступностью и масштабируемостью. Разработке такого SBC (семейства SBC) и посвящена настоящая работа.

## Состояние работ

Разработан специализированный SBC: принципиальные схемы, топологии печатных плат. Выпущено несколько первых образцов, начата сборка небольшого кластера.

Конструкция SBC показана на Рис. 1. Отметим основные особенности:

- конструкция одноплатного компьютера на самом деле двухплатная [1]: маленькая плата с процессором и памятью (SoM – System on Module) требует дорогой технологии, а большая несущая плата сравнительно дешёвая. Такая конструкция позволяет быстро и недорого создавать модификации SBC, если это требуется для конкретного применения. Конструкция безразъёмная, платы соединяются пайкой;
- на SoM находятся процессор (SoC Rockchip RK3328, 4 ядра ARM64 Cortex A53), оперативная память DDR4, flash-память eMMC и микросхема коммутатора Gigabit Ethernet с одним внутренним и тремя внешними портами;

- на несущей плате находятся все крупногабаритные детали, разъёмы Ethernet, SATA, MicroSD и подсистема резервированного питания, включающая два входа внешнего питания и встроенный аккумулятор Li-ion. Ресурс аккумулятора выбран так, чтобы в случае отказа внешнего электропитания узел смог сохранить данные из оперативной памяти на диск и корректно остановить систему до восстановления внешнего электропитания;
- питание от источника 10..15В; можно питать от свинцово-кислотного аккумулятора, работающего в буферном режиме (постоянно находящегося на зарядке).

Сеть Gigabit Ethernet, конечно, не идёт в сравнение в быстрыми сетями кластеров, такими как Infiniband. Зато Ethernet позволяет разнести узлы по разным помещениям с независимым электропитанием, уменьшить физические риски, такие как пожар или затопление. Gigabit Ethernet это массовая технология, позволяющая при небольших затратах организовать отказоустойчивую сеть. Приведём два примера сетей: минимальный на встроенных коммутаторах (Рис. 2 и максимальный на внешних 48-портовых коммутаторах (Рис. 3).

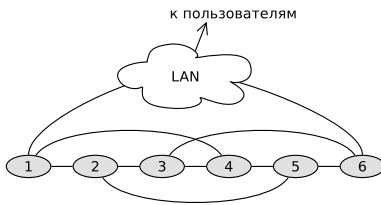


Рис. 2: Пример сети на встроенных коммутаторах.

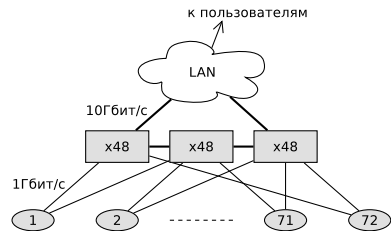


Рис. 3: Пример сети на внешних 48-портовых коммутаторах с диаметром  $D=2$ .

## Программное обеспечение

В настоящий момент на SBC работает Armbian Linux. Используются загрузочные образы для близких по архитектуре компьютеров Rock64 или Firefly ROC-RK3328-CC. В образе заменены ядро Linux, загрузчик uboot и описание аппаратуры компьютера (device tree). Armbian Linux использует репозиторий пакетов Debian GNU/Linux,

т.е. доступен для установки большой объём пре-компилированного свободного программного обеспечения. И ещё в репозитории Debian нет ряда программных продуктов со свободными лицензиями, которые мы планируем применять на сети малых компьютеров для построения систем с высокой доступностью и масштабируемостью. Кратко перечислим их.

### **Riak KV**

Riak KV [4] – распределённое масштабируемое хранилище данных типа «ключ – значение» с высокой доступностью. Одна из свободных реализаций принципов Amazon Dynamo [5]. Полностью децентрализовано: все узлы равноправны. Сохраняет доступность на запись при отказе до  $N - 1$  из  $N$  узлов. NoSQL, не даёт гарантий ACID, реализует «согласованность в конечном счёте» – eventual consistency.

### **Riak CS**

Реализация над Riak KV файлового хранилища, совместимого API Amazon S3. Открывает возможность использовать готовые программные средства, работающие с этим популярным API, например fuse-монтируемые файловые системы 3sfs [7] или s3ql [8].

### **Riak Core**

Riak Core [9][10] – библиотека для распределённого программирования на языке Erlang, выделенная из Riak KV. Обеспечивает мониторинг состояния вычислительного кластера и автоматическое перераспределение нагрузки между узлами при выходе узлов из строя или добавлении в кластер новых узлов, используя метод консистентного хэширования. Riak Core это один из возможных путей к созданию собственных программных систем с высокой доступностью и масштабируемостью.

### **Apache Kafka**

Apache Kafka [11] это распределённый брокер (архитектура издатель-подписчик) и архив сообщений. Позволяет создавать распределённые программные системы с потоковой обработкой данных, соче-

тая различные языки программирования, для которых реализованы Kafka API, в частности Си, Golang, Python, Perl, Erlang.

## Заключение

Приведённый список свободных программных систем для распределённых архитектур, конечно, далеко не полон и отражает в первую очередь предпочтения авторов. Он начинается с систем Riak KV и Riak CS, позволяющих организовать отказоустойчивые масштабируемые хранилища данных, которые используются традиционными, не распределёнными программными системами (например, сервер электронной почты dovecot). В этом случае высокая доступность сервера может достигаться известными методами Linux High Availability [12], есть масштабируемость хранилища, но нет распределения нагрузки на сервер по кластеру. Чтобы достичь распределения нагрузки, новые программные системы нужно сразу проектировать как распределённые системы, в расчёте на мультикомпьютерную архитектуру. Riak Core и Apache Kafka могут быть полезны для реализации таких распределённых программных систем.

Мультикомпьютерные системы на сравнительно недорогих одноплатных компьютерах позволят даже организациям с небольшим бюджетом пользоваться преимуществами распределённых систем – отказоустойчивостью и горизонтальной масштабируемостью.

## Литература

- [1] Елистратов А. В., Коваленко М. Р., Пономарёв А. Ю., Шевчук Ю. В. *Компактный вычислительный модуль для распределённых систем управления и хранения данных*. [https://2021.nscf.ru/Presentations/01\\_Apparatura/03\\_ShevchukYV.pdf](https://2021.nscf.ru/Presentations/01_Apparatura/03_ShevchukYV.pdf)
- [2] PINE64: A Small Form-Factor Single Board Computer Capable of 4K playback <https://www.pine64.org/devices/single-board-computers/rock64/>
- [3] ROC-RK3328-CC Quad-Core 64-Bit Open Source Main Board <https://en.t-firefly.com/product/rocrk3328cc.html>
- [4] Riak KV, a distributed key-value store. <https://github.com/basho/riak>
- [5] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, W. Vogels *Dynamo*:

- Amazon's Highly Available Key-Value Store.* SOSP-2007. <http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf>
- [6] Riak CS: Cloud Storage with S3 API. [https://github.com/TT-Tokyo/riak\\_cs](https://github.com/TT-Tokyo/riak_cs)
- [7] FUSE-based file system backed by Amazon S3. <https://github.com/s3fs-fuse/s3fs-fuse>
- [8] s3ql: Full-featured file system for online data storage <https://github.com/s3ql/s3ql/>
- [9] Riak Core: distributed systems infrastructure used by Riak. [https://github.com/basho/riak\\_core](https://github.com/basho/riak_core)
- [10] Riak Core tutorial. [https://github.com/lambdaclass/riak\\_core\\_tutorial](https://github.com/lambdaclass/riak_core_tutorial)
- [11] Apache Kafka: an open-source distributed event streaming platform. <https://kafka.apache.org/>
- [12] Linux-HA: Open Source High-Availability Software for Linux and other Platforms <http://www.linux-ha.org>

Уймин Антон Григорьевич

Москва, РГУ нефти и газа (НИУ) имени И.М. Губкина

Проект: Лаборатория сетей и систем передачи информации

<https://www.gubkin.ru/faculty/>

[faculty-of-complex-safety-of-the-fuel-and-energy-complex/](https://www.gubkin.ru/faculty-of-complex-safety-of-the-fuel-and-energy-complex/kafedry-i-podrazdeleniya/kbit/)

[kafedry-i-podrazdeleniya/kbit/](https://www.gubkin.ru/faculty-of-complex-safety-of-the-fuel-and-energy-complex/kafedry-i-podrazdeleniya/kbit/)

[laboratoriya-setey-i-sistem-peredachi-informatsii.php](https://www.gubkin.ru/faculty-of-complex-safety-of-the-fuel-and-energy-complex/kafedry-i-podrazdeleniya/kbit/laboratoriya-setey-i-sistem-peredachi-informatsii.php)

## **Опыт развёртывания специализированных лабораторий СПО и ВО на базе отечественного ПО. Проблемы масштабирования решения.**

### **Аннотация**

На основе опыта работы с свободным ПО формируются 5 тезисов, не позволяющих комплексно внедрять свободное ПО в ОУ. Приводится зарубежный опыт решения данных проблем.



Опыт работы с отечественным программным обеспечением ведётся командой #au\_team с 2006 года. Опыт получен на базе нескольких площадок. Площадки указаны в порядке их оснащения отечественным программным обеспечением: УРТК им. А.С. Попова (2006-2020) — Лаборатории: Программно аппаратных средств защиты информации, Технических средств информатизации, Периферийных устройств СВТ, Компьютерных сетей; МАОУ СОШ № 48 (2010-2016) — Класс Информатики; ГБПОУ МИК (2020–2021) — Лаборатории: Программно аппаратных средств защиты информации, Технических средств информатизации, Компьютерных сетей, в школьном подразделении класс Робототехники; РГУ нефти и газа (НИУ) имени И.М. Губкина (2021-настоящее время) — Лаборатория сетей и систем передачи информации. Большинство проектов описаны или презентованы на различных мероприятиях [1],[2]

Целью данного доклада является демонстрация результатов работы и обобщение опыта для дальнейшего его масштабирования. Отдельно рассмотрены причины влияющие на отсутствие массовой поддержки в решении задач импортозамещения.

На базе отечественного программного обеспечения в настоящее время развёрнута и функционирует лаборатория ВУЗа. Решены задачи: работы относительно маломощных рабочих станциях, работы с системами виртуализации (KVM, Virtual BOX), работы со решениями зарубежных вендоров на примере Cisco Packet Tracer [3], GNS 3 и т. д. Лаборатория имеет единый контур управления, на базе проекта Puppet.

Основываясь на опыте предыдущих проектов, было принято решение запустить в 2022 учебном году тестовую лабораторию, на базе которой продемонстрировать практику применения отечественного программного обеспечения. Традиционно, за рамками лаборатории, проект не встретил массовой поддержки:

- 0) Отсутствие идеологии отечественного продукта.
- 1) Отсутствие внутренней мотивации — проблемой является сложность убедить людей выйти из зоны «работает не трожь», к изучению альтернативных инструментов. Отдельно стоит вопрос эргономики свободного ПО, который требует переучиться, и донастроить всё под свои задачи.
- 2) Внешняя перемотивация — проблемой является давление по переходу, которое встречает отторжение.

- 3) Кто бреет бороду? — задача стоит в подготовке кадров, которые будут готовить кадры. При должной подготовке специалиста, его навыки будут более оценены в интеграторе, чем в образовательной организации, а при тяге к преподаванию, в учебных центрах, которые менее нагружены бюрократическими издержками, и не требуют работать с немотивированными слушателями.
- 4) Социальный запрос — очень небольшое количество предприятий готово сделать заказ на специалистов, которых они получают через 3–4 года. Малым предприятиям проще перекупать тех, специалистов, кто уже вырос на реальных производственных задачах, компаниям уровня Ростелеком, Ростех, Росатом, проще создавать корпоративные университеты, проводить мероприятия, как Студенческий ЦЕХ Росатом, тем самым отбирать студентов и доучивать их самостоятельно на своих проектах.

Проблема перехода на отечественное ПО является не технической, а идеологической проблемой. Наша задача убедить применять инструменты, которые разработаны в РФ и для РФ. Стоит обратиться к опыту Cisco который с 1998 года воспитывал специалистов, которые обучали специалистов. Качественный пивок произошёл только в 2012 году. Была решена задача престижа статуса тренеров, задача методических и вспомогательных материалов, интегрированных в примерные образовательные программы, задача оснащения, выдачей бандлов рабочих мест, послуживших основой большинства современных лабораторий сетей в СПО и ВО, а главное работа центров поддержки, которые организовывали энтузиастов.

## Литература

- [1] Уймин А. Г., *СПО в образовательной среде. Как мы переезжали с AltLinux на CentOS*, 2016, <https://www.youtube.com/watch?v=3as1wUcQ1R4>
- [2] Уймин А. Г., *Применение отечественного программного обеспечения для перестройки образовательного процесса вуза в рамках подготовки кадров цифровизации производства*, Цифровая трансформация промышленности: новые горизонты : Сборник научных трудов по материалам 3-й Всероссийской научно-практической конференции, Москва, 10 ноября 2022 года. — Москва: Общество с ограниченной ответственностью «Русайнс», 2022. — С. 398–405.,

- [3] Уймин А. Г., Токарев Г. И., *Инструментальные средства обучения компьютерным сетям. Развёртывание на базе российского программного обеспечения* // Системы управления и информационные технологии. – 2022. – № 4(90). – С. 88-92. – DOI 10.36622/VSTU.2022.90.4.019.

Андрей Савченко

Москва, Базальт СПО

<https://www.altlinux.org/Join>,

<https://www.gentoo.org/get-involved/become-developer/>

## Подготовка дистрибутивостроителей

### Аннотация

Рассматривается опыт и практики подготовки разработчиков дистрибутивов и процесс их вступления в сообщество на примере Alt Join и Gentoo Recruitment. Процесс включает в себя обучение как особенностям дистрибутивов, так и правилам взаимодействия с сообществом. По итогам подготовки проводится достаточно строгий контроль полученных навыков.

### Мотивация

Зачем становятся разработчиками дистрибутивов? Обычно это опытные пользователи, которым чего-то не хватает: пакетов, их оперативного обновления или исправления проблем.

### Технические навыки

Порогом начала обучения являются:

- Работа с VCS (обычно git)
- Умение компилировать программы (желательно знать основные сборочные системы)
- Базовое умение собирать пакеты
- Навыки работы с кодом программ для выбранных языков

## Командная работа

Дистрибутив — это командная работа, важно уметь взаимодействовать с другими людьми, находить общий язык, это не всегда просто. Следует уделять особое внимание изучению порядка взаимодействия в каждом сообществе, культуре разработки свободного кода.

## Полнота охвата

Самой большой технической сложностью является освоение всего многообразия возможных способов работы с пакетами. В Альте это большое количество методов ведения репозитория пакета, в Gentoo это необходимость поддерживать большое число USE флагов и (почти) всех их возможных комбинаций.

Здесь всё постигается опытом и практикой. По сути дела, главная задача ментора — провести подопечного по основным направлениям. Но классического обучения в виде лекций или семинаров здесь нет: кандидат учится сам, ментор отвечает на вопросы и подсказывает что ещё изучить, какие лучшие практики есть для тех или иных аспектов. Это чем-то напоминает дипломную работу.

## Проверка знаний

Наставники тоже люди и могут недоглядеть, поэтому когда ментор решает, что кандидат готов, происходит независимая проверка другим разработчиком, обычно из специальной группы заинтересованных лиц. В ходе такой проверки в том или ином виде проводится интервью с кандидатом, обсуждается проделанная им работа, рассматриваются её недоработки; при необходимости кандидат может получить дополнительные задания.

По решению рецензента о готовности кандидат становится полноценным разработчиком, первое время за его действиями наблюдают более опытные товарищи для коррекции возможных проблем.

## Сроки

И ментор, и кандидат являются волонтерами, поэтому срок подготовки может занимать сильно разное время, зависящее от мотивации

и занятости. Обычно это полгода–год, но дисперсия высока: может быть несколько недель, а может и несколько лет.

Елена Вепрева, Анна Махонина, Алексей Ерпелев, Владимир  
Симонов

Москва, ФГБОУ ВО «Российский государственный социальный  
университет»

<https://rgsu.net/>

## **Разработка учебных проектов моделей подвижных автоматических систем наземного и воздушного размещения с использованием СПО**

### **Аннотация**

Разработаны учебные проекты моделей автоматических систем, осуществляющих движение по поверхности и в воздухе. Синтезированные законы управления позволяют системам избегать препятствий, поддерживать направление движения в сторону объекта, обеспечивать требуемые режимы движения с заданным качеством. В разработке использовалось свободное программное обеспечение

Разработка наземных и воздушных беспилотных транспортных средств предусматривает обязательное создание действующих макетов и прототипов для оценки конструкции и её отработки в лабораторных условиях, на испытательных стендах, полигонах и т.д. При этом модели создаются в САПР, в натурном виде и т. д. Для студентов, обучающихся по направлениям подготовки «Информатика и вычислительная техника», «Управление в технических системах», в особенности для авиационных специальностей участие в таких работах особенно актуально.

Проектирование моделей подвижных автоматических систем наземного и воздушного размещения осуществляется с использованием СПО. Эффективно использование популярных и недорогих аппаратно-программных средств на базе Arduino, Raspberry Pi, а также соответствующих компонент.

Достаточно важным здесь при обучении студентов является разработка закона управления созданным макетом, в особенности управление движением макета. Простейший закон управления — релейный —

не всегда приводит к положительному результату, если объект обладает динамическими свойствами. Этапы синтеза системы управления являются классическими [1]:

- определение целей управления;
- выбор переменных, подлежащих управлению;
- формулировка требований к этим переменным;
- выбор конфигурации системы и исполнительного устройства;
- получение моделей объекта управления, датчиков и исполнительных устройств;
- выбор регулятора и определение ключевых параметров, подлежащих настройке;
- оптимизация параметров и анализ качества системы (при необходимости возврат к выбору конфигурации системы и исполнительного устройства);

Если качество системы приемлемое, процедура синтеза завершается.

В РГСУ в рамках изучения дисциплин, связанных с проектированием информационных систем, создаются и исследуются соответствующие модели подобных систем [2].

Так, разработан действующий макет средства автономной перевозки грузов. Цель разработки состояла не только в проектировании собственно устройства, но и в пробуждении интереса к электронике и робототехнике у обучающихся, посетителей дней открытых дверей вуза, и, кроме того, не создать отторжения из-за сложности проекта. Для достижения поставленной цели практически все соединения, сенсоры, моторы и прочие элементы должны быть по возможности видны, а также и «мозг» — плата обработки информации. Поэтому большая часть корпуса выполнялась вручную, придан яркий и необычный дизайн. Здесь используются датчик линий, не дающий выехать из заданной зоны и предотвращающий падения со стола; датчик дистанции, предотвращающий столкновение с препятствиями; оптико-электронный модуль (датчики освещённости), отвечающий за наведение на источник освещения. Устройство имеет три состояния:

- покой: при отсутствии освещения или при избыточном освещении;
- движение вперёд: скорость движения зависит от датчиков освещённости; повороты осуществляются при определённой разнице

в показаниях левого и правого датчиков освещённости (функция наведения на источник освещения);

- движение назад — при обнаружении препятствия или близости к краю допустимой зоны (например, край стола).

Важную роль играла наглядность макета, поэтому каждое из состояний сопровождалось символами, выводимыми на LED-матрицу, размещённую на видном месте в верхней части макета.

Разработана также модель тренажёра взлёта посадки вертолётного типа. Цель разработки: обучение студентов разработке модели, обладающей динамическими свойствами, а также закона управления. Цель управления здесь состояла в получении приемлемых переходных характеристик для трёх режимов: (а) — взлёта и поддержания заданной высоты; (б) — стабилизации высоты при нагружении и (в) — при освобождении от груза. Переменные, подлежащие управлению: скорость вращения воздушных винтов, которая зависит не только от показаний высотомера (ультразвукового дальномера), но и соответствует предельным значениям напряжения питания. При этом конструкция проектировалась таким образом, чтобы развивается требуемая подъёмная сила. Использовался закон регулирования типа ПИ (пропорционально — интегральный). Проведена оптимизация параметров системы, оценено качество.

Таким образом, представлены результаты разработки учебных проектов моделей подвижных автоматических систем наземного и воздушного размещения с использованием СПО (Wiring, платформа Arduino). Определены области использования разработок; поставлены цели управления; выделены переменные, подлежащие управлению с определением допустимых значений; выбрана конфигурация систем (сенсоры, исполнительные устройства, вычислители, согласующие звенья) и законы управления; оптимизированы параметры системы, оценено качество функционирования. В итоге, разработано несколько макетов, среди которых автоматическое устройство перевозки грузов, движущееся по поверхности, реагирующее на препятствия, следующее за источником света и имеющее несколько режимов функционирования и индикации режимов; тренажёр взлёта-посадки вертолётного типа, осуществляющий стабилизацию высоты полёта, принимающий нагрузку на борт и освобождающийся от полезной нагрузки. Получены и оценены реальные характеристики функциони-

рования моделей. Полученные результаты удовлетворяют поставленным требованиям.



Рис. 1: Программный код разработки представлен здесь.

<https://github.com/Flattershaine>

## Литература

- [1] Дорф Р. *Современные системы управления* / Р. Дорф, Р. Бишоп. Пер. с англ. Б. И. Копылова. — М.: Лаборатория Базовых Знаний, 2002 — 832 с.: илл.
- [2] *Практические аспекты организации занятий по основам программирования в области электроники, автоматики и робототехники при подготовке бакалавров и специалистов инженерных специальностей* / Миронов П. Н., Герус М. И., Аметова М. М. и др. — В сборнике: Инфорино-2018. Материалы IV Международной научно-практической конференции. 2018. С. 155–159.



Анастасия Медведева, Алексей Ерпелев, Максим Каторгин,  
Владимир Симонов

Москва, ФГБОУ ВО «Российский государственный социальный  
университет»

<https://rgsu.net/>

## **Использование СПО в разработке учебных моделей систем определения местоположения объектов различных типов**

### Аннотация

Представлены системы определения местоположения объектов с различными отражающими способностями и свойствами. Системы являются учебными проектами и служат для обучения робототехнике, электронике, программированию с использованием свободного программного обеспечения

Изучение дисциплин профессионального цикла направлений подготовки «Управление в технических системах», «Информатика и вычислительная техника» и ряда других в обязательном порядке сопровождается получением студентами практических навыков разработки реальных аппаратно-программных средств.

Текущая обстановка в стране требует от студентов подготовленности к реальной проектной деятельности в перспективных направлениях по документу «Стратегия научно-технологического развития Российской Федерации» [1], п. 20 разд. а) — ж). Одним из примеров проектных разработок студенческого учебного уровня являются системы пеленгации (определения местоположения) разнообразных объектов. Для разработки таких систем применяются разнообразные датчики информации — видеокамеры, дальномеры различных типов, оптико-электронные блоки и т.д. Данные проекты в учебном отношении эффективны тем, что позволяют студентам: подключать несколько разнотипных датчиков информации; обрабатывать полученную информацию по самостоятельно созданным алгоритмам (включая нормирование данных, фильтрацию, отбрасывание ложных показаний, расчёты координат и т.д.); выводить информацию как на средства отображения информации, так и на другие информационные системы.

Представленные проекты используются для более крупных учебных проектов, например [2] и [3].

В данном материале представлены результаты проектирования нескольких учебных моделей систем определения местоположения объектов различных типов. Первая модель — определение углового местоположения объекта для осуществления азимутального наведения на объект [4]. Здесь решалась задача разработки системы управления макетом летательного аппарата с функциями: разворот по азимуту для автоматического наведения на объект — источник света; измерение дистанции до объекта. Здесь использовались сенсоры: фотоэлектрический сенсорный блок; ультразвуковые дальномеры; инфракрасные дальномеры. В качестве исполнительного устройства применялся серводвигатель. В ходе проектирования на начальном этапе студентами использовались САПР. Далее, на реальной конструкции решались практические проблемы: неустойчивость работы измерителей дистанции (шумы, выбросы), и ряд других. Для решения проблем применялись: экранирование проводников измерительного тракта; в программной части использована двухэтапная цифровая фильтрация [5]. В итоге существенно снижена ошибка измерений и уменьшены выбросы. Макет доказал свою работоспособность и демонстрировался на днях открытых дверей нескольких вузов.

Ещё один действующий макет представляет собой интерактивную систему слежения за подвижными объектами с различной отражающей способностью — посетителями, для применения на выставках, днях открытых дверей и пр.

Система представляет собой подвижное (с помощью первого сервомотора — по азимуту) коромысло с размещёнными на его противоположных концах дальномерами. Указанные дальномеры осуществляют пеленгацию объекта (отслеживают его перемещение) и поворачивают систему вслед за объектом. На коромысле размещён второй сервомотор, осуществляющий перемещение по углу тангажа рекламного луча-указателя, который включается и выключается по случайному закону.

Дополнительная проблема здесь имела место, кроме задач фильтрации сигналов сенсоров, подобрать такой закон управления, чтобы движение системы наведения было плавным, без рывков. Данный макет также доказал свою работоспособность и был неоднократно применён на мероприятиях.

Использованное в настоящих разработках программное обеспечение относится к категории «свободное программное обеспечение», а именно язык Wiring. Применяемая платформа — Arduino.

В настоящее время создаётся прототип аналогичной установки на платформе Raspberry Pi с элементами машинного зрения.

Таким образом, представлены учебные модели систем определения местоположения объектов различных типов — активно излучающих источников света; пассивных объектов с различной отражающей способностью как для ультразвука, так и для оптического диапазона. Следует отметить, что студенты в ходе выполнения проектов в обязательном порядке оформляют отчётную и проектную документацию в соответствии с ГОСТ. Указанная проектная деятельность соответствует следующим компетенциям (ФГОС «Управление в технических системах»): «Готовность к участию в работах по изготовлению, отладке и сдаче в эксплуатацию средств автоматизации и управления»; «Готовность разрабатывать проектную документацию в соответствии с имеющимися стандартами»; «Готовность организовать метрологическое обеспечение производства средств автоматизации и управления».



Рис. 1: Программный код разработки представлен здесь.

[https://github.com/Vladimir28091959/MESTOPOLOJ\\_MEDVEDEVA\\_SIMONOV](https://github.com/Vladimir28091959/MESTOPOLOJ_MEDVEDEVA_SIMONOV)

## Литература

- [1] *Стратегия научно-технологического развития Российской Федерации / Указ Президента Российской Федерации / Режим доступа: электронный. URL: <https://docs.cntd.ru/document/420384257#7E20KD> (дата обращения 10.01.2023 г.).*
- [2] Волохов Т. Д., Краулин Д. К., Кодирбердиев А. М. и др. *Управление азимутальным наведением макета летательного аппарата.* — Современные информационные технологии в образовании, науке и промышленности

- ности: XXIII Международная конференция, XXI Международный конкурс научных и научно-методических работ: Сборник трудов / Отв. редактор и составитель Т.В. Пирязева. — М.: Изд-во «Экон-Информ», 2022. — 224 с. — Стр. 28-30. ISBN 978-5-907681-03-3.
- [3] Вепрева Е. Л., Махонина А. Н., Симонов В. Л. *Разработка систем управления моделями наземных и воздушных беспилотных транспортных средств*. — Современные информационные технологии в образовании, науке и промышленности: XXIII Международная конференция, XXI Международный конкурс научных и научно-методических работ: Сборник трудов / Отв. редактор и составитель Т.В. Пирязева. — М.: Изд-во «Экон-Информ», 2022. — 224 с. — Стр. 140-143. ISBN 978-5-907681-03-3.
- [4] Симонов В. Л., Мякотина С. К., Гурский Г. И., Малькова А. О. *Разработка системы управления макетом летательного аппарата с применением сенсоров*. — 21-я Международная конференция «Авиация и космонавтика». 21-25 ноября 2022 года. Москва. Тезисы. — М.: Издательство «Перо», 2022— 8,06 Мб [Электронное издание]. — Стр. 262-263. ISBN 978-5-00204-819-9.
- [5] *Фильтрация сигналов*. AlexGyver Technologies. [Электронный ресурс] // URL: <https://alexgyver.ru/lessons/filters/> (дата обращения: 10.01.2023 г.).

Александр Крагин, Илья Обрубов, Игорь Воронин  
Москва, Шатура, НИУ ВШЭ, ИПЛИТ РАН

## Классификация опухолевых клеток с использованием моделей машинного обучения в среде Альт Линукс

**Описание предметной области** Спектроскопия комбинационного рассеяния (также известная как рамановская) — метод, который успешно используется в химии для получения структурных «отпечатков пальцев». Колебательная спектроскопия дает ключевую информацию о структуре молекул. В результате полученных спектрограмм, можно анализировать и делать выводы о состоянии биологической ткани - методами машинного обучения.

**Актуальность** Для успешной борьбы с раковой опухолью очень важно обнаружить все раковые клетки на ранних этапах, даже те, которые находятся вне очевидных границ опухоли и являются труднодиагностируемыми. Цель диагностического поиска — своевременно обнаружить и выбрать

тот способ лечения, который бы позволил удалить 100% опасных клеток и минимизировать риск рецидива заболевания. В связи с этим большую важность имеют разнообразные методики классификации тканей. В частности, можно проводить биопсию исследуемой области, получать небольшой образец ткани, а потом анализировать его спектрограмму.

**Исходные данные** В данной работе была рассмотрена задача бинарной классификации с классами “больные” и “здоровые” образцы биологических тканей. Для обучения и тестирования виртуальной модели были использованы 888 спектрограммы, из которых 456 принадлежат классу “больные” и 432 принадлежат классу “здоровые”. Работа выполнялась на сервере `astera.laser.ru` (ИПЛИТ РАН, Шатура) в операционной среде ALT Server 10.1 (FalcoRusticolus), Python 3.9.6, Docker version 20.10.11, ядро Linux `astera 5.10.131-std-def-alt1`

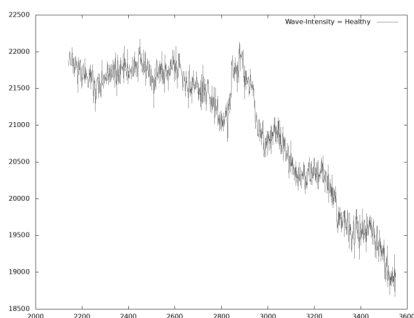


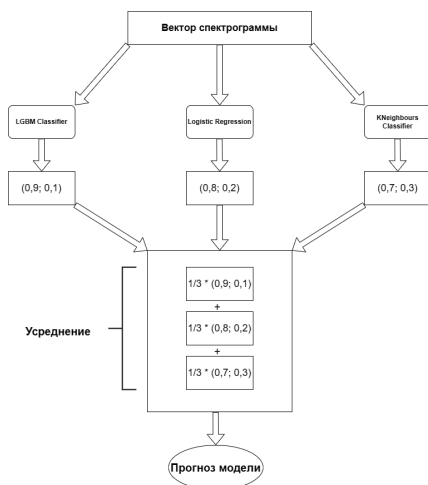
Рис. 1: Визуализация типовой спектрограммы

### Архитектура модели и результаты

В качестве базовых моделей использовались: градиентный бустинг, логистическая регрессия и KNeighbors классификаторы, реализованные в библиотеках LightGBM и SKLearn. После того как эти 3 базовых классификатора делают свои прогнозы, прогнозы усредняются. Благодаря тому, что выбраны классификаторы с разными принципами работы, корреляция между их прогнозами не слишком высока, что позволяет получать ощутимое улучшение точности после усреднения.

**Обучение:** `train` датасет используется для обучения 3 базовых классификаторов независимо друг от друга.

**Валидация:** Используется кросс-валидация с `n_folds = 5`, на каждом шаге модель обучается на  $\frac{4}{5}$  датасета и проверяется на оставшейся  $\frac{1}{5}$  части.



Итоговая точность вычисляется как среднее арифметическое от точности каждого шага.

**Результаты:** На кросс-валидации удалось получить точность 97.6%. Использование популярных фреймворков (LightGBM, SKLearn и NumPy) позволяет создавать приложение для классификации для любой операционной системы.

## Литература

- [1] Marco Riva et al, Glioma biopsies classification using raman spectroscopy and machine learning models o fresh tissue samples, <https://www.mdpi.com/2072-6694/13/5/1073/htm>
- [2] Jason Brownlee, Stacking ensemble machine learning with Python, <https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/>
- [3] Пакет для научных вычислений в Python <https://numpy.org/>
- [4] Линейная регрессия методом наименьших квадратов <https://scikit-learn.org/>.
- [5] Библиотека для ранжирования, классификации, регрессии и других задач машинного обучения для Python <https://github.com/catboost/catboost>

[6] Сервер для дистанционной обработки спектрограмм <http://astera.laser.ru:8888/tree>.

Чашкин Л. Б., Прокофьева Е. Н., Прилепко П. М.,  
Кузенков Ф. Д.

Москва, Национальный исследовательский университет «Высшая школа экономики», департамент компьютерной инженерии, Московский институт электроники и математики имени А. Н. Тихонова

## Использование инструментов и модулей open-source на базе ОС Альт в организации гео-мониторинга

### Аннотация

В настоящее время интеграция open-source и российских программных, технологических разработок становится важным вектором в развитии отечественного IT-сектора. В свою очередь продукты Базальт СПО становятся надёжными, гибкими и высоко-адаптивными платформами для запуска разно-функционального ПО, модулей и инструментов, в частности, свободного доступа. Специалисты Базальт СПО, Alt Linux Team в сотрудничестве с ведущими образовательными, консалтинговыми и научными организациями и центрами оперативно осуществляют поддержку, функционирование, внедрение самых разных интегрируемых и востребованных модулей для решения профессиональных инженерных задач, в том числе геотехнологического характера. Статья посвящена исследованию методов и средств контроля пространственных рисков. В наше время крайнюю актуальность имеет аналитика различных процессов, и практически все профессиональные области накопили и задокументировали достаточно данных для их анализа и прогнозирования. Особенно важным и актуальным развитие в данной области становится для управления данными на предприятиях ресурсно-энергетического сектора и горной промышленности. Здесь одним из самых эффективных инструментов для анализа геоданных являются цифровые модели и карты по горным пространственным объектам. Исследование может послужить примером использования банков данных для анализа и прогнозирования, геоэкспертизы. Модель, созданная в процессе выполнения работы, была выполнена в QGIS с использованием свободных банков данных открытого научного сообщества GISLAB.

## Введение

В условиях реализации стратегии импортозамещения всё более важным становится переход на отечественные ресурсы.

Большое значение представляет ресурсный сектор, поскольку предприятия недропользования и энергетики составляют значимую долю производственной мощности и формирования бюджета страны.

Число компаний и отраслей, в которых в последние годы идёт активная цифровая трансформация, стремительно растёт. Цифровизация ресурсного сектора позволит улучшить операционную эффективность, снизить затраты на поддержку, повысить безопасность, сократить риски остановки производства, поэтому важным направлением здесь является развитие геоинформационных технологий по сбору, хранению, обработке и интерпретации пространственных данных для решения задач инфраструктурного проектирования, рационального использования природных ресурсов и мониторинга экологических ситуаций [1-3].

Геоинформационные технологии находят применение в широком круге научных областей, начиная от традиционных геодезии и картографии, заканчивая современными исследованиями в области глобального потепления и возобновляемой энергетики [4-6]. В частности, одной из актуальных задач здесь является построение цифровых моделей территорий карьеров для проведения экологического мониторинга и своевременной оценки георисков.

## Описание алгоритма

При работе с геоданными первостепенной задачей является выбор инструмента, обладающего достаточным функционалом [7]. Кроме того, важные критерии выбора — это доступность исходного кода, а также наличие приложения, совместимого с операционной системой Альт.

В данной статье будет рассмотрена работа с геоинформационной системой QGIS. QGIS — это продвинутый open-source инструмент подготовки, обработки, визуализации и анализа данных в многочисленных форматах данных, существующих в области геопространственного анализа. QGIS по умолчанию предустановлен в операционной системе Альт Образование для высшей школы.

Перед построением любой цифровой модели необходимо провести интеграцию с одними из доступных карт и осуществить картографическую привязку к пространственным координатам. Для этого в программе QGIS может быть использован модуль QuickMapServices, позволяющий загрузить и работать с разными видами карт.



Далее необходимо получить данные о рельефе из базы данных. Одной из таких баз является проект Shuttle Radar Topology Mission. Shuttle Radar Topology Mission (SRTM) — международный исследовательский проект по созданию цифровой модели высот Земли. SRTM предоставляет данные о высоте рельефа с точностью до 100 метров за 2000 год. В программе QGIS предусмотрен модуль SRTM-Downloader, который может быть использован для получения данных о высоте из проекта SRTM для определённой области.

После получения данных необходимо осуществить их обработку. В программе QGIS доступно наложение этих данных на карту в виде теневого рельефа, что позволяет создать эффект объёмной карты. Далее необходимо наложить на карту цветовой градиент по высоте рельефа. После чего при необходимости существует возможность добавить изолинии на готовую карту рельефа.

Для построения цифровой модели можно воспользоваться модулем `qgis2threejs`, который позволяет перевести полученную карту рельефа в 3D-модель с различными настройками сжатия/растяжения по высоте и ширине.

Разработанный алгоритм был использован для построения карты рельефа территории Вольского карьера в Саратовской области с использованием операционной системы Альт и предустановленной программы QGIS. Результаты представлены на рис. 1 и 2.

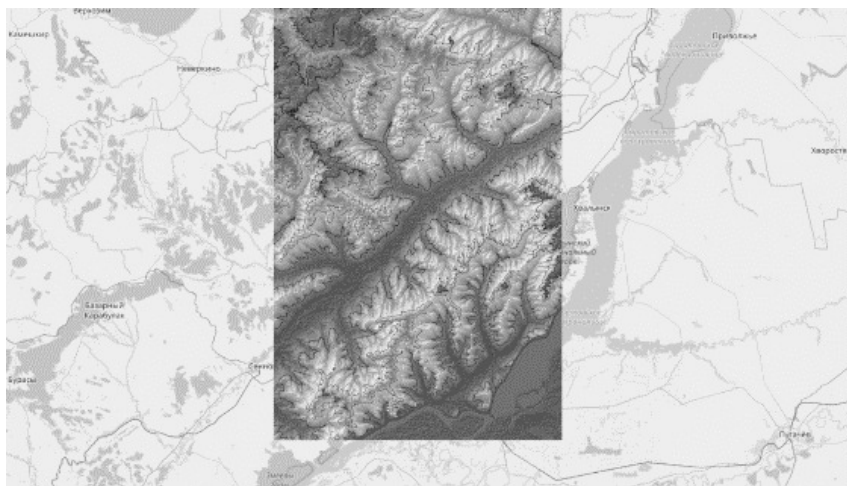


Рис. 1: Карта рельефа Вольского карьера.

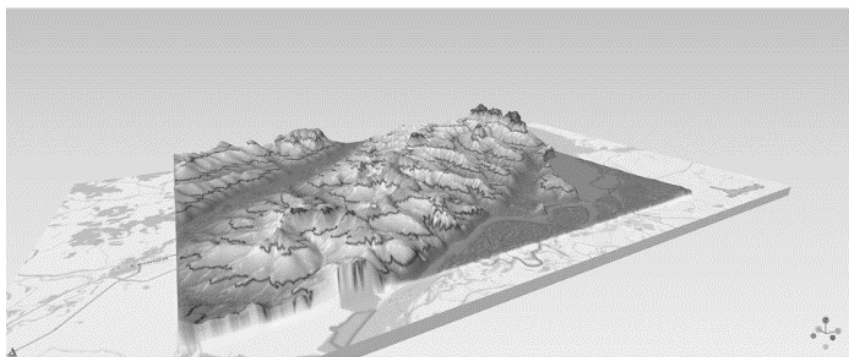


Рис. 2: Цифровая модель Вольского карьера.

## Заключение

Таким образом, операционная система Альт предоставляет возможности для работы с пространственными данными и построения цифровых моделей в виде предустановленной свободно-распространяемой геоинформационной системе QGIS. На основе неё разработан алгоритм построения цифровых моделей карьеров. По алгоритму построена цифровая модель Вольского карьера, точность которой не превышает 100 метров.

Процесс построения цифровой 3D-модели рельефа карьера состоит из 6 шагов:

1. привязка к пространственным координатам, выбор источника данных карт;
2. получение данных о высоте рельефа данной местности;
3. создание теневого рельефа;
4. наложение цветового градиента;
5. добавление изолиний на карту рельефа;
6. построение 3D-модели на основе карты рельефа.

В дальнейших исследованиях для упрощения построения цифровых моделей планируется частичная автоматизация приведённого алгоритма построения цифровых моделей на базе QGIS в виде разработки собственных open-source модулей, совместимых с операционной системой Альт.

## Литература

- [1] Vostrikov A. V., Prokofeva E. N., Goncharenko S. N., Griбанov, I. V. Analytical modeling for the modern mining industry // Eurasian Mining. 2019. No.2(32). P.30–35. DOI 10.17580/em.2019.02.07
- [2] Goncharenko S. N., Duong L. B., Petrov M. V., Stoyanova, I. A. Modeling of parameters of innovation water-protection measures on the basis of industrial-technological indices of coal mining at Vietnam enterprises (2014) Gornyi Zhurnal, (9) , pp.143–146.
- [3] Prokofeva E. N., Vostrikov A. V., Shapovalenko G. N., Alvarez A. The development of effective geomonitoring for mining area with industrial review // Eurasian Mining. 2017. No. 2. P. 61–63.
- [4] Temkin I., Deryabin S., Konov I.: Soft computing models in an intelligent open-pit mines transport control system. Procedia Computer Science, Vol. 120. 2017.
- [5] Temkin, I. O., Klebanov D. A., Deryabin S. A., Konov, I. S.: Method of determining the state of the haul road career in the management of the interaction between robotic elements of the mining transportation complex. Mining journal, №1, P. 78- 82. 2018.
- [6] Rylnikova, M., Ainbinder, I., Radchenko, D. Role of Safety Justification of Mining Development for the Regulatory Framework Formation and Mineral Resources Management 2018 E3S Web of Conferences 41, № 01033
- [7] Степанов С. Ю. Сравнительный анализ открытых геоинформационных систем // Информационные технологии и системы: Управление, Экономика, Транспорт, Право. 2013.

Василий Балашов, Георгий Курячий

Москва, ВМК МГУ, 2022

<https://uneex.org>

## Как мы добывали огонь. Организация аудиторного практикума по курсу «ЯП Python» на останках дистанционного обучения

*...благодаря обыкновенной палочке с гвоздиком, я борюсь за чистоту не сгибаясь...*

Саша Соколов

*«Школа для дураков»*

### Аннотация

Аудиторный практикум по курсу «Язык программирования Python» проводится уже не первый год, при этом постоянно дорабатывается. Попытка свести воедино все накопленные за это время технические инструменты и методические принципы показала, что, во-первых, их не так уж и мало и не все из них очевидны, во-вторых, они, по-видимому, хорошо дополняют друг друга, а по отдельности работают хуже, и в-третьих, включают в себя компоненты дистанционного образования, которыми авторы занимались ещё до того, как это стало трендом.

**Типичные проблемы практикума.** Практические занятия по плотному курсу программирования оказывают неэффективны из-за различия индивидуального стиля восприятия (аудиал, моторный, визуал и т. п.), скорости решения нетривиальных задач (даже самых простых), а также нетвёрдого восприятия теории. Кроме того, много времени тратится на переключение между демонстрацией чего-то на доске и беготнёй по классу. Саму доску обычно тоже плохо видно.

**Структура занятия.** Мы выделили пять видов активностей, связанных с практикумом.

1. Изложение фрагмента теории
2. «Режим падавана»: воспроизведение действий преподавателя
3. «Упражненька»: короткая самостоятельная работа
4. Домашнее задание на неделю
5. «Чатик»: развёрнутые и/или частные вопросы в телеграм-группе

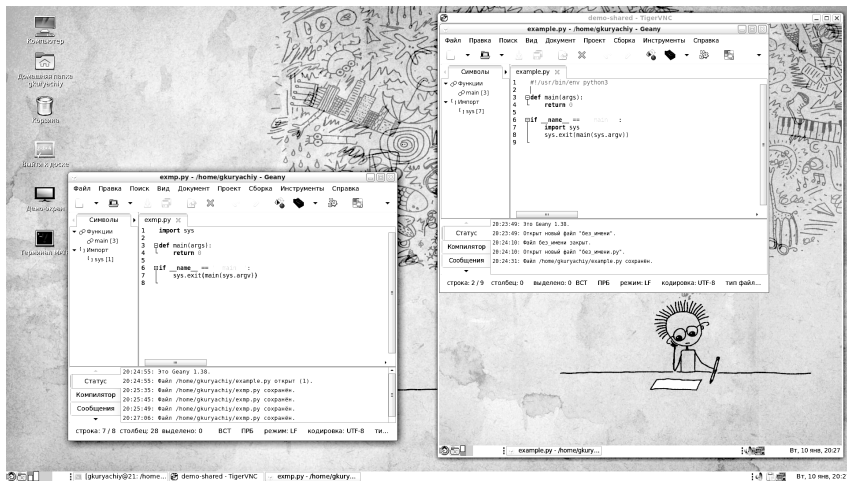
Дополнительно: каждый практикум посвящён тематике предыдущей лекции, доступной онлайн в реальном времени и в записи.

**Методика в этом году.** Мы выдвинули и реализовали тезис «упражненько-ориентированности». «Упражненька» — это содержательно тривиальная самостоятельная работа, которую преподаватель и слушатели делают одновременно (тем самым контролируется время выполнения). Обычно их 7±2 за семинар, перед каждой упражненькой может идти разбор соответствующей методики, после — синхронизация: «Кто ещё не догнал, помашите ручками».

Лекционный материал на семинаре практически не даётся.

Если в силу индивидуальных особенностей студента или неудачной формулировки упражненька оказывается сложной, её всегда можно превратить в демонстрацию.

**«Доска» и «режим падавана».** «Доска» — это трансляция рабочего стола преподавателя, который занимает *половину экрана*. Оставшаяся половина — для самостоятельной работы. Такую доску хорошо видно всем, и преподавателю не надо бегать к ней и от неё. Кроме того, становится возможным последовательное написание демонстрационной программы, которую слушатель воспроизводит в своей части экрана — т. н. «режим падавана», активизирующий моторную и ситуационную память.



В изначальном проекте предполагался также доступ преподавателя к рабочему столу произвольного слушателя по его запросу (т. н. «кнопка помощи»), но он так и не понадобился.

«**Чатик**». Чатик — это две группы в Telegram: внутренний чат 321-й группы ВМК (организационные вопросы и конкретика практикума) и общий чат по лекциям (теория и всё остальное). Всё, что по формату замедляет практикум — сложные/отвлечённые вопросы и дискуссии, вопросы по лекциям и Д/З к лекциям, организационные подробности и т. п. — решается вне семинара.

Таким образом чатик становится неотъемлемой частью учебного процесса. **Важное правило:** вопросы про практикум и лекции задаются в *общем чате*, а не в личной переписке.

**Лекции в YouTube.** Лекции читаются аудиторно, и параллельно идут в виде онлайн-стрима, который потом облагораживается и публикуется повторно. Таким образом, если человек за неделю не удосужился посмотреть лекцию и не задал по ней вопросов, он чётко позиционирует себя как немотивированного.

Стрим формирует отдельное сообщество (порядка 5 тыс подписчиков на момент публикации), для которого доступен чат. Иногда лектор отвечает на вопросы оттуда.

**Связь с ДО.** Часть инструментов и приёмов заимствована из практики дистанционного обучения: загрузочный образ ОС собственного производства с «доской» взят из дистанционных занятий в филиалах, синхронизация и переключение активностей — из методики проведения практикума во время пандемии.

**Истории неуспеха.** Похожая методика была в разные годы применена как минимум трижды на другой аудитории — для начинающих (психфак, геофак, философский факультеты МГУ). Выяснилось, что «лекции для начинающих» неэффективны, был сделан упор на практику. Тем не менее осталась главная проблема — несоблюдение «принципа 1000 упражнений»: времени, которое можно выделить за один семестр на непрофильный курс, не хватает на формирование полноценных навыков написания программ. Кроме того, во всех случаях наблюдается аномально низкая активность в чатах — признак низкой мотивации. Как следствие, до финиша доходят только слушатели, мотивированные познанием, которые в состоянии сами себе организовать дополнительную практику. Таких обычно 5%–10%.

**Итоги** этого года:

- Теперь практически не востребован второй «бегающий» семинарист. *Хорошо это или плохо?*
- Очень мало «троечников»: нет участников, которые «старались, но не осилили».
- Не до конца отлажен механизм тестирования решений Д/З.

Дальнейшее развитие: ревью исходных текстов и онлайн-публикация итогов решения Д/З в реальном времени.

Фомин Станислав Александрович

Москва, ИСПРАН, МФТИ

<https://discopal.ispras.ru>, <http://wiki.4intra.net/>,

<https://github.com/belonesox/cocalc>,

<https://github.com/belonesox/cocalc-on-fedora-ansible>,

<https://github.com/spacedeck/spacedeck-open>

## Современные «интерактивные среды» и «живые лаборатории» — эффективное дистанционное образование по алгоритмам и математическим дисциплинам

### Аннотация

«Ковидный период» и недавно наступившая «эра перемен» сильно подстегнула ВУЗовское дистанционное образование. Раньше нормой были «оффлайн» занятия, если повезло, поддерживаемые книжкой лектора, видеозаписями лекций-семинаров, а «дистанционка» была скорее редко терпимым исключением. Сейчас же нормой становится ситуация, когда преподаватель и даже студенты разбросаны по всему миру. Но мало, созвонившись в каком-нибудь zoom-е, начать читать лекцию, и, заглядывая в чёрные квадратики отключённых видеочаев, пытаться понять, слышат ли преподавателя студенты и вообще, что они делают, мучительно надеясь вовлечь их в процесс. Увы, это просто не работает. Ещё хуже с домашней работой — мало давать студентам «книгу» и надеяться, что они её прочтут — всё в этом устаревшем формате (объём, статичность, неудобство «листов», неинтерактивность, неаудируемость) неэффективно по современным меркам.

Да, есть масса онлайн сервисов, которые уже долгое время помогали аспирантам и студентам во время созвонов, и которые стали ещё популярнее в «ковидную эру» — разделяемые онлайн доски и документы, гугл-доки и гугл-колаб-ноутбуки, . . . но после «войны взаимных интернет-блокировок и санкций» на них нельзя рассчитывать. Например, в РФ программа «Google Suite for Education» закрылась из-за анти-РФ санкций, с другой стороны, десятки и сотни сервисов перестали работать только из-за того, что использовали кеширование и балансировку через сервис Cloudflare (запрещается РКН). Ситуация продолжает ухудшаться, перспективы неутешительны.

Только системы с открытым кодом, поднятые на своей инфраструктуре, предсказуемы и защищены от этих «блокад» — блокировок и санкций. Мы расскажем именно о таких системах, успешно применяемых и развиваемых

автором при преподавании математических и алгоритмических курсов в ИСПРАН и МФТИ и ведении исследовательской и научной работы в отделе математических методов и алгоритмов ИСПРАН.

Системы для организации созвонов, «интерактивные доски», системы для одновременного совместного редактирования кода и текстов, «коллаборативные блокноты»... организуют и «классную» и «внеклассную» работы, граница между которыми уже размыта по всему, что как-то связано с математикой и алгоритмами. А их использование позволяет делать новые типы образовательного контента — на замену унылым толстым книжкам и уродливым слайдам приходят «живые лаборатории» с компактными интерактивными материалами, с которыми можно и «поиграть», и привлечь студентов к совместному творчеству нового.

Конкретно, речь пойдёт о проектах «MediaWiki на стероидах» (Wiki4-Intra.net), SpaceDesk, VS Code Server, CoCalc, Jitsi, об опыте их использования и доработки.

Кулагин Владимир Петрович, Халабия Рустам Фарук,  
Степанова Ирина Владимировна  
Москва, МИРЭА  
<https://www.mirea.ru/>

## Использование среды виртуализации в подготовке системных программистов

### Аннотация

Обучение будущих системных программистов связано с необходимостью овладения ими практических навыков по администрированию операционных систем и модернизации различных системных компонентов, утилит и драйверов, а также изучение методов применения и доработки свободного программного обеспечения. Что довольно сложно обеспечить на имеющемся ограниченном аппаратно-программном обеспечении с предустановленным коммерческим ПО. Это логически приводит к необходимости использования для обучения виртуальных машин. Описываемое программно-аппаратное решение для создания виртуальной среды основано на использовании серверов с ОС Альт Сервер Виртуализации, ОС Альт Сервер и толстых клиентов с ОС Альт Рабочая станция

В настоящее время в Российской Федерации резко выросла потребность в разработке отечественных системных программных продуктов.



Кафедра несколько лет назад разработала основную образовательную программу в области подготовки системных программистов. Данная программа основывается на профессиональном стандарте 06.028 «Системный программист» и государственных образовательных стандартов по направлениям 09.03.02 и 09.04.02. Данные стандарты требуют получения и практического освоения студентами профессиональных знаний, таких как: принципы организации, состав и схемы работы операционных систем, принципы управления ресурсами, архитектура аппаратной платформы, для которой разрабатывается драйвер или системная утилита, методы организации файловых систем, принципы построения сетевого взаимодействия. В процессе формирования учебных программ и обучения студентов мы столкнулись с рядом трудностей, связанных как со спецификой преподаваемых дисциплин, так и с необходимостью ускоренного перехода на свободное обеспечение.

Так, ранее процесс обучения во многом был ориентирован на использование проприетарного программного обеспечения, в частности ОС семейства Windows, и для перестройки учебных программ требуется некоторое время.

Для перехода на отечественное ПО в качестве ОС используется Альт Рабочая станция с дополнительно установленным пакетом «Альт Образование». Однако некоторые проблемы связаны с наличием большого количества компьютеров с предустановленной операционной системой Windows — удаление данной ОС или установка второй ОС на этих машинах ограничено фирмой-производителем.

При использовании свободного ПО а также при обновлении Windows возникли проблемы с использованием дополнительного периферийного оборудования (отсутствие или отказ в обновлении драйверов).

Необходимо обеспечить возможность получения практических навыков работы для всех студентов. При этом надо учитывать, что будущим системным программистам необходимо уметь разрабатывать системные утилиты операционной системы, администрировать операционные системы и сетевое окружение — процесс обучения этим навыкам неизбежно связан с ошибками, которые будут допускать студенты, ошибками, которые могут привести к нарушению работы ОС. Кроме этого, имеющееся на кафедре программное и аппаратное обеспечение не охватывает весь спектр, необходимый для обучения высококвалифицированных специалистов.

Для решения данных проблем было принято решение частично перевести процесс обучения с физических компьютеров на виртуальные машины. Это обеспечит студентам возможность «безопасной» работы с системным ядром различных ОС, построения распределённых систем и т.п. На базе имеющихся двух серверов и персональных компьютеров была создана виртуальная среда для обучения. На первый сервер установлена операцион-

ные система Альт Сервер Виртуализации. Второй сервер с установленной ОС Альт Сервер используется как архив для хранения версий программ и работ студентов, а также для тестирования ПО. Персональные компьютеры функционируют как толстые клиенты под управлением ОС Альт Рабочая станция. Использование толстых клиентов позволяет задействовать персональные компьютеры с другой предустановленной ОС. При обучении каждому студенту выделяется своё рабочее пространство или виртуальная машина, поэтому во время проведения занятий студенты могут заниматься разными задачами, что повышает их самостоятельность. Виртуальные машины конфигурируются на сервере с помощью Proxmox Virtual Environment (PVE).

Полученное аппаратно-программное решение в настоящее время используется для обучения студентов в рамках учебных дисциплин, посвящённых разработке программного обеспечения: «Системное программное обеспечение», «Технологии разработки и сопровождения системных утилит».

На данный момент не хватает дистрибутивов с разными конфигурациями операционных систем и архитектур аппаратной платформы для разработки свободного программного обеспечения, на основе которых можно развёртывать разнообразные виртуальные машины. В дальнейшем планируется реализовать на базе среды виртуализации изучение сетевого окружения и распределённых систем в рамках читаемых на кафедре дисциплин «Технологии сетевого взаимодействия информационных систем», «Системы распределённого управления данными», «Средства контроля и восстановления данных в распределённых системах», «Реконфигурируемые системы», «Системное и инструментальное обеспечение многоагентных систем». Также в дальнейшем реализованная виртуальная среда будет использоваться для обучения и научных исследований распределённого искусственного интеллекта и параллельных вычислений в рамках свободного программного обеспечения.

Жданович Павел Борисович, Жданович Светлана  
Фелорисовна

Волгоград, Волгоградский государственный медицинский университет,  
Волгоградский государственный университет, Волгоградский  
государственный социально-педагогический университет, ООО ИК  
«Сибинтек»

## Распределённое образовательное IaaS-облако для изучения информационных технологий

### Аннотация

Рассказывается о создании и эксплуатации облачной образовательной инфраструктуры на базе свободного ПО, в которой обучающиеся запускают виртуальные машины на своих личных компьютерах.

В дистанционном обучении выделяют следующие аспекты.

Асинхронный: студенты могут выполнять некоторые учебные задачи по своему индивидуальному расписанию. Синхронный: при общении студентов с преподавателями в реальном времени студенты получают такой же объём и качество знаний, как и в кампусе. При обучении информационным технологиям синхронность означает, что обучающиеся получают доступ к рабочему столу преподавателя, а преподаватель — к рабочим столам обучающихся. Кроме того, преподаватель должен полностью контролировать среду выполнения учебных заданий на стороне обучающегося: изменять её настройки, пересылать файлы и т. д.

Средства видеоконференцсвязи (ВКС) этих функций не предоставляют, за исключением функции демонстрации экрана всем участникам конференции. При использовании средств полноценного удалённого управления в учебном процессе требуется, чтобы среды выполнения учебных заданий на стороне каждого обучающегося были одинаковыми, как в аудитории. Этого можно достичь, организовав учебную работу в виртуальной IT-инфраструктуре, где обучающимся предоставляются однотипные виртуальные машины, заранее настроенные для выполнения той или иной учебной задачи. Использование для этой цели услуг IaaS-провайдеров сильно затруднено как с финансовой, так и с организационной точек зрения. Более перспективной задачей является создание образовательной организацией своего собственного, частного облака, однако она не может быть быстро решена с нуля.

В своей практике мы реализовали следующий подход. Для изучения конкретной темы, дисциплины или группы дисциплин создаётся виртуальная машина, на которой предустановлено всё необходимое и только необ-

ходимое программное обеспечение. Образ этой машины распространяется среди обучающихся любым доступным способом, например, через Web-сервер. Скачав образ, обучающийся разворачивает и запускает его на своём компьютере, например, под KVM или Oracle VirtualBox. Запустившись, виртуальная машина подключается к нашему VPN, используя предоставленные гостевые сертификаты и ключ, которые потом заменяются на именные. После этого виртуальная машина получает доступ к нашим сервисам (СУБД, HTTP-серверам файл-серверам и т. д.), а также к виртуальной машине преподавателя, причём этот доступ можно эффективно контролировать. В свою очередь, обучающий и его IT-служба получает полный доступ к виртуальной машине обучающегося, но не к его хост-системе.

Во время онлайн-занятия преподаватель взаимодействует индивидуально с рабочими столами обучающихся (VNC, RDP) и предоставляет свой рабочий стол, когда это нужно. Отметим также, что преподаватель имеет возможность демонстрации рабочего стола любого из обучающихся остальной группе. При выполнении групповых заданий можно разрешить необходимый сетевой трафик между обучающимися. Вербальный и визуальный контакт в ходе занятия осуществляется при помощи средств ВКС.

Описанная система является, по определению, распределённым IaaS-облаком, в котором существенная часть ресурсов выносится на инфраструктуру пользователя (обучающегося), но продолжает управляться провайдером (образовательной организацией или одним преподавателем). Напомним, что распределённые облака вошли, по версии Gartner, в десятку главных направлений развития технологий в 2020 году.

Облако реализуется на основе свободного программного обеспечения. Исключением может являться только программное обеспечение гостевых систем, если содержание обучения предполагает использование проприетарного ПО. Назовём некоторые основные программные компоненты инфраструктурного ПО, использованные нами на практике: Debian 11, OpenVPN, Apache HTTP server, TigerVNC, SSH. Отметим, что свободные ВКС-серверы Apache OpenMeetings и Jitsi Meet также являлись в нашем случае частью виртуальной инфраструктуры. Построение системы на основе ОС Linux является принципиальным с точки зрения управления сетевым трафиком (шейпинг, маршрутизация, межсетевой экран) и автоматизации операций управления, таких как распределение ключей, обмен файлами, создание VNC-подключений для новых студентов, учёт времени работы, посещаемости и пр.

Описанное распределённое образовательное облако создавалось нами в условиях экстренного массового перехода на дистанционное обучение во время пандемии COVID-19, однако оно продолжает эффективно использоваться и по возвращении к очному обучению, где средства ВКС не исполь-

зуются, а также в очно-дистанционном обучении, где ВКС используется для связи с удалёнными слушателями.

А. А. Маркина, А. А. Шульган, Д. А. Костюк  
Брест, Брестский государственный технический университет

## Использование свободного программного обеспечения для ЭЭГ-мониторинга психического состояния пользователя в исследовании UX/UI

### Аннотация

В докладе рассматривается применение электроэнцефалографов потребительского уровня (изделий производства компаний NeuroSky и Emotiv, а также открытого проекта OpenBCI) для оценки эмоционального состояния пользователя в процессе человеко-машинного взаимодействия. Каждое устройство имеет свои преимущества и недостатки для исследования UI/UX. Рассматриваются особенности доступных к получению данных, обсуждаются существующие инструменты с открытым исходным кодом для получения этих данных, а подход к оценке эмоционального состояния, который можно использовать для распознавания положительных и отрицательных эмоций пользователя.

Оценка состояния пользователя с помощью специальных измерительных устройств в процессе работы с программным или аппаратным продуктом может заметно упростить выявление узких мест эргономики (особенности с учетом повысившейся в последние годы доступности биометрических устройств). Наиболее просты в применении средства изменения пульса, отслеживания взгляда, и наиболее примитивные электроэнцефалографы, позволяющие отследить концентрацию внимания. Однако более полноценное детектирование эмоций с помощью электроэнцефалографии (ЭЭГ) является существенно более сложной задачей.

Как известно, ЭЭГ основана на детектировании колебательных электрических процессов, регистрируемых на электродах, размещенных на поверхности головы, и является результатом электрической суммации и фильтрации элементарных процессов в нейронах. Измерения активности коры искажаются тканями и костями черепа, что вносит дополнительный шум и снижает интенсивность сигналов; тем не менее, измерения ЭЭГ дают важную информацию об электрической активности коры головного мозга. Частота сигналов ЭЭГ находится в диапазоне 1–80 Гц, а амплитуды сигналов варьируются от 10 до 100 мкВ [1].

Одной из основных характеристик ЭЭГ является частота. По историческим причинам используют частотную классификацию по визуально-различимым диапазонам ( $\alpha$  — 8..14 Гц,  $\beta$  — 14..40 Гц,  $\theta$  — 4..8 Гц,  $\delta$  — 1..4 Гц,  $\gamma$  — выше 40 Гц и т. д.). В зависимости от диапазона частот, а также амплитуды, формы волны, топографии и типа отклика различают  $\alpha$ -ритм ЭЭГ,  $\beta$ -ритм и т.д. Наиболее информативными для заявленной темы являются  $\alpha$ -ритмы и  $\beta$ -ритмы [2]. Их поддиапазоны не перекрываются источниками естественных шумов организма (таких, как движения глаз и моргание, сердечная деятельность, мышечная активность) [3], а посторонние артефакты от линий электропередач обычно присутствуют выше 50 Гц. Поэтому большая часть постороннего шума в  $\alpha$ - и  $\beta$ -диапазонах значительно снижена. Высокая активность в  $\alpha$ -диапазоне означает расслабление мозга, а  $\beta$ -ритм, наоборот, связан с его активным состоянием. Таким образом, вместе  $\alpha$ - и  $\beta$ -ритмы могут использоваться для обнаружения эмоциональных (возбужденных и валентных) состояний ума. Для выделения  $\alpha$ - и  $\beta$ -полос частот в регистрируемом сигнале традиционно используется полосовой фильтр, а при работе с дискретными сигналами — быстрое преобразование Фурье.

Наиболее бюджетные ЭЭГ выпускаются компанией NeuroSky и оснащены единственным датчиком, и потому их применение в исследованиях UI/UX сводится к оценке изменений в концентрации внимания. Однако на рынке доступно несколько более сложных потребительских ЭЭГ-гарнитур, которые позволяют выполнять оценку эмоционального состояния пользователя, а иногда даже комплектуются коммерческим ПО для такой оценки. Типичным примером с наиболее полным набором датчиков являются гарнитуры Emotiv EPOC/EPOC+ (она чаще используется в экспериментах). Эта гарнитура состоит из 14 электродов для сбора данных и 2 референсных электродов, которые позволяют отстраиваться от помех. Электроды размещены и промаркированы по международной системе 10-20 [3]. Для Emotiv EPOC это позиции AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 и AF4. Также достаточно совершенным решением является проект OpenBCI, доступный как свободное аппаратное обеспечение.

Коммерческие драйвера для гарнитур наподобие Emotiv EPOC/EPOC+ позволяют регистрировать ряд эмоциональных реакций (волнение, разочарование, вовлеченность, интерес, сосредоточенность и т.д.), но их возможности и доступность под GNU/Linux достаточно ограничены. Свободные решения (например, проект Emokit для EPOC/EPOC+) только выводят необработанные сигналы с электродов гарнитуры, поэтому при использовании СПО для ЭЭГ приходится вычислять оценку эмоционального состояния самостоятельно. То же самое верно при использовании гарнитуры OpenBCI (однако при этом OpenBCI является более дорогим решением, а его самостоятельная сборка нетривиальна из-за того что часть использу-

емых радиодеталей снята с производства и требует подбора аналогов по параметрам).

Для самостоятельной оценки эмоциональных состояний с помощью сигналов, регистрируемых электродами ЭЭГ-гарнитуры, можно использовать предложенный Дж. А. Расселом метод описания эмоций в виде точки в двумерном пространстве — циркумплекса. Две координатные оси представляют собой валентность и возбуждение, а расстояние от центральной точки — интенсивность. При классификации эмоций по этому методу необходимо определить, насколько положительные (валентность) и насколько сильные (возбуждение) эмоции ощущаются респондентами. Четыре сектора в этом пространстве — сильные отрицательные, сильные положительные, слабые отрицательные и слабые положительные эмоции. Поэтому классификация эмоциональных состояний предполагает их деление на высокое/низкое возбуждение и положительную/отрицательную валентность, на основании чего выделяют такие эмоции, как счастье, гнев, печаль, покой (например, счастье — это состояние с высоким возбуждением и положительной валентностью, а грусть — с низким возбуждением и отрицательной валентностью).

Уровень возбуждения можно определить, вычислив соотношение  $\beta$ - и  $\alpha$ -ритмов. Высокое возбуждение характеризуется большей мощностью  $\beta$ -ритмов и низкой  $\alpha$ -активностью, поэтому соотношение  $\beta/\alpha$  указывает на состояние возбуждения, в котором находится субъект. В то же время  $\beta$ - и  $\alpha$ -ритмы лучше всего измеряются в лобных и средних отделах головного мозга, поэтому целесообразно использовать сигналы, поступающие от электродов, расположенных на соответствующем участке черепа [2]. Согласно [3], достаточно измерить сигнал в четырех точках в области префронтальной коры: AF3, AF4, F3 и F4.

$$A = \frac{\beta_{AF3+AF4+F3+F4}}{\alpha_{AF3+AF4+F3+F4}}$$

где  $\alpha$  и  $\beta$  — мощность  $\alpha$ - и  $\beta$ -ритмов, AF3, AF4 и т.д. — сигналы от одноименных электродов.

Для оценки валентности, то есть положительного или отрицательного состояния, можно использовать разницу уровней активации двух полушарий коры головного мозга. По данным ряда исследований, положительное состояние связано с активацией областей в левом полушарии головного мозга, а отрицательное — в правом полушарии [4]. Эта связь не прямая: инактивация в левой лобной доле служит индикатором реакции отторжения, которая, в свою очередь, часто связана с негативной реакцией. Точно так же инактивация в правой лобной доле является показателем реакции, которая, в свою очередь, может указывать на положительную эмоцию.

Как уже упоминалось, высокая активность в  $\alpha$ -диапазоне является признаком низкой активности мозга, и наоборот. Таким образом, увеличение

$\alpha$ -ритмов на фоне снижения  $\beta$ -ритмов может быть связано с инактивацией коры [2]. Для измерения такой  $\alpha$ -активности чаще всего используют точки F3 и F4, так как они расположены в префронтальной области, играющей важнейшую роль в регуляции эмоций. При этом следует учитывать, что, согласно имеющимся исследованиям, различия в активности полушарий являются не признаком аффективной валентности, а скорее указанием на мотивационную направленность (притягательность или неприятие стимула) [5]. Однако, поскольку аффективная валентность обычно относится к мотивационной направленности, сравнение активации полушарий по-прежнему служит практическим методом определения валентности. Таким образом, для оценки валентности можно сравнить интенсивности  $\alpha$ - и  $\beta$ -ритмов на электродах F3 и F4 [6].

$$V = \frac{\alpha_{F4}}{\beta_{F4}} - \frac{\alpha_{F3}}{\beta_{F3}}$$

Как уже упоминалось, этот способ оценки эмоций требует доступа к необработанным данным с датчиков. В случае OpenBCI такой доступ является вариантом по-умолчанию; в случае гарнитур Emotive необработанные данные доступны в рамках более дорогих лицензий, предназначенных для исследователей, однако эти же данные позволяет получить свободная библиотека Emokit. В комплекте с Emokit идёт только простой пример консольного вывода значений с датчиков и качества контакта, поэтому на пользователя полностью ложатся также и задачи калибровки контакта электродов, отбора значений их и частотной фильтрации. К Emokit существует несколько фротэндов [7, 8, 9], упрощающих эту задачу. Наиболее развитый из них — EmokitVisualizer [7] — одновременно наименее доступен конечному пользователю из-за крайне чувствительного к вариациям набора устаревших зависимостей (требуется найти работоспособную комбинацию из нескольких модулей для Python, Qt4 и WxWidgets неуказанных автором версий). Более доступен фронтэнд на веб-технологиях CyKITv2 [9] (также находящийся в неактивной разработке). Альтернатива — прямое использование Emokit в собственной программной разработке на Python или Java.

## Литература

- [1] Kandel E. R. et al. *Principles of Neural Science*. Mc Graw Hill, 2012 — 1760 p.
- [2] Niemic C. P. *Studies of emotion: A theoretical and empirical review of psychophysiological studies of emotion* // Journal of Undergraduate Research, v. 1, 2002. pp. 15–18.



- [3] Ramirez R., Vamvakousis Z. *Detecting Emotion from EEG Signals Using the Emotive EPOC Device*. *Lecture Notes in Computer Science*, v. 7670. — Springer, 2012. — p. 175–184.
- [4] Winkler I. et al. *Frontal EEG asymmetry based classification of emotional valence using common spatial patterns* // *World Academy of Science, Engineering and Technology*, vol. 45, 2010. — pp. 373–378.
- [5] Harmon-Jones E. *Clarifying the emotive functions of asymmetrical frontal cortical activity* // *Psychophysiology*, iss. 40(6), 2003. — pp. 838–848.
- [6] Matlovich T. *Emotion Detection using EPOC EEG device* // *Informatics and Information Technologies Student Research Conference*. — Bratislava, Slovakia, April 28, 2016. — pp. 1–6.
- [7] EmokitVisualizer <https://github.com/EmokitAlife/EmokitVisualizer>
- [8] Emokit\_Epoc\_GUI [https://github.com/xribene/Emokit\\_Epoc\\_GUI](https://github.com/xribene/Emokit_Epoc_GUI)
- [9] CyKITv2 <https://github.com/tahesse/CyKITv2>

Владимир Зенкин

Москва, МГТУ им. Н.Э. Баумана

Проект: reportInMarkdown

<https://bitbucket.org/zenkin/reportinmarkdown/>

## Применение языка разметки Markdown для написания отчётов о НИР и других сложноструктурированных документов с регламентированными требованиями к оформлению

### Аннотация

В докладе описан опыт автора по написанию и оформлению технических отчётов по ГОСТ 7.32-2017 с помощью языка разметки Markdown. Для работы предложено применение скриптов автоматизации типовых задач (нумерация рисунков и таблиц, формирование библиографического списка, работа с размерностями величин и т.п.). Преобразование к форматированному виду осуществляется с помощью утилиты randoc. Выполнено сравнение предлагаемой методики с традиционными подходами к написанию технических документов в офисных текстовых процессорах и в  $\text{\LaTeX}$ .

В научной, учебной и инженерной работе существует типовая задача создания сложноструктурированных, форматированных по требованиям документов. К ним относятся технические отчёты, отчёты о НИР, расчётно-пояснительные записки по курсовым проектам, выпускные квалификационные работы и т.п. Помимо увеличения трудозатрат сложная структура документа с многочисленными внутренними ссылками на рисунки, формулы, список литературы чревата возникновением ошибок при составлении документа, когда система ссылок поддерживается вручную.

Для решения данной задачи можно выделить три типовые группы инструментов. К первой относятся офисные текстовые процессоры (MS Word, Writer). Являясь наиболее привычным для пользователя инструментом, они остаются плохо применимыми для больших сложных документов из-за проблем с автоматической нумерацией и внутренними ссылками, неудобством разделения проекта на несколько файлов, проблемами с переносимостью материалов между различными версиями ПО.

Ко второй группе относятся аналогичные инструменты, реализованные с помощью облачных технологий (Google Docs). Основным их достоинством является возможность организации коллективной работы над документом, в том числе синхронной. Однако при этом степень контроля над личными файлами критически снижается, что может быть чрезвычайно важным в условиях работы с коммерческой тайной, личными данными или материалами с грифом секретности.

Другим популярным инструментом является система  $\LaTeX$ . Имея огромные достоинства с точки зрения возможностей автоматизации, организации внутренних ссылок, работы с библиографией,  $\LaTeX$  обладает крайне высоким порогом вхождения, высокой сложностью, неудобен при внесении точечных правок.

В качестве альтернативного инструмента, который мог бы вобрать в себя преимущества первого и третьего подхода, и не слишком пострадал бы от их недостатков, автор предлагает применение формата разметки markdown с его последующей обработкой и компиляцией в формат docx с помощью утилиты pandoc.

Преимуществами markdown по сравнению с разметкой  $\LaTeX$  является:

- отсутствие необходимости в освоении;
- простота при наличии минимально-необходимого функционала;
- большое число редакторов;
- непосредственная визуализация «не текстовых» объектов: формул, рисунков, таблиц;

При этом текст в формате markdown сохраняет большую часть преимуществ  $\LaTeX$ :

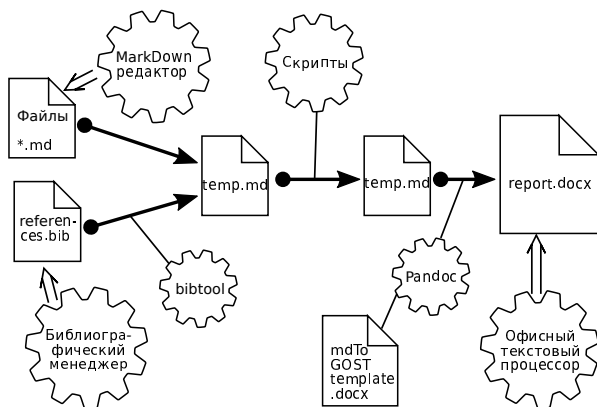


Рис. 1: Схема преобразования файлов проекта в отчёт

- абстрагирование от вёрстки при работе с документом;
- надёжность;
- возможность простой обработки скриптами и утилитами;
- возможность полноценного версионирования;
- независимость от конкретного ПО для редактирования.

Для обеспечения необходимого функционала работы с внутренними ссылками и со списком используемых источников автором был написан ряд скриптов. С их помощью можно проводить автоматизированную нумерацию позиций, указывая в тексте метки вида: !!группа.ключ!!. Для списка литературы поддерживается работа с библиографией в формате bibtex. Общая схема работы используемых скриптов и утилит показана на рисунке 1.

Утилита pandoc осуществляет преобразование файлов проекта в формат docx по шаблону.

Основные преимущества подхода:

- отсутствие необходимости в освоении инструмента работы с текстом;
- WYSIWYM. Привычная работа с документами с «in-line» визуализацией формул, рисунков, таблиц, заголовков и т.п.
- текстовый формат хранения данных повышает надёжность системы, упрощает версионирование, позволяет неограниченно писать и использовать скрипты;
- прозрачная организация внутренних ссылок внутри документа;

| Требование                               | Word | Л <sup>A</sup> T <sub>E</sub> X | riMD |
|--|------|---------------------------------|------|
| Автоматизация внутренних ссылок          | -    | ++                              | +    |
| Работа с библиографическими базами       | -    | ++                              | +    |
| Удобство работы с формулами              | -    | ++                              | ++   |
| Организация совместной работы            | -    | +                               | +    |
| Повторное использование материалов       | -    | +                               | +    |
| Удобство версионирования                 | -    | +                               | +    |
| Отображение картинок и формул в тексте   | ++   | -                               | +    |
| Гибкость в создании финального документа | ++   | -                               | +    |
| Удобство работы с таблицами              | ++   | --                              | +    |
| Низкий порог вхождения                   | ++   | --                              | ++   |

Таблица 1: Требования к инструментам создания отчётов

- возможность разделения файла на части для лучшей организации проекта или для совместной асинхронной работы.

В таблице 1 перечислены основные требования, актуальные для организации рабочего процесса при оформлении отчётов с указанием уровня их реализации в средах текстовых процессоров, с помощью Л<sup>A</sup>T<sub>E</sub>X и с предлагаемой автором методикой.

Таким образом, автор предлагает подход к организации работы по оформлению сложноструктурированных документов научного и технического профиля, направленный на ускорение, повышение удобства и уменьшение числа ошибок. Система реализуется на основе свободного программного обеспечения.

**Карпеш Сергей Валерьевич**

Переславль-Залесский, ИПС им. А.К. Айламазяна РАН

## **Организация записи и вещания научно-технических конференций и роль открытого ПО в данном процессе**

### **Аннотация**

Современный мир дошёл до того, что в объектив камеры попадает абсолютно всё. Любое событие человек пытается записать, сохранить и распространить. В случае с конференциями не все понимают, как это делать. Организация трансляций является следующим уровнем сложности. Принцип формирования изображения тот же, что и у записи, но

обработка должна проводиться уже в реальном времени. Задача не самая простая, но, как показала практика, может быть и хуже. Во время пандемии коронавируса появился термин «удаленный докладчик». Со стороны докладчика очень удобно, но со стороны оффлайн участников и технического персонала, обслуживающего работу конференции, это вызывает лёгкий дискомфорт.

Долгий итеративный процесс съёмок конференции для проекта 0x1.tv показал оптимальный сценарий: несколько камер с разных ракурсов (докладчик, аудитория) и аппаратный захват картинки, передаваемой на проектор. Далее идёт трудоёмкий процесс синхронизации источников и монтаж итогового ролика. Трансляция же эту операцию выполняет в реальном времени, так как все источники приходят в один компьютер. С точки зрения ПО, для этой задачи есть отличное решение в виде OBS Studio, но для получения качественного результата нужно ещё и аппаратное обеспечение, с которым не всё так просто.

Разумеется, можно попытаться обойтись минимумом оборудования и использовать возможности софта. Автором была опробована комбинация из захвата экрана средствами VLC и звука через PulseAudio с дальнейшей передачей потоков по сети на расположенный в центре аудитории ноутбук. Это решение рабочее, однако было большой ошибкой оставить захват звука на расстоянии (обусловлено расположением камеры), так как повлиять при перегрузке линейного входа удалённо нельзя. При грамотном расположении используемого оборудования подобных проблем можно избежать.

Вторая итерация учитывала набитые шишки. Передача по сети содержимого экрана заменена на захват сигнала, идущего на проектор, что позволяет использовать любой ноутбук, а не только предварительно настроенный. Ранее наблюдался дефицит решений в области захвата, особенно поддерживаемых в Linux. Профессиональные устройства жёстко привязаны не только к Windows, но ещё и к ПО (например, невозможно использовать вне Adobe Premier), а только зарождающийся сегмент «бытовых» устройств стабильностью похвастаться не мог. Из подходящих (на раннем этапе погружения автора в тему) было решение от производителя Eriphan. Это VGA (до FullHD) захват с интерфейсом USB 2.0 и некоторыми «приколами», вроде понижения частоты кадров при повышении разрешения (эффект не критичен для слайдов). Производитель выпускал драйвер под Linux, с которым устройство становилось доступным через v4l2. Со временем появились качественные решения от производителя Magewell, которые подключаются к шине PCI-E и имеют на борту как VGA, так и HDMI (с бонусами в виде передачи аудио и работы по дешёвой витой паре). Правда версия драйвера под Windows имеет больше возможностей по сравнению с версией под Linux. Переход на HDMI сильно упрощает схему подключения, убирает лишние провода, а вместе с ними и земляные петли, которые вносят

высокочастотный шум в звук с микрофонов. На момент публикации уже доступны устройства захвата с интерфейсом USB, которые прикидываются веб-камерой и работают на базовом драйвере в любой операционной системе. Качество изображения на динамической картинке у таких устройств сильно уступает относительно устройств без сжатия, подключаемых к PCI-E, но стоимость при этом менее \$10 вместо \$300.

Пандемия коронавируса внесла серьёзные коррективы в работу конференций. Сейчас почти все конференции проходят в гибридном режиме, когда минимум 1 докладчик подключается удалённо. На первый взгляд, это не должно вызывать проблем, но организаторам теперь предстоит присутствующим лично участникам показать онлайн доклады, а удалённым участникам оффлайн. В описываемом случае копию ещё и транслировать.

На рынке много сервисов, предоставляющих удобную связь. Самыми распространёнными являются Zoom и Skype. Существуют и свободные проекты, вроде BigBlueButton, которые можно поднять на своём оборудовании. При проведении конференции в гибридном формате самым сложным является переключение между онлайн и оффлайн режимами. В этот момент вместо картинки с ноутбука на проектор требуется подать демонстрацию экрана удалённого докладчика и, при наличии, выводить изображение с камеры на проектор/телевизор для комфорта аудитории. Выполнить эти операции возможно только при наличии отдельных потоков, с которыми будет легко производить различные манипуляции.

Лидером в данном вопросе является Skype, который достаточно давно (как минимум с 2018 года) позволяет получить все интересующие потоки через технологию NDI (передача изображения с малой задержкой по локальной сети). Клиент Skype создаёт по одному NDI потоку на каждого участника звонка, демонстрацию экрана и дополнительно «любой активный». Все свободные проекты не обладают подобным функционалом, однако работают в браузере и можно добиться похожего результата средствами OBS Studio с пагином. Он позволяет использовать NDI потоки в качестве источников и формировать потоки из любого уже имеющегося источника сигнала. Практика показала, что это оптимальный вариант для быстрого переключения между режимами онлайн/оффлайн, который позволяет

Последней проблемой, не дающей оператору спокойно наслаждаться докладом, является передвигающийся по сцене докладчик, за которым приходится следить камерой. Ручное направление камеры можно заменить на моторизованную поворотную платформу. Автору сильно упростило задачу наличие профессионального жёсткого штатива, на голову которого можно легко закрепить напечатанные на 3D принтере шестерни и моторы. Для генерации шестерёнок использовался проект OpenSCAD, контроллером выступила плата Arduino под управлением открытой прошивки для ЧПУ станков grbl. Хотя поворотное устройство и не похоже на ЧПУ фре-

зерный станок, но требования к прошивке одинаковы: перемещение по осям в координаты с плавным разгоном и торможением. В случае автора камера имеет порт подключения проводного пульта управления zoom-ом, что сильно упростило задачу. Вместо модификации прошки grbl проще выделить отдельный микроконтроллер под реализацию протокола LANC, на который с основного подаётся ШИМ сигнал, где 0% заполнения соответствуют максимальной скорости удаления, 100% максимальной скорости приближения и 50% остановка. Решение выглядит крайне забавно, если учесть единственный источник ШИМ сигнала в прошивке grbl: скорость вращения шпинделя. Аппаратно проблема слежения за докладчиком решена, но ПО требует доработки.

Развитие данного «проекта выходного дня» видится следующим: автоматическое слежение за докладчиком при помощи OpenCV и перемещение в координату, определяемую кликом на изображение с камеры общего вида.

А. Д. Дельман, А. М. Ильин, Д. П. Князихин, Э. А. Макаров,  
А. Н. Непейвода, А. С. Терентьева, М. Р. Терюха,  
А. А. Чибизова, К. К. Шевченко  
Москва, МГТУ им. Н.Э. Баумана

Проект: Chipollino <https://github.com/StarikTenger/Chipollino>

## Интерактивный конвертер Chipollino для наглядного изучения теории автоматов

В теории формальных языков существует множество различных способов представить регулярные языки, начиная от классических конечных автоматов и кончая полугруппами преобразований. Знакомство с этим разнообразием представлений не только расширяет кругозор, но и формирует связь между различными областями теоретической информатики. Однако в рамках учебного курса на теорию автоматов отводится крайне мало времени, и попытка вместить в эти часы хотя бы базовые понятия о каждом из представлений приводит к тому, что они выглядят как мало связанные между собой разрозненные факты. При этом, если студент пропустил общеизвестную тему (например, по минимизации ДКА), его можно отправить «смотреть индусов», но некоторые из тем представлены в основном в специальной литературе, и если студент не успел понять такую тему на лекции, то он вынужден разбираться в ней по единственному примеру из слайдов или по статьям. Несложную и красивую теорию оказывается трудно оставить на самостоятельный разбор из-за недостатка наглядных материалов.

Изначально проект возник из намерения перенести некоторое количество теории в лабораторные работы, в связи с чем наряду с индивидуальными вариантами лабораторных появилась возможность взять групповую задачу. Наглядное представление было лишь одной из подзадач и в основном требовалось для отладки алгоритмов преобразования различных представлений друг в друга (тем более что студенты уже владели базовыми навыками использования Graphviz и зачастую сами добавляли вывод диаграмм в свои лабораторные работы). Чуть позже стало ясно, что если эту сторону проекта лучше развить, то им могут воспользоваться не только его непосредственные разработчики, но для этого недостаточно строить dot-представление автоматов, а также нужно демонстрировать промежуточные выкладки в форме таблиц, списков и пошагового описания преобразований. Эти представления удобно записывать в  $\text{\LaTeX}$ . Окончательная конвертация диаграмм [6] тоже осуществляется в  $\text{\LaTeX}$ , с помощью библиотеки Tikz. Дополнительно к конвертеру, в частности, для упрощения тестирования, добавлен модуль порождения случайных регулярных выражений и задач.

В настоящий момент конвертер охватывает следующие представления регулярных языков (см. Рис. , (a)):

- регулярные выражения;
- недетерминированные и детерминированные автоматы;
- трансформационные моноиды;
- префиксные грамматики.

На вход конвертеру подаётся текстовый файл с последовательностью инструкций, преобразования по которым будут залогированы. Пример файла с инструкцией представлен ниже. Восклицательные знаки указывают, что преобразование логируется.

```
R = {(aa*)*(a*)*a*b}
A = MergeBisim.Antimirov R !!
```

Результат преобразования регулярного выражения R в автомат Антимирова посредством построения частичных производных и слияния по бисимуляции состояний в автомате представлен на Рис. (б), (в).

Большое количество различных представлений и возможность автоматически проанализировать лично построенный пример регулярного выражения позволяет на практике разобраться со свойствами преобразований, рассматриваемых в курсе<sup>1</sup>. Например, убедиться, что алгоритм минимизации

<sup>1</sup>Неисчерпаемое число взаимосвязей между различными преобразованиями в теории автоматов и вдохновило на идею проекта-конвертера.



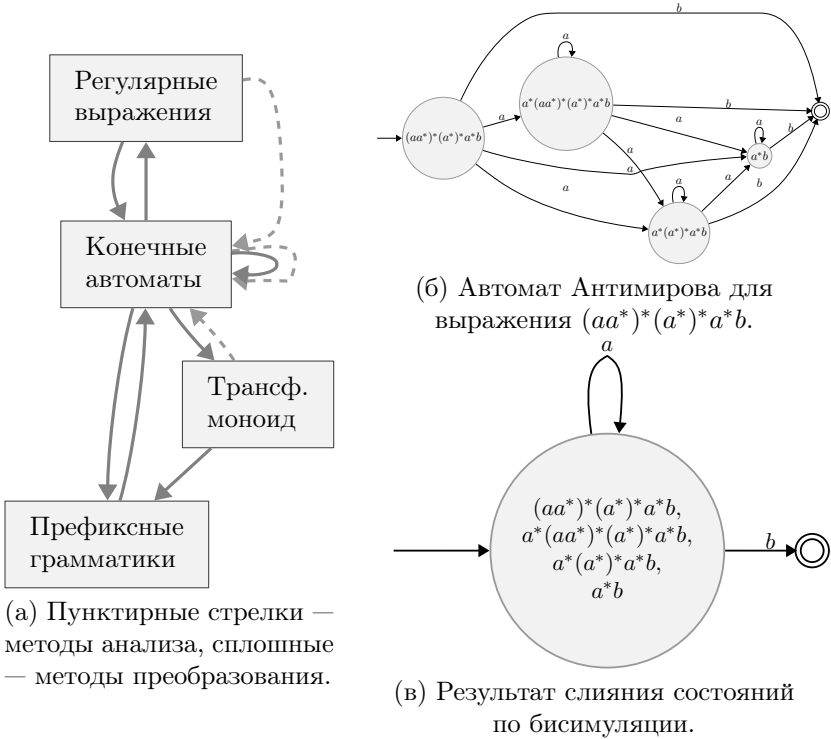


Рис. 1: Логика конвертера Chirpollino и пример преобразований

ции Брзозовски (посредством цепочки обращений и детерминизаций) эквивалентен классическому алгоритму минимизации [3], и сравнить их скорость для случайно порождённых автоматов; или что автомат Глушкова получается из автомата Томпсона после удаления пустых переходов [4]. Ещё один интересный опыт был связан с поиском нижней оценки числа минимальных состояний в недетерминированном конечном автомате: эксперименты показали, что наиболее известная оценка из работы [1] может быть легко улучшена (и, оказывается, улучшение уже существовало [2], причём опубликованное в том же журнале, но из-за использования различной терминологии сходство было замечено не сразу).

Определённые сложности проекта связаны с тем, что изначально он задумывался как лабораторная работа, от которой не требуется таких качеств, как масштабируемость и удобный пользовательский интерфейс. Так-

же из-за большого охвата теоретических конструкций не все из них пока что охвачены подробнымилогами — эта задача остаётся на будущее.

## Литература

- [1] *I. Glaister, J. Shallit* Lower Bound Technique for the Size of Nondeterministic Finite Automata // Information Processing Letters, 59, 75–77, 1996.
- [2] *J.-C. Birget* Intersection and Union of Regular Languages and State Complexity // Information Processing Letters 43, 185–190, 1992.
- [3] *B. W. Watson* A Taxonomy of Finite Automata Minimization Algorithms // Computing science notes, Vol.9344, Technische Universiteit Eindhoven, 1993.
- [4] *C. Allauzen, M. Mohri* A Unified Construction of the Glushkov, Follow, and Antimirov Automata // TR2006-880, Courant Institute of Mathematical Sciences, 2006.
- [5] *Ссылка на репозиторий проекта* <https://github.com/StarikTenger/Chipollino>
- [6] *Конвертер dot2tex* <https://github.com/kjellmf/dot2tex>

Михаил Маркушевич, Алексей Краснов

г. Москва, Государственное бюджетное общеобразовательное учреждение города Москвы «Школа № 1352»

## **О решении проблемы дефицита учебно-методической литературы, посвящённой методике эффективного использования СПО в учебном процессе российской школы**

### Аннотация

В настоящий момент процесс миграции учебного процесса российской общеобразовательной школы на отечественное программное обеспечение интенсифицировался, примером чего является массовая поставка в школы Московской области персональных компьютеров под управлением отечественной ОС Альт Образование, базирующейся на ядре Linux. С другой стороны, рассматриваемый процесс сталкивается с рядом проблем, одной из которых является наличие дефицита

учебно-методической литературы, посвящённой методике эффективного применения СПО в учебном процессе российской школы. Авторы статьи отмечают некоторые возможные пути, направленные на решение обозначенной проблемы.

В текущей ситуации, когда в результате возрастающего давления экономических санкций со стороны ряда недружественных западных государств активизировался процесс перехода российских образовательных организаций (как среднего, так и высшего образования) на отечественное свободное программное обеспечение, крайне важным, с точки зрения авторов, является обеспечение поддержки учебного процесса высококачественной учебно-методической литературой, посвящённой эффективной методике обучения как информатике, так и другим учебным предметам на основе использования данного типа программного обеспечения.

Многие педагоги, к сожалению, не имеют необходимого опыта применения как прикладного, так и системного свободного программного обеспечения в своей профессиональной деятельности в виду наличия колоссальной инерции в плане использования проприетарных популярных операционных систем семейства Windows, а также офисных пакетов, таких как Microsoft Office, ставших de-facto стандартом по умолчанию до недавнего времени в отечественном педагогическом сообществе. Несмотря на более частое упоминание в российских школьных учебниках свободных программных сред в последние годы, тем не менее, можно говорить о наличии частичного дефицита эффективных полноформатных методик обучения различным информационным технологиям на базе СПО.

Отметим, что благодаря Библиотеке Alt Linux, российским педагогам доступна литература, содержащая элементы методик обучения с использованием таких свободных программных сред как Maxima, Octave, Scilab, GIMP, Gnumeric, OpenOffice.org, языкам программирования Free Pascal, Python, C++ в среде Qt Creator, основам операционной системы Linux на примере Линукс Юниор, Alt Linux. Достаточно большое количество литературы посвящено методике обучения основам программирования на базе блочного языка Scratch [1]. Однако совершенно очевидно, что многие из свободных программных сред, которые могли быть успешно используемы в учебно-воспитательном процессе российской общеобразовательной школы в настоящее время не имеют соответствующих разработанных и опубликованных методик их эффективного применения. К таковым хотелось бы отнести операционные системы Альт Образование, офисный пакет LibreOffice, ПО для трёхмерной компьютерной графики Blender и др.

Авторами статьи приложены усилия к компенсации указанного частичного дефицита методик в области информационных технологий, в частности, методики обучения школьников обработки цифрового звука на базе использования аудио редактора Audacity [2], векторной графики с исполь-

зованием графического редактора LibreOffice Draw [3], растровой графики с использованием графического редактора GIMP [4], 3D графики с использованием отечественной системы компьютерного черчения Компас-3D [5]. Кроме того, подготовлены к печати книга «Рисую в LibreOffice Draw» и статья по методике обучения созданию и обработке текстовой информации с использованием текстового редактора LibreOffice Writer в журнале «Информатика в школе».

Когда мы говорим об эффективной методике обучения тем или иным информационным технологиям (обработки графики, текста, звука, видео и т.д.), базирующимся на свободном программном обеспечении, то, с точки зрения авторов, *методику обучения можно приблизительно рассматривать как процесс передачи от преподавателя обучающемуся некоего необходимого и достаточного набора алгоритмов создания тех или иных информационных объектов, соответствующих изучаемой информационной технологии, включающих в себя знание и умение использовать стандартный инструментарий той программной среды (сред), которые являются оптимальными для данных целей на основе сравнительного анализа доступного программного обеспечения.*

Именно описанный выше набор алгоритмов создания информационных объектов представляет насущный интерес для отечественных преподавателей информатики, так как даёт им возможность сразу начать эффективно использовать свободное программное обеспечение в учебном процессе в ситуации нехватки времени на подробное ознакомление с новыми для них программными инструментами. Оперативная разработка и публикация данных методик является серьёзным вызовом для сообщества российских специалистов и педагогов-исследователей, работающих в области применения СПО в образовательной сфере.

## Литература

- [1] Сорокина Т. Е. Информатика. 5–6 классы. *Практикум по программированию в среде Scratch.* / Т. Е. Сорокина, А. Ю. Босова; под ред. Л. Л. Босовой. — М.: БИНОМ. Лаборатория знаний, 2019. — 144 с.
- [2] Маркушевич М. В. *Элементы методики преподавания темы «Создание и обработка звуковой информации» с использованием свободного звукового редактора Audacity* // Информатика в школе. № 1. 2022. С. 55–66
- [3] Федосов А. Ю., Маркушевич М. В. *Элементы методики преподавания векторной графики на основе свободного графического редактора LibreOffice Draw на уровне основного общего образования* // Информатика в школе. № 9. 2021. С. 4–13

- [4] Маркушевич М. В. *Элементы методики преподавания темы «Растровая компьютерная графика» с помощью свободного графического редактора GIMP на уровне основного общего образования // Информатика в школе.* № 6. 2022. С. 12–29.
- [5] Маркушевич М. В. *Элементы методики преподавания компьютерной трёхмерной графики на уровне основного общего образования с использованием КОМПАС-3D LT V12 // Информатика в школе.* № 3. 2022. С. 53–66.

Драгунов Алексей Владиславович, Драгунов Константин Алексеевич, Гладченко Леонид Сергеевич, Петров Владислав Андреевич

Псков, Государственное бюджетное учреждение Псковской области «Региональный центр информационных технологий», ФГБОУ ВО «МГТУ «СТАНКИН», Псковский государственный университет  
<https://it.pskovedu.ru/>

## Развитие региональной информационной образовательной среды, основанной на свободном программном обеспечении

### Аннотация

Система общего и профессионального образования в России претерпела с начала века уже несколько перезагрузок комплексного подхода к компьютеризации, информатизации, цифровизации. Авторы статьи непосредственно участвуют в этих процессах, выполняя задачи, определяемые федеральными органами исполнительной власти, развивая региональную цифровую экосистему образования. В статье описывается новый опыт Псковской области по созданию комплексного цифрового решения для региональной системы образования, учитывающий сегодняшние требования к цифровой трансформации образования.

### Введение

В статье «Информационная образовательная среда региона, основанная на свободном программном обеспечении: технологии и содержание», которая вышла в сборнике трудов 13й конференции «Свободное программное обеспечение в высшей школе», мы рассказали о структуре системы и

lowcode — платформе собственной разработки, на которой она построена [1]. За прошедшие с того времени четыре года произошли серьёзные изменения, в том числе — изменились подходы к цифровизации образования — не только появились новые задачи, но и новые требования, в частности по обеспечению информационной безопасности.

## Безопасность

С 1 января 2023 года в соответствии с Законом «Об образовании в Российской Федерации» персональные данные обучающихся в системе общего образования и СПО могут обрабатываться только в государственных информационных системах. Отметим, что это не стало чем-то новым, тк государственными информационными системами должны были быть система доступности дошкольного образования и навигаторы дообразования. Теперь это касается всех подсистем, где обрабатываются персональные данные детей, включая электронные журналы, системы электронного обучения (в тч LMS), системы автоматизации учреждений СПО и цифровые платформы ЦОПП. Поскольку все эти подсистемы являются частью региональной информационной образовательной среды (РИОС), то в 2022 году была решена задача аттестации её в качестве региональной ГИС. Это потребовало от нас организации защищённого серверного узла. Было важно расположить более 30 наших подсистем в защищённом региональном «облаке», обеспечить их взаимодействие, управление и выделить защищённый сегмент, в который пользователи будут иметь доступ к расширенным данным только с защищённых рабочих мест. Перед нами стоял выбор между двумя технологиями — виртуализация и контейнеризация. Был сделан выбор в пользу Docker для размещения большинства подсистем и KVM для виртуализации небольшого числа виртуальных машин. В качестве серверной операционной системы был выбран «Альт 8 сервер» с поддержкой виртуализации/контейнеризации. Кроме опыта эффективной поддержки со стороны сообщества, одной из причин такого выбора была доступность сертифицированной версии, включавший в формуляр средств виртуализации. Непростой задачей стал выбор решения для организации безопасности на подключаемых компьютерах в образовательных организациях и органах управления образованием. После сравнения нескольких решений, было выбрано российское решение КриптоКом, построенное на защищённой реализации OpenVPN. Итоговая стоимость комплекта ПО для рабочих станций под операционными системами Linux и Windows составила около 15 тысяч рублей (сертифицированный сетевой клиент, антивирусное ПО, IP-сканер, средство защиты от НСД и межсетевой экран). К большому сожалению, ни одна сертифицированная версия российской операционной системы не позволяет использовать её в качестве защищённого рабочего места без этих

дополнительных затрат, просто ограничившись приобретением антивирусного ПО. В конце 2022 года проведена подготовка к проведению во всех ОО и ОУО оценки соответствия требованиям к защите информации, которое будет организовано для всех созданных защищённых рабочих мест.

## Новые возможности

В таблице 1 приведена структура РИОС по состоянию на конец 2022 года. Описание подсистем доступно по адресу <https://it.pskovedu.ru/> Проект в целом был представлен на конференции DIGITAL INNOPOLIS DAY – 2022 [2]

Табл. 1. Структура РИС «ЦОПО»

|   |  |  |
|---|--|--|
| <b>Общий сегмент</b><br>Идентификация и аутентификация, интеграция с ЕСИА<br>Реестр ОО и ОУО<br>Открытые данные «Региональный образовательный портал»<br>Мониторинг образования<br>Управление информатизацией<br>Приемка ОО к новому учебному году<br>Библиотека: инвентаризация обеспеченности учебниками и электронный заказ<br>Управление проектами и задачами в образовании Псковской области<br>Интерактивные учебные материалы<br>Электронное обучение Цифровой урок<br>Онлайн-трансляции<br>Кодификаторы содержания образования<br>Отдых детей в каникулярное время<br>Мероприятия в образовании | <b>Общее образование</b><br>Электронные журналы и дневники<br>Подача заявлений в ОО<br>Питание школьников  | <b>Оценка качества</b><br>РИС «Оценка качества образования»<br>Регистрация на экзамен<br>Результаты ГИА и КК             |
|   | <b>Дошкольное образование</b><br>Доступность дошкольного образования   | <b>Дополнительное образование</b><br>Навигатор и персфинансирование<br>Цифровой школьный музей                           |
|   | <b>Профессиональное образование</b><br>Подача заявлений и зачисление в профессиональные образовательные организации<br>Цифровая платформа ЦОПП<br>Цифровой колледж | <b>Развитие педагогических кадров</b><br>Банк педагогических практик<br>Аттестация педагогов<br>Аттестация руководителей |

За три последних года мы реализовали в регионе инвентаризацию обеспеченности учебниками и их электронный заказ; региональную образовательную платформу, позволяющую педагогам готовить и проводить занятия в синхронном интерактивном формате, используя верифицированный образовательный контент (ФГИС «Моя школа») и курсы Moodle; собственные видеосервера на базе nginx, обеспечивающие видеотрансляций ГИА (ЕГЭ, ОГЭ — замена сервисов Ростелекома), открытых уроков, олимпиад;

цифровую платформу центров опережающей профессиональной подготовки (ЦОПП); навигатор дополнительного образования и многое другое.

Уникальные особенности

1. Lowcode — гибкое изменение структур данных, процессов и интерфейсов без программирования
2. Конструктор взаимодействия с ЕПГУ (СМЭВ)
3. Вывод всех массовых социально значимых услуг на ЕПГУ
4. Поддержка цифровых административных регламентов — реализованы общие механизмы и МСЗУ «Аттестация педагогов»
5. Реализована интеграция с витринами ЕПГУ Минцифры (ожидается возможность вывода в продуктивную среду)

## Используемое свободное ПО

Ключевые опенсорс-решения, использованные при разработке системы:

1. Альт 8 Сервер — для целей аттестации использован в защищённом исполнении (<https://altsp.su/>)

2. Средства разработки:

2.1 PHP 5.4 — 8.x (<https://www.php.net/>) — основной язык, выбор которого, с одной стороны, был обусловлен историческими причинами. Остаётся современным эффективным языком с низким порогом вхождения и хорошей производительностью;

2.2 Jquery (<https://jquery.com/>) — использовался при разработке собственной lowecode-платформы для быстрой разработки и сопровождения отраслевых систем;

2.3 Open JDK (<https://openjdk.org/>) — на нём мы разрабатываем новую версию нашей lowecode-платформы;

2.4 C++, Qt (<https://gcc.gnu.org/>) — модули подсистемы распознавания для собственной подсистемы обработки бланков;

2.5 Python — модули обучения нейронок на примерах бланков, аналитический функционал;

3. СУБД

3.1 PostgreSQL (<https://postgresql.org>) — основная СУБД для большинства подсистем;

3.2 MongoDB (<https://mongodb.com/>) — хранилище для высоконагруженной базы данных оценок электронных журналов;

3.3 Firebird (<https://firebirdsql.org>) — база данных для настольных клиент-серверных приложений для работы с данными, входящих в состав платформы;

4. Образовательные решения



### Интеграция РИС «ЦОПО» с федеральными системами и ЦПО



Рис. 1: Интеграция подсистем РИС «ЦОПО»

4.1 LMS Moodle (<https://moodle.com>) — интегрирована с РИОС — подсистема электронного обучения на базе Moodle доступна каждому учащемуся и педагогу

4.2 Xerte Toolkit (<https://www.xerte.org.uk>) — конструктор интерактивных учебных модулей, отличная замена learningapps/h5p. Доступен каждому педагогу и учащемуся

#### 5. Другое

5.1 Tensor flow (<https://www.tensorflow.org/>) — используем для распознавания рукописных полей на бланках

5.2 Tesseract (<https://tesseract-ocr.github.io/>) — используем для распознавания печатных полей на бланках

### Интеграция

Реализованные интеграции представлены на рисунке 1. Подробно об использовании платформы для ГИА и интеграции с ЕПГУ и ФИС ГИА приводится в [3]

## Резюме и выводы

Чтобы достичь цифровой трансформации системы регионального образования — состояния перехода количества в новое качество сделано уже много на федеральном и на региональном уровне. Предлагаем обратить внимание на следующие задачи:

1. Интеграция и доступ к результатам использования цифрового образовательного контента:

1.1 обязательное требование к кодификации образовательного контента для общего образования, претендующего на верификацию и допускаемого к использованию и интеграции, с использованием универсальных кодификторов ФИПИ, тематических классификаторов общего образования;

1.2 обязательное требование к образовательным ресурсам и платформам, содержащим контент по предоставлению «цифрового следа» и поддержке стандартных протоколов интеграции, в зависимости от формата, модели и технологии использования контента (SCORM, xAPI, LTI и др.)

2. Улучшение интеграции систем Рособрнадзора с федеральными и региональными системами в сфере образования. Субъекты Российской Федерации, вузы и сузуы пока не получили возможности, которые были запланированы ещё на этапе эксперимента по введению ЕГЭ. В частности, регионы пока не получают информацию о поступлении своих выпускников, а вузы и сузуы — информацию об образовательных дефицитах зачисленных на первый курс студентов. Также пока не реализован алгоритм оптимального автоматического зачисления абитуриентов на основании заранее заявленных приоритетов при подаче единого заявления в вузы через ФИС ГИА. Не регламентировано получение результатов ГИА из базы данных ФИС ГИА с учётом изображений бланков и необходимой для анализа информации об обработке бланков и результатах верификации.

3. Обеспечение интеграции систем, размещённых на сайте <https://eds00.ru> «Единое содержание общего образования» («Тематический классификатор общего образования», «Конструктор учебных планов», «Конструктор рабочих программ») с РИС «Цифровое образование Псковской области»

4. Возможность предоставления информации в множество систем мониторинга и статистики из региональных информационных систем путём загрузки данных, без необходимости организовывать внесение данных с уровня образовательной организации. Большинство запросов, получаемых с федерального уровня содержат информацию, которая уже есть в тех или иных федеральных системах. Система постоянных параллельных запросов цифровых данных может быть представлена в виде перевёрнутой пирамиды, где внизу, в вершине находятся школы и педагоги. В настоящее время можно отметить несколько систем, с ко-

торами РИС не могут взаимодействовать и которые используются для получения данных: <https://1sep.edu.ru/>, <https://monitoring.fcprc.ru/>, <https://sas.ficto.ru/>, <https://is.cminp.ru/>, <https://fis-oko-vhod.ru/>, <https://еип-фкис.рф> (мониторинги и рейтинги — ФГБУ «ФЦОМОФВ»), <https://fcdtk.ru/museums>, <https://miccedu.ru/>.

Это только малая часть информационных систем сбора данных, с которыми приходится иметь дело образовательным организациям. В большинстве случаев эти системы не предусматривают возможности централизованной загрузки данных и экспорта информации на региональном уровне. Они не основываются на единой НСИ, не предоставляют сервисов для интеграции с внешними информационными системами.

Более 10 лет назад нами был разработан перечень принципов, которые помогли бы снизить бюрократическую нагрузку на систему образования, но к сожалению, пока они не реализованы. Считаем эту задачу одной из наиболее важных.

## Литература

- [1] Алексей Драгунов, *Информационная образовательная среда региона, основанная на свободном программном обеспечении: технологии и содержание*, / Свободное программное обеспечение в высшей школе. Сборник материалов Тринадцатой конференции. 2018/ Переславль, 26–28 января 2018 года / М. МАКС Пресс, С. 96–102. <https://www.elibrary.ru/item.asp?id=32528495&pff=1>
- [2] Алексей Драгунов, *Региональная государственная информационная система как инструмент цифровой трансформации сферы образования*, <https://cloud.pskovedu.ru/s/4CQB66ygywRHQaS>
- [3] Алексей Драгунов, *Цифровые сервисы ГИА в интеграции с ЕПГУ и ФИС ГИА*, <https://cloud.pskovedu.ru/s/8giyBsZR3HaTbPa>

Яков Шпунт  
Москва, ComNews

## Особенности российских операционных систем, используемых в ходе проектов по импортозамещению

### Аннотация

Минцифры svelo количество российских операционных систем до трёх: Astra Linux, Alt и RedOS. Причём велика вероятность того, что образовательные учреждения могут просто поставить перед фактом миграции на одну из этих трёх систем. Тут ничего страшного, но к этому надо быть готовым и знать о ключевых особенностях этих дистрибутивов.

Похоже, что процесс импортозамещения в сфере образования всё же начинается. Причём образовательные учреждения могут переводить на российское ПО в ходе комплексных проектов, вместе с другими подведомственными региональным органам власти организациям. Обычно в ходе таких миграций проходят мероприятия по переподготовке и обучению кадров, но под эти программы могут попадать не все преподаватели и технические специалисты, да и качество подготовки может хромать.

Вместе с тем, всё не так страшно, как может показаться на первый взгляд. Между разными дистрибутивами всё же больше сходств, чем различий, но к этим различиям нужно быть готовыми. Собственно, цель данного доклада и состоит в том, чтобы эти различия подсветить.

### Astra Linux

Данный дистрибутив довольно часто выбирают для миграции с Windows. Интерфейс в стиле Windows XP. Базируется на Debian, можно (но без острой необходимости лучше этого не делать) подключить соответствующие репозитории.

**Положительные особенности.** Низкие системные требования, будет работоспособна даже на типичном компьютере 15-летней давности с одноплатным процессором и 2 Гбайт оперативной памяти. Широкое распространение в госструктурах и крупных компаниях, так что в случае проблем можно обращаться к знакомым сотрудникам, например, «Сбербанка».

**Нейтральные особенности.** Использует пакетную базу Debian. По умолчанию использует репозитории разработчика, где выбор ПО в целом

достаточен. Большая часть разработчиков стороннего ПО Astra Linux поддерживает в первую очередь.

**Особенности, которые можно считать отрицательными.** Неприятный внешний вид. Скучные возможности по настройке. Не самые новые версии пакетов. Малопопулярен у домашних пользователей и в целом не лучший выбор для установки на личные ПК учащихся.

## Альт Образование

Дистрибутив от «Базальт СПО», ориентированный на сферу образования. В отличие от других дистрибутивов данного разработчика, может использоваться не только для организации рабочих мест, но и серверов.

**Положительные особенности.** Включает «из коробки» большой выбор ПО для сферы образования всех уровней, включая серверное. Самый большой российский репозиторий ПО. Подробная документация. Дистрибутив вполне можно предложить ученикам для установки личные ПК или ноутбуки.

**Нейтральные особенности.** Предлагается два графических рабочих стола: XFCE и KDE. Последний крайне не рекомендуется использовать на не самых новых компьютерах, а также там, где предполагается обучение младших школьников.

**Отрицательные особенности.** Установка ПО может вызвать сложности при незнании особенностей работы с пакетным менеджером дистрибутивов компании. По крайней мере, если необходимое ПО не удалось найти в репозитории или в виде flatpak.

## РедОС

Стоит отметить, что всё сказанное можно отнести и к дистрибутивам от НТЦ РОСА, которые используют ту же основу, что и РедОС. По большому счёту, эти дистрибутивы отличаются только темами оформления.

**Положительные особенности.** Основу составляет CentOS, для которого накоплен обширный багаж ПО. Может использоваться как для организации рабочих мест, так и для развёртывания серверов.

**Нейтральные особенности.** Возможности для настройки «под себя» сильно ограничены.

**Отрицательные особенности.** Процесс адаптации дистрибутива под нужды сферы образования находится на начальной стадии. Малый выбор ПО в официальных репозиториях. Утилиты для администрирования сложны и неудобны. Не самый привлекательный вариант для установки на личные компьютеры.

## Что предложить для личных ПК?

Из российских операционных систем, не входящих в Реестр российской ПО, можно предложить три варианта: Runtu/Runtu Lite, Simply Linux, Green Linux. Они лучше подходят для неподготовленных пользователей и имеют обширные репозитории ПО, которые локализованы в России. Runtu и Green Linux можно использовать тогда, когда в учебном заведении установлена Astra Linux. Simply Linux имеет общую базу с Альт Образование, есть немало общего и с РедОС.

**А. И. Федосеев**

Москва, Ассоциация участников технологических кружков

<https://foss.kruzhok.org>

## Всероссийский конкурс open source проектов: опыт проведения и перспективы

### Аннотация

С 2021 года в России проводится Всероссийский конкурс open source проектов школьников и студентов. Участниками его первого запуска стали более 400 команд из 68 регионов России. Школьники и студенты принимали участие в одном из двух треков: с собственными проектами или с контрибьюшенами в один из существующих проектов, опубликованных на платформах GitHub или GitLab. В качестве партнёров и экспертов конкурса приняли участие представители ведущих отечественных ИТ-компаний и FOSS-проектов. В 2022/23 году в конкурсе появятся новые номинации и тематические направления, но главное — будет запущен акселерационный этап для новых проектов и расширена образовательная программа для новичков. Открытым остаются вопросы системного включения вузов в конкурс, сочетание конкурса с существующими образовательными программами и практиками применения FOSS в вузе.

## Предпосылки запуска конкурса

В последние годы актуальность свободного и открытого программного обеспечения становится всё более заметной. Это можно увидеть по заявлениям профильных чиновников [1] и росту популярности свободных программ в школах и вузах [2]. Не менее важно и то, что открытые архитектуры, доступные свободные инструменты и языки программирования стано-

вятся де-факто основой современной культуры разработки программного обеспечения и высоких технологий [3].

Парадоксально, но одновременно с этим в отечественном образовании продолжает сохраняться консервативное отношение к свободному ПО как философии и принципу организации процесса разработки. Участие школьников и студентов в таких проектах является скорее исключением, чем распространённой практикой. Владение git или Python многими молодыми людьми не приводит к их включению в действующие сообщества разработчиков свободных программ. Ещё более серьёзной проблемой становится популярность у школьников и студентов поверхностного «стартаперства»: желания заменить красивыми презентациями реальные инженерные результаты.

Эти выводы сделаны по итогам более чем семилетнего опыта организации в России Национальной технологической олимпиады (ранее — Олимпиады НТИ) [4] и Кружкового движения Национальной технологической инициативы [5], участниками которого стало уже более 400 тысяч школьников и студентов: в рамках финалов олимпиады, хакатонов, проектных школ и кружков по всей стране молодые люди вместе с наставниками осваивают современные технологии и применяют их для решения учебных, соревновательных и реальных задач.

На наш взгляд, включение школьников и студентов в существующие сообщества разработчиков свободного ПО и запуск ими собственных проектов под свободной или открытой лицензией даёт несколько принципиальных возможностей:

- получение опыта разработки свободных программ и работы в профессиональных сообществах разработчиков;
- наличие в своём портфолио реального проекта, которым можно гордиться и использовать при устройстве на работу;
- деятельное знакомство с ценностями разработки свободного или открытого кода и решения реальных задач, личное проживание максимы «Show me the code».

## Конкурс проектов

Первый запуск Всероссийского конкурса open source проектов школьников и студентов состоялся с февраля по июнь 2022 года при поддержке Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации и Министерства высшего образования и науки. Заявки на конкурс подавались через платформу «Талант» Кружкового движения НТИ [6]. Участники конкурса предлагали на конкурс описания своих проек-

тов в привязке к коду, опубликованному на одной из популярных платформ GitHub или GitLab.

Участники могли выбрать для себя один из двух треков:

«**Создатели**» — проектные команды и индивидуальные участники, которые опубликовали свой проект;

«**Контрибьюторы**» — индивидуальные участники, которые предложили значимые улучшения к одному из популярных open source проектов.

Участие в Конкурсе не было ограничено строгой тематикой, но включало в себя специальных направлений от партнёров Конкурса, которыми стали ведущие технологические компании, такие как Яндекс, Сбер, Ростелеком, Positive Technologies, КРОК, Контур, лидеры в области отечественного open source — Базальт, PostgresPRO, CyberOK, Astra Linux, Открытая мобильная платформа, — а также институты развития и ведущие университеты — Фонд содействия инновациям, МГУ им. Ломоносова, НИУ ВШЭ, Университет ИТМО, МИЭТ и другие.

Оценка работ на Конкурсе проходила в 2 этапа: общественная и экспертная. Общественная оценка осуществлялась путём подсчёта голосов участников сообщества на платформах репозиториях GitHub или GitLab, её результаты выводились на лидерборд. Затем проекты оценивались профессиональным экспертным сообществом, представителями ведущих отечественных технологических компаний и популярных open source проектов. В критерии экспертной оценки входили не только качество кода, но и ценность разработки для других людей, востребованность проекта, следование распространённым практикам разработки и гайдлайнам.

Все участники, прошедшие этап экспертной оценки, получили также обратную связь от экспертов по дальнейшим шагам в развитии проекта или собственной профессиональной траектории.

## Итоги первого года

Участниками конкурса стало 429 человек: как индивидуальных участников, так и объединившихся в команды. В трек «Создатели» было подано 363 проекта, а в трек «Контрибьюторы» — только 20 заявок. Всего конкурс охватил представителей из 68 регионов страны. Среди наиболее активных регионов: г. Москва (81), г. Санкт-Петербург (48), Московская область (31), Республика Татарстан (24) и Новосибирская область (17). Средний возраст участников — 18 лет.

Нужно отметить значительный разрыв между работами лидеров и большей части поданных на конкурс работ: проекты победителей трека «Создатели» представляют собой достаточно серьёзные и интересные публичные



проекты [8, 9, 10, 11, 12], тогда как в «хвосте» оказались проекты, представляющие собой просто загруженный код учебного проекта на GitHub. В треке «Контрибьюторы» оказалось намного меньше участников и, соответственно, конкуренция оказалась серьёзнее [13].

Победители и призёры конкурса были награждены призами, в том числе старшекласники получили дополнительные баллы ЕГЭ для поступления в вузы.

## Код для всех

По итогам отборочного этапа конкурса В мае 2022 года в ситуации недоступности для разработчиков и менторов из России программы «Google Summer of Code» возникла инициатива по запуску аналогичной отечественной программы стажировок для участников open source сообщества под названием «Код для всех» [7].

Программа продолжила Всероссийский конкурс open source проектов, на участие в ней было подано 540 заявок в том числе от активных участников конкурса. В итоге для участия в стажировках было отобрано 22 человека. Участники программы включились в существующие проекты под руководством менторов из технологических компаний. Партнёрами программы выступили: PostgresPro, Belsoft, CyberOK и Яндекс. Стажёры получили не только ценный опыт, но и стипендию или вознаграждение от партнёров программы.

## Конкурс в 2022/23 учебном году

В 2022/23 учебном году конкурс стартовал второй раз. Среди нововведений конкурса добавление отдельного этапа акселерации проектов, обновление образовательной программы, сопровождающей конкурс и добавление третьего трека — внедрение open source решений. Подготовительный этап для начинающих разработчиков стартовал 23 декабря, а приём заявок на конкурс начнётся 30 января 2023 года.

Одним из вызовов для развития конкурса является отсутствие системной работы вузов с разработкой свободного и открытого ПО, что приводит к низкой популярности конкурса и малой распространённости open source разработки среди студентов. Возможным решением проблемы может стать появление специализированного курса для вузов, посвящённого культуре open source разработки, тесно связанного с готовым циклом конкурса. Другим инструментом популяризации может стать поддержка свободного ПО со стороны студенческих сообществ, молодёжных конструкторских бюро и клубов свободного ПО в вузах. Так или иначе, масштабное распространение культуры свободного ПО в России станет возможным только в сочетании

таких мероприятий как Всероссийский конкурс и системной работы в образовательных организациях.

## Литература

- [1] Стратегию развития ПО с открытым кодом в РФ обещали разработать к сентябрю — <https://www.interfax.ru/russia/770426>
- [2] Внедрение российских ОС в школы и вузы ускорилось в 6 раз — [https://www.cnews.ru/news/top/2022-08-16\\_vnedrenie\\_glavnoj\\_rossijskoj](https://www.cnews.ru/news/top/2022-08-16_vnedrenie_glavnoj_rossijskoj)
- [3] Nadia Eghbal, Roads and bridges. The Unseen labor behind our digital infrastructure, 2016
- [4] Национальная технологическая олимпиада — <https://ntcontest.ru>
- [5] Кружковое движение Национальной технологической инициативы — <https://kruzhok.org>
- [6] Сайт Всероссийского конкурса open source проектов школьников и студентов — <https://foss.kruzhok.org>
- [7] Сайт программы «Код для всех» — <https://foss.kruzhok.org/code-for-all>
- [8] Веб-приложение, собственный клиент для IT-площадки Хабрахабр — <https://github.com/jarvis394/geekr>
- [9] Вики-движок — <https://github.com/bouncepaw/mycorrhiza>
- [10] Инструмент для автоматического удаления фона с фотографий — <https://github.com/OPHoperHP0/image-background-remove-tool>
- [11] Утилита для взаимодействия с Android-телефонами — <https://github.com/fast-geek/Android-Tool>
- [12] Программа для поиска схожих участков кода в исходных кодах на основе сравнения абстрактных синтаксических деревьев — <https://github.com/OSLL/code-plagiarism>
- [13] Вклад победителя конкурса в проект cython — <https://github.com/cython/cython/pulls/0dminnimda>

Евгений Ковалев

Москва, Московский педагогический государственный университет (МПГУ)

## **О необходимости изучения студентами методических аспектов внедрения свободного программного обеспечения и механизмов импортозамещения**

### Аннотация

В статье рассматривается необходимость изучения студентами ИТ-направлений бакалавриата (на примере направления «Прикладная информатика») методики исследования предметных областей для обоснования выбора программных решений в области СПО. Автор приводит методические аспекты изучения механизмов внедрения, направленные на импортозамещение прикладного ПО, моделирования бизнес-процессов в различных предметных областях. Полученные прикладные компетенции студенты могут реализовать в исследовательских проектах и курсовых работах

В настоящее время актуальной задачей при обучении студентов ИТ-направлений, а также педагогических направлений, связанных с преподаванием информатики, является реализация принципов импортозамещения в сфере программного обеспечения. Задачи, которые нужно решать будущим специалистам, лежат в области обоснования и выбора программного обеспечения, проведения анализа изменений бизнес-процессов, которые влечёт переход на новое программное обеспечение, адаптации сотрудников к нему и повышение эффективности деятельности предприятия. Результатом такой учебной работы будет исследование виртуального предприятия и анализ программного обеспечения, которое требуется ему для замещения импортного ПО.

Основой для такого предприятия может быть:

1. Виртуальное предприятие выбранной сферы производства (пищевая, мебельная промышленность, строительство и т.п.)

2. Собственный стартап (в области ИТ или сферы услуг) и ведение его предполагаемой деятельности. Для этого на учебной практике, предшествующей изучению специальных курсов, студенты изучают основы бизнес-планирования и программные средства управления проектами.

При этом основной задачей может быть получение и накопление в общем репозитории компетенций в области внедрения свободного программного обеспечения. Среди видов работ, которые могут проводить при исследовании студенты: проведение исследований бизнес-процессов и выявление потребностей в специализированной ПО, создание базы знаний по типовым

бизнес-процессам, связанным с внедрением свободного ПО, создание базы данных по проектам внедрения, развитие компетенций для поддержки и развития свободного ПО, расчёт стоимости перехода на свободное ПО.

Методика проведения исследования при выполнении задания на производственной практике предполагает:

- исследование базовых понятий в области экономики и права, необходимых для функционирования предприятия;
- исследование выбранной предметной области;
- выделение процессов и функций, подлежащих автоматизации средствами информационных систем;
- информационное моделирование в методологии IDEF0 или DFD, построение моделей «КАК ЕСТЬ» и «КАК БУДЕТ»;
- формирование требований к конфигурации и платформе информационной системы предприятия;
- анализ работы и формирование отчётных документов.

Модель перехода на свободное ПО:

- Анализ бизнес-процессов в предложенной предметной области, выделение ключевых целей и задач ПО;
- Моделирование бизнес-процессов в известных нотациях (BPMN);
- Проект внедрения ПО;
- Расчёт стоимости владения ПО;

В дальнейшем при подготовке курсовой работы или выпускной квалификационной работы (диплома) по результатам проведённого исследования студентам предлагается разработать собственную конфигурацию программного решения. При этом полученные компетенции в области экономических знаний и правовых норм помогут выполнить грамотное теоретическое обоснование и описание предметной области предприятия, что зачастую является проблемой для ряда студентов.

При продолжении обучения в магистратуре, например, на базе дисциплин магистерской программы 090403 — Прикладная информатика, магистерская программа «Информационное моделирование бизнес-процессов и социально-экономических систем» можно предложить студентам исследование и систематизацию рынка свободного ПО.

Примерные критерии анализа ПО:

- находится в реестре отечественного ПО;
- работает на отечественных программных и аппаратных платформах;
- оценка транзакционных издержек при внедрении;

- наличие (объём) рынка труда, анализ потребностей в специалистах;
- цифровой двойник специалиста и развитие необходимых компетенций для подготовки к внедрению и использованию ПО.

В качестве итогового исследования можно предложить тематику разработки проектов и управление ИТ-проектами в области импортозамещения ПО, разработки типовых моделей перехода на свободное ПО и реализация моделей импортозамещения.

Апробация проекта внедрения может проходить на производственной практике в компании-партнёре программы и оцениваться внешними экспертами, которые привлекаются для независимой оценки достижений, полученных на практике.

Как итоговый результат методики – осознанное выполнение заданий по автоматизации учётной политики предприятия и подготовка к самостоятельному открытию предприятия «с нуля» и основам ведения бизнеса. Немаловажным является и то, что большой процент студентов, на основе выполненных таким образом заданий, продолжают разрабатывать в рамках выпускных квалификационных работ готовое программное решение. Оно может быть внедрено и представлено потенциальному работодателю в качестве своего портфолио.

## Литература

- [1] Брукс, Ф., Мифический человеко-месяц, или как создаются программные системы. <http://www.lib.ru/CTOTOR/BRUKS/mithsoftware.txt>
- [2] Ковалёв Е. Е., Ковалёва Н. А. Формирование профессиональных компетенций бакалавров направления «Прикладная информатика» при реализации дистанционного обучения с использованием программных разработок на платформе «1С:Предприятие». Информатика и образование. 2021. № 2 (321). С. 55–61.
- [3] Ковалёв Е. Е., Ковалёва Н. А. Разработка системной модели достижения индикаторов профессиональных компетенций при обучении бакалавров направления «Прикладная информатика». // Актуальные проблемы обучения математике и информатике в школе и вузе. Материалы V международной заочной научной конференции. Под общей редакцией Л. И. Боженковой, М. В. Егуповой. 2020. С. 362–368.

Алексеев Е. Р., канд. тех. наук, доцент, Грушевский С. П.,  
доктор пед. наук., профессор, Дога К. В., студентка  
Краснодар, Кубанский Государственный Университет  
math.kubsu.ru, 1aer.ru

## Использование дистрибутива Альт Образование при информационной подготовке будущих учителей математики и информатики в Кубанском Государственном Университете

### Аннотация

Представлен опыт использования дистрибутива Альт Образование в бакалавриате на факультете математики и информатики Кубанского Государственного Университета при обучении будущих учителей математики и информатики

В 2022 году Кубанский Государственный Университет (КубГУ) и компания «Базальт СПО» заключили договор о сотрудничестве, в рамках которого Кубанскому Государственному Университету было предоставлено право использовать 50 экземпляров дистрибутива Альт Образование 10.

Начиная с 2022–23 уч. года дистрибутив Альт Образование был установлен в компьютерных классах факультета математики и компьютерных наук. С первого сентября 2022 года началось его полноценное использование в учебном процессе.

Кафедра информационных образовательных технологий (ИТО) выпускает учителей математики и информатики (направление подготовки — 44.03.05 Педагогическое образование (с двумя профилями подготовки), направленность — Математика, информатика). Был перестроен весь учебный процесс для бакалавров набора 2022 года. Информационная подготовка будущих учителей начинается с дисциплин:

1. Программирование (4 семестра, 1–2 курс бакалавриата, 4 экзамена)
2. Программное обеспечение ЭВМ (2 семестра, 1 курс бакалавриата, зачёт, экзамен)

При разработке новой учебной программы было учтено, что студенты не сдавали ЕГЭ по информатике. В связи с этим очень важна роль курса «Программное обеспечение ЭВМ», в котором будущие учителя приобретают навыки работы с современными операционными системами, офисным программным обеспечением и различными программами для работы Сети Интернет.

Первый семестр этого курса посвящён получению базовых навыков работы на компьютере под управлением ОС Альт Линукс (Альт Образование). Студенты изучают особенности файловой системы ОС семейства Linux, права доступа к файлам и каталогам, работу в терминале, файловый менеджер mc. В качестве зачётного задания предлагается решить задачу по управлению файлами и каталогами в терминале.

В качестве методического учебного пособия студентам было предложено «Первое знакомство с ОС Linux» [1], которое получило положительные отзывы на объединённой конференции в Переславле-Залесском в мае 2022 года «Свободное программное обеспечение: от обучения до разработки 2022».

В этом же семестре студенты знакомятся с языком разметки Markdown. Документы в этом формате используются для оформления отчётов по лабораторным работам в курсе «Программирование». Знания, полученные в этом семестре, станут базой для дальнейшего освоения ОС семейства Linux и будут широко использоваться в курсе «Программирование».

Второй семестр курса «Программное обеспечение ЭВМ» целиком посвящён приобретению пользовательских навыков работы в ОС семейства Linux (Альт Образование). Студенты познакомятся с графическим интерфейсом пользователя в Linux на примере Альт Образования. Критерием сдачи этого блока курса является изучение курса «Альт Образование 10 (рабочий стол xfce). Курс для пользователей» (<https://kurs.basealt.ru/course/view.php?id=89>). Кроме того, студенты изучают прикладное программное обеспечение, которое является необходимым для учителей математики и информатики.

1. Графические программы для рисования блок-схем (afce, dia).
2. Язык разметки HTML, включая особенности представления математических формул в web-документах.
3. Язык разметки LaTeX.
4. Офисный пакет LibreOffice. Студенты изучают текстовый процессор LibreOffice Writer (включая работу с редактором формул LibreOffice Formula), программу работы с презентациями LibreOffice Impress. Отдельное внимание в этом блоке уделяется решению математических задач с помощью LibreOffice Calc).

В завершении этого курса выполняется лабораторная работа, посвящённая сравнительному анализу интерфейса и возможностей LibreOffice и Microsoft Office.

Курс «Программирование» посвящён изучению алгоритмики, программированию на языках «C(C++)» и Python. При этом ориентирован на работу под управлением ОС семейства Linux (Альт Образование). Первый

семестр посвящён изучению синтаксиса C(C++), алгоритмам разветвляющихся и циклических процессов. В качестве компилятора используется gcc/g++. В качестве среды программирования студентам предлагается текстовый редактор Geany. Второй семестр является самым сложным. Здесь студенты знакомятся с указателями, массивами, двойными указателями, матрицами и функциями. Изучаются особенности хранения данных в памяти компьютера. На реальных задачах мы рассматриваем то, как способы размещения элементов матриц в памяти компьютера влияют на быстрдействие программ. Кроме этого, в этом же семестре будущие учителя разрабатывают программы на C(C++) решения задач линейной алгебры.

В третьем семестре студенты более глубоко осваивают работу с файлами в C/C++, знакомятся с построением статических и динамических библиотек. В завершении семестра будущие бакалавры изучают особенности графического вывода информации в консольных кроссплатформенных приложениях на C/C++.

Начиная, со второго семестра планируется использование курса «Язык программирования Си в школьном курсе информатики: работа в Geany и Code::Blocks» (<https://kurs.basealt.ru/course/view.php?id=15>) в учебном процессе.

Четвёртый семестр посвящён изучению языка программирования Python. Студенты знакомятся с базовыми элементами языка, а также библиотеками numpy, scipy, matplotlib, sympy. Все студенты получают индивидуальное задание по разработке полноценного приложения математической направленности на Python 3. Отдельную проблему представляет вопрос выбора среды программирования для Python. В качестве IDE предлагается использовать Geany или IDLE. Однако разработка учебных математических приложений предъявляет особые требования к среде разработки. В итоге в качестве одной из сред разработки студентам будет предложен Jupyter Notebook, который кроме расчётных зон позволяет создавать зоны в формате Markdown, LaTeX. Кроме того, используя Jupyter Notebook, можно выводить результаты вычислений в графическом виде и виде формул в формате LaTeX. По существу Jupyter Notebook предоставляет будущему учителю с одной стороны среду для разработки программ на Python, а с другой — интерактивную среду для разработки электронных конспектов.

Параллельно с курсом «Программирование» на втором курсе изучается дисциплина «Математические пакеты и их применение в естественнонаучном образовании». Планируется, что студенты будут изучать пакеты Scilab (Octave) и язык программирования Julia. В качестве IDE для Julia предлагается Jupyter Notebook и Pluto.

Таким образом, планируется, что по завершении второго курса бакалавриата будущие учителя математики и информатики:

1. Станут квалифицированными пользователями ОС семейства Linux.



2. Изучат алгоритмику и программирование на языках C(C++) и Python3.
3. Приобретут навыки разработки электронных конспектов по математике и информатике с использованием Python и Julia.

Работа выполнена при финансовой поддержке Кубанского научного фонда в рамках научного проекта № ППН-21.1/10 «Цифровая дидактика для предметного обучения, воспитательной работы учащихся и профессиональной подготовки учителей».

## Литература

- [1] Алексеев, Е. Р. Учебное пособие «Первое знакомство с ОС Linux» / Е. Р. Алексеев, К. В. Дога, Ю. Н. Номоконова // Объединённая конференция «СПО: от обучения до разработки» : Сборник тезисов конференции, Переславль-Залесский, 19–22 мая 2022 года. — Москва: ООО «МАКС Пресс», 2022. — С. 53-56. — EDN FDDCWV.

Алексей Федосеев

Москва, Ассоциация кружков

Проект: Qt, ОС Аврора <https://www.qt.io>, <https://auroraos.ru>

## Совместные образовательные программы по основам open source разработки Кружкового движения и ОС Аврора

### Аннотация

В 2022 году стартовала совместная инициатива Кружкового движения и компании «Открытая мобильная платформа» для создания и апробации образовательных программ по разработке приложений с открытым исходным кодом для школьников и студентов, включая знакомство с современными инструментами и культурой открытой разработки, фреймворком Qt для создания переносимых приложений, дистрибутивами GNU/Linux, мобильной разработкой и особенностями ОС Аврора для мобильных устройств. Инициатива включает в себя создание набора семестровых программ, которые будут запущены в рамках кружков дополнительного образования, на уроках технологии и информатики в школе, в колледже или на младших курсах вуза. Программы позволят пройти путь от основ программирования, администрирования дистрибутивов GNU/Linux и веб-дизайна до создания

полноценных программных проектов в команде будущих специалистов. Набор программ рассчитан на двух-трёхлетнюю продолжительность и разные уровни подготовки участников. Весной 2023 года стартует апробация программ на базе площадок Кружкового движения, а с августа запланировано масштабирование в школах и кружках по всей стране. Доклад включает в себя общую логику программ, принципы их создания и последующего развития, возможности для участия общества.

## Введение

Низкий уровень культуры разработки свободного программного обеспечения среди школьников и студентов вызван в числе прочего с отсутствием доступных образовательных программ и методических материалов для наставников кружков, преподавателей школ и вузов по ключевым технологиям стека свободного ПО и связанных с ним практиками разработки. При наличии огромного объёма материалов по языкам программирования, веб-разработке или базам данных в опоре на открытый стек, сама суть и инструменты разработки свободного ПО остаются за редким исключением [1] не охваченными такими ресурсами — большая часть подобных тематик (ОС GNU/Linux, инструменты отладки, git и совместная разработка, создание переносимых приложений и т. п.) доступны в виде платных курсов для профессионалов и начинающих разработчиков [5, 6] или в англоязычных учебных материалах [7, 8]. Более того, курсы и материалы, созданные для самообучения профессионалов, зачастую требуют серьёзной доработки и методического оснащения при переводе в программы для среднего и высшего образования.

В связи с большой потребностью в энтузиастах и будущих специалистах в области разработки свободного ПО в 2022 году стартовала совместная инициатива Кружкового движения Национальной технологической инициативы [2] и компании «Открытая мобильная платформа» [3] для создания и последующей апробации образовательных программ по open source для школьников и студентов, включая знакомство с современными инструментами и культурой открытой разработки, фреймворком Qt для создания переносимых приложений, дистрибутивами GNU/Linux, мобильной разработкой и особенностями ОС Аврора для мобильных устройств.

За основу был взят опыт ведущих кружков и сообществ Кружкового движения и команды Открытой мобильной платформы по обучению школьников и студентов основам разработки программ. В октябре 2022 года в Москве был организован семинар по обучению разработке на базе open source инструментов для ОС Аврора, участниками которого стали представители ведущих кружков дополнительного образования, реали-

зующих программы по близким темам: Академии цифровых технологий (г. Санкт-Петербург), Иркутского государственного университета, Кванториума (г. Томск), Регионального школьного технопарка (г. Астрахань), Школьного акселератора «ЮнИТ Урал» (г. Екатеринбург), и др. В рамках семинара ведущие методисты обсудили содержание и форматы недостающих образовательных программ, поделились опытом включения школьников и студентов в работу с open source инструментами, а также существующими наработками в программах для школьников, которые могут быть применены при обучении будущих профессионалов более старшего возраста. Итогом обсуждения стал следующий проект набора семестровых программ.

## Общая архитектура проекта

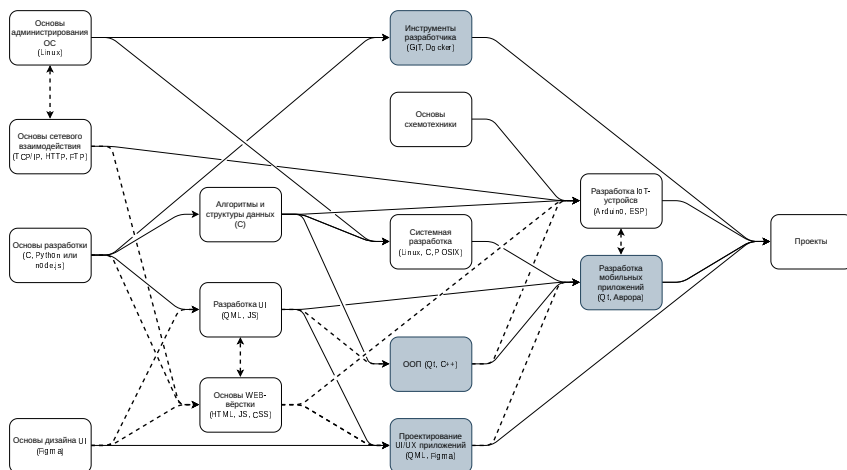
Проект состоит из связанного набора образовательных программ, рассчитанных на широкую аудиторию от старших классов школы до студентов младших курсов. Программы будут запущены в рамках кружков дополнительного образования, на уроках технологии и информатики в школе, в колледже или на младших курсах вуза. Набор программ рассчитан на двух-трёхлетнюю продолжительность и должен учитывать различные уровни подготовки участников. За основу был взят модульный принцип — проект собирается из семестровых курсов по 72 а.ч.

Программы позволят пройти путь от основ программирования, администрирования дистрибутивов GNU/Linux и веб-дизайна до создания полноценных программных проектов в команде будущих специалистов. Ядром инициативы станет стек разработки под мобильную ОС Аврора, включающий множество популярных open source инструментов, таких как git, gcc/g++ и низкоуровневое программирование под Linux, Qt/QML, инструменты для разработки под ОС Аврора.

На схеме представлен набор программ в виде графа зависимостей по требованиям к знаниям и навыкам обучающихся. Зелёным цветом выделены программы, которые будут разработаны и апробированы в первую очередь — в весеннем семестре 2023 года.

Методические материалы программ будут опубликованы под свободной лицензией, чтобы сообщество наставников и преподавателей могло участвовать в их последующей доработке.

В качестве иллюстрации рассмотрим один из разрабатываемых курсов — по теме введения в культуру разработки свободного и открытого ПО. Задача этого курса состоит не только в том, чтобы познакомить будущих разработчиков с инструментами для разработки, тестирования и отладки, совместной работы с кодом, сборки проектов и т. п., но также дать элементы культуры создания свободных программ. Важной дополнительной задачей



данного курса станет его применение в рамках Всероссийского конкурса open source проектов школьников и студентов, который в 2023 году будет проводиться второй раз [4]. Для участников конкурса прохождение курса станет важным дополнением к подготовке собственных проектов.

В программу данного модуля будут включены такие темы как:

- Основы свободного ПО, базовые определения и принципы организации. Обзор лицензий и типов проектов для которых они подходят. Стадии жизни проекта и инструменты для их достижения.
- Система контроля версий git. Знакомство с репозиториями и платформами размещения исходного кода программ на примере GitHub и GitLab. Чтение и использование в своем проекте кода из git-репозиторияев.
- Стандарты документирования кода (markdown, man, doxygen и др.). Оформление кода. Виды стилей. Средства автоматической генерации документации.
- Знакомство с рабочим окружением. Path пути к системному окружению. Создание виртуального окружения для проекта.
- Системы сборки и конфигурирования: make, qmake, cmake, autotools.

## Планы апробации и внедрения

Проект предполагает две стадии апробации разработанных программ:

1. Весной 2023 года на базе площадок Кружкового движения пройдёт первичная апробация разработанных программ.
2. На 2023/24 учебный год запланировано масштабное пилотирование программ в школах и кружках по всей стране, в рамках мероприятий Кружкового движения и при подготовке к Национальной технологической олимпиаде.

Итогом пилотирования программ и их внедрения в образовательные организации ожидается повышение вовлечённости преподавателей, школьников и студентов в создание собственных свободных/открытых приложений, в том числе для ОС Аврора. Это начинание будет поддержано за счет Всероссийского конкурса open source проектов школьников и студентов и других инструментов Кружкового движения.

## Литература

- [1] Books:Main page — ALT Linux Wiki, [электронный ресурс] [https://www.altlinux.org/Books:Main\\_page](https://www.altlinux.org/Books:Main_page), дата посещения 10.01.2023
- [2] Главная страница | Кружковое движение [электронный ресурс] <https://kruzhok.org>, дата посещения 10.01.2023
- [3] ОС Аврора | Открытая мобильная платформа [электронный ресурс] <https://www.omp.ru/os-aurora> дата посещения 10.01.2023
- [4] Всероссийский конкурс open source проектов школьников и студентов [электронный ресурс] <https://foss.kruzhok.org>, дата посещения 10.01.2023
- [5] Курсы и вебинары Свободно-распространяемое ПО (Open Source) в Специалисте [электронный ресурс] <https://www.specialist.ru/vendor/spo>, дата посещения 10.01.2023
- [6] Обучение DevOps | Университет Иннополис, [электронный ресурс] <https://team-project.university.innopolis.ru/devops>, дата посещения 10.01.2023
- [7] Open Source Software Development, Linux and Git | Coursera, [электронный ресурс] <https://www.coursera.org/specializations/oss-development-linux-git>, дата посещения 10.01.2023
- [8] Introduction to Git and GitHub | Coursera, [электронный ресурс] <https://www.coursera.org/learn/introduction-git-github>, дата посещения 10.01.2023

Кирилл Чувиллин

Москва, ООО «Открытая мобильная платформа»

Проект: Qt, ОС Аврора <https://www.qt.io>, <https://auroraos.ru>

## Обучение школьников кроссплатформенной разработке на основе Qt и Qt Quick

### Аннотация

Qt — это мультиплатформенный фреймворк, предоставляющий широкий набор библиотек и инструментов разработки. Многие дистрибутивы Linux используют Qt как основной фреймворк, в том числе российские операционные системы для ПК, а также мобильная ОС Аврора, разрабатываемая компанией «Открытая мобильная платформа». Обучение разработке на Qt охватывает множество аспектов создания приложений, которые достаточно подробно рассматриваются в студенческом курсе, разработанном нашей компанией. При этом часть фреймворка, обеспеченная технологией Qt Quick и языком QML, с одной стороны позволяет создавать функциональные приложения, а с другой — достаточно компактна для освоения школьниками. В докладе представляются технические аспекты Qt Quick, которые удобно использовать для обучения школьников разработке мобильных и десктопных приложений.

### Введение

Компания «Открытая мобильная платформа» (ОМП) [4] является разработчиком средств, позволяющих выстроить доверенную инфраструктуру для работы с мобильными устройствами. Функционирование такой инфраструктуры обеспечивается, в том числе, операционной системой Аврора и платформой управления устройствами Аврора Центр. ОС Аврора [1] — это российский дистрибутив Linux, предназначенный для использования на смартфонах и планшетах корпоративными заказчиками и государственными компаниями. В совокупности технологии разработки для ОС Аврора сосредоточены вокруг ядра Linux, POSIX-совместимых интерфейсов [5], а также фреймворка Qt [6].

Со стороны ОМП реализован ряд мер, нацеленных на обеспечение качественной и быстрой разработки ПО. Он включает поставку инструментов разработки и документации, организацию технической поддержки и консалтинга по разработке, предоставление примеров приложений [12], а также проведение учебных мероприятий [11]. Среди сотрудников ОМП есть

специалисты по разработке ПО, которые также имеют опыт преподавания в вузах. Совместно с партнёрами нами были подготовлены материалы для проведения учебных курсов в вузах и организациях дополнительного образования по системной разработке и разработке прикладного ПО для ОС Аврора.

В 2022 году мы также получили ряд запросов от образовательных организаций, работающих со школьниками, на разработку и внедрение новых учебных материалов по разработке для среднего образования. В том числе в октябре мы провели семинар совместно с Кружковым движением Национальной технологической инициативы [2], на котором с методистами обсудили содержание и форматы недостающих образовательных программ, а также участие ОМП в подготовке методических материалов.

Результатом обсуждения стал набор компетенций на направлениях разработки и дизайна интерфейса, которыми полезно обладать школьникам, интересующимся IT-специальностями, и соответствующих программ, среди которых:

- Основы администрирования ОС;
- Основы сетевого взаимодействия (TCP/IP, HTTP, FTP);
- Основы разработки (C, Python или node.js);
- Основы дизайна UI (Figma);
- Алгоритмы и структуры данных (C);
- Разработка UI (QML, JS);
- Основы WEB-вёрстки (HTML, JS, CSS);
- Инструменты разработчика (GIT, Docker);
- Системная разработка (Linux, C, POSIX);
- ООП (Qt, C++);
- Проектирование UI/UX приложений (QML, Figma);
- Разработка мобильных приложений (Qt, Аврора).

В докладе акцент сделан на кроссплатформенной разработке прикладного ПО, основой которой в данном контексте являются технологии Qt, а также языки программирования C++ и QML.

## Технологии Qt для школьников

Qt — это мультиплатформенный фреймворк, предоставляющий широкий набор библиотек и инструментов разработки ПО [6]. Примеры программ, написанных на Qt: KDE, Skype, Telegram, Virtual Box, Google Earth и др. Многие дистрибутивы Linux используют Qt как основной фреймворк,

в том числе Ubuntu, Fedora, Red Hat и отечественные Заря, ROSA, ALT Linux. Не исключением является и ОС Аврора, разрабатываемая компанией «Открытая мобильная платформа».

Технологии, которые используются при разработке на Qt, довольно обширны и включают как различные подходы к шаблонам проектирования (предлагая концепцию сигналов-слотов, а также механизм делегатов для работы с представлениями), так и API для взаимодействия с различными подсистемами и периферией.

В основе фреймворка Qt лежат парадигма объектно-ориентированного программирования, язык C++ и метаобъектный компилятор. Его удобно использовать как наглядный пример в курсах по C++ или ООП. Кроме того, есть успешный опыт использования Qt для обучения алгоритмическому программированию и работе с графикой [10].

Но фреймворк Qt также предоставляет технологию Qt Quick [8], которая, с одной стороны, обладает достаточно низким порогом входа, доступным школьникам, а с другой — позволяет создавать полноценные приложения. Кроме того, она на практике знакомит с концепцией декларативного программирования, которая сейчас активно используется при создании мобильных приложений.

В докладе рассматриваются технические аспекты, которые важны для формирования методических материалов учебных программ.

## Программа курса по Qt Quick

В этом разделе приведён вариант программы для обучения школьников, который охватывает все важные аспекты разработки приложений с использованием технологии Qt Quick.

1. Основы Qt Quick.
  - a) Знакомство с фреймворком Qt и технологией Qt Quick.
  - b) Инструменты разработки.
  - c) Структура проекта.
  - d) Знакомство с языком QML.
  - e) Основные визуальные типы.
  - f) Позиционирование элементов.
  - g) Обработка событий.
  - h) Определения новых свойств объектов.
  - i) Взаимодействие с пользователем.
2. Расширенные возможности Qt Quick.
  - a) Подключение JS-библиотек.



- b) Трансформации и состояния.
  - c) Анимации.
  - d) Работа с компонентами.
  - e) Модели.
  - f) Представления.
  - g) Многопоточность.
3. Организация UI приложений.
- a) Библиотеки системных компонентов.
  - b) Навигация в приложениях.
  - c) Работа с текстом.
  - d) Элементы управления.
  - e) Меню.
  - f) Масштабирование приложений.
  - g) Стилизация приложений.
  - h) Локализация приложений.
4. Использование системных API.
- a) Работа с файлами.
  - b) Мультимедиа.
  - c) Камера.
  - d) Работа с HTTP.
  - e) Web-сокеты.
  - f) Датчики.
  - g) Геолокация.
  - h) Работа с базами данных.
  - i) Взаимодействие приложений.
  - j) QML-плагины ОС Аврора.

## Литература

- [1] Аврора — первая российская мобильная операционная система, [электронный ресурс] <https://auroraos.ru>, дата посещения 19.04.2022
- [2] Сертификаты, патенты и лицензии ОС Аврора, [электронный ресурс] <https://auroraos.ru/certificates>, дата посещения 19.04.2022
- [3] Каталог приложения для ОС Аврора, [электронный ресурс] <https://auroraos.ru/applications>, дата посещения 19.04.2022

- [4] Открытая мобильная платформа — официальный сайт, [электронный ресурс] <https://www.omp.ru>, дата посещения 19.04.2022
- [5] PosixP1003.1 - Standard for Information Technology–Portable Operating System Interface (POSIX(TM)) Base Specifications, Issue 8, [электронный ресурс] [https://standards.ieee.org/project/1003\\_1.html](https://standards.ieee.org/project/1003_1.html), дата посещения 19.04.2022
- [6] Qt | Cross-platform software development for embedded & desktop, [электронный ресурс] <https://qt.io>, дата посещения 19.04.2022
- [7] Stack Overflow - Where Developers Learn, Share, & Build Careers, [электронный ресурс] <https://stackoverflow.com>, дата посещения 19.04.2022
- [8] QML Applications | Qt 5.15, [электронный ресурс] <https://doc.qt.io/qt-5/qmlapplications.html>, дата посещения 19.04.2022
- [9] Nash, John, "Non-cooperative Games "Annals of Mathematics y. 1951, vol. 54-2 pp.286–295
- [10] Артём Проскурнёв, Проект «qtSimpleGraph». Работа с графическими примитивами, СПО: от обучения до разработки. Объединённая конференция. Переславль-Залесский, 15–18 июня 2021 г. : сборник тезисов конференции, –М. ООО «МАКС Пресс», 2021 г. С. 100–103.
- [11] Кирилл Чувиллин, Обучение разработке на Qt для мобильных устройств. Что нужно студентам и профессионалам, СПО: от обучения до разработки. Объединённая конференция. Переславль-Залесский, 15–18 июня 2021 г. : сборник тезисов конференции, –М. ООО «МАКС Пресс», 2021 г. С. 89–90.
- [12] Кирилл Чувиллин, Проекты с открытым исходным кодом как основа передачи компетенций разработчикам приложений для ОС Аврора, СПО: от обучения до разработки. Объединённая конференция. Переславль-Залесский, 19–22 мая 2022 г. : сборник тезисов конференции, –М. ООО «МАКС Пресс», 2022 г. С. 158–161

Валерий Баканов

Москва, МИРЭА, НИУ ВШЭ

<http://vbakanov.ru/dataflow/>, <http://vbakanov.ru/spf@home/>

## Практикум по изучению скрытого в алгоритмах параллелизма и его рационального использования в вычислениях

### Аннотация

Использование параллелизма при вычислениях в настоящее время является основным методом повышения производительности при обработке данных [1]. План реиндустриализации России включает разработку новых микропроцессоров, в частности систем ЭЛЬБРУС, реализующих принцип выполнения машинных команд с явным параллелизмом (EPIC, *Explicitly Parallel Instruction Computing*) методом применения сверхдлинного машинного слова (VLW, *Very Long Instruction Word*).

Расширение использования процессоров ЭЛЬБРУС в России потребует разработки Инструментального Программного Обеспечения (ИПО) разнообразного применения (компиляторы и интерпретаторы с разных языков программирования, системы поддержки и отладки и др.). Для будущих разработчиков ценно как понимание сути самого свойства параллелизма в алгоритмах (программах) и методов его нахождения и, что особенно важно, рационального использования в вычислениях. Именно для этого и создан данный ПРАКТИКУМ, в настоящее время используемый в нескольких Университетах России.

Рассматриваемый ПРАКТИКУМ фактически является комплексом авторских программ-симуляторов разных этапов создания параллельных программ - от анализа алгоритмов на наличие в них внутреннего (скрытого) параллелизма и его параметров до разработки рациональных методов эквивалентных преобразований алгоритмов с целью максимально эффективного применения их (найденного на предыдущем этапе исследований) потенциала параллелизма при вычислениях. Конечной целью применения ПРАКТИКУМА является построение стремящихся к оптимальным логических планов (каркасов) выполнения программ на параллельных вычислительных системах с заданными параметрами. Входящие в ПРАКТИКУМ программные системы фактически являются исследовательскими инструментами (*исследовательской платформой*) класса OpenSource<sup>1</sup> и предла-

---

<sup>1</sup><https://github.com/Valery-Bakanov>

гаются для свободного распространения (и проистекающего из этого дальнейшего совершенствования) совместно с инструкциями пользования, методическими разработками, планами исследований и др. при обучении.

Исходной сущностью для ПРАКТИКУМА является алгоритм [2] и его свойства со стороны внутренне присущего атрибута параллелизма. Собственно алгоритм представляется в императивном стиле с применением ассемблероподобного языка (при этом явная информация о последовательности выполнения операторов и наличии параллелизма отсутствует) с порядком следования операндов согласно соглашениям AT&T, условность выполнения частей программы реализована методом предикатов [3]. Для выявления параллелизма использована агентная модель (входящая в ПРАКТИКУМ программа D-F, реализующая выполнение операторов программы в порядке готовности к выполнению всех операндов данного оператора) и метод построения и управляемого целенаправленного преобразования специальных сечений (ЯПФ, *Ярусно-Параллельная Форма*) информационного графа алгоритма (программа SPF@home). Программные модули ПРАКТИКУМА разработаны с использованием языка C/C++ в стиле GUI (для пакетной обработки реализована работа с командной строкой)<sup>2</sup>.

В общем случае задача получения оптимального (по заданным критериям) плана (расписания) параллельного выполнения конкретного алгоритма (программы) является *NP*-полной [4] и не допускает прямого разрешения. Поэтому программная система SPF@home использует *эвристический подход* к решению поставленной задачи, заключающийся в эквивалентном (не изменяющем информационные связи операторов) целенаправленном изменении ЯПФ рассматриваемого алгоритма (программы).

В качестве реализации этой идеи применён метод разработки сценариев преобразования ЯПФ на скриптовом языке Lua [5]. Lua-сценарии являются языковым выражением эвристического подхода и предполагают итерационный характер их совершенствования (что позволяет постепенно улучшать их качество в отношении получения более близкого к оптимальности результата).

Усложнённое применение ПРАКТИКУМА даёт возможность решать оптимизационные задачи (целевыми параметрами могут являться показатели качества получаемых планов параллельного выполнения программ и вычислительная сложность их получения [6] и др. параметры) при определённой конфигурации поля параллельных вычислителей.

ПРАКТИКУМ позволяет решать также обратные задачи - напр., определение рациональных параметров параллельной вычислительной системы

---

<sup>2</sup> [http://vbakanov.ru/dataflow/content/install\\_df.exe](http://vbakanov.ru/dataflow/content/install_df.exe) ,  
[http://vbakanov.ru/spf@home/content/install\\_spf.exe](http://vbakanov.ru/spf@home/content/install_spf.exe)

исходя из баланса производительности и стоимости собственно системы для заданного алгоритма (класса алгоритмов).

Данный ПРАКТИКУМ и методики (приёмы выявления скрытого параллелизма и его параметров в произвольных алгоритмах, способы построения рациональных планов выполнения параллельных программ на заданном поле вычислителей) ряд лет применяются при обучении студентов в указанных университетах России и позволили повысить компетенции учащихся в области параллельной обработки данных.

## Литература

- [1] Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2004. — 608 с.
- [2] AlgoWiki. Открытая энциклопедия свойств алгоритмов. Под ред.: Воеводин В., Донгарра Дж. URL: <http://algowiki-project.org> (дата обращения: 5.1.2023).
- [3] Баканов В.М. Управление динамикой вычислений в процессорах поточковой архитектуры для различных типов алгоритмов. // Журнал "Программная инженерия", — М.: 2015, № 9, с. 20-24.
- [4] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, Книга по Требованию, 2012. — 420 с.
- [5] Иерусалимски Роберту. Программирование на языке Lua. — М.: ДМК Пресс, 2014. — 382 с.
- [6] Баканов В.М. Вычислительная сложность построения рациональных планов выполнения программ на заданном поле параллельных вычислителей. *Russian Technological Journal*. 2022; 10(6):7–19. DOI: <https://doi.org/10.32362/2500-316X-2022-10-6-7-19>

Павловский Владимир Владимирович  
Москва, РГУ нефти и газа (НИУ) имени И.М. Губкина

## **Взаимодействие высшей и средней школы при внедрении СПО в общеобразовательные учреждения**

В области информационных технологий требует в настоящее время сочетания детальной проработки, применения современных методических средств, а также возможностей быстрой адаптации к меняющимся требованиям отрасли.

Значительной проблемой в ходе организации обучения является создание методических материалов для применения изученного материала на практике, которые позволили бы закрепить полученные знания, сформировать необходимые требуемые образовательной программой компетенции и создать мотивацию к получению новых знаний и к качественному их усваиванию [1].

На примере лабораторного практикума «Технические средства информатизации: практикум для СПО», разработанного на базе отечественного свободного программного обеспечения, прошедшего апробацию в УРТК им. А.С. Попова, ГБПОУ МИК и РГУ нефти и газа (НИУ) имени И.М. Губкина можно сформулировать тезис, что большинство интеграторов и сервисных центров применяют методики, пришедшие от западных компаний HP, Dell, Cisco, Super Micro и т.д. Что в текущих реалиях является не актуальным. Разработанный практикум может быть использован преподавателями и студентами профессиональных образовательных организаций для облегчения входа в профессию. Лабораторный практикум может быть использован при освоении дополнительных образовательных программ повышения квалификации педагогических работников образовательных организаций, реализующих программы среднего профессионального образования из числа преподавателей и мастеров производственного обучения [2].

Лабораторный практикум был разработан в целях повышения уровня умений и знаний в области профессиональной деятельности:

- эксплуатация и обслуживание серверного и сетевого оборудования;
- диагностика и мониторинг работоспособности программно-технических средств;
- обеспечение целостности резервирования информации и информационной безопасности объектов сетевой инфраструктуры;
- организация и проведение работ по обеспечению защиты автоматизированных и телекоммуникационных систем в организациях различных структур и отраслевой направленности;

- эксплуатация, техническое обслуживание, сопровождение и настройка компьютерных систем и комплексов;
- обеспечение функционирования программных и аппаратных средств защиты информации в компьютерных системах и комплексах.

Структурно лабораторный практикум представляет собой комплект лабораторных работ, включающий описание задания; необходимое оборудование, приборы, материалы и программное обеспечение для выполнения задания; методику выполнения задания; сетевые диаграммы; схему оценивания выполненного задания, являющегося одним из вариантов заданий, выполняемых в процессе прохождения промежуточной аттестации, включая квалификационные экзамены по профессиональным модулям. Основой комплекса является демонстрация функционала на классических Windows или проприетарных решениях, и демонстрация как эти же работы могут быть выполнены средствами свободного программного обеспечения. В качестве системного программного обеспечения выбран дистрибутив Simply Linux [3], применяемый как в режиме LiveCD так и в установленном варианте [4].

Для каждого задания лабораторных работ имеются пошаговые инструкции выполнения элементов заданий, пояснения выполняемых действий и объяснения последствий неправильных действий и приёмов; скриншоты экрана; ссылки на электронные ресурсы для получения необходимой информации и инструкций для выполнения заданий.

В результате выполнения заданий по лабораторным работам, обучающиеся смогут систематизировать и закрепить знания и умения, а преподаватели смогут оценить уровень подготовки обучающихся к самостоятельной работе и уровень практических навыков для участия в сдаче не только промежуточной аттестации, но и итоговой государственной аттестации.

Таким образом, мы приближаемся к решению основной задачи — сделать методические материалы гибкими, информативными, целостными и актуальными.

В настоящий момент комплекс готовится к изданию в издательстве Издательский центр «IPR MEDIA» в 2023 году. Технические средства информатизации : практикум для СПО / А. Г. Уймин. — Москва : Ай Пи Ар Медиа ; Саратов : Профобразование, 2023. — (Среднее профессиональное образование). — Текст : электронный. Ай Пи Ар Медиа ISBN 978-5-4497-2023-8. Профобразование ISBN 978-5-4488-1589-8

Получена положительная рецензия кафедры информатики и защиты информации Владимирского государственного университета им. А.Г. и Н.Г. Столетовых.

## Литература

- [1] Трунилова Вера Николаевна, Роль практических занятий в повышении качества освоения дисциплины «Технические средства информатизации» // Известия Великолукской государственной сельскохозяйственной академии. 2017. № 5.
- [2] Уймин, А. Г. Применение отечественного программного обеспечения для перестройки образовательного процесса вуза в рамках подготовки кадров цифровизации производства / А. Г. Уймин // Цифровая трансформация промышленности: новые горизонты : Сборник научных трудов по материалам 3-й Всероссийской научно-практической конференции, Москва, 10 ноября 2022 года. — Москва: Общество с ограниченной ответственностью «Русайнс», 2022. — С. 398–405.
- [3] Уймин, А. Г. Инструментальные средства обучения компьютерным сетям. Развёртывание на базе российского программного обеспечения / А. Г. Уймин, Г. И. Токарев // Системы управления и информационные технологии. — 2022. — № 4(90). — С. 88-92. — DOI 10.36622/VSTU.2022.90.4.019.
- [4] Simply Linux 10.1 Документация. Руководство пользователя <https://docs.altlinux.org/ru-RU/alt-simply/10.1/html/alt-simply/index.html>

Диденко Денис Владимирович

Щёлково, Государственное бюджетное профессиональное образовательное учреждение Московской области «Щёлковский колледж»

## Использование шаблона текстового процессора LibreOffice Writer для подготовки ВКР в колледже

### Аннотация

В докладе рассмотрен опыт внедрения и успешного применения при подготовке выпускных письменных квалификационных работ свободного программного обеспечения «Libreoffice Writer» в ГБПОУ МО «Щёлковский колледж»

## Общие сведения

В ГБПОУ МО «Щёлковский колледж», структурном подразделении №5 и 6 в процессе внедрения в работу продуктов из набора свободного



программного обеспечения остро встал вопрос о работе над ВКР и ВПКР (выпускная квалификационная работа и выпускная письменная квалификационная работа).

В принципе, данный вопрос стоит остро всегда, вне зависимости от того каким офисным программным обеспечением пользуются студенты.

В данном случае, я посчитал, что данное решение послужит тем толчком, который сподвигнет студентов выбрать для работы именно свободный офисный редактор? постольку поскольку стили документа — сильная сторона LO.

Кроме того, в структурном подразделении №5 ГБПОУ МО «Щелковский колледж» ведутся занятия с обучающимися 8-х и 9-х классов школ Щёлковского муниципального района по программе «Путевка в жизнь» (это некий аналог советской системы Учебно-производственного комбината). Данная программа, позволяет получить профессию «Оператор ВМ и ВМ», кроме этого ряд других профессий начального профессионального образования.

В своей работе я использую настроенный совместно со студентами шаблон LibreOffice Writer с расширением .ott. Работа с ним реализуется исключительно с использованием стилей и обычное форматирование документа практически не используется.

Отсутствие сегрегации по моему мнению — основное достоинство свободного программного обеспечения. Я считаю, что каждый студент вне зависимости от достатка должен иметь равные возможности по сравнению с товарищами по реализации учебных возможностей, посредством программного обеспечения. И учебное заведение должно в обязательном порядке выставлять требования такого характера, которые будут способствовать реализации возможностей каждого, а не ограничивать их.

Конечно, здесь есть над чем ещё работать и путь к поставленной цели тернист и долог. Например, в требованиях спущенных свыше указан проприетарный шрифт Times New Roman Правообладателем шрифта является расположенная в США компания Monotype Imaging Inc. Все это порождает множество вопросов, относительно целесообразности данного решения.

## Стилевая структура шаблона

Шаблон состоит из настроенных стилей (Заголовок 1, заголовок 2 и заголовок 3), стилей подписей изображений и стилей содержимого таблицы, настроенных номеров страниц (настроенных через расширение Pager) и оглавления. Все эти настройки по нашему мнению в достаточной мере отражают возможности реализации требований к выпускной работе.

## Титульный лист

Это наверное единственная область, где мы отказались от использования стилей, так как некоторые пользователи упорно пытаются оформить некоторые вещи стилями заголовков, что неминуемо ведёт к попаданию фрагментов титульного листа в оглавление.

## Оглавление

Данная область не требует традиционного редактирования и тем более удаления.

Оглавление изменяется 2-мя щелчками мыши (ПКМ — обновить указатель — ЛКМ) каждый раз по завершении редактирования документа.

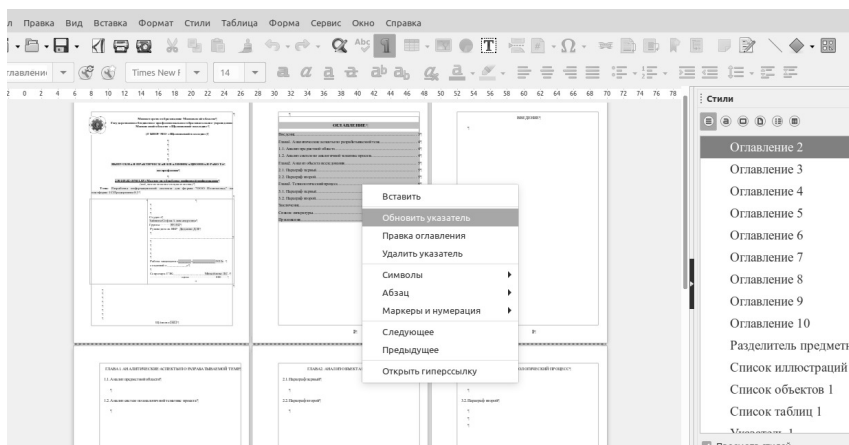


Рис. 1: Работа с оглавлением (не удаляем).

## Основные разделы ВКПР

Основные разделы ВКПР у нас это:

1. Введение
2. Основная часть (сам заголовок не печатается, но раздел состоит из глав и параграфов)
3. Заключение

4. Список литературы

5. Приложения

Разделы оформляются заголовком 1 уровня.

## Главы

Формируют структуру основной части, идут с автоматической нумерацией (ГЛАВА 1, ГЛАВА 2 и т. д.). Главы оформляются заголовком 2 уровня.

## Параграфы

Формируют структуру глав основной части, также идут с автоматической нумерацией, которая подчиняется главам. Параграфы оформляются заголовком 3 уровня.

## Подписи к изображениям

По умолчанию они оформлены стилем «подпись» и соответственно в шаблоне данный стиль приведён в соответствии требованиям.

## Содержимое таблицы

Данный стиль приведён в соответствии требованиям к работе и оформление любой таблицы или приведение к необходимому виду готового объекта занимает 2-3 щелчка мышью и выбором стиля «Содержимое таблицы».

## Этапы работы с шаблоном

### *Копирование*

На наш взгляд, вставку текста и иных объектов лучше делать поэтапно чтобы избежать какие-либо неприятные шероховатости, а именно копировать из иных мест текст отдельно от изображений и вставлять в режиме «только текст». Изображения же мы вставляем как обычно.

### *Сброс стилей*

После набора текста всей работы, изображений и таблиц имеет смысл выделить всё от введения и до конца, выбрав пункт в окне выбора стиля (очистить форматирование) — таким образом вся работа за исключением оглавления и титульного листа сбросится на стиль «базовый», который настроен для основного текста работы.

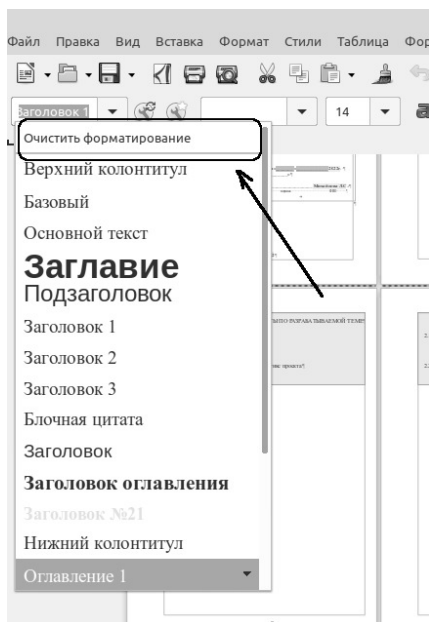


Рис. 2: Сброс стилей.

### *Стили заголовков*

Далее, мы отмечаем разделы (введение, заключение, список литературы и приложения) стилем «Заголовок 1». Главы, которые подчиняются разделам и сосредоточены в основной части работы — отмечаются стилем «Заголовок 2». Параграфы же, которые являются подзаголовками глав — отмечаются как «Заголовок 3».

### *Работа с изображениями*

Подписи к изображениям программа сама отмечает стилем «Название». Отмечаем отсутствие обтекания и формируем подпись к рисунку, выбрав категорию «Рисунок»

### *Работа с таблицами*

Готовую таблицу выделив всю и отметив стилем «Содержимое таблицы» мы приведём сразу к виду который соответствует требованиям.

### *Завершающий этап*

По окончании всех операций, необходимо вернуться к оглавлению и нажав на него ПКМ выбрать пункт контекстного меню «обновить указа-

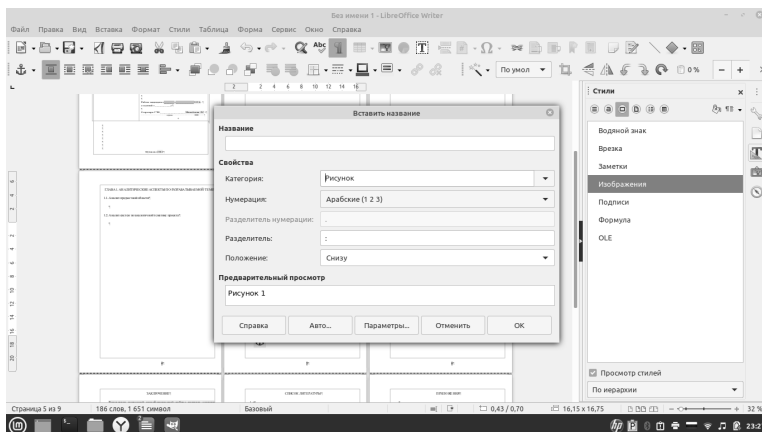


Рис. 3: Вставка названия.

тель» — этим пунктом нужно завершать все правки работы, для того, чтобы в него занеслись новые пункты и обновились номера страниц.

## Выводы

По результатам апробации используемого при подготовке к защите ВПКР шаблона документа Libreoffice Writer был сделан однозначный вывод, что процесс подготовки ВПКР довольно успешно реализуется в среде офисного свободного программного обеспечения Libreoffice Writer.

Оформление работы в данном случае занимает немного времени и самое главное, последующие возможные правки могут занимать считанные минуты.

Воронин Игорь Вадимович, Воронин Ростислав Игоревич  
Шатура, Москва, ИПЛИТ РАН, МГТУ им. Баумана (студент)

## Настройка домашнего помощника Home Assistant в среде Альт Линукс

### Аннотация

**Home Assistant** – это одна из самых популярных систем организации, автоматизации и управления устройствами умного дома и интернета вещей от большого количества различных брендов. Система написана на Python и имеет открытый исходный код. В докладе разбираются пути установки системы с минимальными затратами сил и времени, протоколы связи и адаптация подключения разнообразных устройств и исполнительных механизмов, используемых в быту, и на производстве.

Каждый из нас, живёт в своём доме. И все мы хотим чтобы наша жизнь была удобной и комфортной. Для этого разработаны и используются всечелские домашние помощники, например такое популярное приложение с открытым исходным кодом для организации умного дома — Home Assistant.

Это ПО в Альт Линукс р10 уже можно установить из репозитория командой:

```
#apt-get install hass-core
```

После установки и запуска в системе будет доступен процесс статус которого можно посмотреть командой:

```
# systemctl status hass
```

К нему можно присоединиться из браузера по порту 8123, например так: <http://192.168.0.132:8123/>

Проект развивается трудом большого количества энтузиастов из открытого сообщества, он отлично подходит для работы на контроллере Raspberry Pi или OrangePi ну и конечно его можно развернуть на локальном сервере. С помощью этого приложения можно управлять множеством устройств в своём доме прямо с мобильного телефона, наблюдать за их состоянием, или запрограммировать их работу по расписанию. Например вы можете включать свет только в тех комнатах где кто-то из людей сейчас находится, чтобы экономить электроэнергию и не жечь лампочки впустую. Запускать работу робота пылесоса только тогда когда ни кого нет в доме, чтобы он прибирался в ваше отсутствие. Открывать-закрывать шторы жалюзи на окна, оберегая тем самым мебель от яркого солнца. Используя умную розетку вы можете не переживать — выключили ли вы утюг выходя из дома, поскольку вам всегда доступно дистанционное выключение

этой розетки используя мобильный телефон. А так же можете посмотреть количество энергии прошедшей за выбранный период через эту розетку.

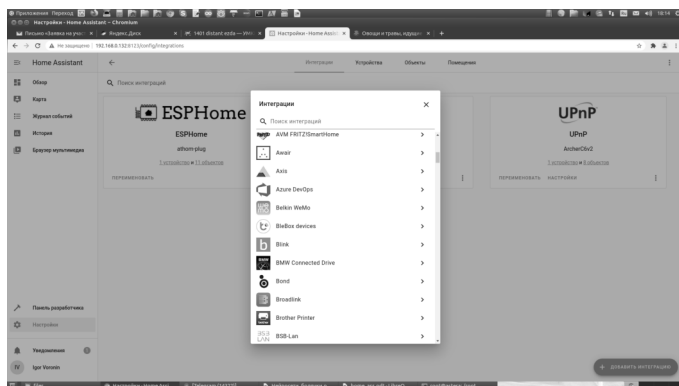
Если вдруг система водоснабжения даст протечку, то датчики диагностики воды сразу оповещает вас о нештатной ситуации с водопроводом, а сама система автоматически перекроет кран на входе в дом — чтобы не залить соседей снизу, и не допустить утечку воды.

Если у вас есть солнечные панели на крыше вашего дома, то Домашний ассистент — сможет управлять накоплением солнечной энергии. Home Assistant помогает вам вносить вклад в борьбу с климатическим кризисом. Одним из элементов борьбы с климатическим кризисом является обеспечение того, чтобы наши дома были энергоэффективными и использовали низкоуглеродные источники энергии. Home Assistant помогает контролировать потребление электроэнергии, и тем самым, экономить деньги.

Так же такие роботизированные системы позволяют управлять и контролировать работой котла отопления, тогда можно настроить режим работы системы отопления в максимально комфортном режиме зимой и с максимальной экономией газа.

А ещё можно с помощью Home Assistant в выращивать в гараже под домом или в подсобном помещении разнообразные овощи и травы употребляемые в пищу зимой — контролируя при этом расход электро энергии на освещение, тепла на отопление и состав раствора ЕС, таким образом, чтобы растения развивались в наиболее благоприятной им среде.

Для связи с нужными объектами в системе предусмотрена интеграция с различными устройствами и объектами. Это могут быть самые разнообразные решения от различных поставщиков



В случае если вы сами разработали устройство и хотите его интегрировать в свой Home Assistant, то это можно осуществить используя интеграцию ESP Home, правда в этом случае ваше устройство — должно быть создано с использованием контроллера семейства ESP, чтобы оно могло связаться с сервером где установлен сам Home Assistant по wifi или Ethernet. Тогда при настройке этой связи надо будет всего лишь указать IP адрес — полученный от роутера, порт и прописать API ключ к этому устройству, для его аутентификации.

Так же для связи с устройствами доступны протоколы mqtt. В этом случае публикуется сообщение mqtt.publish, в топике и после отправки сообщения его можно посмотреть mqtt\_client-ом, получить и обработать необходимым сценарием. Для работы таких сценариев — предусмотрены разнообразные режимы автоматизации в Home Assistant по событиям — триггерам или по условиям, которые задаются пользователями.

## Литература

- [1] <https://packages.altlinux.org/en/p10/srpms/hass/>
- [2] <https://www.home-assistant.io/>
- [3] <http://umkikit.ru/>> Поддержка >> Умный контроллер котла SmartTherm>

Сергей Мартишин, Марина Храпченко

Москва, Институт системного программирования им. В.П. Иванникова РАН

Проект: Название проекта(ов) URL проекта

## Реализация агрегированного графического запроса конфиденциальных данных из облачной базы данных с использованием языков Go и R

### Аннотация

При работе с конфиденциальными данными на облаке необходимо обеспечить невозможность их раскрытия на этапе запроса к облаку, ответа облака, визуализации данных. Симметрические запросы, призванные получить статистические данные, являются дедуктивно безопасными. Для реализации агрегированного графического запроса предлагается использовать языки Go и R. Простота установки, наличие значительного количества пакетов делает их удобными для разработки приложений.



В настоящее время облачные сервисы стали неотъемлемой частью работы с информацией. Помимо удобства доступа и экономии на оборудовании, программном обеспечении и техподдержке, это обеспечивает возможность организации совместной работы с хранящимися на облаке данными.

В научных проектах при работе с данными на облаке следует учитывать, что данные могут быть конфиденциальными, работа с которыми должна быть построена в соответствии с определёнными правилами. Например, медицинские данные содержат персональную информацию, анализ которой представляет интерес для учёных и практикующих врачей. Очевидно, что работа с такими данными должна осуществляться не только в соответствии с правами доступа. Также необходимо обеспечить невозможность раскрытия конфиденциальной информации на этапе запроса к облаку, ответа облака, визуализации данных.

В предыдущих работах авторы исследовали модель облачных вычислений над конфиденциальными данными [1]. В [2] было сформулировано определение дедуктивной безопасности: клиент базы данных, получив ответы на все доступные ему запросы, не может получить дедуктивно (в виде логического следствия) конфиденциальные данные пользователей. Также в [2] было доказано, что симметрические запросы являются дедуктивно безопасными, поскольку призваны собирать лишь статистические данные, но не раскрывают никакой персональной информации [3]. Решением является хранение конфиденциальных данных на защищённых серверах, а агрегированная информация, полученная с помощью дедуктивно безопасных запросов, может обрабатываться на облаке для её последующего использования.

Также ранее был исследован вопрос представления полученных агрегированных данных в виде, удобном для восприятия человеком [4]. Для анализа статистической информации и её визуализации широко применяется язык R [4]. Интерпретируемый объектно-ориентированный язык R является СПО (свободное программное обеспечение) с открытым исходным кодом в рамках проекта GNU.

К достоинствам языка R относится простота установки. Под Linux установка выполняется стандартной командой: `# dnf install R`. Также для работы с R существуют графические пользовательские интерфейсы. Например, среда разработки RStudio [5], распространяемая под свободной лицензией GNU GP. Для её установки следует выполнить команды:

```
# dnf install rstudio-desktop
# dnf install rstudio-server
```

Для визуализации информации языком R на вход подаётся уже агрегированные данные в стандартном формате, например, JSON. Эту информацию можно передавать на облако для её визуализации. Основные статистические методы реализованы в качестве стандартных функций R.

Подход, изложенный в [6] требует использования сервера Apache. Предлагаемое ниже использование языка Go не требует установки стороннего сервера. Создание приложений на Go подробно рассмотрены авторами в [7]. К достоинствам языка также относится поддержка многопоточности и параллельного программирования.

Установка Go на Linux происходит в режиме администратора:

```
# dnf install golang
```

В дальнейшем работа осуществляется не в режиме администратора (приглашение #), а в режиме пользователя (приглашение \$). Запуск программы на Go осуществляется с помощью команды:

```
$ go run имя_файла.go
```

Программа на Go использует пакеты. Для объявления пакета используется ключевое слово `package`. Исполнение программы начинается с запуска пакета с именем `main`, который должен содержать функцию `main()`, являющейся входной точкой в приложение.

Go предлагает широкий спектр пакетов, наиболее часто используются пакеты [8]:

- `fmt` — пакет, содержащий функции ввода/вывода,
- `io/ioutil` — пакет, предоставляющий базовые интерфейсы и функции ввода/вывода,
- `log` — пакет ведения журнала, записывающий стандартную ошибку и печатающий дату и время каждого зарегистрированного сообщения,
- `net/http` — пакет служит для выполнения клиент-серверных функций HTTP. `encoding/base64` пакет, реализующий кодировку `base64`, как указано в RFC 4648.
- `net/url` — пакет, анализирует URL-адреса и реализует экранирование запросов.

При наличии готовых пакетов с нужной функциональностью, они могут быть импортированы в программу с помощью оператора `import` [8].

Однако в тех случаях, когда код использует внешние пакеты, эти пакеты (распространяемые как модули) становятся зависимостями [8]. Чтобы отслеживать добавляемые зависимости и управлять ими, необходимо поместить код в директорию проекта в отдельный модуль и сделать директорию проекта активной. Далее, чтобы добавить зависимости в собственный модуль следует выполнить команду `$ go mod init "путь_к_модулю"` [7].

Для обработки запросов в сети Интернет Go чаще всего использует функцию пакета `http`:

```
HandleFunc(pattern string, handler func(ResponseWriter, *Request)).
```

Функция `http.HandleFunc()` сообщает веб-серверу, что необходимо выполнить функцию `func(ResponseWriter, *Request)` (второй параметр), которая будет обрабатывать запрос, для пути (`pattern string`). В свою очередь,

функция `func(ResponseWriter, *Request)` также принимает два параметра: `ResponseWriter` — поток ответа и `*Request` — информацию о запросе [8].

Наличие встроенного сервера, большого количества пакетов и удобства обработки запросов, делает применение языка Go простым и эффективным в научных проектах для создания запросов конфиденциальных данных из облачной базы данных. Язык R обеспечивает визуализацию в сети Интернет агрегированных данных, не раскрывая конфиденциальную информацию. Базовый код проекта представлен в <https://github.com/otd13isp/Pereslav12023>. В файле `readme.txt` проекта даны подробные комментарии по установке СПО и порядке выполнения команд для создания проекта.

## Литература

- [1] Варновский Н. П., Мартишин С. А., Храпченко М. В., Шокуров А. В. *Пороговые системы гомоморфного шифрования и защита информации в облачных вычислениях*, Программирование. 2015. № 4. С. 47–51.
- [2] Варновский Н. П., Захаров В. А., Шокуров А. В., *О дедуктивной безопасности запросов к базам конфиденциальных данных в системе облачных вычислений // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика*. 2017. № 1. С. 38а-44.
- [3] Мартишин С. А., Храпченко М. В. *Организация облачных вычислений над конфиденциальными данными на СПО. // Материалы конференции / АНО «Национальный суперкомпьютерный форум», Институт Программных Систем РАН, Институт Логики, Базальт СПО. — Переславль, 7–9 февраля 2020 года / отв. Ред. Чёрный В.Л. — М.: МАКС Пресс, 2020. с. 171–174*
- [4] The R Project for Statistical Computing [Электронный ресурс] — Режим доступа: <https://www.r-project.org>, свободный. Яз. англ. — Дата обращения: 11.03.2022.
- [5] RStudio [Электронный ресурс] — Режим доступа: <https://www.rstudio.com>, свободный. Яз. англ. — Дата обращения: 11.03.2022.
- [6] Мартишин С. А., Храпченко М. В. *Визуализация агрегированных конфиденциальных данных в научных проектах в сети Интернет с использованием языка R. // Объединённая конференция «СПО: от обучения до разработки»: материалы конференции / Переславль-Залесский, 19–22 мая 2022 г. / отв. ред. Чёрный В. Л. — М.: МАКС Пресс, 2022. — 208 с. ISBN 978-5-317-06790-8. с. 201–205*
- [7] Мартишин С. А. // *Базы данных: проектирование и разработка информационных систем с использованием СУБД MySQL и языка Go :*

учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ИНФРА-М, 2022. — 306 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1830834. ISBN 978-5-16-017213-2 (print), ISBN 978-5-16-106422-1 (online).

- [8] Go [Электронный ресурс] — Режим доступа: <https://go.dev/>, свободный. Яз. англ. — Дата обращения: 15.12.2022.

---

Научное издание  
Восемнадцатая конференция.  
Свободное программное обеспечение в высшей школе

Оформление обложки: *А.С.Осмоловская*  
Верстка: *В.Л.Чёрный*  
Редактура: *В.Л.Чёрный*

Издательство ООО «МАКС Пресс»  
Лицензия ИД N 00510 от 01.12.99 г.

Отпечатано с готового оригинал-макета  
Подписано в печать 23.01.2023 г.  
Формат 60x90 1/16. Усл. печ. л. 8,25.  
Тираж 170 экз. Изд. № 001.

119992, ГСП-2, Москва, Ленинские горы,  
МГУ им. М.В. Ломоносова, 2-й учебный корпус, 527 к  
Тел. 8(495)939-3890/91. Тел./Факс 8(495)939-3891

Отпечатано в полном соответствии с качеством  
предоставленных материалов в ООО «Фотоэксперт»  
109316, г. Москва, Волгоградский проспект, д. 42,  
корп. 5, эт. 1, пом. I, ком. 6.3-23Н