

АНО «Институт логики, когнитологии и развития личности»
ООО «Базальт СПО»
Институт Программных Систем РАН

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

Переславль, 25–27 января 2019 года

Сборник тезисов конференции

Москва,
МАКС Пресс,
2019
ISBN 978-5-317-06037-4

УДК 004.91:378
ББК 32.97:74.48
Ч-54

Программный комитет:

А. Е. Новодворский — председатель,
А. А. Савченко,
Г. В. Курячий,
Т. Н. Губина

Четырнадцатая конференция «Свободное программное обеспечение в высшей школе»: Материалы конференции / Переславль, 25–27 января 2019 года. / отв. ред. Черный В. Л. — М.: МАКС Пресс, 2019. — 144 с. : ил.

ISBN 978-5-317-06037-4

В книге собраны тезисы конференции, одобренные Программным комитетом четырнадцатой конференции «Свободное программное обеспечение в высшей школе».

Издательство ООО «МАКС Пресс»
Лицензия ИД N 00510 от 01.12.99 г.
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)939-3893
Изд. номер № 003

ISBN 978-5-317-06037-4

© Коллектив авторов, 2019

Программа конференции

25 января, пятница

11.30-14.00 Заселение, обед, регистрация участников

13:30 Автобус от гостиницы Переславль

Дневное заседание 14.00–18.30

14.00 А. В. Смирнов. Открытие. Информация оргкомитета

14.10–14.40 В. А. Сухомлин и др.

Методические основы создания системы развития
цифровых навыков на базе платформы Альт 8

14.40–15.10 Н. Н. Непейвода

Третье издание «Прикладной логики» 10

15.10–15.40 А. Г. Кушниренко и др.

Основные понятия программирования в полуторогодовом
курсе для дошкольников и первоклассников 11

15.40–16.10 Г. В. Курячий

Дистанционное online-преподавание
компьютерно-ориентированных дисциплин 14

16.10–16.30 Кофе-пауза

16.30–16.50 Е. Р. Алексеев

Использование свободных программ на кафедре
информационных образовательных технологий
Кубанского Государственного Университета 15

16.50–17.10	М. Райко и др. Для кого и как можно проводить олимпиады по кооперативному программированию	17
17.10–17.30	И. А. Хахаев Возможности совершенствования образовательного дистрибутива АЛЪТ	22
17.30–17.50	Т. Н. Губина Методическая поддержка педагогов по переходу на российские операционные системы семейства Альт: направления работы, достигнутые результаты, планы на будущее	25
17.50–18.10	С. А. Голубев Свободный проект: работа со СМИ	29
18.10–18.30	А. В. Федорчук Linux для гуманитариев	32

26 января, суббота

09.30 Автобус от гостиницы

Утреннее заседание 10.00–13.00

10.00–10.20	В. Л. Симонов, А. В. Ерпелев и др. Использование свободного ПО в учебном процессе на примере разработки устройств «умного дома» с применением микроконтроллеров Arduino и Iskra JS...	34
10.20–10.40	М. В. Нидилько и др. Разработка проекта на микроконтроллерной плате.....	36
10.40–11.00	И. В. Воронин, В. В. Воронина От основ к созданию роботов. Знакомство с робототехникой в школе	38
11.00–11.20	Д. А. Костюк и др. Построение практикумов по изучению архитектуры и периферийных устройств на основе шины USB	43
11.20–11.40	Кофе-пауза	

11.40–12.00	Р. А. Симаков	
	Использование СУБД Ред База Данных в МИВлГУ	49
12.00–12.20	И. Е. Панченко и др.	
	PostgreSQL в высшем образовании	51
12.20–12.40	А. Г. Михеев	
	Проект RunaWFE Free. Реализация транзакций для курса процессного управления предприятием	53
12.40–13.00	М. О. Петрова	
	Чего мы ждём от специалистов, проходящих сертификацию, и как им к этому подготовиться	58
13:00	Автобус в гостиницу	
13.00–15.00	Перерыв на обед	
14:40	Автобус от гостиницы	

Вечернее заседание

15.00–18.20

15.00–15.20	Н. О. Бешапошников и др.	
	Применение методов распознавания в практикумах по программированию	61
15.20–15.40	А. Г. Леонов, Н. О. Бешапошников и др.	
	Проект двуязыковой пиктограммно-текстовой учебной среды программирования ПиктоМир-К	64
15.40–16.00	М. В. Быков	
	diglossa.js — система подготовки и публикации дву-(много)-язычных текстов	66
16.00–16.20	А. А. Савченко	
	Взгляд изнутри на Google Summer of Code	70
16.20–16.40	Кофе-пауза	
16.40–17.10	С. М. Абрамов	
	Проект «Чуткий дом» как еще один пример тесной взаимной зависимости IT-науки, IT-бизнеса и IT-образования	74

17.10–17.40	А. С. Кремер	
	Обучение навыкам внедрения отечественных технических средств в национальную ИКТ-инфраструктуру	76
17.40–18.00	Р. Г. Ставцев	
	Лаборатория электроники Байкал. Программное и аппаратное окружение	80
18.00–18.20	С. А. Фомин	
	OBS — швейцарский нож передачи знаний. Боевые приемы Open Broadcaster Software	80
18.40	Автобус в гостиницу	

27 января, воскресенье

09.30 Автобус от гостиницы

Утреннее заседание

10.00–14.00

10.00–10.20	А. Н. Пустыгин, А. А. Ковалевский и др.	
	Применение программных инструментов с открытым исходным кодом для анализа исходных текстов программ	91
10.20–10.40	Л. Н. Чернышов, В. Н. Лукин	
	Дистанционное обучение программированию на Google-платформе	95
10.40–11.00	М. Каторгин	
	Использование свободного программного обеспечения в студенческом проекте метеостанции на базе платы Arduino MEGA	99
11.00–11.20	С. А. Мартишин, В. Л. Симонов, М. В. Храпченко	
	Использование свободного программного обеспечения HTML5 Boilerplate, Bootstrap, Initializr для обучения студентов технологиям создания клиентской части web-приложений	101
11.20–11.40	В. В. Лаптев и др.	
	Обучающая среда по программированию на базе СПО	106
11.40–12.00	Кофе-пауза	

12.00–12.20	Пикова А.В.Бойко Л.А.Симонов В.Л. Использование свободного программного обеспечения для проектирования устройств диагностирования психофизического состояния человека	112
12.20–12.40	М. В. Маркушевич Основные методические подходы к построению методики преподавания информатики в основной школе, разработанной на базе свободного программного обеспечения	115
12.40–13.00	А. Ю. Никитин, Н. Н. Никитина Использование офисных пакетов OpenOffice и Myoffice в учебном процессе в Калужском государственном университете им. К.Э.Циолковского	120
13.00–13.20	Н. Е. Говорухин Web хостинг на основе персонального компьютера, создание доменных имен третьего уровня .	123
13.20–13.40	В. А. Суховерхий Linux как решение проблемы устаревания компьютеров в учебном заведении	124
13.40–14.00	А. М. Андрианов Проектное командное обучение с использованием СПО	128
14.30	Автобус в Москву	

Вне программы

Т. О. Сундукова Роль свободного программного обеспечения в преподавании дисциплины «Технологии разработки баз данных» в вузе	134
С. А. Мартишин, М. В. Храпченко Изучение технологий распределенного кэширования данных на базе СПО в высоконагруженных системах . .	137
А. Крюков Выбор ПО для учебного процесса	141

Владимир Сухомлин, Елена Зубарева, Алексей Якушин
Москва, Московский государственный университет имени М.В. Ломоносова

Методические основы создания системы развития цифровых навыков на базе платформы Альт

Аннотация

В докладе представлены результаты исследования методических основ проекта создания системы развития цифровых навыков на базе платформы Альт и СПО, включая модель, классификацию, языки описания цифровых навыков; организацию фонда образовательных ресурсов; профиль метаданных и семантическую модель образовательного ресурса.

Одной из центральных проблем построения цифровой экономики является вопрос подготовки кадров с необходимыми цифровыми навыками и создания образовательных технологий, ориентированных на выявление спроса на востребованные общие и специализированные цифровые навыки и обучение этим навыкам [1, 2]. В интересах безопасности и конкурентоспособности государства также актуальной становится проблема импортозамещения в области информационных технологий с максимально широким охватом сфер деятельности. Решение этих проблем возможно на пути создания национальной системы развития цифровых навыков (СРЦН) с помощью фонда образовательных ресурсов (ОР), разработанных на базе платформы Альт и продуктов свободного программного обеспечения (СПО).

К числу основных задач, решаемых на базе СРЦН, относятся такие задачи, как выявление, анализ, классификация, спецификация, фондирование и сопровождение актуальных навыков по отраслям; создание фонда образовательных ресурсов (на базе системы Альт и СПО — Фонд ОР Альт) и образовательных технологий, ориентированных на развитие навыков; реализация образовательных процессов подготовки актуальных навыков и сертификацию.

На начальном этапе создании СРЦН или её компонент центральными становятся вопросы разработки концептуальных и методических решений, таких как: разработка концепции цифровых навыков, системы их классификации, языков спецификации навыков, организации фонда актуальных навыков, принципов построения фонда ОР

Альт, профиля/стандарта представления метаданных ОР, семантической модели ОР, определяющей дидактическое содержание ОР и т.п.

В интересах создания СРЦН авторами проведена исследовательская работа по разработке определённых выше методических аспектов и получены следующие основные результаты:

1. Разработаны: видение, миссия, цель создания и задачи СРЦН на базе фонда ОР Альт.
2. Разработана метамодель для описания цифровых навыков, отражающая функционал, динамику развития, многомерность данного понятия [3].
3. Предложен метод классификации и описания цифровых навыков в области ИКТ, основанный на использовании языка описания навыков SFIA7 (международный стандарт 2018 года [4, 5]), расширенном механизмом профилирования навыков.
4. Предложен метод описания навыков рабочего места посредством конкретизации базовых навыков или их профилей с помощью спецификаций требований к навыку, связанных с рабочим местом, временем использования, производственным контекстом.
5. Выполнен анализ и отбор систем классификации образовательных программ и ресурсов для использования в метаданных для фонда образовательных ресурсов.
6. На основе анализа международных образовательных стандартов в области ИТ [6] разработана концепция семантической модели описания дидактического содержания образовательных ресурсов/программ, на основе иерархического представления результатов обучения для заданного ОР.
7. На основе анализа международных стандартов метаданных для ОР [7] разработан базовый репозиторий метаданных для создания Фонда ОР Альт.
8. Разработана концептуальная модель и архитектура системы развития цифровых навыков, требования к функциональности основных её компонент.
9. Представлены примеры использования разработанных решений на примере ОР существующего репозитория ОР.

Литература

- [1] Ministerial Declaration on the Digital Economy («Cancún Declaration»), 2016. <http://www.oecd.org/sti/ieconomy/Digital-Economy-Ministerial-Declaration-2016.pdf>
- [2] Послание Президента Федеральному Собранию, 2016. <http://www.kremlin.ru/events/president/news/53379>
- [3] Сухомлин В.А., Зубарева Е.В., Якушин А.В. Методологические аспекты концепции цифровых навыков // Современные информационные технологии и ИТ-образование. 2017. Том 13, № 2. С. 146-152. DOI: 10.25559/SITITO.2017.2.253
- [4] Дрожжинов В.И. SFIA — система профессиональных стандартов в сфере ИТ эпохи цифровой экономики // Современные информационные технологии и ИТ-образование. 2017. Том 13, № 1. С. 132-143. DOI: 10.25559/SITITO.2017.1.466
- [5] The SFIA Foundation — SFIA. <https://www.sfia-online.org/en>
- [6] Сухомлин В.А. Международные образовательные стандарты в области информационных технологий // Прикладная информатика. 2012. № 1(37). С. 33–54. <https://elibrary.ru/item.asp?id=17363662>
- [7] Позднеев Б.М., Тихомирова В.Д. Стандартизация метаданных электронных образовательных ресурсов // Открытое образование. 2015. № 1(108). С. 55–59. <https://elibrary.ru/item.asp?id=23242192>

Н. Н. Непейвода

Переславль-Залесский, ИПС РАН

Третье издание «Прикладной логики»

Издательство «Директ-медиа» выпускает в 20189 году третье, переработанное и дополненное издание учебного пособия «Прикладная логика». Книга дополнена четвёртой частью, посвящённой приложениям логики прежде всего в информатике. В частности, в ней впервые в монографической литературе на русском языке (и, возможно, в мире) даётся описание языка доказательного программирования AGDA и советы по программированию на нём.

Издание будет свободно доступно в электронной форме студентам и преподавателям вузов, научным работникам.

Если первый опыт взаимодействия будет удачным, предполагается издание там же давно ожидаемого «Системного и логического анализа» и монографии «Уроки конструктивизма: анализ достижений и неудач».

В выступлении будет представлена презентация по этим книгам.

Н. О. Бесшапошников, А. Г. Кушниренко, А. Г. Леонов, М. В. Райко

г. Москва, ФГУ ФНЦ НИИСИ РАН

Проект: ПиктоМир <https://piktomir.ru/>

Основные понятия программирования в полуторагодовом курсе для дошкольников и первоклассников

Аннотация

В ходе доклада будет перечислен набор основных понятий и приёмов программирования, и методика их изложения в полуторагодовом курсе для дошкольников и первоклассников. Методика существенно использует современные методы распознавания изображений и сигналов.

Согласно действующему ФГОС[1], в системе образования РФ курс информатики является обязательным в 7—9 классах и в 10—11 классах. Одной из центральных тем курса информатики является программирование. Действующие федеральные документы предусматривают, что к концу 9 класса школьники должны выйти на невысокий, но и ненулевой уровень практического владения программированием. К сожалению, практика показывает, что этот уровень выходит лишь незначительная доля учащихся. Одной из причин этого факта является катастрофически поздний возраст знакомства с программированием, предусмотренный действующим ФГОС основной школы. Опыт работы отдела учебной информатики НИИСИ РАН показывает, что оптимальным для знакомства с базовыми понятиями программирования, является старший дошкольный возраст.

Мы разрабатываем *четырёхлетнюю программу изучения алгоритмики и программирования*, которая нацелена на достижение следующего измеримого результата: 100% обучаемых к моменту окончанию начальной школы должны выйти на уровень практического владения программированием, предусмотренный действующим сегодня ФГОС к окончанию 9 класса.

Достижение возможно более высоких *предметных* результатов по информатике и программированию не может быть единственной целью. По нашему опыту, раннее изучение программирования в деятельностно-игровой форме оказывается прекрасным инструментом для ранней выработки таких *метапредметных навыков*, как системное мышление, способность к выстраиванию длинных цепочек умозаключений, готовность к продолжительной концентрации усилий в одном направлении. Вместе с тем, достижение измеримых предметных навыков также важно. Именно поэтому мы выбрали в качестве измеримого предметного навыка один из центральных навыков обязательного курса информатики — навык практического программирования.

Полуторагодовой курс изучения основных понятия программирования дошкольниками и первоклассниками, обсуждаемый в докладе, запланирован, как первая вводная часть следующего четырёхлетнего цикла:

- одно занятие в неделю в течение учебного года в дошкольном учреждении,
- одно занятие в неделю в весеннем полугодии учебного года для первоклассников,
- одно занятие в неделю в течение учебного года для второклассников и третьеклассников.

Домашние задания в курсе не задаются, половина времени каждого занятия посвящается практической работе с компьютерами и учебными пособиями-роботами. На каждом занятии ребёнок выполняет не менее 4—5 практических заданий на компьютере (планшете). Задание выдаётся одному ребёнку, или команде из двух детей, каждый из которых работает на своём компьютере над своей частью задания.

Как показывает наш опыт работы с выпускниками МПГУ[2] в выравнивающем курсе «Азы программирования», существует базовый набор простейших понятий и приёмов программирования, без которого невозможно сознательное усвоение материала. Именно изучению

этого базового набора в максимально эффективной для старшего дошкольного и младшего школьного возраста и посвящён обсуждаемый курс.

Основные понятия программирования вводятся с помощью парадигмы программного управления исполнителями — роботами. Составляемые программы описывают процесс управления одним или несколькими исполнителями без обратной связи или с обратной связью.

В течение первого года обучения, за редкими исключениями составляются программы управления без обратной связи. На этом этапе используются две конструкции языка программирования — числовой повторитель и подпрограмма. Программы на первом году обучения составляются в пиктограммной форме: с помощью пиктограмм команд и конструкций языка, материализованных либо в виде материальных объектов (Пикто-Квадратиков и ПиктоКубиков), соединяемых ребёнком в программу на столе или магнитной доске, либо с помощью виртуальных пиктограмм, перемещаемых ребёнком на экране планшета с сенсорным экраном.

Третье полугодие курса посвящено программам с обратной связью и использованию чисел в программах управления роботами.

Бесценными для повышения скорости и качества усвоения материала на первых занятиях оказались два нововведения: распознавание и последующее выполнение компьютером программ, составленных из материальных объектов и управление роботом с помощью звуковых сигналов в режиме пульта и в режиме программного управления[3].

Эти нововведения будут продемонстрированы во время доклада.

В соответствии с поставленной целью — выйти на определённый уровень владения навыками практического программирования — на четвёртом году обучения должна использоваться какая-то традиционная текстовая система программирования, например, КуМир. Однако наш опыт показал, что переход от пиктограммного стиля программирования к текстовому оказывается для обучаемых трудным. Для облегчения этого перехода в НИИСИ РАН начата разработка смешанной пиктограммно-текстовой системы ПиктоМир-К[4], которой на настоящей конференции будет посвящён отдельный доклад.

Работа выполнена по теме 0065-2018-0017Госзадания 2018 года в отделе учебной информатики ФГУ ФНЦ НИИСИ РАН

Литература

- [1] Федеральный государственный образовательный стандарт основного общего образования [Электронный ресурс]: федер. стандарт от 17.12.2010 № 1897. URL: <https://rg.ru/2010/12/19/obrstandart-site-dok.html>
- [2] Кушниренко А. Г., Леонов А. Г., Райко М. В. Курс «Азы программирования» для будущих учителей и студентов естественных Факультетов [Электронный ресурс] / Презентация доклада на конференции 31.01.2016. URL: <https://docplayer.ru/36377633-Kursazy-programirovaniya-dlya-budushchih-uchiteley-i-studentov-estestvennyh-fakultetov.html>
- [3] Беспашников Н. О., Леонов А. Г., Прилипка А. А. Цифровизация образования — Новые возможности управления образовательными треками // Вестник Кибернетики, № 2. 2018. С. 157–163
- [4] Кушниренко А. Г., Леонов А. Г., Пиктограммно-текстовая система программирования ПиктоМир-К для дошкольников и младших школьников // Вестник кибернетики. 2017. № 4 (28). С. 167–172.

Георгий Курячий

Москва, ВМК МГУ им. М. В. Ломоносова

Дистанционное online-преподавание компьютерно-ориентированных дисциплин

Мои дистанционные и недистанционные занятия по Python со студентами (НПО Восход, Ереванский филиал МГУ, факультет Психологии МГУ) с недавних пор устроены так.

Все студенты сидят за специально оборудованными компьютерами. Используется сетевая загрузка, так что в другое время там хоть винда может быть. Преподаватель сидит за таким же.

Пол-экрана занимает т. н. «доска» — ещё один специальный рабочий стол некоторого компьютера (выделенного или преподавательского, неважно, главное, что он только пол-экрана занимает), к которому у всех участников процесса есть доступ по сети.

Изначально студенты могут только наблюдать за происходящим, а преподаватель может этим экраном управлять. Он открывает сайты,

редактор, пишет программы, отлаживает их и т. п., короче, проводит часть семинара. Студенты наблюдают и воспроизводят действия преподавателя. Если что-то не получается, им помогает ассистент (если он есть), или сам преподаватель.

Потом даётся задание (скажем, написать такую-то программу), и все, включая преподавателя, это задание решают. Каждый, кто решил, говорит «Я решил!». В какой-то момент преподавателю надоёдает ждать и последний решивший «выходит к доске» — получает доступ по управлению общим рабочим столом, показывает и отлаживает решение. Преподаватель помогает и исправляет недочёты.

И так до конца семинара.

В случае дистанционных занятий применяется аудио- и текстовый чат. В аудитории выделяется один компьютер, который выводится на колонки (вариант: каждый сидит в гарнитуре), преподаватель дистанционно использует гарнитуру. Для надёжного общения (формулировка задач, ответы на важные вопросы) используется групповой текстовый чат.

Коротко о технической стороне. Общий рабочий стол — `vncserver` из пакета `Tiger VNC`, клиент к нему — `vncviewer` из того же пакета или `remmina`; сервер чата — `murmur`, клиент к нему — `mumble` (есть отличный клиент для android — `plumble`), тематическая начинка для собственно занятий — по необходимости, платформа бездисковой загрузки — `ALT Server/ALT Workstation`.

Евгений Алексеев

Краснодар, Кубанский государственный университет

Использование свободных программ на кафедре информационных образовательных технологий Кубанского государственного университета

Аннотация

Представлен опыт использования свободного программного обеспечения при подготовке учителей математики и информатики в Кубанском государственном университете

Кафедра информационных образовательных технологий (ИОТ) Кубанского государственного университета (КубГУ) готовит учителей математики и информатики для Краснодара и всего региона.

Подготовка учителей на кафедре ИОТ двухуровневая: бакалавриат (5 лет) и магистратура (2 года).

В бакалавриате информационный блок дисциплин подготовки будущих учителей включает в себя такие дисциплины, как программирование, программное обеспечение ЭВМ, теоретические (математические) основы информатики, практикум по решению задач на ЭВМ, компьютерные сети, локальные сети, системные работы на компьютере, технологии web-программирования, математические пакеты и их применение в естественно-научном образовании, компьютерная графика, информационные системы, компьютерное моделирование.

Кроме того, студенты с применением ЭВМ изучают курсы «Численные методы», «Основы нейронных сетей», «Исследование операций». В шестом и восьмом семестрах будущие учителя защищают курсовые работы «Теория и методика обучения математике», «Теория и методика обучения информатике».

Обучение в основном ведётся на базе ОС Windows с применением как проприетарного, так и свободного ПО.

Двухгодичный курс программирования будущим учителям читается в основном на базе свободного программного обеспечения. Будущие учителя изучают алгоритмизацию и программирование на двух языках: Pascal и Python. В последние годы курс алгоритмизации и программирования на Паскале изучался на базе системы программирования PascalABC.net (<http://pascalabc.net>). В 2018/2019 программирование на Паскале читается с использованием Free Pascal и Lazarus. В основе лежит курс «Язык программирования Pascal: работа в средах FreePascal, Geany, Lazarus» на сайте <https://kurs.basealt.ru>. Курс был разработан автором по заказу компании Базальт СПО.

Курсы, посвящённые математическим пакетам, читаются кафедрой ИТО КубГУ всем студентам факультета математики и компьютерных наук. Начиная с 2018-19 учебного года, наряду с пакетом MathCAD при изучении этих курсов используются пакеты Scilab, GNU Octave и Maxima. Будущие учителя получают индивидуальное задание — разработать небольшую тему для школьников из курса математики средней школы в пакете Maxima. Maxima является универсальным программным средством, в котором можно представить методическое пособие, включающее блоки с теоретическим материалом, графическую и расчётную (вычислительную) части.

На третьем и четвёртом курсе студенты педагогического направления изучают курсы «Теория и методика обучения математике», «Теория и методика обучения информатике», которые заканчиваются защитой курсовой работы. Кроме изучения теории и методики обучения математике и информатике, студенты разрабатывают приложения и информационные материалы, которые могут использоваться в практической деятельности учителя. Начиная с 2018-19 учебного года, студентам были предложены темы курсовых, позволяющие внедрять свободное программное обеспечение в практику школы и университета. В 2018-19 учебном году студенты разрабатывают следующие темы:

1. Использование свободных математических программ при решении уравнений и неравенств в курсе математики средней школы.
2. Использование свободных математических программ и XHTML редактора eXeLearning при изучении раздела «Свойства элементарных функций» в курсе математики средней школы.
3. Операционная система семейства Linux на компьютере учителя математики.
4. Разработка специализированного калькулятора «Системы счисления» для курса «Математические основы информатики» .
5. Разработка системы скриптов для установки свободного ПО на компьютер учителя-предметника.

Результаты ряда тем курсовых работ будут использованы при разработке курса «Системы компьютерной математики Maxima, SciLab», который запланирован к разработке компанией Базальт СПО.

М. В. Райко, И. Н. Грибанова, Я. Н. Зайдельман,
Н. О. Бесшапошников
г. Москва, ФГУ ФНЦ НИИСИ РАН

Проект: ПиктоМир <https://piktomir.ru/>

**Для кого и как можно проводить олимпиады по
кооперативному программированию**

В докладе рассказывается об опыте организации с помощью свободно распространяемой системы программирования ПиктоМир олимпиад по кооперативному программированию для младших школьников и школьников средней школы.

Педагогическая практика показывает, что коллективные активности, более продуктивны, чем индивидуальные. Коллективные активности бывают двух типов: кооперативные и состязательные. Кооперативная активность направлена на достижение некоторой общей цели членами одного коллектива, одной команды, состязательная активность требует от участников соревнования и для младших возрастов может оказаться неприемлемой. Однако состязания повышают мотивацию детей, полностью отказываться от них нежелательно. Выходом из этого противоречия могут служить такие соревнования, где состязаются команды, а не отдельные индивидуумы. В простейшем случае — пары, выполняющие кооперативные, т.е. командные, задания. В таких командных соревнованиях состязательные и коллективные активности сочетаются и дополняют друг друга: внутри команды работа ведётся кооперативно, а состязательность обезличена и возникает только на уровне команд. При этом важно то, что опыт победы или проигрыша переживается ребёнком не индивидуально, а в окружении партнёров по команде.

Традиционной формой соревнований в нашей стране являются предметные олимпиады. Такой тип соревнований хорошо проработан методически, поддерживается государством на уровне федерального закона. [1]

Наш опыт показывает, что всё многообразие олимпиад удобно разделить на три типа:

- олимпиады для профессиональной ориентации, цель: популяризация профессий, связанных с программированием, развитие творческого потенциала учащихся, желающих связать свою будущую профессию с программированием, формирование практического опыта использования программирования для решения творческих задач
- олимпиады в клубах и кружках, цель: мотивация детей к более глубокому изучению программирования.
- олимпиады как элемент регулярного учебного курса, цель: проверка уже полученных знаний, повышение интереса

детей к учебному курсу, дополнительная мотивация детей для дальнейшего изучения данного курса

Отдел учебной информатики НИИСИ РАН в 2018 году провёл ряд олимпиад указанных выше трёх типов. Система ПиктоМир[2] и задания олимпиад являются свободно распространяемыми и наш опыт можно использовать для проведения олимпиад всех трёх типов.

Олимпиады для профессиональной ориентации

Необходимым условием для проведения таких олимпиад является знакомство участников с азами программирования и системой программирования ПиктоМир.

Пример подобной олимпиады — олимпиада для учащихся 8—9 класса «Кооперативные игры по программированию», организованных в рамках муниципального приоритетного проекта «Алгоритмическое мышление: от 0 до 11» на базе МБОУ Сургутский естественно-научный лицей.

10 команд, по 2 человека в каждой, выполняли творческие задания, направленные на демонстрацию навыков в области программирования, умения работать в команде и решать сложные профессиональные задачи. Олимпиада проводилась в один день. Участникам необходимо было выполнить 7 задач. Победители были отмечены дипломами, а остальные — сертификатами участника.

Аналогичная олимпиада ранее была проведена для учащихся 5—11 классов в Зимней Пушчинской Школе.

Олимпиады в клубах, кружках

Данный тип олимпиад не предполагает знакомства ни с азами программирования, ни с системой ПиктоМир.

Примеры таких олимпиад:

Олимпиада в «Переславском клубе интеллектуальных игр» среди учащихся 7—8 классов. Олимпиада проходила в 2 дня. Первый день — знакомство с системой Пиктомир и командная тренировка. Участники были разбиты на пары и получили серию заданий на коллективную разработку. В каждом задании были задействованы два виртуальных робота, команда должна была выполнить задание, запрограммировав совместные действия этих роботов. Второй день —

основная олимпиада, для которой был подготовлен пакет из 7 заданий нарастающей сложности. На решение заданий отводилось полтора часа. Команда-победитель выполнила все 7 заданий, остальные команды выполнили по 5 заданий.

Олимпиада в Творческой мастерской по робототехнике «Start Junior» г. Москвы среди учащихся 2—5 классов. Олимпиада проводилась в один день. Первый час — знакомство с системой Пиктомир и тренировка в парах. Второй час — собственно олимпиада. Для олимпиады был подготовлен пакет из 5 заданий нарастающей сложности. Все команды сумели решить 4 задачи из 5 и заслуженно получили дипломы успешных участников.

Все перечисленные Олимпиады вызвали у детей большой интерес и желание продолжить занятия с роботами.

Олимпиады в учебном курсе

Примеры:

Олимпиада в рамках курса «Алгоритмика для младшекласников» в школе 199 г. Москвы. Олимпиада длилась 45 минут. Для олимпиады был подготовлен пакет из 11 заданий нарастающей сложности. Команда победителей решила 11 заданий, остальные команды выполнили по 8—10 заданий. Основные трудности у детей возникли с одним из заданий. На него команды потратили больше всего времени и не все команды справились. Наличие подобного трудного задания хорошо тем, что стимулирует контакты детей по завершении олимпиады.

Олимпиада в 7 математическом классе школы 179 в рамках курса «Информатика основной школы». Олимпиада проводилась в два дня. Первый день — знакомство с системой Пиктомир и тренировка в парах. Во второй день — олимпиада продолжительностью 1 час.

Для олимпиады был предложен пакет из 7 заданий нарастающей сложности. Команда-победитель решила 6 задач, остальные команды — 4 задачи.

Для проведения описанных выше олимпиад требуется организовать локальную сеть WiFi ПиктоМир-компьютеров в помещении, где проводится олимпиада.[3] [4] Это требует предварительных усилий и предварительного тестирования. Возможна и организационно более простая схема проведения командной олимпиады. По такой схеме были проведены олимпиада в летнем лагере детский клуб Жени Кац

«Солнышково» и в Московской школе 199 в рамках летнего лагеря «Московская смена».[5]

Олимпиады проводились как завершающее мероприятие для цикла из 10 занятий. Дети были разбиты на команды по два человека. Каждой команде был выдан один планшет, локальная сеть не организовывалась. На общем планшете, передавая его от одного участника другому, члены команды составляли две программы управления двумя виртуальными роботами, решающими некоторую общую задачу. Олимпиада длилась 45 минут.

Основываясь на опыте работы с детьми на предшествующих занятиях, авторы заданий олимпиады старались подобрать задачи так, чтобы хотя бы одна команда показала выдающийся результат, а все остальные команды показали отличный результат. Этот план удалось претворить в жизнь. В итоге все дети были горды своими успехами, понимая, что успехи эти заслуженные.[6]

Опыт показал, что участие в описанных выше олимпиадах оказалось комфортным для участников, в том числе и для не вышедших в победители. Все участники уходили с ощущением успеха, заслуженно достигнутого в решении трудных задач. Не было ни одной олимпиады, в которой хотя бы одно задание не было выполнено ни одной командой. Не было ни одной команды, которая выполнила бы менее половины заданий.

Литература

- [1] Федеральный закон от 29.12.2012 N 273-ФЗ (ред. от 25.12.2018) «Об образовании в Российской Федерации». Глава 11, статья 77
- [2] Главная страница сайта системы ПиктоМир, <https://www.niisi.ru/piktomir/>
- [3] Бесшапошников Н.О., Кушниренко А.Г., Леонов А.Г. Введение в кооперативное программирование: персональная ответственность — коллективный результат [Электронный ресурс] / Доклад на 14-ой международной конференции SECR2018, Москва, 12.10.2018. <https://2018.secrus.org/program/submitted-presentations/cooperative-programming-tasks/>
- [4] Леонов А. Г., Бесшапошников Н. О., Машенко К. А., Прилипко А. А. МетаМир — система для проведения индивидуальных и командных олимпиад по алгоритмике и программированию для дошкольников и младших школьников // Сборник научных статей по итогам международ-

ной. науч.-практ. конф. 22-23 декабря 2017 г. Санкт-Петербург: Изд-во «КультИнформПресс», 2017. С. 49–55.

- [5] Кушниренко А. Г., Леонов А. Г., Грибанова И. Н., Райко М. В. Проведение цикла занятий «Алгоритмика» в летнем лагере для дошкольников и младшекласников // Труды НИИСИ РАН. 2018. Т. 8. № 4. С. 181–185
- [6] Н. О. Бесшапошников, А. Г. Кушниренко, А. Г. Леонов, М. В. Райко, И. Н. Грибанова. Кооперативно-параллельное выполнение заданий при проведении дошкольных и школьных командных олимпиад по алгоритмике и программированию // Воспитание и обучение детей младшего возраста. И-во: Издательство "Мозаика-Синтез— М. № 7, 2018. С. 29–31. eISSN: 2308–6408

И.А.Хахаев

Санкт-Петербург, СПбГЭТУ «ЛЭТИ»

<https://etu.ru/>

Возможности совершенствования образовательного дистрибутива АЛЬТ

Аннотация

Формулируются предложения по повышению качества образовательных дистрибутивов «Альт» и их привлекательности для пользователей по результатам эксплуатации при проведении занятий в техническом вузе. Обсуждается механизм реализации предложений.

Дистрибутив «Альт Образование 8.2» использовался в течение осеннего семестра 2018/2019 учебного года на кафедре вычислительной техники (ВТ) СПбГЭТУ «ЛЭТИ» при проведении практических и лабораторных занятий по дисциплинам

- Архитектура вычислительных систем
- Вычислительные системы
- Объектно-ориентированное программирование
- Организация ЭВМ
- Программирование
- Распределённые программные системы.

Также дистрибутив был развёрнут в режиме сервера (с набором штатных серверных приложений) на аппаратных средствах Huawei RH2288 V3 для поддержки samba, ftp и удалённого доступа с графическим интерфейсом (XRDP и X2Go). По результатам эксплуатации дистрибутива стали понятны возможные направления совершенствования продукта, которые могут быть реализованы как для актуальной платформы, так и при разработке дистрибутива на следующей (9-й) платформе.

1. Улучшение компоновки и состава
2. Совершенствование документации
3. Оптимизация настроек
4. Доработка Центра управления системой.

В части компоновки и состава предлагается рассмотреть возможность отказа от варианта дистрибутива «всё в одном» и вернуться к комплекту «рабочее место + сервер», при этом для сервера не использовать systemd (прототип — «Школьный сервер» платформы 7). Кроме того, можно исключить из дистрибутива для рабочего места компонент QtCreator, добавить возможности выбора загрузчика (LILO, GRUB Legacy, GRUB2) в соответствии с назначением системы и программной конфигурацией, добавить компонент для подготовки электронных учебников (eXeLearning) и заменить утилиту создания копий экрана xfce-screenshooter на другую (например, Shutter).

В части документации предлагается добавить описание настроек графической среды для XFCE и KDE, в раздел описания средств разработки для C/C++ добавить подраздел про установку всех компонентов Qt, требуемых для работы полноценной работы в среде QtCreator (примеры, комплекты и т. п.), а также добавить раздел про серверные приложения, их назначение и настройку, хотя бы по одной конфигурации для каждого серверного приложения.

В части оптимизации настроек полезными в смысле улучшения качества и привлекательности дистрибутива для пользователей могут быть следующие модификации:

1. в случае дополнительной установки KDE оставить XFCE в качестве сеанса по умолчанию
2. реализовать получение корректного разрешения от графического сервера при первоначальной настройке графической среды

3. в варианте с единственной средой XFCE разрешить повторный запуск и сохранение настроек для Code::Blocks
4. обеспечить установку шрифта droid sans при установке системы или изменить шрифт по умолчанию в GTK-приложениях и в XFCE
5. изменить настройку GTK-диалогов для открытия и сохранения файлов так, чтобы каталоги перечислялись в начале
6. при установке серверных приложений, включая alterator-fbi, сделать автоматический запуск соответствующих служб сразу после установки и при загрузке системы без дополнительных действий пользователей
7. на «домашней» странице web-сервера добавить ссылки на Центр управления системой (alterator-fbi), mediawiki, owncloud, ftp-хранилище
8. доработать конфигурации сервера XRDP и сервера X2Go для обеспечения подключения каталогов с клиентских машин в подкаталоги «домашнего» каталога на сервере.

В части доработки Центра управления системой предлагаются следующие меры

1. в локальной и web-версиях Центра управления системой добавить возможность импорта списка пользователей из текстового файла (интерфейс к команде newusers)
2. синхронизировать функции ЦУС в локальном варианте и в варианте alterator-fbi
3. реализовать автоматическую установку модуля управления сервисом/службой при установке пакетов соответствующего сервиса/службы.

Со своей стороны СПбГЭТУ «ЛЭТИ» в рамках соответствующего договора может помочь в решении указанных задач в части документации, а в случае организационной и технологической поддержки подключения студентов к процессу разработки — и в части оптимизации настроек при согласовании списка задач.

Татьяна Губина

Москва, ООО «Базальт СПО»

Проект: Портал дистанционной поддержки пользователей ОС Альт
kurs.basealt.ru

Методическая поддержка педагогов по переходу на российские операционные системы семейства Альт: направления работы, достигнутые результаты, планы на будущее

Аннотация

За последние несколько лет наблюдается рост интереса образовательных учреждений к отечественному программному обеспечению, переход на использование которого осуществляется как силами отдельно взятых школ, вузов, так и централизованно через Департаменты образования. В докладе речь пойдет о направлениях работы Базальт СПО с образовательными организациями, которые выбирают дистрибутивы ОС Альт для построения своей инфраструктуры. Большое внимание будет уделено вопросам методической поддержки педагогов по переходу на дистрибутивы Альт.

Одно из направлений работы компании Базальт СПО на рынке программного обеспечения в РФ — разработка дистрибутивов для образовательных организаций.

Разработчики компании более 15 лет участвуют в обеспечении российским и свободным ПО образовательных учреждений. Многие преподаватели школ и вузов стали членами сообщества наших разработчиков.

Внедрение в школы Московской области

С 2001 года компания выполняет поставки своих программных решений в школы РФ.

За последние 2 года школы Московской области получили более 39000 комплектов, включающих в себя ноутбуки с установленной операционной системой (ОС) Альт Линукс СПТ 7.0, Альт 8 СП с дополнительным набором образовательного программного обеспечения (ПО).

В настоящее время ведётся активная работа и уделяется большое внимание поддержке педагогов в вопросах перехода на разрабатываемые программные решения, оказывается техническая и методическая поддержка школам Московской области, а также работа со школами из других регионов.

Концептуальная основа построения процесса подготовки педагогов к работе в ОС Альт

В 2017 году в помощь учителям и преподавателям компания ООО «Базальт СПО» разработала и запустила Портал дистанционной поддержки пользователей ОС Альт kurs.basealt.ru с учебными и методическими материалами.

В основу структуры электронного контента Портала для построения процесса методической поддержки учителей был положен перечень ИКТ-компетенций педагога, приведённый в Приложении № 1 профстандарта педагога [1]. В соответствии с 3 компонентами профессиональной ИКТ-компетентности педагога (основана на Рекомендации ЮНЕСКО «Структура ИКТ-компетентности учителей», 2011 г.) Портал дистанционной поддержки включает в себя:

- учебные курсы, которые позволят сформировать теоретические знания и практические умения в области применения ОС Альт и входящих в неё прикладных программ при решении профессиональных задач (общепользовательский компонент);
- методические курсы, содержащие организационные и методические вопросы построения информационной среды школы, организации образовательного процесса на основе программных решений Альт (общепедагогический компонент);
- учебные специализированные курсы, узкой направленности, которые помогут освоить программы, относящиеся к категории специального программного обеспечения и системам программирования (предметно-педагогический компонент).

Разработка контента каждого из курсов шла с ориентиром на ИКТ-компетенции учащихся, которые в соответствии с действующими ФГОС и примерными основными образовательными программами должны быть сформированы за время обучения в школе.



Рис. 1: Главная страница портала kurs.basealt.ru

Портал дистанционной поддержки пользователей ОС Альт

Это веб-ресурс, который поможет педагогу интегрироваться в современную информационно-образовательную среду образовательного учреждения, построенную на отечественных программных решениях Альт.

Входной точкой для понимания и целостного восприятия концепции перехода на отечественное программное обеспечение является курс «Импортозамещение ПО в учебном процессе». На курсе доступны:

нормативная база перехода на отечественное программное обеспечение;

таблицы соответствия «образовательная цель (по предметам) и ПО из дистрибутива Альт»;

перечень и описание основных элементов формируемой ИКТ-компетенции учащихся и ПО из дистрибутива Альт;

таблицы соответствий ПО, используемого в образовательных целях под управлением ОС Windows и ПО из дистрибутива Альт;

руководства пользователя, видеообзоры некоторых программ;

Анализ возможностей преподавания школьного курса информатики на базе Альт Образование 8 и пр.

В рамках работы конференции будут представлены возможности портала, содержание и структура уже разработанных курсов, варианты сотрудничества нашей компании с педагогическим сообществом в части разработки методических и учебных материалов для учителей.

Достигнутые результаты

В настоящее время наблюдается рост интереса пользователей к нашему portalу, а также их активности на курсах.

Количество курсов на портале, материалы которых уже доступны для изучения — 20.

Количество зарегистрированных пользователей — более 400.

Наибольшей популярностью пользуются курсы:

Альт Рабочая станция (Mate) (128 участников) — выдано 18 сертификатов.

Альт Образование (81 участник) — выдано 16 сертификатов.

Записано и размещено на портале более 70 видеуроков (скринкастов).

Планы на будущее

На этом наша работа не останавливается, на текущий момент ведётся разработка и в ближайшее время будут открыты материалы ещё 6 курсов, многие из них будут представлены в видеоформате.

Видеоматериалы будут также размещены на официальном канале YouTube компании.

В перспективе планируется сделать Портал kurs.basealt.ru площадкой для накопления и обмена опытом и знаниями педагогов в области организации образовательного процесса на базе ОС семейства Альт, доступной для всех заинтересованных лиц.

Мы открыты для всестороннего сотрудничества с образовательными организациями: разработка курсов, методических материалов по программным решениям Альт для использования в учебном процессе.

Компания Базальт СПО готова и дальше предоставлять свои технические, организационные и кадровые возможности для организации внедрения свободного и российского ПО, а также участвовать в

совершенствовании содержания образования с учётом вопросов импортозамещения и всестороннем сотрудничестве с образовательными организациями.

Литература

- [1] Профессиональный стандарт «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)». Приказ Минтруда России от 18.10.2013 №544н.,

Сергей Голубев

Раменское

Что следует знать о СМИ студентам технических специальностей, участвующих в свободных проектах

Аннотация

В докладе рассматриваются некоторые аспекты информационной поддержки свободного проекта. Автор считает, что эти сведения помогут преподавателям вузов максимально эффективно подготовить студентов технических специальностей к практической работе в данном направлении.

В настоящее время часть студентов и даже школьников так или иначе участвуют в свободных проектах: кто самостоятельно, кто в рамках практических занятий. Успех их дальнейшей работы определяется не только качеством кода, чему их научат в вузе, но и массой иных факторов, в том числе эффективным участием в информационной поддержке проекта. Для этого выпускник должен прежде всего знать, как работают современные отечественные СМИ. В настоящее время наблюдается ряд устойчивых тенденций, вызванных спецификой развития этого вида бизнеса в России.

Технические СМИ трансформируются в деловые

Вероятнее всего, вызванный кризисом 2008-го года закат сугубо технических СМИ, специализирующихся на популяризации отраслевых решений, завершился их практически полной ликвидацией. Экономические реалии таковы, что независимые издания вынуждены

ориентироваться исключительно на бизнес-сектор, причём выступающий не в качестве читателя, а в качестве заказчика.

Для сохранения видимости независимости и сохранения прежнего объёма инфопотока, как правило, используется самый дешёвый контент. Это могут быть переводы и/или практически неизменённые пресс-релизы от различных партнёров.

Важно, что обратный процесс вряд ли возможен, поскольку трансформация привела к заметной смене кадров — все ушли в PR. К тому же, свято место пусто не бывает — техническая ниша уже занята.

Новые информационные ресурсы

Прежде всего, освободившуюся нишу заняли сайты типа «Хабрахабр». На них компании могут самостоятельно рассказывать о собственных решениях, причём делать это значительно оперативней и компетентней. Более того, теоретически никто не запрещает заинтересованным разработчикам организовать там некое пишущее сообщество, способное создавать оригинальные и более-менее объективные материалы.

На официальных сайтах различных компаний появились блоги. В частности, блог компании ЛК полностью закрывает потребности читателя в информационных материалах, делая ненужными аналогичные публикации в СМИ.

Наконец, соцсети оказались вполне удобным инструментом для информирования достаточно широкой аудитории. Многие компании успешно эксплуатируют этот информационный канал, тем более, что он гарантирует быструю обратную связь с пользователем.

Методы работы СМИ в современных условиях

Главное концептуальное изменение заключается в резком падении интереса к читателю, как одному из главных кормильцев. В первую очередь это вызвано снижением популярности бумажных изданий чуть ли не до нуля, в результате чего читатель перестал покупать материальные экземпляры.

Что касается рекламодателя, то у профильных СМИ он и раньше был больше похож на спонсора, то теперь он превратился в благодетеля без всяких оговорок. Если читатель бумажных экземпляров хотя бы смотрел всю рекламу, то в сети есть масса способов её заблокиро-

вать, причём все методы борьбы с блокировщиками эффективными назвать нельзя.

Таким образом, у СМИ не осталось вариантов. Только публикация оплаченных статей позволяет им как-то удержаться на плаву.

Впрочем, солидные СМИ какие-то приличия при этом стараются соблюдать. Компания заказывает написание текста аффилированному с изданием PR-агентству и размещает его в СМИ от имени своего сотрудника. Редакция как бы и ни при чём — прислали интересный материал, который и был размещён на сайте.

Тем более, что такой метод позволяет создавать действительно интересные и полезные статьи, поскольку текст проходит множество стадий контроля. Разумеется, имеет значение и более высокий гонорар автору.

В результате не случилось никакой трагедии, поскольку довольны все участники процесса, включая читателей. Дорожащие репутацией СМИ всё-таки следят за тем, чтобы текст не получился откровенно рекламным — в большинстве случаев получается вполне объективный и грамотный материал, который интересно читать.

Не исключено, что именно таким путём у нас будут сформированы настоящие отраслевые СМИ, существующие на средства отрасли и выражающие её интересы. По крайней мере, этот сценарий представляется более вероятным, чем создание профильных СМИ отраслевыми ассоциациями, решающими в настоящее время несколько иные задачи.

А как же СПО?

После пика, вызванного государственными проектами, интерес СМИ к СПО постепенно падал и в настоящее время приблизился к нулевой отметке. Не исключено, что соответствующие разделы в изданиях перешли в режим минимальной поддержки «чтобы было».

СПО рассматривается исключительно в контексте импортозамещения, причём не в технологическом, а в бизнес-политическом аспекте.

Как строить работу со СМИ

Если у свободного проекта недостаточно средств для договора с хорошим PR-агентством, то никак. Разумеется, это не означает, что

СМИ про него вообще не будут писать, но построить таким образом сколько-нибудь внятную и управляемую информационную политику невозможно.

Но в этом нет ничего страшного, поскольку на сегодняшний день есть масса других способов решения той же задачи: «Хабрахабр», соц-сети, собственные блоги. Получится и дешевле, и эффективней.

Алексей Федорчук

Москва

Проект: Блогосайт, Cinia <https://alv.me/>, <https://www.cinia.ru>

Linux для гуманитариев

Аннотация

В докладе затрагиваются два аспекта взаимодействия СПО и гуманитарных областей человеческой деятельности (научной и литературной). Первый — пригоден ли Linux и его приложения для использования представителями гуманитарных профессий? И второй — если пригоден, то как?

Традиционно считается, что работа в ОС Linux предъявляет более значительные требования к технической подготовке применителя, нежели в иных операционках. И потому эта система подходит в качестве настольной для специалистов, обладающих соответствующими знаниями и навыками — то есть инженеров и научных работников естественно-научного профиля. Но не пригодна для использования специалистами-гуманитариями и представителями так называемых творческих профессий.

Однако это не так. Ибо Linux располагает мощными средствами работы с текстами. Не случайно прародительский UNIX «в мирных целях» впервые был применён именно для подготовки технической документации. С тех пор средства эти (как универсальные утилиты CLI, так и специализированные приложения) значительно усовершенствовались. И дополнились инструментами для иллюстрирования текстов. Впрочем, иллюстрирование — тема очень отдельная, и зависящая от того, что именно иллюстрируется.

Поэтому Linux представляется очень удобной средой для всех, кто в своей практической деятельности связан с работой с текстами и их

иллюстрированием — от переводчиков и журналистов до, страшно сказать, юристов и даже поэтов.

Ну а потребность в технических знаниях для практического использования Linux'a и его приложений сильно преувеличена. Разумеется, некий минимум таких знаний был бы не лишним. И желательно этот минимум получать если не за школьной партой, то за вузовской скамьей. Но поскольку в наших условиях это не реально — вполне можно обойтись и чисто рецептурными знаниями. Каковые нынче приобретаются очень легко как из сетевых, так и печатных материалов.

Тем более что знать тут надо не так уж и много. Во-первых, кое что об инструментах. Когда речь заходит о работе с текстами, обычно вспоминаются так называемые текстовые процессоры. Хотя трудно представить себе что-то менее подходящее для сочинения нарративных текстов, нежели то, что называется word processor (text processor — это совершенно другие приложения). И потому выбор гуманитария-текстовика — это текстовый редактор.

Каковых, казалось бы, бессчётное количество. Однако вспомним о CLI-утилитах для работы с текстами. Создавать тексты с их помощью — занятие не самое весёлое (хотя и возможное). Но зато для обработки текстов уже созданных (контекстного поиска, замены, разделения, объединения) они подходят более чем.

Тут уж желательна визуализация результатов выполнения команд. И это резко снижает число участников в выборе текстового редактора — ограничивая его теми, что имеют встроенный терминал. Благо, по случайному совпадению, они же оказываются и в числе самых функциональных.

Так что, «резюмируя сегодняшний базар», можно сказать: да, Linux очень пригоден для его применения гуманитариями. И не в последнюю очередь потому, что позволяет легко и просто совмещать инструменты CLI и GUI.

Алексей Ерпелев, Владимир Симонов, Анастасия Рубанкова
г. Москва, ФГБОУ ВО «Российский государственный социальный
университет»

Использование свободного ПО в учебном процессе на примере разработки устройств «умного дома» с применением микроконтроллеров Arduino и Iskra JS

Аннотация

В докладе рассматривается использование свободного программного обеспечения для платформ Arduino и Iskra JS при обучении студентов направлений подготовки ИВТ, ИСТ и ПИН основам программирования, электроники и робототехники. Была выбрана тематика «умного дома» и проектирование устройств для него, как наиболее универсальная. Макет умного дома реализован на базе плат Iskra JS (JavaScript), Arduino (C++), а также различных модулей: ультразвукового дальномера, светодиодов (LED-лент и отдельных светодиодов), ИК-приёмника, зуммеров, сервоприводов, датчиков-измерителей климатических параметров.

Современное технологическое развитие дало возможность создавать приборы, устройства и их комплексы для улучшения условий социальной жизни человека [1]. Такие технологии можно объединить общим названием — «умный дом» (УД). К устройствам управления умным домом относятся контроллеры, сенсоры, актуаторы, индикаторы различных типов и иные устройства. Данные устройства работают не только на проприетарном, но и на свободном программном обеспечении. Так, например, микроконтроллер Arduino работает на свободном ПО Arduino IDE, которое позволяет на языке программирования C++ реализовывать сложные по своей структуре устройства и макеты. Данный микроконтроллер используется в обучении студентов по специальностям ИВТ, ИСТ и ПИН для освоения профильных компетенций — способностью сопрягать аппаратные и программные средства в составе информационных и автоматизированных систем (ПК-5) и способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2).

С целью демонстрации использования свободного программного обеспечения в учебном процессе на примере разработки устройств

«умного дома», авторами была поставлена и решена задача разработки соответствующего макета (платы IskraJS (JavaScript), Arduino (C++), ультразвуковой (УЗ) дальномер, светодиоды (LED-лента и отдельные светодиоды), ИК-приёмник, зуммеры, сервоприводы, датчики-измерители климатических параметров и др).

Разработанный экспонат представляет собой макет дома с «умными» функциями. Размеры экспоната 400x600x500 мм, при этом выполняется следующее: управление внутренним освещением от срабатывания УЗ дальномера; регистрация перемещения человека по комнатам датчиками движения; управление открытием/закрытием дверей и освещением от ИК-пульта; озвучивание от зуммера случайными музыкальными композициями; управление воротами УД с помощью RFID, а также ряд иных функций.

При выполнении проекта «умный дом» студенты изучают языки программирования, получают навыки сборки устройств из электронных компонентов и опыт проектирования схем устройств в виртуальных средах, например TinkerCad. Также студенты учатся составлять алгоритмы работы модулей и устройств, обработки информации; указанные алгоритмы реализовываются с использованием нескольких языков программирования с выдачей сигналов на исполнительные устройства. Таким образом, студентами осваиваются соответствующие компетенции ФГОС.

Логическим завершением проектирования является 3D-прототипирование для изготовления корпусных элементов (компетенция №45 Worldskills), что является актуальным требованием современного обучения в вузах.

Анализируя полученный результат, можно отметить его эффективность и привлекательность.

Разработанный экспонат демонстрирует возможности свободного программного обеспечения для освоения студентами компетенций по направлениям подготовки ИВТ, ИСТ и ПИН. При этом технологии УД являются базовыми для изучения студентами технических специальностей программирования, электроники, автоматике, основ робототехники и прототипирования [3, 4]. Развитие данного направления «умный дом» в обучении студентов предполагает дальнейшее усложнение проекта в сторону увеличения количества используемых функций (шлагбаум, сканер отпечатка пальца и т.д. и усложнения алгоритмов) и применения более сложных технологий обмена данными, что

реализуется на базе одноплатных компьютеров, например Raspberry Pi.

Литература

- [1] Обзор системы умного дома Samsung SmartThings // Smart.Home Market URL: <https://smart-home.market/obzor-sistemy-umnogo-doma-samsung-smarththings-s3026> (дата обращения: 02.12.2018).
- [2] Сайт, посвящённый разработкам в области электроники — URL: <http://amperka.ru/> (дата обращения: 02.12.2018).
- [3] Проектирование студентами высших учебных заведений реальных устройств при изучении ряда дисциплин, связанных с обработкой информации и данных [Текст]* / Симонов В.Л., Аметова М.М., Хмыров Н.А. и др. // Информационные технологии в образовании: материалы науч.-практ. конф. — Саратов, 2017.— С. 296–298.
- [4] Справочник по отечественным и иностранным электронным компонентам / Режим доступа URL: <http://www.chipinfo.ru/> (дата обращения 02.12.2018).

Марина Нидилько, Джамиля Митарова
г.Москва, РГСУ

Проект: Разработка проекта на микроконтроллерной плате

Разработка проекта на микроконтроллерной плате

Аннотация

На сегодняшний день существует множество различных свободных программных продуктов, которые используются в разных областях и сферах человеческой деятельности. Сфера программирования не является исключением. Так, к примеру, существуют свободное ПО для программирования микроконтроллерных плат.

Остановимся подробнее на том, что же такое свободное программное обеспечение, микроконтроллерные платы и как они связаны между собой, а также какие проекты можно собрать из этого сочетания.

СПО — это свободное программное обеспечение, пользователи которого имеют права на распространение, установку, изменение и использования программного продукта. При этом если на программное обеспечение есть исключительные права, то свободное использование обеспечивается за счёт свободной лицензии.

Микроконтроллерная плата это маломощный компьютер, роль процессора в котором исполняет микроконтроллер. Микроконтроллер — это микросхема, которая предназначена для управления электронными устройствами. Микроконтроллер на одном кристалле сочетает в себе функции процессора, периферийных устройств, устройств ввода-вывода, а также ОЗУ и ПЗУ.[2]

Рассмотрим наиболее известные микроконтроллерные платы для проектирования электронных устройств.

- Arduino — плата с микроконтроллером и парой десятков контактов для подключения всевозможных компонентов от лампочек до различных датчиков. В микроконтроллер зашит загрузчик, что позволяет программировать без использования специального программатора. Такая конструкция позволяет создавать различные проекты. Программируется Arduino на упрощённой версии языка C++, известной как Wiring. Для разработки программного кода используется бесплатная среда Arduino IDE. [1]
- IskraJS — также как и Arduino, плата с микроконтроллером и контактами для подключения датчиков. Плата обладает встроенным интерпретатором для языка программирования JavaScript. Для написания кода используется среда разработки Espruino Web IDE.[3]

При помощи аппаратной части (микроконтроллерной платы) и программного компонента (написанного в свободно распространяемой среде разработки) можно создавать от простых проектов (например, гирлянда) до сложных.

Примером сложного проекта можно привести систему контроля и управления доступом на защищённую территорию. Для реализации данного проекта нам необходимо подключить к плате считыватель RFID, электромеханический замок, кнопки вызова, микрофон и динамики для связи. Возможности системы можно расширить, добавляя другие компоненты. Для работы подключённых устройств к плате, необходимо написать программу, которая объединит все компоненты в единую гармоничную систему. В итоге мы получим проект следую-

щего принципа действия: человек, имеющий право доступа, подносит ключ к RFID считывателю, далее после проверки подлинности ключа, электромеханический замок отпирается. В случае, если ключ неверный, замок остаётся закрыт. Если у человека нет ключа, он вызывает диспетчера при помощи кнопки вызова и в этом случае диспетчер принимает решение о доступе данного человека на территорию. Если принято решение впустить человека, с пульта управления посылается сигнал и замок отпирается.

Таким образом, мы видим, что при помощи свободного программного обеспечения для программирования микроконтроллерных плат можно создавать различные интересные и полезные проекты.

Литература

- [1] Arduino, 2018, <https://www.arduino.cc/en/Main/Software>
- [2] Микроконтроллер, 2018, <https://ru.wikipedia.org/wiki/Микроконтроллер>
- [3] Iskra JS/Узнать больше, 2018, <http://amperka.ru/product/iskra-neo>

Воронин Игорь Вадимович Воронина Вероника Вадимовна
Шатура, Моск.область, г. Павлово, Нижегородской области, Институт
Проблем Лазерных и Информационных технологий Российской Академии
Наук, МБОУ СОШ №7

Проект: Образовательный проект УМКИ, Сайт фестиваля РоботоБУМ
<http://umki.vinforika.ru/>, <http://robotobum.ru>

От основ к созданию роботов. Знакомство с робототехникой в школе

Аннотация

Рассматриваются способы привлечения школьников в сферу интересов технического творчества. Анализируется возможность использования в образовательном процессе учебного пособия для занятий по робототехнике «Программирование для детей. От основ к созданию роботов». Предлагается решение в виде комплекта образовательной робототехники УМКИ, как пакет в школьном дистрибутиве УМКИ.

Параметр	Число охваченных людей
Число школьников, занятых творчеством	500 000
Число школьников, планирующих поступать в технический вуз	200 000
Число абитуриентов, поступивших в технический вуз	50 000
Число студентов, которые выберут обучение по профилю будущего стартапа	10 000
Число студентов, запустивших свой стартап	2 000
Всего число людей, работающих в стартапах	300
Успешный стартап, хотя бы 2 или 3 команды	24

Рассматривая тенденции и перспективы развития современного школьного образования, можно отметить, что знакомство с робототехникой полезно тем, что учащиеся получают знания и навыки, которые будут полезны и при дальнейшем обучении в вузе и при работе на производстве. После получения высшего технического образования молодые люди вливаются в технологический процесс и либо работают по найму, либо открывают своё дело — запускают стартап. Одним из главных залогов технологического процветания страны можно считать организацию множества успешных стартапов.

Для решения такой задачи, а также с целью повышения уровня подготовки выпускников высших учебных заведений, в университеты и технические вузы должен поступать качественный абитуриентский состав. Отсюда следует понятный вывод: для того, чтоб в достаточном количестве в высшие учебные заведения пришло нужное количество грамотных молодых людей, заниматься их подготовкой необходимо с как можно раннего возраста. Воронка задействованных в научно-техническом творчестве школьников приведена в таблице.

Понятно, что, ученик в старшей школы, это уже практически, сформированная личность, поставившая себе ближайшие цели, и

убеждать его, большее время уделить занятиям программированием а, например, не юриспруденцией или общественными науками — бесполезно, поэтому начинать техническую подготовку детей надо в средней или младшей школе, а, первые занятия по знакомству с робототехникой целесообразно проводить даже с дошкольниками.

Учебное пособие издательства Питер: «Программирование для детей. От основ к созданию роботов», (авторы И.Воронин, В.Воронина) разработано как раз с целью, дать возможность наибольшему количеству ребят попробовать себя в техническом творчестве. Базовые основы, описанные в книге помогут юному читателю самостоятельно создать свой самый первый проект, который в будущем сможет оказаться базой для собственного стартапа.

Книга разделена на девятнадцать эпизодов. Каждый эпизод, отдельная тема для самостоятельного изучения читателем или педагогом может быть использован в качестве руководства при проведении занятий с детьми. Оно состоит из разделов, которые называются: эпизоды, по аналогии со Звёздными войнами. В конце каждого эпизода есть теоретические и практические задания, которые тщательно подобраны к темам.

В книге по шагам от самых простых, базовых понятий ребёнок постепенно подводится к пониманию основ робототехники. В книге рассмотрены следующие темы.

- Основы теории информатики и робототехники: например, знакомство с различными системами счисления, представлено в интересной и увлекательной форме — показано, как закодировать сложные данные наборами ноликов и единичек, как с помощью построенных двоичных кодов можно управлять движением игрушечной машинки.
- Темы, связанные с кодированием информации разбираются на конкретных примерах, в которых с помощью мобильных приложений предлагается научиться читать и создавать собственные QR-коды и штрих-коды.
- Базовые понятия логики, такие как, дизъюнкция, конъюнкция, инверсия рассматриваются не только в теоретическом плане, но и разбирается возможность практического построения действующих моделей из электронных компонентов детского конструктора.

- В интересных заданиях, в результате выполнения которых создаются забавные картинки, разбираются понятия системы координат — тема необходимая для позиционирования робота на плоскости.
- На примерах связанных с разработкой моделей используемых в быту устройств, исследуются несложные электрические цепи.
- Материалы второй части книги содержат уже более усложнённый материал, по сравнению с первой, базовой частью: в частности, происходит первое знакомство с программированием контроллеров на базе Arduino, и программирование датчиков к ним.
- В этой части книги изучаются базовые функции ввода-вывода данных, разбираются отличия между аналоговым или цифровым портами, происходит знакомство с разнообразными протоколами беспроводной связи, подробно описывается разнообразие понятия драйвера и обсуждается его необходимость при управлении моторами.
- И ещё многое-много другое.

Эти знания помогут ребятам подготовиться и достойно выступить на соревнованиях и хакатонах по робототехнике, а также принять участие и различных научно-технических мероприятиях, тем самым будет формироваться навык, который им пригодится в дальнейшем при возможном запуске собственного стартапа.

Описываемое издание входит в учебно-методический комплект образовательной робототехники УМКИ, (пакет в школьном дистрибутиве УМКИ с использованием сред визуального программирования SNAP и КУМИР, который включён в базовую поставку Альт Образование), в качестве книги для ученика.

Книга «Программирование для детей. От основ к созданию роботов», продаётся в популярных интернет-магазинах и получила очень хорошие отзывы от педагогов и родителей: «Вот бы мне в своё время такую книгу! Где так легко и доступно рассказывается о сложных вещах. Кстати, книгу можно и родителям порекомендовать, которые хотят разобраться в теме и понимать как там всё работает и устроено», «Расскажу своё мнение как педагога по робототехнике. В книге подробно, по разделам расписано о создании роботов, их программированию и управлению, причём разными способами. Каждый эпизод книги можно использовать в качестве руководства для проведения

занятий с детьми. В конце каждого эпизода есть теоретические и практические задания, которые очень тщательно подобраны к темам и мне очень-очень понравились (сама их выполняла, очень интересно). Мне было очень понятно *как это делать*. Думаю, что детям начиная с 5-го класса всё будет так же понятно. Вообще я занимаюсь с детьми робототехникой, используя проприетарные образовательные конструкторы от датской компании. Но, прочитав «Программирование для детей», планирую написать новую образовательную программу, пойти с ней к директору и просить необходимое оборудование для её реализации. В книге есть даже ссылки на полезные ресурсы, где можно скачать всё необходимое программное обеспечение для обучения *бесплатно* и ссылки, где можно купить необходимые компоненты для практических заданий».

Используемое в обсуждаемом учебном пособии оборудование, построено на базе AVR микропроцессоров ATmega — контроллеров Xbee, Arduino с наборами датчиков и электронными компонентами позволяющими создавать программируемые модели роботов.

Книга «Программирование для детей. От основ к созданию роботов», как и весь учебно-методический комплект «УМКИ» базируется на программном обеспечении под лицензией GNU, которое совершенно свободно и бесплатно можно установить на компьютер, смартфон, передать друзьям, ученикам и коллегам по работе. Пакеты программного обеспечения входят в репозиторий Восьмой платформы (p8).

Литература

- [1] Стратегия развития отрасли информационных технологий в Российской Федерации на 2014-2020 годы и на перспективу до 2025 года
- [2] Игорь Воронин о стартапах, ВРО и Лего <http://edurobots.ru/2016/07/igor-voronin-o-startapax-vro-i-lego/>
- [3] Воронин И.В., Воронина В.В. Программирование для детей. От основ к созданию роботов, г.2018, ISBN 978-5-4461-0555-7, Издательство Питер
- [4] Книжный портал LiveLib <https://www.livelib.ru/review/1047123-programmirovanie-dlya-detej-ot-osnov-k-sozdaniyu-robotov-veronika-voronina#comments>
- [5] Отзыв на книгу <https://www.labyrinth.ru/reviews/show/1779774/>

Д.А. Костюк¹, К. Опасяк²,
Г.А. Четвёркина¹, В.М. Лаппо¹

Брест; Варшава, ¹Брестский государственный технический университет;

²Варшавский политехнический университет, Samsung R&D Insitute Poland

Построение практикумов по изучению архитектуры и периферийных устройств на основе шины USB

Аннотация

Рассматриваются особенности применения Linux-устройства в роли конфигурируемого USB-гаджета для практического изучения студентами архитектуры и протоколов шины USB. Подход нацелен на абстрагирование от низкоуровневой реализации в рамках ОС для максимальной фокусировки на особенностях обмена данными и стандартных классах устройств, применении средств отладки и мониторинга USB-трафика. За счёт работы в пространстве пользователя и использования виртуальных файловых систем ConfigFS и FunctionFS достигается упрощение учебных задач, благодаря чему изучение USB-протоколов, несмотря на их значительно большую сложность, требует времени, сравнимого с изучением legacy-интерфейсов.

Введение

Классический подход в изучении низкоуровневого программирования и архитектуры предполагает поэтапное изучение процессора и соответствующего ему языка машинных команд, архитектуры вычислительной системы и периферийных устройств (их принципов действия, интерфейсов подключения и протоколов обмена данными). Взаимодействие с классическими (т. н. legacy) интерфейсами (COM, PS/2, LPT/IEEE 1284) позволяет изучить работу с портами ввода-вывода и прерываниями, но в многозадачных ОС данную задачу осложняет необходимость написания драйверов, работающих полностью или частично в привилегированном режиме, и взаимодействующих с другими частями ОС.

В своё время успешное решение данной задачи с использованием модулей ядра GNU/Linux позволило авторам сохранить прежнюю схему курса в условиях современной и более чем актуальной платформы [1, 2]. Однако изучаемые студентами классические интерфейсы начинают создавать проблему при выполнении учебных задач из-за их

постепенного вытеснения интерфейсом USB, полностью завершившегося на ноутбуках и почти полностью — на настольных компьютерах.

Особенности изучаемого материала

По сложности USB очень отличается от классических интерфейсов, представляя собой сеть с собственным стеком протоколов (рис. 1-а).

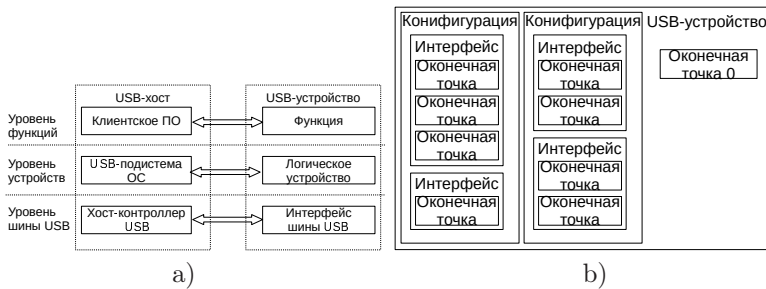


Рис. 1: Уровни организации интерфейса USB (а) и иерархическое представление USB-устройства (б)

В логическом плане USB-устройство всегда является *хабом* либо *функцией*. Функции реализуют функционал устройства, хабы — дополнительные точки подключения, а композитные устройства сочетают несколько функций через внутренние хабы. Каждая функция имеет как минимум одну *конфигурацию*, каждая конфигурация — как минимум один *интерфейс*, а уже интерфейс может иметь *оконечные точки* (endpoints) (рис. 1-б). Оконечные точки различают по номеру, а также по направлению передачи данных в соответствующем *канале* (pipe). Каждая функция реализует минимум две оконечных точки с номером 0, доступные сразу по включению устройства; остальные оконечные точки становятся доступны после конфигурирования.

Взаимодействие с этой иерархией осуществляется с помощью структур данных определённого формата — *дескрипторов*. Каждое устройство имеет как минимум один дескриптор конфигурации, каждый дескриптор конфигурации — минимум один дескриптор интерфейса, а каждый дескриптор интерфейса — до 15 дескрипторов оконечных точек (плюс строковые дескрипторы на разных языках). Об-

мен данными с устройством всегда инициируется хостом и осуществляется в виде транзакций приёма и передачи данных. Передаваемые наборы служебных пакетов и пакетов данных зависят от типа передачи. По типу передачи каналы (и, соответственно, оконечные точки) подразделяются на *поточные* (bulk), *управляющие* (control), *изохронные* (isochronous) и *прерывания* (interrupt).

Взаимодействие с устройством всегда начинается с управляющего канала 0, который позволяет прочесть краткую информацию об устройстве, включая коды производителя и модели (Vendor ID и Product ID), а также список других оконечных точек, после чего драйвер выполняет конфигурирование устройства и далее — информационный обмен со сконфигурированными оконечными точками по одному из четырёх протоколов передачи. При этом устройство может принадлежать к одному из стандартных классов (например USB-HID), либо представлять собой некоторую проприетарную реализацию.

Специфика разработанного практикума

Из приведённого краткого описания очевидно, что разработка драйвера USB-устройства оказывается очень сложной задачей даже для студентов, уже умеющих создавать простые модули ядра. Осложняет задачу также то, что многоуровневой логической организации соответствует не самый простой API, а также существенная разница в реализации программного обеспечения для хоста и устройства. Поэтому учебные курсы архитектуры ЭВМ зачастую ограничиваются теоретическим изучением особенностей шины USB без практических заданий, касающихся создания какой-либо законченной реализации.

Представляемый лабораторный практикум решает данную проблему за счёт абстрагирования от реализации драйвера в рамках ОС и максимального переноса логики работы в пространство пользователя для концентрации на изучении логической структуры и принципов взаимодействия с USB.

Заметим, что работа с USB-устройствами в пространстве пользователя достаточно распространена, в частности, на этапе прототипирования и отладки драйвера устройства, а также в случаях, когда производительность не является проблемой. В GNU/Linux данный уровень абстракции обеспечивает популярная библиотека libusb, однако её API также требует времени на изучение, что создаёт дополни-

тельную когнитивную нагрузку, а следовательно увеличивает время, необходимое на освоение материала. Поэтому в разработанном курсе ей отведена не самая большая роль, а взамен упор делается на взаимодействие с USB на уровне виртуальных файловых систем ConfigFS и FunctionFS.

Виртуальная файловая система ConfigFS позволяет создавать и изменять из пространства пользователя объекты ядра. В нашем случае она используется в качестве интерфейса для описания функций и конфигураций, составляющих композитное USB-устройство (в частности, аналогичный подход используется Android-смартфонами). При этом создание сложной иерархической структуры, схематично показанной на рис. 1-b, выполняется на уровне файловой абстракции, созданием/изменением соответствующих файлов и каталогов [3]. Для создания типовых функций устройства в пространстве пользователя ConfigFS используется совместно с виртуальной файловой системой FunctionFS (рис. 2).

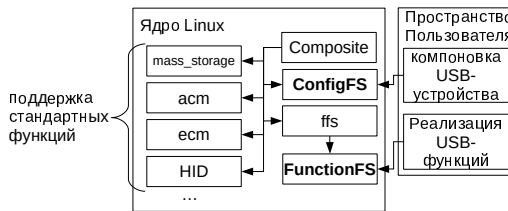


Рис. 2: Использование ConfigFS и FunctionFS

Программный инструментарий и выбор аппаратной платформы

Кроме соответствующих модулей ядра, библиотеки libusb и стандартного тулчейна GNU/Linux, в лабораторном практикуме используется утилита lsusb для получения информации о подключённых устройствах, а также анализатор сетевых пакетов Wireshark для исследования обмена информации хоста с устройством (с помощью модуля ядра usbmon).

Выбор аппаратного обеспечения для проведения практикума предопределён тем, что в часть функционала, отвечающая за поведение USB-устройства, требует наличия контроллера, способного работать в

режиме USB-device, отсутствующего на стандартных ПК. Таким контроллером обычно оснащены одноплатные компьютеры на базе процессора ARM, поэтому разумным выбором оказываются компьютеры Raspberry Pi либо им подобные; в нашем случае в этой роли выступал комплект отладочных плат на процессоре Ecuipos для практикума по программированию встраиваемых систем [4].

Заметим, что при необходимости упростить требования за счёт отказа от взаимодействия с реальным USB-устройством в пользу его эмуляции, выбор эмулятора и эмулируемой платформы не является тривиальным. Очевидный вариант с запуском USB-устройства в виртуальной машине и подключение эмулируемого USB-контроллера к контроллеру хост-системы представляет собой задачу, прямо противоположную той, которую решают системы виртуализации, включая QEMU (проброс реального USB-устройства в виртуализованное окружение). Однако запуск учебных программ в режиме эмуляции легко решается с помощью модуля ядра `dummy_hcd`, который перенаправляет трафик USB-устройства обратно в систему на виртуальный USB-хаб, создавая закольцованное устройство. Благодаря этому драйвер USB-хоста и драйвер USB-устройства могут работать на одной и той же хост-системе (рис. 3).

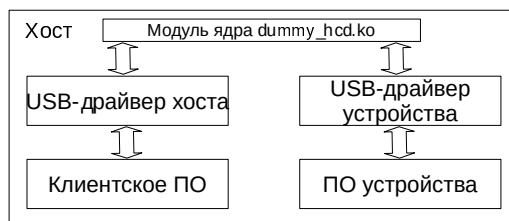


Рис. 3: Реализация информационного обмена с USB-устройством на хосте

Структура практикума

Структура разработанного практикума включает 5 лабораторных работ. В первой работе наглядно демонстрируется, как происходит обмен информацией, студенты знакомятся утилитой `lsusb` и библиотекой `libusb` для получения информации об устройстве, а также используют

анализатор пакетов Wireshark для отслеживания простейшего взаимодействия с устройством (нажатия клавиши). Во второй работе рассматриваются особенности передачи данных по шине USB и соответствующий функционал libusb. В третьей работе рассматривается конфигурирование устройства с помощью файловой системы ConfigFS. Четвёртая работа позволяет реализовать стандартную функцию с помощью файловой системы FunctionFS. Наконец, целью завершающей пятой работы является реверс-инжиниринг разработанного проприетарного драйвера с помощью Wireshark.

Литература

- [1] *Костюк Д.А.* Изучение низкоуровневого программирования и вычислительной архитектуры на базе платформы GNU/Linux // Свободное программное обеспечение в высшей школе: тезисы докладов V конференции. Переславль, 30–31 января 2010 г. / Москва, Институт логики, 2010. — С. 32–35.
- [2] *Костюк Д.А., Жук А.М.* Построение практикумов по программированию периферийных устройств и архитектуре ЭВМ на базе GNU/Linux // Тези міжнародної науково-практичної конференції FOSS LVIV–2011, Львів, 1–6 лютого 2011 р. – Львів: Вид. ЛНУ ім. І. Франка, 2011. — С. 73–75.
- [3] Pietrasiewicz A. Make your own USB gadget // Открытые технологии: сборник материалов Тринадцатой Международной конференции разработчиков и пользователей свободного программного обеспечения Linux Vacation / Eastern Europe 2017, Гродно, 22–25 июня 2017 г. — С. 19–22.
- [4] *Костюк, Д.А.* Построение практикумов по программированию встраиваемых систем / Д.А. Костюк, И.С. Кутень, В.С. Разумейчик // Десятая конференция «Свободное программное обеспечение в высшей школе»: тезисы докладов Переславль, 24–25 января 2015 года. — М.: Альт Линукс, 2015. — С. 14–18.

Роман Симаков

Муром, ООО «РЕД СОФТ»

Проект: СУБД Ред База Данных, Муромский институт ВлГУ

<http://www.reddatabase.ru/>, <http://www.mivlgu.ru/>

Использование СУБД Ред База Данных в МИВлГУ

Аннотация

Компания РЕД СОФТ давно и тесно сотрудничает с кафедрой «Информационные системы» Муромского института Владимирского государственного университета. Основной состав Муромского отделения компании составляют выпускники этой и других кафедр вуза. Два основных продукта компании РЕД СОФТ — СУБД Ред База Данных и ОС РедОС, в свою очередь, активно и плодотворно используются в образовательных процессах. Прослеживается необходимость масштабировать этот опыт на другие вузы.

История взаимодействия

Филиал компании в Муроме открылся в 2006 году с целью развития СУБД Firebird в самостоятельный продукт. Кафедра «Информационные системы» (далее кафедра ИС) стала основным поставщиком кадров. Аспиранты и талантливые студенты старших курсов получили возможность заниматься реальным проектом. Будучи преподавателями, разработчики вместе с собственным ростом привлекали к работе над проектом и других студентов. Не все справлялись, но к счастью, встречалось много талантливых специалистов, которые и пополняли ряды сотрудников.

Часть сотрудников продолжила и педагогическую деятельность, воспитывая новые кадры. Таким образом связь производства и образования всегда присутствовали в жизни компании РЕД СОФТ и МИВлГУ.

Операционная система

С 2014 года РЕД СОФТ разрабатывает свою операционную систему Ред ОС. К настоящему моменту времени импортозамещение в сфере ОС идёт полным ходом. Многие отечественные разработчики предлагают свои решения, в основном на базе Линукс. Это означает

что те студенты, которые сейчас поступают учиться, а после получения диплома пойдут работать в госструктуры, должны будут обладать всеми необходимыми навыками. Знать Windows не только недостаточно, но во вполне определённом круге организаций даже бесполезно! Это подтолкнуло состав кафедры ИС к глубокой переработке рабочих программ. Компания РЕД СОФТ предоставила образовательные лицензии и провела курс обучения преподавателей кафедры.

Много вопросов возникает по программам аналогам. Это действительно сложный вопрос. Ведь заменять программы в лабораторных надо теми, которыми они импортозамещаются! Этот вопрос до сих пор открыт и вероятно здесь можно объединять усилия разных вузов и компаний поставщиков решений.

Тем не менее некоторые дисциплины постепенно переводятся на работу с Ред ОС — там где аналоги очевидны или сами программы кроссплатформенные, отдельные курсы администрирования ОС или сетей, программирование и т.п.

СУБД для бакалавров

СУБД Ред База Данных является основной СУБД при подготовке бакалавров на курсах «Управление данными» и «Базы Данных» где подготавливаются пользователи, прикладные программисты, администраторы. Исторически использовался Interbase, который был включён в состав Delphi — основного языка разработки на курсах программирования. Далее его заменил Firebird. Delphi давно не преподаётся, но это не мешает использовать Firebird/RedDatabase в учебном процессе.

В результате курсов студенты обучаются всем необходимым навыкам по работе с СУБД Ред База Данных. В тоже время они могут выяснить любой нюанс поведения СУБД, почему сделано так, а не иначе, ведь им преподают разработчики!

Магистратура для СУБД

Тем, кому интересно и учиться, и развиваться далее, предлагается поступить в магистратуру. По программе магистров есть дисциплина «Разработка СУБД». Здесь в полной мере раскрывается весь смысл взаимодействия и открытости кода СУБД Ред База Данных. Безусловно любой вуз может изучать исходники открытых СУБД, до-

рабатывать её, но когда есть «доступ» к разработчику, он тебя обучает, есть у кого спросить, то и разрабатывать, и обучатся эффективнее, быстрее и интереснее.

РЕД СОФТ в свою очередь получает возможность в буквальном смысле слова готовить себе кадры. Если магистр обучается и разбирается, то это уже готовый, ценный специалист с гарантированным трудоустройством.

Заключение

Симбиоз производства и образования возможен и идёт на пользу. Компания, производящая открытые продукты, легче может пойти в вузы и более целенаправленно развивать нужную ей компетенцию. От этого выигрывают все:

1. Студенты имеют возможность на этапе обучения решать «боевые» задачи, изучают продукты, которые востребованы в государстве и коммерческом секторе.
2. Институт готовит выпускников, ориентированных на отечественные продукты, согласно текущей ситуации в стране. Это конкурентноспособные специалисты на рынке труда.

Компания РЕД СОФТ готова масштабировать свой опыт на другие вузы и открыта для любых предложений.

Иван Панченко¹, Евгений Моргунов², Борис Новиков³, Павел Лузанов, Егор Рогов¹

¹Москва, ²Красноярск, ³Санкт-Петербург, ¹Postgres Professional, ²СибГУ им Решетнева ³СпбГУ

<https://postgrespro.ru/education>

PostgreSQL в высшем образовании

Аннотация

После продолжительной работы при поддержке компании Postgres Professional были выпущены учебники по СУБД для высшей школы, базирующиеся на СУБД PostgreSQL. Начало работы над этими учебниками анонсировалось ранее на этой конференции. В докладе будут обсуждены сами учебники, опыт работы над ними, будет рассказано об

учебных курсах на их базе и вообще о состоянии преподавания СУБД в России.

Обучение в области компьютерных наук и прикладной информатики в большой степени «захвачено» крупными поставщиками программного обеспечения – такими как, например, компании Oracle и Microsoft. Действуя в своих интересах, они навязали свои продукты в качестве единственных примеров для изучения, что привело к снижению кругозора выпускников и снижению качества образования. К тому же, продукт с закрытым кодом не позволяет узнать, что у него внутри, и тем более попробовать сделать что-то своё. Такое образование скорее пригодно для прикладной магии, чем для специалистов в области информационных технологий: они получают набор приёмов для решения конкретных задач, без глубокого их понимания и без умения создавать новое.

Напротив, при использовании Open Source, всегда есть возможность познакомиться с внутренним устройством продукта с любой степенью подробности и глубины, и попробовать свои силы в его изменении. Поэтому проводимая компанией Postgres Professional работа по продвижению СУБД семейства Postgres в образовании имеет большое значение в масштабах всей отечественной высшей школы.

Эта деятельность началась с перевода документации (которая имеет английский оригинал в формате SGML) на русский язык. Это около 1500 страниц текста, и работа заняла примерно год. При этом была разработана технология с использованием GNU gettext, упрощающая перевод последующих версий.

Затем была П. Лузановым и Е. Роговым разработана свободно распространяемая брошюра «PostgreSQL для начинающих» [1], которая разошлась более чем в 1.5 тыс. печатных экземпляров, при этом оставаясь доступной для скачивания в формате PDF.

Е.П. Моргунов в СибГАУ им. Решетнёва а течение нескольких лет вёл преподавание языка SQL на базе СУБД PostgreSQL, и на базе его курса им при поддержке и содержательной помощи компании Postgres Professional был написан учебник «PostgreSQL. Основы языка SQL» [2], выпедший в 2018 году в издательстве BHV. Это первая часть курса, в который войдут также ещё две книги, посвящённые расширенным возможностям SQL и прикладному серверному программированию.

Профессор СПбГУ Б.А. Новиков, при поддержке Postgres Professional, написал учебник «Основы технологий баз данных» [3], являющийся более глубоким университетским курсом, описывающим теорию баз данных, используемые в них алгоритмы, и их связь с прикладным использованием СУБД. Первая часть этого учебника вышла в издательстве ДМК, вторая часть в настоящее время находится на редактировании.

Все эти учебники, будучи изданными на бумаге, остаются доступными в формате PDF на сайте Postgres Professional.

В начале 2019-го учебного года по двум выпущенным учебникам были прочитаны факультативные курсы на факультете ВМК МГУ и на ФКН НИУ ВШЭ, видеозаписи которых будут опубликованы.

В нескольких университетах страны Postgres уже используется в преподавании, и мы надеемся на расширение их круга.

Литература

- [1] П. Лузанов, Е. Рогов, И. Лёвшин Postgres – первое знакомство. <https://postgrespro.ru/education/books/introbook>
- [2] Е. П. Моргунов PostgreSQL. Основы языка SQL. <https://postgrespro.ru/education/books/sqlprimer>
- [3] Б. А. Новиков, Е. А. Горшкова Основы технологий баз данных. <https://postgrespro.ru/education/books/dbtech>

Михеев Андрей Геннадьевич

Москва, НИТУ МИСиС, ООО «Процессные технологии»

Проект: RunaWFE Free <http://runawfe.org/>

Проект RunaWFE Free. Реализация транзакций для курса процессного управления предприятием

Аннотация

Для обучения студентов по специальности «бизнес-информатика» в системе RunaWFE Free реализован механизм, основанный на особенностях процессного управления, поддерживающий выполнение нескольких действий бизнес-процесса в рамках одной транзакции. При

этом для работы с данными не требуется установки полноценного сервера баз данных. При наличии на компьютере LibreOffice Writer или MS Excel, студенты могут хранить данные в виде таблиц, расположенных на листах формата MS Excel, и корректно изменять данные действиями бизнес-процесса, объединёнными в одну транзакцию.

Процессное управление

Процессное управление предполагает, что деятельность предприятия можно представить в виде множества выполняющихся бизнес-процессов. Оно эффективно для предприятий, в производственной деятельности которых происходит многократное повторение заранее известных цепочек действий, совершаемых различными исполнителями. Такими предприятиями является большинство офисных компаний, занимающихся работой с документами. Также использование процессного подхода эффективно на предприятиях, деятельность которых описывается чёткими регламентами, например, — в органах государственного управления.

В НИТУ МИСиС для обучения студентов построению бизнес-процессов был создан курс процессного управления предприятием. В рамках курса студенты изучают теорию процессного управления, получают практический опыт разработки и исполнения бизнес-процессов. В частности, в рамках курса производится обучение студентов работе с внешними данными. Курс читается студентам, обучающимся по специальности «бизнес-информатика». Так как они не знают теорию баз данных, в свободной системе RunaWFE Free [1] для них был разработан упрощённый механизм работы с данными [2], позволяющий проводить отдельные операции с реляционными таблицами, расположенными на листах документов формата MS Excel.

Однако, оказалось, что возможности выполнения отдельных операций с данными недостаточно. В итоговых контрольных работах курса требуется выполнение нескольких действий бизнес-процесса в рамках одной транзакции. Поэтому в 2018 г. в системе RunaWFE Free был реализован упрощённый механизм транзакций, основанный на особенностях процессного управления.

Реализация механизма транзакций для узлов-действий бизнес-процессов в системе RunaWFE Free

Объясним реализацию транзакций в системе RunaWFE Free на примере бизнес-процесса обновления остатков товаров на складе. Учебный бизнес-процесс представляет собой цикл, обновляющий остатки (См. Рис. 1). Его подпроцесс «Обновить остатки» является транзакционным — обведён двойной рамочкой.

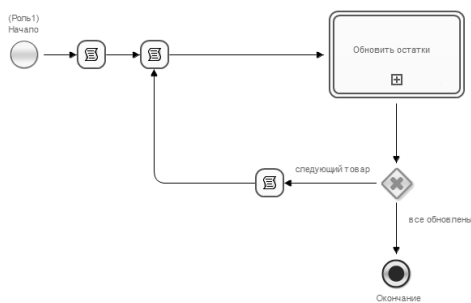


Рис. 1: Бизнес-процесс обновления остатков товаров

Схема подпроцесса приведена на Рис. 2. Подпроцесс-композиция обновления остатка состоит из трёх операций — «чтение», «модификация», «запись». При его выполнении необходимо гарантировать что во время исполнения этих операций не будет изменено текущее количество обрабатываемого товара, пока не выполнится запись нового значения

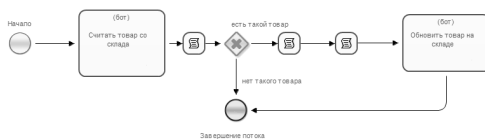


Рис. 2: Подпроцесс-композиция обновления остатка

Обновление остатков товара выполняется при помощи трёх последовательных операций — «чтение», «модификация», «запись»:

- Считывание текущего количества товара из внешнего хранилища («Чтение») в переменную
- Вычитание количества заказанного товара из полученного текущего количества («Модификация»)
- Обновление остатка во внешнем хранилище («Запись»)

Для корректности данных в таблице необходимо гарантировать что во время выполнения этих операций до записи нового значения не будет изменено текущее количество обрабатываемого товара.

Для этого в свойстве «Транзакционный» для подпроцесса-композиции устанавливается значение «Да» (См. Рис. 3). Этот признак обеспечивает блокировку задач бота используемого внутри композиции (бота работы с внешним хранилищем) до момента времени, когда все точки управления выйдут из неё.

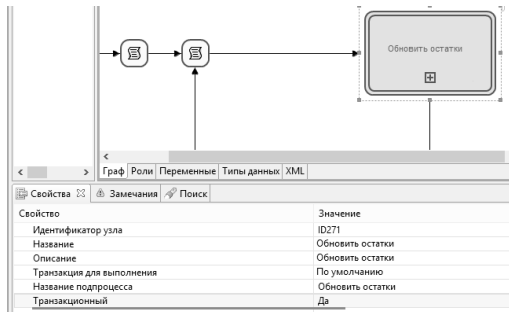


Рис. 3: Признак транзакционности подпроцесса-композиции.

Для считывания текущего количества товара из внешнего хранилища используется задача бота с обработчиком «Внешнее хранилище данных», действие — SELECT (См. Рис. 4).

Для изменения количества используется формула-обработчик в задаче-сценарии (См. Рис. 5)

Для изменения текущего количества товара во внешнем хранилище используется действие UPDATE (См. Рис. 6).

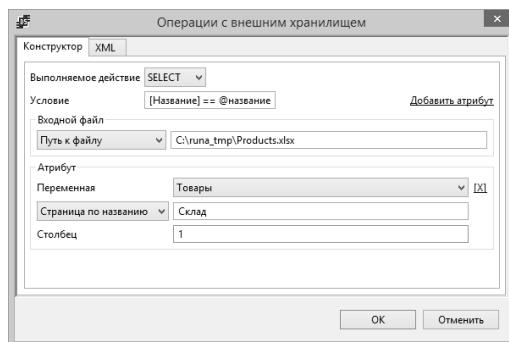


Рис. 4: Считывание текущего количества.

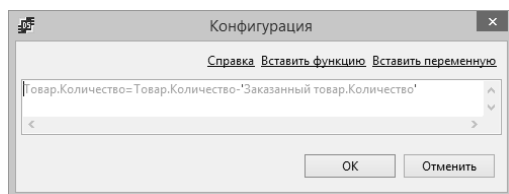


Рис. 5: Изменение количества.

Для правильной работы внутреннего подпроцесса-транзакции на сервере для бота «Бот работы с внешним хранилищем» должно быть установлено свойство «Транзакционный» (см. Рис. 7)

После начала выполнения ботом работы с внешним хранилищем задания «считать товар со склада» и до выхода всех точек управления из подпроцесса-транзакции (или срабатывания таймаута) бот будет выполнять задания только из этого подпроцесса, остальные задания он будет игнорировать. После того, как все точки управления уйдут из подпроцесса-композиции, блокирование заданий бота будет снято.

Литература

- [1] Ссылка на сайт проекта RunaWFE Free <http://wf.runa.ru/> <http://runawfe.org/>

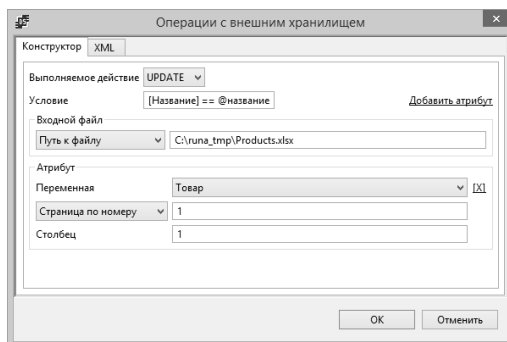


Рис. 6: Изменение текущего количества товара во внешнем хранилище.

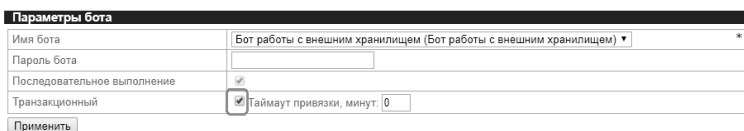


Рис. 7: Признак транзакционности для бота.

- [2] Михеев А. Г. Методика обучения работе с данными в курсе процессного управления // Открытое образование, 2016 — № 6 С. 4 — 8

Мария Петрова

Санкт-Петербург, «Базальт СПО»

<https://www.basealt.ru/partners/integration/certification/>

Чего мы ждём от специалистов, проходящих сертификацию, и как им к этому подготовиться.

Аннотация

Любая система оценки качества или соответствия — это всегда некоторая формальность и неизбежный поиск компромисса между

идеальными ожиданиями и суровой реальностью. Развивая программу обучения персонала, проводящего информационно-технологическое обслуживание разрабатываемой компанией «Базальт СПО» программной платформы, мы попытались облегчить этот поиск, сделать со своей стороны шаги навстречу требованиям отрасли, приблизить процесс сертификации к решению типовых задач. В докладе хочется рассказать об опыте сближения обучения, сертификации и эксплуатации программных продуктов, обсудить перспективы развития в этом направлении.

Персонал отечественных предприятий начал рассматриваться в качестве объекта сертификации примерно на рубеже тысячелетий. Сертификация специалистов, как процедура подтверждения соответствия определённым профессиональным стандартам или квалификационным требованиям, не заменяет базовое образование и не ставит его под сомнение, сертификат не противопоставляется практическому опыту, а напротив, подтверждает его. Тем не менее, обычно сертификация тесно связана с дополнительным обучением, которое может быть самостоятельным или проходить в специализированном учебном центре.

Авторизованные учебные курсы для подготовки к сертификационному экзамену по продуктам компании «Базальт СПО» направлены на получение необходимых знаний и навыков для работы с линейкой ОС Альт. И это главное в процессе обучения. Мы всегда помним, что сертификация не является самоцелью, а лишь инструментом подтверждения квалификации системных администраторов, на чьи плечи несомненно ляжет большая часть ответственности при реализации плана перехода на российское программное обеспечение. Не только пользователи информационных систем, но и ИТ-специалисты долгие годы находились в пространстве зарубежных решений и технологий, и проложить короткий путь к использованию отечественного ПО — это задача, наверное, по масштабам сродни строительству магистрали через всю страну. Тем не менее, мы понимаем важность обучения при переходе к новому программному окружению и готовы вести работу в этом направлении.

При небольшом объёме учебного курса особое значение приобретают практические работы. Грамотно составленное задание на лабораторную работу, возникающие вопросы и найденные решения в ходе её выполнения, непосредственное общение с преподавателем — согласитесь, всё это трудно переоценить. Мы целенаправленно с первых

шагов готовим слушателей к написанию сценариев на языке командного интерпретатора. Обычно подобные практические задания состоят из нескольких частей: стартового развёртывания рабочей среды, последующего тестирования основного процесса и заключительного очищающего сценария. Позже, при прохождении процедуры сертификации, специалист встретится с аналогичной схемой тестовых заданий, но ещё раз подчеркну, что приоритетом обучения является не слепое натаскивание на прохождение теста, а формирование осмысленных, устойчивых и воспроизводимых профессиональных навыков. Лабораторные работы заведомо не отличаются повышенной трудностью, но зато прекрасно укладываются в философию Unix: «пишите программы, которые делают что-то одно и делают это хорошо; пишите программы, которые бы работали вместе».

Другой отличительной чертой базальтовских курсов становится практика использования live-usb, с которых в любом классе можно загрузить актуальную версию системы, а потом уже после окончания курса ещё раз проработать всю историю команд, если возникнет необходимость.

Так же, для последующей самостоятельной подготовки слушателям предлагается полный набор так называемых скринкастов. Запись живых занятий с комментариями и ответами преподавателя является обязательным элементом авторизованных курсов.

Сочетание перечисленных особенностей помогает достичь хороших результатов обучения и действительно подготовить соискателей к успешному прохождению сертификации. Процедура сертификации проходит согласно Регламенту, который доступен на сайте <http://www.basealt.ru/>. Она начинается, если считать подачу заявки и оформление договора нулевым этапом, с личного собеседования с кандидатом. Затем следуют выполнение задания и подготовка сопроводительных инструкций, проведение автоматизированного тестирования, обсуждение результатов и, как итог, заслуженный документ о присвоении статуса сертифицированного специалиста.

Никита Бесшапошников, Михаил Дьяченко, Анатолий Кушниренко, Михаил Кузьменко, Александр Леонов, Максим Матюшин, Кирилл Прокин

Москва, Московский государственный университет, ФГУ ФНЦ НИИСИ РАН, Московский педагогический государственный университет, АО «АйТи Сфера»

Применение методов распознавания в практикумах по программированию

Аннотация

Технологии по распознаванию образов с помощью нейронных сетей можно применять и в образовательных процессах. При этом автоматизированные практикумы, как компоненты цифровой образовательной среды, дополняются новыми функциями, ускоряющими и упрощающими процесс обучения. Обсуждаются возможные точки приложения технологий машинного обучения в автоматизации учебного процесса на примере практикумов по освоению основ программирования для дошкольников и младших школьников.

Процессы цифровой трансформации проникли в практически все сферы жизнедеятельности человека, от науки и производства до домохозяйств и образования. Последнее имеет особую ценность в современном быстроменяющемся мире. Объём информации, необходимый для усвоения учащимися всё возрастает, а динамика смены профессиональных ориентиров в потребностях общества увеличивается с невероятной скоростью. При этом вычислительные мощности, достигнутые современной микроэлектроникой, позволяют использовать для решения широкого круга задач не только суперкомпьютеры, серверы и десктопы, но и мобильные системы, смартфоны и планшеты.

Предпрофессиональную подготовку жителя XXI века сейчас нужно начинать как можно раньше, ещё в дошкольном возрасте. Человек, не имеющий четвёртую компетентность, — программирование (кодирование), — в дальнейшем останется функционально неграмотным и будет выброшен на «обочину жизни» точно также, как ранее там оказывались не умеющие читать, писать, считать (три другие необходимые компетенции XX века). В пропедевтическом курсе по алгоритме (программированию) для дошкольников и младших школьников, который с успехом осваивают 100% обучаемых, дети

осваивают основы составления алгоритмов и терминологию, без которых невозможно дальнейшее развитие, интеллектуальный рост и образование полноценного члена современного общества. Эти курсы и практикумы разработаны группой учёных в ФГУ ФНЦ НИИСИ РАН и широко апробированы в России. Учащиеся на практических занятиях в системе ПиктоМир «собирают» программы по управлению виртуальными роботами из команд-пиктограмм, осваивают основы алгоритмизации и знакомятся с основами программного управления роботом-исполнителей [1].

Обучение в системе ПиктоМир рекомендуются начинать в возрасте 6 лет, что соответствует старшей группе дошкольных образовательных организаций. Здесь сказываются не только ограничения СанПиН на время работы с планшетными компьютерами, но и эргономические ограничения опорно-двигательного аппарата малышей. Однако, даже дети трёх-четырёх лет с удовольствием складывают картинки из деревянных кубиков, при этом развивая ещё и мелкую моторику.

В качестве эксперимента, детям предлагалось собирать программу для управления виртуальным роботом не на экране планшета, а прямо на столе из деревянных кубиков с нарисованными на них командами. Основной вопрос, который при этом оставался не решённым, как исполнить и проверить составленную программу на компьютере? Здесь на помощь пришла технология распознавания образов, фотографий программ, составленных учащимися.

Ученик составлял программу, фотографировал её на мобильное устройство и результат, распознанная программа, появлялась в ПиктоМире, как будто её составили вручную. При этом задача получения в системе ПиктоМир программы ученика, составленной из кубиков, фактически сводилась к созданию алгоритма эффективного распознавания картинок-пиктограмм на кубиках одинакового размера, разложенных на плоскости при ручной съёмке в 3-х мерном пространстве при различных сопровождающих искажениях.

Первоначально были разработаны алгоритмы с использованием библиотеки `openCV` для детектирования границ пиктограмм по цвету и дальнейшую классификацию на плоскости по заранее предложенному в задании шаблону программы. Такой автоматизированный подход в условиях хорошей освещённости границ изображений позволил эффективно распознавать составленную программу. Напротив, в условиях неравномерной освещённости, присутствии в кадре посторонних

объектов, незначительных перспективных искажениях эвристические алгоритмы определения относительного местоположения пиктограмм давали высокий процент ошибок [2].

В силу сложности решаемой задачи в дальнейшем были применены методы машинного обучения. Стандартные модели машинного обучения (SVM, Random Forest, модель логистической регрессии и т. д.) достаточно хорошо показали себя в задачах, связанных с обработкой скалярных или векторных признаков небольшой размерности. Для обработки изображений были использованы нейронные сети большой глубины: SSD Mobilenet, Faster RCNN Resnet, Faster RCNN Inception. Разработанные алгоритмы распознавания дали очень хороший результат и позволили предложить решение вопроса переноса детектирующего алгоритма на конечные мобильные устройства [3].

Работы выполнены в рамках темы «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов» (тема госзадания РАН 0065-2019-0010).

Литература

- [1] Кушниренко А. Г., Леонов А. Г., Райко М. В. Проведение годового цикла занятий «алгоритмика для дошкольников» в подготовительных группах ДОУ. // ВОСПИТАНИЕ И ОБУЧЕНИЕ ДЕТЕЙ МЛАДШЕГО ВОЗРАСТА. Издательство: Издательство «Мозаика-Синтез» (Москва) 2018
- [2] Беспалошников Н. О., Кузьменко М. А., Леонов А. Г., Матюшин М. А. Автоматизация разметки набора данных для нейронных сетей // Вестник Кибернетики №4(32) 2018
- [3] Беспалошников Н. О., Кузьменко М. А., Леонов А. Г., Матюшин М. А. Некоторые вопросы эффективности детерминированных алгоритмов распознавания образов с помощью библиотеки OpenCV // Труды НИИСИ РАН. 2018. №2. Том 8.

Никита Бешапошников, Анатолий Кушниренко, Александр
Леонов, Александр Малый

Москва, Московский государственный университет, ФГУ ФНЦ НИИСИ РАН,
Московский педагогический государственный университет

Проект двуязыковой пиктограммно-текстовой учебной среды программирования ПиктоМир-К.

Аннотация

В эпоху всемирной цифровизации одной из фундаментальных компетенций становится программирование (алгоритмизация, кодирование). По действующим ФГОС основы алгоритмики в курсе Информатика и ИКТ начинают систематически изучаться в конце основной школе. При радикальном снижении возраста знакомства с программированием, критичной становится сложность перехода от пиктограммных языков программирования к текстовым. Предлагается методика и подходы к раннему обучению программированию, начиная с дошкольных образовательных организаций, а также программный комплекс — двуязыковая пиктограммно-текстовая учебная среда программирования ПиктоМир-К, при использовании которой существенно упрощается процесс перехода от пиктограммного стиля программирования к текстовому, что позволяет перенести освоение основ алгоритмизации в младшие классы.

Несмотря на то, что изучение основ программирования традиционно происходит в основной школе, мировая и отечественная практика говорит о возможности снижения возраста знакомства с азами алгоритмизации на дошкольный уровень. Дети возраста 6+ способны эффективно осваивать алгоритмическую грамотность в игровой форме. При использовании системы пиктограммного программирования ПиктоМир, разработанной учёными в ФГУ ФНЦ НИИСИ РАН, дошколята успешно осваивают методы программирования, с использованием таких управляющих конструкций как повторитель и вспомогательный алгоритм, в рамках годового курса, самостоятельно составляя в годовом курсе 120-150 программ [1].

Программирование в системе ПиктоМир с успехом используется в качестве пропедевтики курсов по программированию и для студентов вузов. В полугодовом выравнивающем курсе «Азы программирования» первые несколько занятий проводятся в среде ПиктоМир, после чего происходит переход в комфортную, как нам ранее казалось,

учебную систему программирования КуМир по специально разработанной методике. Опыт показал, однако, что переход от пиктограммного стиля программирования даже по специально разработанной методике оказывается достаточно сложным не только для детей, но и для взрослых.

Методика перехода включает в себя решение одних и тех же задач в различных программных системах (текстовой и пиктограммной), на разных языках программирования, что практически исключает алгоритмическую сложность из перечня испытываемых обучающимися трудностей. Однако интерфейсные различия в текстовой и пиктограммной системах программирования, а также появление новой сущности — синтаксических ошибок, отсутствующей в пиктографической системе, создают серьёзные проблемы для обучающихся.

Двухязыковая пиктограммно-текстовая учебная среда программирования ПиктоМир-К предназначена для облегчения такого перехода для учащихся, путём предоставления возможности одновременного использования пиктограммного и текстового представления программы, и исключения возможности появления синтаксических ошибок, не только в пиктограммном, но и в текстовом режиме. Последнее обеспечивается синтаксически ориентированным редактором. При составлении простейших учебных программ процесс редактирования практически не усложнится, а результат, текст на школьном алгоритмическом языке, окажется полностью совместим с системой КуМир[2]. Естественно, что все программные исполнители систем ПиктоМир и КуМир окажутся доступны для использования в среде ПиктоМир-К.

При работе в системе, в текстовом режиме исключено появление синтаксических ошибок, так как в процессе редактирования программа составляется из образцов-пиктограмм, которые можно добавить только в специально выделенные места текста. Перетаскиваемая пиктограмма раскрывается в соответствующий ей текст на школьном языке программирования. Полученный такими преобразованиями текст может содержать выделенные места для вставки пиктограмм-выражений. Синтаксические конструкции также имеют поля для вставки логических или арифметических выражений. При этом, интерфейс написания логических и арифметических выражения на каждом шаге поддерживает корректный результат. Например, при вводе выражения $(3 + 5) \times 2$, невозможны некорректные промежуточные выражения такие, как «3+», «(3 + 5» или «(3 + 5)×», по-

скольку на каждом шагу выражение автоматически модифицируется так, чтобы что быть синтаксически корректным. [3].

Работы выполнены в рамках темы «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов» (тема госзадания РАН 0065-2019-0010).

Литература

- [1] Кушниренко А. Г., Леонов А. Г., Райко М. В. Проведение годового цикла занятий «алгоритмика для дошкольников» в подготовительных группах ДОУ. // ВОСПИТАНИЕ И ОБУЧЕНИЕ ДЕТЕЙ МЛАДШЕГО ВОЗРАСТА. Издательство: Издательство «Мозаика-Синтез» (Москва) 2018
- [2] Бешапошников Н. О., Дедков А. Н., Ерёмин Д. Б., Леонов А. Г. Система программирования Кумир 2.x // Труды НИИСИ РАН. 2015. №1. Том 5.
- [3] Кушниренко А. Г., Леонов А. Г. Архитектура смешанной пиктограммно-текстовой системы программирования для дошкольников и младших школьников // Вестник кибернетики, издательство СурГУ (Сургут), том 28, № 4, 2017.

Михаил Быков

г. Москва, <http://diglossa.org>

<http://diglossa.org/diglossa.js>

diglossa.js — назад к тексту

Аннотация

diglossa.js построена на голом исходном авторитетном тексте, в который не вносятся изменения и который хранится в системе контроля версий (github.com). К тексту последовательно пристраиваются слои (layers), привязанные к конкретному месту текста якорями. Таким образом можно получить результат любой сложности, сохранив исходный текст в неприкосновенности. Добавочными слоями к тексту могут быть: предисловия, комментарии, пагинации, параллельные тексты, логические разметки, примечания издателей, любая мета-информация и т.п. Этот подход даёт возможность использования свободной лицензии на

результат при ограниченных лицензиях на исходные тексты. Использовать `diglossa.js` не сложнее, чем хорошо известный `FB2-Reader`, подготовка текста занимает минуты.

Лирика

Носители письменного текста начинались с глиняных табличек и листьев растений, сейчас носители тех же самых текстов превратились в энтерпрайз-системы с джавой, обслуживаемые коллективами учёных и айтишников. Тексты были редки и дороги, и сейчас доступны обрывки и некачественные списки, фейки. Собственно авторитетные тексты так же редки и ещё более дороги, чем в момент своего появления. Тексты обрастают грязью, как корабли ракушками. Создаются всё более сложные системы, начиная с `MS-Word`. Текст как просто текст в обороте отсутствует. Даже просто скопировать аккуратно текст обычными средствами невозможно — если нужно аккуратно скопировать (например `ssh-ключ`), приходится использовать специальную утилиту — `ssh-copy-id`. А кроме неё, ничего для работы с просто текстом нет.

Количество информации удваивается каждые два года. Измеряется уже то-ли в зета-байтах, то-ли в пета-байтах, не помню. Но мы умнее и грамотнее в той же пропорции очевидно не становимся. Из этого следует, кстати, что наши знания с информацией не связаны, их графики не коррелируют. И спросите в любой аудитории, что такое информация, и что такое знания, сколько публики, столько будет и ответов.

Сами слова обрастают грязью, как корабли ракушками. В 5 веке китайский монах Сюаньцзан написал трактат «Записки о западных странах», в 15 веке У Чэнъэнь написал его пересказ «Путешествие на Запад», толстый том, один из 4 классических китайских романов, со всеми тантрическими чудесами, демонами и заклинаниями, в 21 веке в Голливуд в Китае снимает серию блокбастеров «Путешествие на Запад — покорение демонов» с миллиардными оборотами. О взрывном увеличении потока фейков даже и говорить скучно. Грязью обрастает любое слово, видимо, это закон природы.

Недавно ещё в почте использовался текст, сейчас это без исключения `html`. Текста без `www-urlencoded` больше нет. Текст, как сказал когда-то Ницше, умер.

Технические подробности

Утилита `diglossa.js` имеет целью работу с текстом. Можно использовать её простейшим образом. Можно взять любой табличный редактор, типа MS-Excel, и в нём расположить тексты (один или более) в колонках, сохранить файл и открыть его в Диглоссе. Читатель сразу получает удобный интерфейс, полнотекстовый поиск и интеграцию с утилитами типа Морфея (при проводе мышью над словом сразу получает результат обработки слова в Морфее). В тексты можно вносить исправления и сразу видеть результат. Убедившись, что результат вам нравится, то есть смыслы всех текстов в каждой строке соответствуют друг другу, можно опубликовать результат в вебе в один клик.

Но кроме этого, можно также опубликовать исходные тексты, в виде текстов, в специальной репозитории на github.com, или в любой другой системе контроля версий. Следующий читатель может загрузить любой из текстов, или все, уже из репозитория.

Вариант с табличным редактором добавлен для удобства работающих с текстами. Основным же вариантом является первичное чтение текстов из файлов дерева директорий, или из сети. Таким образом можно получить структуру текста любой сложности. Работа начинается с чтения файла манифеста, `anu-name.json`, в котором есть единственное обязательное поле: путь к дереву директорий. Или адрес ресурса. Может быть добавочный `the-same-name-local.json`, который содержит сведения, которые публиковать не нужно.

Если все ресурсы указаны как удалённые, для распространения (публикации) результата достаточно опубликовать только файл манифеста. И затем открыть его в Диглоссе.

К исходному тексту могут быть добавлены не только параллельные тексты целиком, но и любой текст к любому месту в исходном тексте. Это достигается системой слоёв (`layers`) и якорей. Собственно, параллельные тексты являются одним из примеров слоя. Другие примеры: предисловия и справочники, комментарии, пагинации, логические разметки, картинки и примечания издателей, любая метаинформация и режим учебника, когда исходный текст представляется вставками, цитатами в обрамлении нового текста. (В публикуемых пакетах по большей части все эти возможности пока не реализованы). Использовать сложную систему слоёв без необходимости не обязательно. Например, в большинство текстов логическая разметка может быть внесена попросту в текст. Сейчас нет возможности добавлять в

Electron.js плагины, но в близком будущем обработку слоёв можно будет добавлять по желанию.

Поскольку используются только известные и популярные web-технологии (Node.js, Electron.js и CouchDB), то можно так же добавить свои стилевые таблицы. CouchDB нужен потому, что Diglossa.js и Morpheus.js работают в связке, и имеют возможность создания на лету локального словаря для конкретного текста или группы текстов. При этом читатель будет иметь возможность вносить добавления и исправления в словари на лету. Следовательно, нужна система синхронизации между всеми мобильниками, десктопами и серверами, что и обеспечивает CouchDB. Кроме этого, Node.js и CouchDB используют JSON как основную структуру представления данных, то есть дерево. А в языкознании все данные имеют структуру дерева. Почему обычно в этих задачах используется Питон, для меня загадка.

При этом исходный текст в обычной работе никак не затрагивается. А если в него действительно вносится изменение (исправляется опечатка), в репозитории результат появится в виде git merge request, и может быть принят или отклонён.

Лицензия

Таким образом, появляется возможность использовать свободную лицензию на результат. Старожилы конференции помнят, что несколько лет назад обсуждалась возможность использования свободных лицензий для художественных, а не программных тестов, музыкальных и изобразительных произведений и т.д. Вывод был сделан не утешительный, — свободные лицензии лицензии в художественных текстах использовать невозможно, поскольку сама их художественность связана с личностью автора, и любое внесение изменения автоматически нарушит не отменяемое никакой лицензией авторское право. В случае же данного подхода появляется пространство для работы над текстом, не вносящее изменений в текст. И если лицензия исходного текста позволяет только его распространение, то он может быть включён в результат, сам имеющий свободную лицензию.

Лирика

Таким образом, diglossa.js является прообразом библиотеки будущего, когда распространяется не книга в переплёте, а только лишь

необходимая часть текста с необходимым аппаратом. Вспомните здесь об усилиях, предпринимаемых государством по сохранению системы закорючлых бумажных библиотек. Превращению простого текста в чудовищные информационные системы может быть противопоставлена культура распространения авторитетного текста со свободной лицензией. А касте служащих информационных систем — сообщество грамотных читателей.

Литература

- [1] Брукс, Ф., Мифический человеко-месяц, или как создаются программные системы., <http://www.lib.ru/CTOTOR/BRUKS/mithsoftware.txt>

Андрей Савченко
Москва, Базальт СПО

Взгляд изнутри на Google Summer of Code

Аннотация

Программа Google Summer of Code существует уже 15-й год и является хорошим способом для студентов присоединиться к разработке разнообразных свободных проектов, получить богатый технический и социальный опыт. Проекты в свою очередь получают новых разработчиков и решение своих задач. В данной работе рассказывается о том, как устроен этот процесс глазами ментора и администратора одного из таких проектов в рамках GSoC, даются практические рекомендации по участию в программе.

Что такое GSoC

Программа Google Summer of Code (GSoC)[1] предоставляет студентам возможность заняться летом разработкой СПО за разумную стипендию вместо иных подработок.

Основные цели программы:

- Привлечение новых разработчиков в сообщество СПО.
- Способствование разработке СПО проектов.
- Предоставить студентам опыт реальной разработки и взаимодействия с сообществом.

Заявка

Принять участие в качестве организации может любой активно развиваемый СПО проект, имеющий в наличии хотя бы двух менторов (разработчиков, готовых быть наставниками для студентов, участвующих в организации).

Программа GSoC нацелена на диверсификацию участников: ежегодно количество участвующих организаций возрастает. При этом никому не гарантируется ежегодное участие: даже организации-ветераны могут не взять, чтоб дать возможность поучаствовать другим, например, Mozilla не приняли в 2015 году.

Сейчас самое время подать заявку! Окно открыто до 6 февраля 20:00 UTC[2].

Если нет желания или возможности принять участие в качестве самостоятельной организации, можно попроситься к одной из «зонтичных» организаций, берущих под своё крыло различные проекты по определённым тематикам. Доступен неофициальный список[3] таких организаций по 2018 году.

Каждая организация подготавливает идеи работ для студентов и наставников студентов для данных работ. Но, кроме этого, у студентов есть право выдвигать собственные проекты. Это сложнее, поскольку требует проработки идеи со стороны самого студента и не всегда возможно найти наставника для таких проектов, но, как показывает практика, если студент предложил свою идею — он по уровню существенно выше среднего.

Список принятых организаций будет доступен 26 февраля. Заявки студенты могут подавать с 25 марта по 9 апреля[2]. Рекомендуется заранее связываться с проектами, в которых студент заинтересован для того, чтоб освоится с проектом, его сообществом, инфраструктурой.

Отбор студентов осуществляется организациями на конкурсной основе. Обычной практикой в дополнение к рассмотрению заявки студента являются небольшие задания, цель которых в проверке знаний студента и навыков работы с инструментарием. Традиционный способ: попросить исправить ряд багов.

Каждая организация запрашивает $[min..max]$ интервал слотов и самостоятельно распределяет по ним желаемые студенческие заявки. Google не назначает принятых студентов, но может отклонить назначенную организацией заявку; обычно это происходит из-за проблем с документами, например, если студент не предоставил документы, что

он является студентом на момент объявления результатов конкурсного отбора (в 2019 году это будет 6 мая[2]).

Самая важная часть студенческой заявки — это детально проработанный план работ, именно по нему проводится экспертиза соответствия студента и проекта. Желательно, чтоб план содержал недельный список планируемой работы. Кроме того, заявка должна содержать обоснование необходимости проведения работ и планируемый результат. Так же важно указывать контакты, часы доступности (поскольку разные участники проекта обычно из разных временных зон) и дни планируемого отсутствия (например, экзамены).

Студенты пишут заявку самостоятельно, но могут предоставлять наставникам черновики для комментариев. Хорошую заявку сложно написать с первого раза, поэтому процесс, обычно, итерационный и начинать его рекомендуется заранее.

Для всех участников очень важно соблюдать установленные сроки подачи заявок и отчётов. Они известны заранее[2] ещё до начала GSoC. При несоблюдении сроков заявка или отчёт приниматься не будут со всеми вытекающими последствиями. Исключений нет и не будет ни для кого ни при каких условиях и ни по каким причинам. Поэтому настоятельно рекомендуется подавать все документы заранее.

Участие

Официально период написания кода начинается 27 мая и длится до 26 августа[2]. По обоюдному согласию между организацией, наставником и студентом возможно начать работу раньше, например, для компенсации времени, выпадающего из-за сдачи экзаменов.

Важной частью работы является регулярное общение между наставником и студентом. Обычно это ежедневные краткие ответы на вопросы и детальное общение в чате еженедельно.

От студента ожидается полноценная работа (30-40 часов) 5 дней в неделю с обязательными еженедельными отчётами о проделанной работе, кроме того, многие организации практикуют краткие ежедневные отчёты: итоги за день и план на завтра.

Затраты времени наставника сильно варьируются в зависимости от уровня подготовки и самостоятельности студента (поэтому не следует брать слабых студентов!), но обычно находится в интервале от

4 до 10 часов в неделю. Поэтому редко когда наставник может вести двух и более студентов.

Основной этап работ делится на три части с ежемесячными отчётами студентов и наставников — они взаимно оценивают работу друг друга и при неуспеваемости студента он может быть исключён из программы. При необходимости по обоюдному согласию наставников и студентов план работ может быть скорректирован для учёта возникших сложностей или прорывов.

Итоги

В результате выполнения работ должен быть сформирован репозиторий с созданным кодом под СПО лицензией. Организация не обязана принимать его в основной репозиторий, но это рекомендуется. Всячески приветствуется закрепление студентов в проектах и продолжение их вклада в будущем — в этом основная цель программы.

Литература

- [1] Google Summer of Code
<https://summerofcode.withgoogle.com/>
- [2] Google Summer of Code 2019 Timeline
<https://developers.google.com/open-source/gsoc/timeline>
- [3] List of GSoC Umbrella Organizations
<https://nikhita.github.io/list-of-gsoc-umbrella-orgs>



Рис. 1: Взаимосвязь IT-науки, IT-бизнеса и IT-образования.

Абрамов С. М.

Переславль, ИПС имени А.К. Айламазяна РАН

Проект «Чуткий дом» как ещё один пример тесной взаимной зависимости IT-науки, IT-бизнеса и IT-образования

Аннотация

Проект «Чуткий дом» рождён Институтом программных систем имени А.К. Айламазяна РАН (один из лидеров российской IT-науки) и компанией «Ботик-технологии» (один из старейших Internet-провайдеров в России), и он ярко иллюстрирует следующую идею (Рисунок 1): невозможно в разрыве друг от друга развивать IT-науку, IT-бизнес и IT-образование, а совместная слаженная работа этих трёх граней даёт отличные результаты в каждой из них.

Проект «Чуткий дом» вырос из бизнес-потребностей компании «Ботик-технологии», осознания некоторого запроса рынка и наличия научного задела и практического опыта как ИПС имени А.К. Айламазяна РАН, так и компании «Ботик-технологии». Большое количество предыдущих проектов накопило базу, позволившую в дальнейшем сформулировать идею «Чуткого дома» и различные детали реализации проекта.

В 2014 году ФАНО России вместе со Сколково проводили конкурс по отбору проектов коммерциализации результатов научных исследований. В рамках участия в данном конкурсе мы и завершили формирование видения проекта «Чуткий дом». Проект стал финалистом конкурса, была создана компания «Чуткий дом», которая в 2015 году получила статус участника Сколково.

В чём основная идея продукта «Чуткий дом»? Если кратко, то «Чуткий дом» это умный дом, который (вдобавок к обычной функциональности умных домов) умеет:

- чувствовать (и даже предчувствовать) проблемы с поднадзорным оборудованием в наших домах;
- оповещать хозяина о проблемах, обеспечивая возможность обеспокоить соответствующее оборудование и, тем самым, парировать проблему.

Целевые объекты: квартиры, коттеджи, дома, поместья, любые иные здания: образование, медицина, торговля, промышленность, бизнес-центры, госучреждения, гостиницы, фермы, овощехранилища. . .

Решаемая проблема: в зданиях много всяких устройств и систем; часто не ясно, они работают нормально или есть проблемы (или проблемы скоро появятся).

Система «Чуткий дом» за счёт сети из сенсоров и исполнительных механизмов реализует **все традиционные возможности обычных систем умного дома:**

- удалённый мониторинг и удалённое управление устройствами и системами дома;
- поддержка параметров в заданных границах: термостатирование, светостатирование, поддержка заданных параметров воздуха и т.п.;
- сценарии настроек — сценарии по пространству, по времени/событию.

За счёт машинного обучения, искусственного интеллекта система «Чуткий дом» реализует самообучение — накопление шаблонов поведения исправного поднадзорного оборудования — и, за счёт этого, обеспечивает «чуткость»:

- классификацию «всё нормально / что-то не так»;
- упреждающий прогноз «через ... некоторое время... возможна проблема»;
- парирование угроз — возможность обесточить соответствующее оборудование.

Какое оборудование в доме может быть поднадзорным? — Любое оборудование: отопительный котёл (дизельный, газовый, пеллеточный и т.д.), накопительные электрические нагреватели воды, водяные помпы (системы водоснабжения), нагревательные шнуры для подогрева труб (защита от заморозки труб), дренажные насосы, септики с аэрацией, холодильники, стиральные или посудомоечные машины, кондиционеры, вентиляционные системы и т.п.

В докладе будет более подробно рассказано про технические аспекты проекта «Чуткий дом», про место и роль Open Source в проекте, про текущую стадию, планы и перспективы коммерциализации, и, конечно, про роль в проекте взаимодействия IT-науки, IT-бизнеса и IT-образования.

Кремер Аркадий Соломонович

г. Москва, Московский технический университет связи информатики (МТУСИ)

Обучение навыкам внедрения отечественных технических средств в национальную ИКТ-инфраструктуру

Аннотация

Существующая организация учебного процесса в вузах не способствует решению важнейших задач подготовки специалистов в сфере развития национальной ИКТ-инфраструктуры с использованием отечественных технологий, а также обеспечения безопасности и доверия при её использовании. Такой мощный ресурс, как образование, используется недостаточно эффективно и целенаправленно для содействия

решению задач обеспечения технологической независимости и информационной безопасности Российской Федерации.

В докладе рассматривается опыт разработки и внедрения в образовательный процесс базовой кафедры Ассоциации документальной электросвязи в МТУСИ типового отечественного модуля изучения интернет-технологий. Ключевой особенностью модуля является использование при его создании российских программно-аппаратных средств и отечественной среды разработки. Реализация проекта позволит совместить обучение основам технологий сети Интернет с изучением возможностей отечественных технических средств и подготовкой кадров для их разработки, внедрения и использования.

В процессе происходящей в настоящее время в Российской Федерации цифровой трансформации подавляющее большинство систем принятия решений и бизнес-процессов в ключевых отраслях экономики и сфере государственного управления реализуются или планируются к реализации с использованием информационно-коммуникационных технологий сети Интернет. В различных информационных системах, доступ к которым осуществляется с использованием сети Интернет, уже сейчас хранятся и обрабатываются значительные объемы информации, в том числе касающейся вопросов государственной политики и обороны, финансовой и научно-технической сферы, частной жизни граждан. Электронные средства массовой информации, информационные системы, социальные сети, доступ к которым осуществляется с использованием сети Интернет, стали частью повседневной жизни россиян. Пользователями российского сегмента сети Интернет в 2017 году являлись более 85 млн. человек.

Глобализация современных информационно-коммуникационных сетей и информационных систем, вынужденное применение при их построении иностранного оборудования и заимствованного программного обеспечения, а также существенное увеличение количества автоматизированных систем управления производственными и технологическими процессами в условиях интенсивного совершенствования средств и методов применения информационных и коммуникационных технологий в противоправных целях, формируют новые угрозы безопасности Российской Федерации, усложняют решение задач по обеспечению защиты интересов граждан и государства в информационной сфере. В этих условиях особое значение должно уделяться подготовке квалифицированных кадров, способных разрабатывать и

внедрять отечественные информационно-коммуникационные технологии.

На содержание этой деятельности существенное влияние оказывают агрессивная политика крупнейших иностранных вендоров-производителей программного обеспечения и оборудования, а также недостаточное внимание, уделяемое технологическому оснащению процесса обучения. В большинстве случаев используются демонстрационные возможности обычного компьютера с комментариями, основанными на доминирующей в информационном пространстве информации о зарубежных технических решениях. Последствием сложившейся ситуации является повсеместное внедрение в учебный процесс изучения продукции иностранных производителей, сформировавших развитую экосистему продвижения своих решений. При этом не уделяется достаточного внимания изучению возможностей и перспектив внедрения отечественных технологий.

Важным направлением решения этой проблемы является создание готовых унифицированных решений, которые могут быть использованы при формировании и реализации образовательных программ. Такой подход обеспечит повышение качества изучения интернет-технологий и будет способствовать повышению привлекательности и конкурентоспособности использующих такие решения образовательных организаций. Изучение интернет-технологий станет целенаправленным, то есть появятся основания для планомерного повышения технической грамотности и активности специалистов по разработке отечественных решений в области интернет-технологий и их применению в условиях цифрового развития.

Руководствуясь этими соображениями, сотрудники базовой кафедры АДЭ в МТУСИ «Технологии электронного обмена данными» предложили разработать и внедрить в образовательный процесс кафедры типовой отечественный модуль изучения интернет-технологий, который должен стать основой для формирования профессиональных знаний и компетенций в области построения, развития и управления сетью Интернет. Важнейшей составляющей данного проекта являются российские программно-аппаратные средства. Использование типового модуля в учебном процессе позволит сформировать у слушателей следующие компетенции (включая, но не ограничиваясь):

- понимание реализации стека сетевых протоколов и принципов работы сети передачи данных на базе IP, организации адресации и маршрутизации, сетевых интерфейсов;
- понимание принципов построения и управления сетью Интернет, взаимодействия сетей связи и обмена трафиком;
- понимание принципов функционирования системы DNS, системы координации и администрирования доменных имен верхнего уровня (TLDs);
- понимание преимуществ и недостатков открытой модели разработки программного обеспечения, сути свободных лицензий и их ограничений, процессов разработки и внедрения новых сервисов и протоколов;
- умение работать с сетевыми утилитами, анализаторами трафика и другими специальными инструментами.

Проект получил необходимое финансирование со стороны Координационного центра национального домена сети Интернет и находится на стадии реализации. В качестве аппаратной основы для формирования модуля используются российские вычислительные комплексы Эльбрус 101-РС и Эльбрус 801-РС, разработанные в ИНЭУМ им. И.С. Брука. В качестве системного ПО в этих вычислительных комплексах применяется операционная система Альт компании Базальт СПО. Выбор ОС Альт обусловлен наличием у Базальт СПО полностью поддерживаемого собственного репозитория (Sisyphus), что позволяет обеспечивать полный жизненный цикл ПО. В конце 2018 года в офисе компании Базальт СПО был сформирован тестовый учебный класс и проведены пробные занятия с обозначенной программно-аппаратной конфигурацией. По результатам были проведены необходимые доработки и отладка прикладного программного обеспечения, сам комплекс был перевезен и смонтирован в МТУСИ.

Кроме аппаратно-программного комплекса, реализующего основные интернет-технологии, модуль включает учебно-методические материалы по теоретической части курса и комплексу лабораторных работ. Апробация модуля на базовой кафедре АДЭ запланирована на первое полугодие 2019 года. Важной перспективой данного проекта является обеспечение возможности его тиражирования и использования во всех вузах.

Ставцев Роман Геннадьевич
Москва, АО «БАЙКАЛ ЭЛЕКТРОНИКС»
<http://baik.al>

Лаборатория электроники Байкал. Программное и аппаратное окружение

Аннотация
Краткий обзор «Лаборатория электроники Байкал», опыт использования в учебном и исследовательском процессе.

Стас Фомин
Москва

OBS — швейцарский нож передачи знаний. Боевые приёмы Open Broadcaster Software.

Аннотация
С одной стороны, сейчас всё отлично для удалённого образования, будь то полностью дистанционные курсы, или просто *flipped classroom*, когда «односторонние» лекции заменила видеозапись или электронные статьи. У всех есть ноутбук, планшет, или мощный смартфон, позволяющий просматривать как тексты, так и видеолекции, и даже мобильные интернет-тарифы не напрягают просмотром образовательного видео.

Но почему-то до сих пор есть мнение, что изготовление добротных видеолекций это какой-то *rocket science*, требующий «студий», дорогого видеоборудования и сложного софта. В вузах заводят «департаменты видеопроизводства», но и статьи на технарских ресурсах внушают, что для достойного качества видеоконтента нужны тонны сложных и дорогих устройств и куча места (см. например, [1, 2]).

В результате, «образовательный видеоконтент», вместо того, чтобы быть самым дешёвым по производству, с трудозатратами примерно равными его длительности, который можно сделать где угодно, превращается в сложное мероприятие с визитом в студию, с привлечением кучи лишних людей, и на выходе опять таки получается что-то невнятное — тяжёлые двухчасовые лекции с запуганным преподавателем, стоящим перед камерами и боящемся пошевелится.

Мы же утверждаем, что добротный информационный видеоконтент, которые будет лучше оффлайн лекций:

- Обычно состоит из комбинации экрана с наложенной «говорящей головой с руками», несущими основную невербалику. При этом легко добавить как настоящие доски с формулами, так и виртуальные. См. [5].
- Производить дёшево — не требует ни больших трудозатрат, ни дорогого оборудования, ни сложного софта.
- Можно делать где угодно, хоть в офисе, хоть дома. Сидя, и не напрягаясь. В комфортном для себя ритме. «Бодрость» же лекции можно придать автоматически.
- Готово к публикации сразу же после записи!
- Отсутствие «профессионалов» видеозаписи — только плюс, почти все навыки профессионалов, которые полезны для сериалов, свадеб и спортивных трансляций — не нужны при изготовлении МООСов.

О том, что для этого нужно, и как это сделать — мы и поговорим в нашем докладе.

Введение

Лет пять-десять назад для всего вышеперечисленного мы предлагали использовать наш фреймворк `SeminarAssembler` [3, 4], но технологии не стоят на месте.

Хотя `SeminarAssembler` остаётся очень эффективным для автоматической записи конференций¹, появился OBS, `Open Broadcaster Software`, ещё проще решающий большую часть «видеоконтентных» задач:

- Надо просто записать скринкаст? — не парьтесь, ставьте, даже такую простую задачу можно и нужно доверить OBS. Нет смысла изучать 100500 других способов скринкастинга, собранных или не собранных в ваш линукс-мак-винду.
- Надо сделать «профессиональный» МООС или клип-обзор, т.е. скринкаст с говорящей головой в углу — да, конечно можно.
- Транслировать просто видео — конечно!
- Транслировать видео с экраном — всегда!

¹Когда есть куча видеисточников разбросанных по большому залу, запись из которых надо мучительно сводить, и особенно, когда есть куча залов, в которых всё должно сниматься автоматически, без постоянного присутствия оператора

- Оперативно переключаться между разными планами «экран», «экран с головой в углу», «голова с экраном в углу», «просто камера», «камера2+камера3», — ... !
- Показать экран другого компьютера или вообще странного медицинского устройства — ... тоже можно.
- И всё это максимально эффективно, с использованием видеокарты для упаковки и наложения оверлеев, чтобы не грузить процессор, с предустановленными параметрами транслирующих сервисов, ...

Более того, именно дешевизна и простота позволит вам делать правильный формат видеозаписи — короткие видеоролики, по 5-15 минут, удобные для фрагментарного просмотра, а не полутора-часовые академические лекции. Т.е. никакого видеомонтажа-склеек-вырезок — только запись, опциональное ускорение и публикация.

И речь идёт не только о лекциях на массовую аудиторию. Даже если аудитория состоит из одного человека, часто удобно сделать серию коротких видеороликов, чтобы не писать длинных писем, не тратить дни на детальную постановку задач (для студентов, для программистов, неважно) или документирование сделанного. Кстати, если совсем никто не будет смотреть — это всё равно гораздо эффективней, чем писать отчёт (по гранту, по работе), который никто не будет читать .

Рассмотрим самые распространённые задачи.

«Личный» видеоклип

Самый быстрый, с точки зрения автора, способ зафиксировать свои знания, хотя часто бывает также эффективно и для получателя.

Некоторые утверждают, что читать в два раза быстрее, чем смотреть видео, и местами, когда гуглишь какие-то очевидные ответы, а гугл тебе подсовывает вместо «стековерфлоу с одной строчкой», длинные видео каких-то косноязычных индусов, которые полчаса разливают воду, путаясь открывают нужную софтину... начинаешь проклипать не только их, но и изобретателей ютуба.

Но. Если эти знания не настолько тривиальны, если нужно зафиксировать их здесь и сейчас, чтобы немедленно передать знания другим людям, или наоборот, застраховаться от внезапного ухода автора из компании (о чём я давно говорил в куче докладов, например в [8],

то нет ничего эффективней, чем быстро набросать маймап или слайды с краткими тезисами, открыть код и нужные софтины, а дальше, в режиме «говорящей головы на фоне слайдов-кода-софтин» быстро озвучить тезисы, показать, «как оно работает» с живыми комментариями. Плотность информации — 60 тысяч знаков за полчаса минимум, поверьте тому, кто неоднократно писал стенограммы. Причём ключевое — будет на слайдах-тезисах-майндмапах-лайвдемо, и отпечатается на зрительных нейронах, а остальное, возможно непечатным русским слогом, пойдёт аудиоканалом.

- Нужно всего лишь — <https://obsproject.com/> OBS (Studio или Classic, обычно неважно, но начните с Studio).
- USB-вебкамера — мимика, жесты, демо с предметами тоже может пригодиться. Совсем экономить на вебкамерах не стоит, но вебкамы от Microsoft и Logitech — обычно нестыдно.

В идеале, при записи себя, желательно:

- Два монитора — на одном будет интерфейс OBS, чтобы переключать сцены-планы монтажа, включать-останавливать запись.
- Раскладной зелёный экран сзади, покупается на алиэкспрессе за копейки, можно пристроить в любом офисе, кроме самых убогих опенспейсов.
- Хотя даже в этом убогом опенспейсе, или дома, в общем, в ситуации когда сзади нет стены, существуют раскладные зелёные экраны 1.5×2м, которых можно прикрепить к креслу, убив тем самым двух зайцев — защитить свой экран от любопытных взглядов сзади, и получить хромакей вокруг тела и головы.
- В OBS настраиваете несколько стандартных планов — «чисто экран», «камера на фоне-чего-нибудь», «экран+камера в углу», для камеры сделать фильтр «хромакей» (один раз подобрать для офисного освещения, обычно сойдёт навсегда, до переезда).
- Как настраивать «планы» в OBS — интуитивно понятно, двигайте эти прямоугольники мышью, параллельно попробуйте зажимать SHIFT, CONTROL... и откроете для себя кроппинг или ресайз без нарушения пропорций.

Видеомонтаж — не нужен. Лучше записывать небольшими блоками 2-15 минут, отвечающими на конкретный вопрос — «Введение в

XXX», «Как на XXX сделать YYY» и т.п. Подготовили материалы, запустили запись, рассказали... остановили, отдыхаете. Последовательность просмотра можно потом выстроить в вашей LMS², CMS, или что у вас там вместо этого.

А в качестве вишенки на торте — можно элементарно улучшить вашу запись, просто ускорив, процентов на 20%-70% (кто знает, насколько вы небыстры).

```
ffmpeg -y -i "$1" -filter_complex "setpts=PTS/1.5;atempo=1.5" \\  
-vcodec libvpx-vp9 -tile-columns 6 -frame-parallel 1 \\  
-auto-alt-ref 1 -lag-in-frames 25 \\  
-acodec libvorbis -ab 64K -vb 1M -f webm "$1.webm"
```

Так можно сделать из заикающегося тугодума популярного блогера ютуба, и заодно получить нативно воспроизводимое в любом браузере видео в WebM-формате.

При съёмке личного клипа есть конечно много тонкостей и гитик:

- Надо приучить себя смотреть в незаметную вебкамеру, а не на монитор. Если пишете на ноутбуке с отдельной камерой — лучше наклонить экран, и смотреть в камеру напрямую, иногда подглядывая.
- Удобно использовать ноутбук с пером, тачскрин-монитор или моноблок. См. [5]
- Если нет — сделайте хотя бы максимально крупный курсор и пользуйтесь им.
- Поставьте либо хороший USB-микрофон, либо берите звук с гарнитуры или прищепки. Постарайтесь, чтобы в комнате параллельно не было шумной ругани и рядом не печатал 3D-принтер. См. [9].
- Скорее всего, надо будет подогнать на $\pm 200\text{ms}$ аудио, чтобы попасть «в губы». Тут надо итерационно записывать себя на камеру, например, считать до 10, так, чтобы губы были крупно видны, потом просматривать всё это в плеере VLC, используя клавиши «j» и «q» (сдвиг звука на $\pm 50\text{ms}$), пока звук не попадёт «в губы». Потом полученную задержку внести в настройки аудиомиксера.

²Learning Management System

Если вы снимаете что происходит в другом устройстве — например, вы хотите показать экран при работе с BIOSом ноутбука, . . . или ещё какого-то устройства, то вам будут нужны фреймбgrabберы для VGA, DVI или HDMI. См. [6].

Лекции и семинары

Перейдём к следующему уровню — съёмка семинаров-лекций, или внутрикомпанийских сборищ и прочих *таунхаллов*.

Рассмотрим именно простой³ случай, когда у нас одна камера, экран и звук с микрофона или воздуха. Т.е. технически, от предыдущего случая отличие в том, что лектор не сидит, а движется, и его сложно снять вебкамерой с без оптического зума⁴, возможно используется проектор для слайдов и демо, возможно используется флипчарт, маркерная или грифельная доска. Скорее всего используется какое-то звукоусиление, и звук надо брать не с воздуха, где он соберёт всё эхо большого помещения, шум кондиционеров и аудитории, а с микрофона. Тут уже нужен будет оператор записи, хотя ничего хитрого делать не надо — после настройки оборудования вполне справится студент.

Тут у вас стоит задача завести в одно место видео с камеры, экран и звук. Надеюсь, вам не надо объяснять, что снимать одной камерой человека и экран проектора — в не очень сильно подготовленном помещении⁵ обычно глупо — вы получите белый, пересвеченный экран, где ничего не разобрать, и чёрного человека на этом фоне. Поэтому камерой надо снимать докладчика так, чтобы в камеру не попадал экран (а если попадал, то только небольшая часть), а экран брать по настоящему, цифровым, чтобы не потерять ни пикселя.

Итак, завести звук относительно просто, кабель для звука может быть сколь угодно длины, коннектором 6.3мм, колокольчиками или XLRом он войдёт в микшерный пульт, другим концом он пойдёт либо напрямую в LineIn звуковой платы на материнке компа или ноута —

³Случай с огромными залами конференций, кучей камер, дублированием записи экранов и звука. . . можно наверно пытаться сделать также, но это уже падает либо в необходимость профессиональных видеокоманд, либо в использовании моего SeminarAssembler, как делаю я при съёмке всех этих конференций, но это вам будет сложно. Тут мы начнём с основ.

⁴А они почти все такие

⁵Затемнение перед экраном проектора и в зале, точечная подсветка докладчика проекторами сверху

тут надо потестировать, настроить уровни, проверить, что внезапные шумы от чего-то⁶, всё не сломают, ну и замерить расхождение между звуком и видео, чтобы указать коррекцию в OBS. Следующий уровень крутизны — внешняя USB звуковая карта, так будет качественней и надёжней. Ну и рекомендую параллельно писать звук с микшера на диктофоны, см. [9], если в случае личного видео — «запорол, выругался, перезаписал», то с мероприятием, где ведущий лектор — не вы, это уже не пройдёт.

Теперь самое сложное. Экран и камера.

- Камера должна быть с зумом, должна быть возможность её перенацеливать на зал, маркерные доски, менять масштаб.
- Камера должна быть удаляема от компа с OBS хотя бы на 3-5 метров — иначе не снять стоящего-перемещающегося лектора.

В целом, стоит задача довести до единого компа источник экрана и видео, причём так, чтобы камера снимала спикера. Обычно, ноутбук докладчика находится рядом с ним (на кафедре, куда он может подглядывать, чтобы не смотреть на экран, поворачиваясь задом к зрителям). Соответственно, тут возможны разные варианты:

- Комп с презентацией-лайв-демо, примерно там, где докладчик. OBS запускать там же, экран берётся прямо с компа, но тогда надо выносить камеру, хотя бы на метров 5.
- Комп с OBS отдельно от компа докладчика, в принципе, можно снимать на USB-камеру (хотя и неудобно), но тогда надо решать вопрос с захватом экрана. Тащите фреймграбберы для VGA, DVI или HDMI (См. [6]) заводите в них видеосигнал, либо HDMI насквозь, как с Elgato, либо мучайтесь с сплиттингом DVI или VGA видеосигнала, и хватайте в Eriphian-фреймграбберы.

Какую же камеру выбрать?

Из вебкамер с настоящим зумом и как-то управляемых, я помню только Logitech Orbit, а все остальные «настоящие» камеры, с нормальной оптикой и оптическим зумом... обычно как USB не работают (т.е. подключение по USB и при этом представление вебкамерой, а не, например, стораджем снятого — таких не очень знаю). Т.е. я знаю

⁶Например, радиомодуль — у меня был один факап из-за USB WiFi, воткнутого недалеко от входа в звуковуху

например, как можно использовать DV-over-USB как для захвата через USB с помощью dbgrab DV-потока, но в OBS мне его не удалось завести. Т.е. USB-вебкамера только в варианте, когда вот докладчик совсем не бегаёт, недалеко от компа-ноута с OBS. Редко, но такое бывает. Типа поставить стол с ноутбуком, а ещё лучше — кафедру, запретить выходить из-за этого стола и ходить, оглядываться назад, и снимать снизу вверх (не очень хороший ракурс, совсем не секси), с кафедры уже приличней.

Моя рекомендация — самый простой способ получить цифровое видео от настоящей камеры с зумом, которую можно поставить на нормальный штатив, и не покупая 100500 странных и дорогих оцифровщиико-конвертеров-фреймграбберов (HDMI2USB, SDI2USB)... это Firewire, он же IEEE1394.

Сейчас несколько задвигаемый стандарт... но он до сих пор:

- Есть в куче ноутбуков, не только эппловых — HPшные, Thinkpad-ы, Dell-ы, до 2014 года выпуска — почти все, да и сейчас полно.
- Новый ноут не обязателен, для записи 1280x720 и оверлей-сборки в OBS хватит и мобильного Core i5, даже без поддержки видеокарты, т.е. вполне можно выделить под это старый ноут, их всегда можно найти или купить за копейки.
- PCI-карта для десктопа, если системная плата не поддерживает, стоит считанные копейки.

Теперь о камерах с поддержкой Firewire — они двух типов, DV и HDV, вроде уже не производятся, но стоят сущие копейки на вторичном рынке. При том, что у них отличная оптика, может быть большая слабошумная матрица, неплохой звук, всякие удобства управления.

- *DV-камеры* — разрешение всего 720x576⁷. Это минус, наверно некруто, если снимать современную свадьбу, но если монтировать с экраном на канвас 1280x720, когда спикер пойдёт либо урезанным квадратиком в левый нижний угол, либо полосой слева или справа от экрана → это выше крыши. Также это достаточно, если вдруг снимать маркерные доски или флипчарты. Я лично любитель MiniDV-камер от Panasonic (см. [10]), у меня их были десятки, и хотя я для съемки докладов для 0x1.tv перешёл на HDV-камеры 1080i⁸, я их часто использую как стра-

⁷720x576 в случае PAL и 720x480 в случае американок, т.е. NTSC

⁸Canon HV20-30-40, фотоаппараты Panasonic Lumix

ховочные (снимать другой угол зала, куда внезапно может уйти докладчик, или флишчарт крупным планом и т.п.). Заодно, запись можно страховать, записывая на двухчасовую⁹ miniDV-кассету.

- *HDV-камеры*: Все те же плюсы, что и у DV-камер, разрешение 1080p, единственное — кассеты нужны получше и они только часовые.

Рассмотрим, для простоты, проблему захвата в OBS DV и HDV-камер под виндой. Тут не совсем всё очевидно.

- DV-камеры: старый добрый OBS-classic захватывает их без проблем стандартно, как «Video Capture Device», но при переходе к OBS Studio ребята что-то сломали¹⁰ и пока ещё не могут починить. Что обидно, что не работает и обходной путь — <https://obsproject.com/mantis/view.php?id=1089#c2864>, захват через VLC-плагин.
- HDV-камеры: OBS-classic их не видит. OBS studio через «Video Capture Device» их тоже не видит, но есть обходной путь.
 - сначала надо поставить и запустить VLC, сделать в нём Capture, добиться, чтобы он захватывал с HDV-камеры. Запустите VLC, CTRL-C, выберите нужную камеру. . .
 - Ну, или чтобы не погружаться в тонкости настройки... запретите остальные вебкамеры, тогда `vlc dshow://`
 - . . . будет вынужден захватывать HDV-камеру, без вариантов.
 - потом в OBS studio выбрать источник типа VLC Source,
 - . . . и вуаля — там будет видео с камеры.

В настройках трансляции и записи ничего сложного нет, хотя да... пишете видео только в стриминг-форматы, устойчивые к ситуации «всё внезапно сдохло». С AVI, или ещё хуже mp4, вы можете внезапно попасть. Пишите всё в FLV, с битрейтом побольше (хоть 10Mb x264 кодека), переходировать для хранения или заливки куда-нибудь лучше потом.

⁹80-минутную в LP-режиме

¹⁰<https://obsproject.com/mantis/view.php?id=709>

Кейс с таунхоллом

Для простоты, ещё раз рассмотрим конкретный кейс. Про вот это всё меня спрашивали ребята из компании В.

- есть десктоп, используемый для показа слайдов и лайвдемо, куда я, кстати, вставил Firewire карту, и она ещё там.
- тот десктоп с виндой — поставьте туда виндовый OBS classic, ну и вообще, он должен там быть.
- OBS будет захватывать экран прямо с компьютера (там, где презентация и лайвдемо докладчика).
- Звук тоже с микшерного пульта, недалеко, 6.3mm → 3.5mm.

Осталось добавить камеру. DV-камеру типа GS-80 можно купить на avito рублей за 700 (сам продаю), благородную GS-400 — за пару тыщ. Добавим 5-метровый Firewire шнур — на ебей или aliexpressе всегда можно купить и 10 метровый, редко, но попадаются и 25 метровые¹¹ Тогда, как я писал выше, надо использовать OBS classic, он вполне подойдёт для этой задачи.

Дальше как обычно, делайте несколько типов заготовленных сцен «чисто камера, вокруг корпоративные дизайнерские рамки», «чисто экран», «экран + кропшированная камера в левом нижнем углу», «экран и кроп-камеры в левом верхнем», «вертикальнополосый-кроп-камеры + обрезанный под слайды экран»... и поехали.

Вполне один и тот же человек, сможет, стоя за камерой, посматривать вперёд, на монитор с OBS, переключая сцены с помощью беспроводной мыши или чего-нибудь типа этого.

Если камера ОК, но нельзя задействовать один комп для докладчика и запись, — вперёд за фреймграбберами (см. [6])

Если же хотите помучаться с другими камерами... тут смотрите, что они умеют — большая часть консьюмерских камер умеют отдавать HDMI — его можно попытаться захватывать через Elgato. Увы, получается не всегда, если не использовать какие-нибудь конвертеры и скейлеры (типа продукции Kramer).

Более профессиональные камеры обязательно имеют SDI выход — ну и нужен будет профессиональный фреймграббер с SDI, типа продукции BlackMagic.

¹¹У меня лично есть даже 25 метровый шнур для таких трансляций, и его не просто было достать...

Литература

- [1] «Как писать подкасты, что за железо нужно? Автор статей, текстов и постов сейчас вам всё расскажет»
<https://habr.com/ru/company/jugru/blog/434298/>
- [2] «Как мы делали студию для записи онлайн-курсов»
<https://habr.com/ru/company/jugru/blog/434298/>
- [3] SeminarAssembler — эффективная съёмка, монтаж и публикация лекций и конференций
<http://0x1.tv/Seminar-assembler-oseduconf>
- [4] «Корпоративный Болливуд», или видеопоток знаний на конвейере
<http://wiki.4intra.net/Swp12-stas-fomin-bolliwood>
- [5] Магия пера или эффективная свобода преподавания со стилусом
<http://0x1.tv/20140126-4>
- [6] «Захват экрана — за ценой не постоим»
<http://0x1.tv/Screenrecording>
- [7] «Конференции-трансляции-таунхоллы-семинары-видеоблоги-скринкасты — ответ один. Швейцарский нож OBS, Open Broadcaster Software»
<http://0x1.tv/Obs-as-swiss-army-knife>
- [8] «Страховка от автобуса» или пересадка сознания при групповой нейрохирургии
<https://whalerider.ru/2012/abstracts/381.html>
- [9] Главное в видео доклада — это ... вовсе не видео
<http://0x1.tv/Audio-important-video>
- [10] MiniDV камеры от Panasonic
<http://wiki.4intra.net/Panasonic-minidv>

Пустыгин А.Н., Кузьминых К.М., Ковалевский А.А.,
Пустыгина Е.А., Егоров Д.Ю.

Челябинск, Челябинский государственный университет

Применение программных инструментов с открытым исходным кодом для анализа исходных текстов программ

Аннотация

Использование внутреннего представления исходного текста открывает путь для систематического изучения и преобразования кода программ с целью его изучения и проверки. Рассмотрены утилиты с открытым кодом, использующие внутреннее представление исходного кода.

С целью актуализации выполняемых разработок были предприняты усилия для обзора открытых инструментов, пригодных в качестве базы сравнения с разрабатываемыми прототипами. Были рассмотрены существующие открытые инструменты для анализа, использующие построение синтаксического дерева разбора (AST):

1. PyCharm — IDE с открытой версией (подписная free licence) [1].
2. Vulture — анализатор кода в программах Python [2], использует модуль ast стандартной библиотеки и создаёт абстрактные синтаксические деревья для всех файлов исходного кода в проекте.
3. ObjectWeb ASM (частью входит в Java Development Kit 6) [3] — API для представления кода JAVA в виде дерева.
4. Joern — анализатор исходного кода [4], способный построить: абстрактное синтаксическое дерево, граф управления, граф вызовов, граф структуры каталога исходного кода, граф программных зависимостей для исходного текста на языке C++. Инструмент работает в связке с системой управления графовой NoSQL базой данных Neo4j [5] — графовая система управления базами данных с открытым исходным кодом, реализованная на Java. Считается самой распространённой графовой СУБД, формат данных собственный, специализированный. собственный язык запросов Cypher, но запросы можно делать и другими способами, например, напрямую через Java API и на языке Gremlin, созданном в проекте с открытым исходным кодом TinkerPop.

С целью ознакомления с возможностями базовых инструментов был выполнен проект по применению Joerg, как наиболее функционального, для решения типовых анализа потока управления.

Для обработки информации, полученной с помощью инструмента Joerg, применялась написанная программа пост-обработки на Python. В функционал пост-обработки входят:

- построение трассы исполнения программы по исходному тексту;
- получение списка всех информационных объектов программы по исходному тексту;
- проверка наличия заданных конструкций в исходном тексте программы;
- формирование перечня маршрутов исполнения функциональных объектов по исходному тексту программы;
- контроль связей функциональных объектов в исходном тексте программы по информации;
- контроль связей функциональных объектов в исходном тексте программы по управлению.

Параметры запуска и результаты тестирования программы пост-обработки доступны в открытом источнике [6].

Под функциональными объектами понимаются объекты классов, локальные и глобальные переменные проекта, непосредственно участвующие в алгоритме. Под маршрутом исполнения функционального объекта понимается его «линия жизни» от момента создания до указанного места в исходном коде. Результатом исполнения скрипта пост-обработки будет перечень всех маршрутов исполнения, в которых, так или иначе встречается использование указанного функционального объекта, либо сообщение о том, что такие маршруты отсутствуют.

Программа пост-обработки может контролировать связи функциональных объектов проекта по управлению и по информации. Для этих задач необходимо выбрать два функциональных объекта: исследуемый и второстепенный.

Под связью функциональных объектов по управлению понимается любое воздействие исследуемого объекта на второстепенный, в том числе: модификация (чтение, запись) данных второстепенного объекта, вызов методов второстепенного объекта и любое другое воздействие на данные второстепенного объекта.

Под связью функциональных объектов по информации подразумевается любое взаимодействие второстепенного объекта на данные исследуемого объекта.

Поиск и анализ такого рода связей между объектами является неотъемлемой частью анализа текста любого программного продукта. Он даёт возможность убедиться в отсутствии недокументированных возможностей программы, отсутствии избыточности исходного кода.

Пример использования пакета из Joern и инструмента пост-обработки

Анализируемый текст (Python):

```
ifstream file("graphFile");

unsigned int n = 0;
unsigned int oriented;
file >> n >> oriented;

G.resize(n);

while (!file.eof()) {
    int a, b;
    file >> a;
    file >> b;

    G[a].push_back(b);
    if(oriented == 0) {
        G[b].push_back(a);
    }
}

file.close();
```

Исследование прототипа сравнения показало ограничение его функциональности по глубине построения синтаксического дерева разбора, а также недостаточный уровень документирования инструмента Joern.

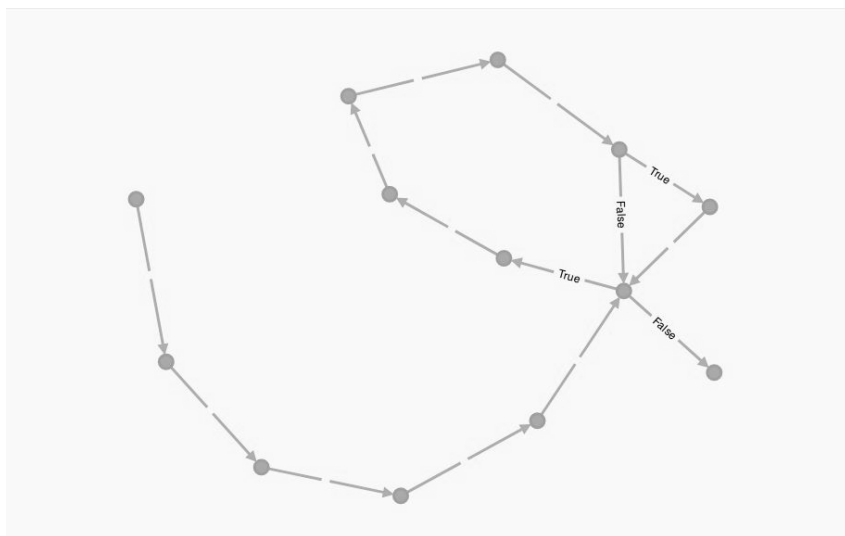


Рис. 1: Граф трасс исполнения анализируемого текста

Литература

- [1] The Python IDE for Professional Developers <http://www.jetbrains.com/pycharm/index.html> (просмотрено 09.01.2019)
- [2] Vulture — pythondead-code-removal <http://github.com/jendrikseipp/vulture> (просмотрено 09.01.2019)
- [3] ASM — bytecode manipulation and analysis framework <http://asm.ow2.org/> (просмотрено 09.01.2019)
- [4] A robust parser for C/C++ storing abstract syntax trees <http://github.com/octopus-platform/joern> (просмотрено 09.01.2019)
- [5] Graph Databases for Connected Data <http://neo4j.com> (просмотрено 09.01.2019)
- [6] <https://github.com/Oskal174/staticProgramAnalysis/>

Лукин В.Н., Чернышов Л.Н.

Москва, Московский авиационный институт, Финансовый университет при правительстве РФ

<https://github.com/LevChern/eduprocess>

Дистанционное обучение программированию на Google-платформе

Низкий уровень квалификации современных бакалавров и магистров, которые формально считаются специалистами в области информационных технологий, ни у кого не вызывает сомнений. Фундаментальных причин тому несколько, отметим некоторые: слабость выпускников школ, неадекватный подход к оценке качества образования, невысокое качество учебных программ, недостаточный уровень студенческих занятий. Смириться с существующим положением невозможно: они приводят к незапланированной перегрузке преподавателя. В первом случае время тратится на начальную подготовку, во втором оно убивается на бессмысленную «отчётность», в третьем приходится расходовать его на не слишком актуальные темы. А четвёртая причина в значительной степени следствие предыдущих: у преподавателя нет времени на нормальные занятия. Возможности избавиться от первых трёх причин нет. Посмотрим, как можно выкроить время для занятий, дающих качественные знания.

Для контрольных мероприятий на каждую тему необходим пакет однотипных задач. При современных технологиях списывания задачи для каждого экзамена должны быть уникальными. Таким образом, объём работы становится чрезмерным, а перегрузка преподавателя на подготовку курса, осложнённая необходимостью выполнять бюрократические работы, приводит к выпуску некачественных программистов. Возникает естественная идея автоматизировать как процесс генерации заданий, так и процесс проверки работ.

Проверка ответов на задания по дисциплинам программирования имеет свои особенности: разнообразие видов занятий, большое количество упражнений. Для лабораторных работ особой разницы между очной и заочной формой выполнения нет: в обоих случаях студенты выполняют работы дома, время аудиторных занятий идёт на приём работ. Понятно, что для контроля необходим набор типовых задач. Их уникальность обеспечивается генерацией вариантов на некотором

наборе «базовых» задач, которые накапливаются от семестра к семестру.

Рассмотрим вариант процесса подготовки и проведения контрольных мероприятий в автоматизированной системе.

При большой нагрузке преподавателя естественно использование облачных платформ. Но предлагаемые средства поддержки учебного процесса на порталах вузов неудовлетворительны. Существующие системы дистанционного образования могли бы использоваться, но им не хватает гибкости. Платформа Google имеет модуль для ведения учебного процесса Classroom, но она также не решает всех задач.

Более эффективно реализовать функции поддержки учебного процесса на языке Google Apps Script, предназначенном для автоматизации работы с online-приложениями. Имея подходящий набор функций, преподаватель может приспособить их под свои потребности. Рассмотрим вариант подобного набора.

Создание первоначальной структуры

Исходные данные:

- список групп;
- список студентов;
- задания
- указания к выполнению
- шаблон отчёта по ЛР.

Структура папки «Дисциплина-2018»:

Таблица ГРУППА-1-работы (без доступа студентам)

ГРУППА-1 (доступ студентам на чтение)

ГРУППА-1-посещение

ГРУППА-1-оценки

ГРУППА-1-вариант

Вопросы и ответы (доступ студентам на комментирование)

Работы студентов

 Фамилия 1 (доступ студенту на запись)

 Фамилия 2 . . .

 . . .

Скрипт, прикрепленный к списку групп, генерирует такие структуры для каждой группы. Для группы указывают дату первого занятия, число занятий и периодичность, в таблицах посещений создаются

колонки по дням занятий. В оценочной таблице создаются колонки с названиями заданий. Предусматриваются оценки за работы и отчётность по ним.

В таблице со списком группы для каждого студента создаётся ссылка на папку с его работами и дата последнего изменения файлов. В папке «Дисциплины» размещаются документы для выполнения работ. Ссылки на них размещаются в таблице «Общие документы». В указаниях к выполнению описываются правила выбора вариантов работ и критерии оценки.

Подготовка и генерация заданий

Составляется список заданий, доступный для студентов, преподаватель назначает индивидуальное задание каждому из них. В таблице размещаются ссылки на описание вариантов заданий. В папку студента помещается описание задания.

Генерация вариантов заданий аналогична способу, описанному в [3]. Задание составляется на основе нескольких типовых, в каждом из которых предусматриваются альтернативные значения или фразы. Оно формулируется так, чтобы части текста могли заменяться шаблонами.

Например:

«Напишите код HTML-страницы с функциями на JavaScript. На экранной форме располагаются %k% элемента выбора (%pos%), поле ввода и поле-флажок (%pos%) и кнопка (%pos%). В первом элементе выбирается %typ1%, во втором — %typ2%. . . .»

Для каждого шаблона задаются варианты его значений, которые при генерации случайным образом его заменяют. Результат при необходимости может корректироваться.

Выполнение работ и оценка

Результаты работы студент выкладывает в свою папку, а в журнале отмечает, что сделано. Преподаватель просматривает журнал, скачивает папку с программами и проверяет их. Если студент указал ссылку на программу, она может запускаться непосредственно. Для просмотра хода работ используется таблица, в которой указаны даты изменения.

Оценка работы состоит из нескольких отметок, например, отдельно оценивается программа и отчёт по ней. Общепринятая практика — снижение отметки за нарушение сроков по определённым правилам. Простейший способ вычисления итоговой отметки — нормированная

сумма всех отметок по заданиям. Но преподаватель может задать и свой способ.

Интеграция с другими приложениями

Помимо приложений на Google-платформе, можно использовать и другие внешние приложения, например, системы тестирования. При этом возникает необходимость интеграции с ними. Обычное web-приложение, серверная компонента которого написано на PHP, может взаимодействовать с Google-приложением с помощью пакета `google-api-php-client`. Так тестовая система может автоматически размещать результаты тестирования в Google-таблицы с успеваемостью.

Практика применения

Описанный инструментарий используется авторами не первый год и показал свою полезность. Сама Google-платформа даёт много возможностей и удобна для многих целей учебного процесса (см., например, [5]). В нашем случае применение скриптов даёт ещё большую гибкость.

По Закону РФ о персональных данных информация о студентах может храниться только на территории РФ. Это можно обойти простым обезличиванием — вместо фамилий студентов хранить их номера по списку группы. Или взять с них письменное согласие на размещение определённых персональных данных (фамилия, успеваемость и т.п.). Кроме того, есть информация, что сервера Google будут размещаться и в России¹.

Литература

- [1] Лукин В. Н., Чернышов Л. Н. Проблемы подготовки студентов в области информационных технологий: контроль качества. Издательский дом «Народное образование», ж-л «Образовательные технологии» — № 3, 2018.
- [2] Чернышов Л.Н. Облачные технологии Google в учебном процессе. Материалы XI-ой международной конференции по неравновесным процессам в соплах и струях (NPNJ'2016). 25–31 мая 2015 г. — Алушта, Крым. М.: Из-во МАИ, 2016, с.578–580.
- [3] В. В. Лаптев, В. В. Толасова. Генерация вариантов заданий для лабораторных работ по программированию. Вестник АГТУ. Сер. Управление, вычислительная техника и информатика. — № 1, 2010.

¹<https://xakep.ru/2015/04/10/google-servers/>

- [4] Ананченко И. В. Облачные технологии в высшем образовании // Современные наукоёмкие технологии. — №5, 2015, — с. 48-52 URL: <http://www.top-technologies.ru/ru/article/view?id=35037> (дата обращения: 05.01.2018).
- [5] Карпова Н. А. Использование Google на уроке. URL: <https://docplayer.ru/45369243-Ispolzovanie-google-na-uroke.html> (дата обращения: 05.01.2018)

Максим Каторгин

Москва, Федеральное государственное бюджетное образовательное учреждение высшего образования Российский государственный социальный университет (РГСУ)

Использование свободного программного обеспечения в студенческом проекте метеостанции на базе платы Arduino MEGA

Аннотация

Разработано программно-аппаратное средство для осуществления экспресс-контроля основных параметров рабочего места в помещении типа читальный зал: освещённости, температуры и влажности. Основа разработки — свободное ПО, плата Arduino с сенсорами и LED-индикацией. В ходе проектирования проходят обучение студенты направлений «Информационные системы и технологии», «Информатика и вычислительная техника» в области программирования, электроники и основ робототехники

Современные техника и технологии позволяют автоматизировать многие функции помещений — создаются так называемые «умные дома». Однако возрастающая сложность таких систем, их естественный износ или случайные поломки приводят к сбоям и частичному выходу систем из строя, в результате часть физических параметров среды обитания может выйти из допустимых пределов.

В соответствии с СанПиН [1] должны проверяться основные параметры помещений, предназначенных для работы людей, и для этого разработаны многочисленные приборы, которые, однако, или дороги, или сложны в использовании. Поэтому актуальность разработки состоит в том, чтобы разработать недорогое эффективное устройство

экспресс-оценки основных параметров помещения (например, температуры, влажности, освещённости) с желательной яркой и доступной индикацией измеренных параметров. Выбран тип помещения — читальный зал (что, однако не исключает применения разработки и для других типов помещения (по категории работ и периода года) при соответствующей перестройке параметров контрольно-измерительного комплекса.

Разработка выполнена на базе платформы Arduino и представляет собой контрольно-измерительный комплекс с функциями измерения уровней основных параметров рабочего места в помещении: освещённости, температуры и влажности; сравнения результатов измерений с допустимыми значениями; вывода результатов (например, в виде цветовой LED-индикации).

Индикация вхождения в какой-либо из диапазонов допуска осуществляется цветовой кодировкой: «ниже нормы» — синий цвет; «норма» — зелёный цвет; «выше нормы» — красный цвет.

Калибровка осуществлялась в соответствии с данными разработчика аппаратуры — фирмы «Амперка» [2]. Датчик температуры и влажности выполнен на основе микросхемы SHT31 от компании Sensirion по запатентованной технологии CMOSens®, откалиброван и отличается высокой точностью измерений. Модуль измерителя освещённости основан на фоторезисторе GL5528 [2, 3].

Разработанное устройство выполнено в переносном варианте.

Разработка является студенческим проектом, выполненным автором на втором курсе, и предполагается дальнейшее развитие проекта в направлении как расширения количества измеряемых параметров (уровней загазованности, ультрафиолетового излучения, скорости обмена воздушного потока и ряда других), так и фиксации измерений в виде записи в текстовые файлы, передачи по сети; индикация фактически измеренных уровней в виде цифровой индикации и т.д. [4, 5]. Также предполагается учёт особенностей условий помещения для лиц с ОВЗ.

Литература

- [1] СанПиН 2.2.1.1278-03 «Гигиенические требования к естественному, искусственному и совмещённому освещению жилых и общественных зданий». URL: <http://docs.cntd.ru/document/901859404> (дата обращения: 10.01.2019).

- [2] Сайт фирмы «Амперка» URL: <http://amperka.ru/> (дата обращения: 10.01.2019).
- [3] Торо Карвинен, Киммо Карвинен, Вилле Валтокари. Делаем сенсоры. Проекты сенсорных устройств на базе Arduino и Raspberry Pi / М.: ООО «И.Д. Вильямс», 2016. — 448 с.
- [4] Аметова М. М., Каторгин М. К., Симонов В. Л., Учакин Д. Д., Хмыров Н. А., Шмелёв А. Н., Шукин Ф. О. Обучение программированию студентов технических специальностей в области основ электроники, автоматики и робототехники / Материалы XXIX Международной конференции «Современные информационные технологии в образовании», 26 июня 2018 г. / Департамент образования города Москвы, Фонд новых технологий в образовании «БАЙТИК» / Изд. полиграфического центра Московского издательско-полиграфического колледжа им. И. Фёдоров. — гг. Москва — Троицк. — 555 с. — с. 150–152.
- [5] Миронов П. Н., Герус М. И., Симонов В. Л., Аметова М. М., Шукин Ф. О., Хмыров Н. А., Юров Н. Н., Виноградов Д. А., Кошеварова Н. А. Практические аспекты организации занятий по основам программирования в области электроники, автоматики и робототехники при подготовке бакалавров и специалистов инженерных специальностей / Материалы IV международной научно-практической конференции «Информатизация инженерного образования «Инфорино-2018»», 23–26 октября 2018 г., Москва. — М.: Издательство МЭИ, 2018. — 556 с.: ил. С. 155–159.

Сергей Мартишин, Владимир Симонов, Марина Храпченко
Москва, Федеральное государственное бюджетное учреждение науки
Институт системного программирования им. В.П. Иванникова Российской
академии наук (ИСП РАН), Федеральное государственное бюджетное
образовательное учреждение высшего образования Российский
государственный социальный университет (РГСУ)

**Использование свободного программного
обеспечения HTML5 Boilerplate, Bootstrap, Initializr
для обучения студентов технологиям создания
клиентской части web-приложений**

Представлены возможности программных продуктов СПО: HTML5 Boilerplate, Bootstrap, Initializr для разработки клиентского приложения в клиент-серверных облачных вычислениях. Программные продукты используются при обучении студентов направлений «Информатика и вычислительная техника», «Информационные системы и технологии» и других для создания обучающимися полноценных современных распределённых web-приложений, в особенности их фронтэнд части

В настоящее время объёмы генерируемой человечеством информации настолько велики, что для её хранения и обработки необходимо использовать современные технологии, среди которых наиболее широкое распространение получили облачные вычисления. Последние требуют значительных вычислительных ресурсов, поскольку основной объём вычислений производится на сервере, а данные выдаются клиентам по запросу в режиме реального времени. Это приводит к появлению новых технологий, позволяющих разработчикам сделать процесс создания программного обеспечения более универсальным.

Студентам направлений подготовки «Информатика и вычислительная техника», «Информационные системы и технологии» и ряда других необходимо знать вышеуказанные современные технологии, поскольку соответствующие ФГОС ВО содержат ряд компетенций, таких как «способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования» (ПК-2). Соответствующие разделы входят в состав таких дисциплин, как «базы данных», «распределённые и удалённые базы данных», «разработка интернет-приложений», «проектирование и администрирование информационных систем», «программирование мобильных устройств».

При обучении студентов основное внимание уделяется серверной части как основе приложения, поэтому в своих предыдущих работах авторы уделили много внимания именно вопросам разработки серверной части [1, 2]. В то же время для каждого пользователя в значительной мере именно клиентская часть определяет эффективность использования данного приложения, и от оценки пользователя зависит популярность и востребованность продукта, что в свою очередь требует от производителей уделить особое внимание интерфейсу и простоте использования клиентской части.

Доступ к информации может осуществляться с различных типов устройств: стационарных компьютеров, планшетов, мобильных теле-

фонов и пр. Задача разработчиков состоит в том, чтобы пользователь смог на любом клиентском устройстве получить доступ к ресурсам в привычном для себя виде.

Для разработки клиентского приложения наиболее простыми и удобными являются следующие программные продукты СПО: HTML5 Boilerplate, Bootstrap, Initializr. Данные продукты позволяют разработчикам решить большую часть задач при реализации клиентской части.

HTML5 Boilerplate [3, 6] — это фронтэнд шаблон для разработчиков. Он состоит из набора файлов, содержащих готовые решения, созданные специалистами или взятыми из практики. Файлы обеспечивают основу для любого создаваемого веб-сайта и могут быть скачаны бесплатно. Также шаблон предлагает готовую структуру файлов и папок для будущего проекта и основные конфигурации для работы с сервером. Использование шаблона значительно ускоряет разработку и улучшает её качество.

Bootstrap (Twitter Bootstrap) [4] — свободное программное обеспечение (free and open-source software, FOSS) для создания фронтэнд (front-end) частей сайтов и веб-приложений. Он включает в себя различные шаблоны для создания форм, кнопок, меню и иных компонентов web-интерфейса.

Следует помнить, что Bootstrap ориентирован на достаточно новые версии CSS и HTML, поэтому необходимо быть внимательными при использовании устаревшими браузерами, требующими дополнительной поддержки. Для установки необходимо зайти на официальный сайт [4] и загрузить Bootstrap. После перенаправления на другую страницу, содержащую следующие различные варианты установки, необходимо выбрать нужную. Для стандартной загрузки выбирают Bootstrap — релиз с уже скомпилированными файлами. Остальные возможности предусмотрены для разработчиков более высокой квалификации, знающих языки стилей LESS и Sass.

Большим удобством при использовании фреймворков является возможность создавать кроссбраузерные приложения, включая приложения для экранов с различным разрешением (от самых маленьких экранов смартфонов до полноразмерных экранов), наличие понятного алгоритма создания наиболее часто встречающихся компонентов: меню, таблиц, текстовых полей, кнопок и многого другого. В то же время код генерируемой фреймворком страницы является избыточ-

ным, поскольку содержит даже неиспользуемые стили и возможности.

Заметим, что ниже рассмотрены только основные примеры использования Bootstrap для размещения контента и создания макета страницы. Более подробно общие принципы создания универсальных приложений с использованием фреймворка можно найти в специализированной литературе.

Для работы Bootstrap использует такое понятие, как компоненты — готовые блоки, использующие HTML и/или JavaScript с предопределёнными стилями. Наиболее часто применяются компоненты для создания сетки (колонок), навигационной панели (`nav`, `navbar`), для отображения большого центрального блока (`jumbotron`), контейнера (`container`) и пр. Можно использовать как готовый компонент, так и создать свой собственный, либо изменить вид готового компонента.

Выбор между Bootstrap и HTML5 Boilerplate зависит от конкретного проекта, поскольку именно его требованиями определяется выбор средств разработки, каждое из которых имеет свои преимущества. Однако возможность объединить их во многих случаях является несомненным достоинством. Поэтому во многих случаях для быстрого создания интерфейса используется Initializr [5]. Заметим, что Initializr автоматически интегрирует Boilerplate и Bootstrap друг с другом, и пользователь избавлен от необходимости производить дополнительные настройки. При помощи Initializr можно легко сгенерировать шаблон, позволяя выбрать только необходимые для проекта части. Очень важно, что Initializr позволяет значительно проще, чем при помощи только Bootstrap и HTML5 Boilerplate, создавать макет страницы. Однако, разработка новых версий СПО Initializr активно продолжалась до 2018 года, когда разработчики предложили в дальнейшем использовать Bootstrap и HTML5 Boilerplate напрямую. Тем не менее, последние доступные на сайте (<http://www.initializr.com/>) версии Initializr будут работать достаточно продолжительное время, до внесения браузерами или ОС кардинальных изменений, как правило, добавлений, в свою концепцию.

Таким образом, студенты получают навыки, необходимые для создания полноценных современных распределённых web-приложений, в особенности их фронтэнд части.

Литература

- [1] Мартишин С.А., Симонов В.Л., Храпченко М.В. Изучение студентами комплекса MEAN для создания серверной и клиентской частей современных web-приложений (тезисы доклада). // Материалы X Всероссийской научно-практической конференции «Информационные технологии в науке, технике, образовании» / 1–2 ноября 2018 года, Саратовский НИГУ им. Н.Г Чернышевского, г. Саратов. — с. 225–229.
- [2] Мартишин С.А., Симонов В.Л., Храпченко М.В. Современные технологии облачных вычислений над распределёнными данными (тезисы доклада). // Материалы Всероссийской научно-практической конференции «Информационные технологии в науке, технике, образовании» / 17 марта 2017 года, Саратовский НИГУ им. Н.Г Чернышевского, г. Саратов. — с. 49–53
- [3] HTML5 Boilerplate [Электронный ресурс]. — URL: <https://html5boilerplate.com/> — Яз. англ. — Дата обращения: 27.11.2018.
- [4] Bootstrap [Электронный ресурс]. — URL: <http://getbootstrap.com> — Яз. англ. — Дата обращения: 27.11.2018.
- [5] Initializr [Электронный ресурс]. — URL: <http://www.initializr.com> — Яз. англ. — Дата обращения: 27.11.2018.
- [6] h5bp/html5boilerplate.com [Электронный ресурс]. — URL: <https://github.com/h5bp/html5-boilerplate/> — Яз. англ. — Дата обращения: 27.11.2018.

Валерий Лаптев, Константин Обухов, Игорь Мурзаев
г. Астрахань, Астраханский государственный технический университет

Обучающая среда по программированию на базе СПО

Аннотация

За 5 лет эксплуатации была накоплена информация о достоинствах и недостатках учебного языка программирования и среды Semantic IDE. Принято решение реализовать полноценную обучающую среду для программистов на базе СПО силами студентов направления «Программная инженерия». Архитектура системы состоит из 2 систем: исполняющей и управляющей. Исполняющая подсистема — это интегрированная среда программирования на учебном языке, разрабатываемая на базе VSCode. Редактор дорабатывается до семантического редактора Semantic IDE. Программы на учебном языке транслируются в C++, который компилируется gcc в исполняемый модуль. Front-end транслятора реализуется с использованием ANTLR на языке C#, backend реализуется «в лоб» на C++. Исполняющая система является клиентом для серверной части — управляющей подсистемы. Управляющая подсистема, в свою очередь, состоит из подсистем генерации заданий, проверки решений и оценки решений. Система строится на базе БД PostgreSQL, функционал реализуется на языке C# с использованием .NET Core. Подсистема генерации осуществляет генерацию разнообразных заданий (тестовых и на программирование) на основе настроек, шаблонов, и с учётом достигнутого уровня обученности. Подсистема проверки первоначально проверяет решение на плагиат, а затем осуществляет проверку и оценку решения. Подсистема построена как олимпиадная система, но оценивание осуществляется на основе нескольких моделей. В частности, оценивается не только правильность решения, но и качество кода на основе метрик, и деятельность программистов в исполняющей системе.

После введения ЕГЭ в 2009 году в АГТУ на ИТ-специальности стали поступать студенты, которые не изучали программирование в школе. Возникла задача построить обучение программированию с нуля. В результате анализа средств обучения было принято решение разработать учебный язык программирования и реализовать среду для начального обучения программированию на этом языке.

К 2012 году была разработана среда программирования Semantic IDE, которая стала использоваться в учебном процессе. Основные концепции:

- среда поддерживает работу с одно- и многомодульными программами;
- среда обеспечивает независимый от платформы механизм накопления программных компонент на учебном языке;
- среда обеспечивает разработку обучающих материалов непосредственно в редакторе без привлечения сторонних средств;
- редактор кода оперирует конструкциями языка программирования и объектами программы; изменение ключевых слов в коде невозможно;
- ошибки определяются в момент набора кода программы;
- редактор позволяет набирать код и в русской, и в английской лексике;
- обеспечена возможность переключения лексики и синтаксиса языка программирования, причём это не приводит к повторному анализу кода программы.

Разработка велась силами студентов направления «Программная инженерия» под руководством автора. Основной разработчик был один, но иногда привлекались другие студенты. По итогам 5-летнего использования среды Semantic IDE в учебном процессе можно сделать следующие выводы:

- Участие в реальном проекте сильно повышало мотивацию студентов.
- Русская лексика учебного языка упрощает для начинающих понимание и усвоение основ программирования.
- Отсутствие синтаксических ошибок при наборе кода позволяет больше внимания уделять решению задачи, и скорость набора программы существенно выше, чем в традиционном текстовом редакторе.
- Отсутствие операций с буфером обмена не позволяет использовать «копипаст» и стимулирует студентов писать функции.
- Нетрадиционный язык программирования и нетекстовый формат файла программы значительно снижает количество попыток нечестного выполнения заданий.

Однако анализ самого проекта показывает, что имеется ряд серьёзных недостатков:

1. Проектная документация отсутствует, поэтому проект сложно развивать.
2. Проект был реализован с использованием бесплатных средств от Микрософт, поэтому работает только на платформе Windows и перенос его на Linux или в Интернет по трудоёмкости сравним с новой разработкой.
3. Среда является локальной, предназначена для одного пользователя и не содержит никаких средств для коллективной работы.
4. Учебный язык интерпретируется, исполняемый файл не создаётся, и программы работают только в среде, что не всегда удобно.

Кроме того, практика обучения показала, что многие начинающие программисты испытывают трудности при переходе от дружественной среды Semantic IDE к профессиональным средам с языком C++, даже несмотря на возможности переключения лексики и синтаксиса в Semantic IDE. Поэтому было решено разработать более развитую обучающую среду с использованием свободного ПО. Основные концепции:

1. Система должна обучать деятельности — программированию.
2. Система должна управлять деятельностью обучаемого в процессе обучения.
3. Управление деятельностью должно быть адаптивным на основе контроля усвоения учебного материала.
4. Система должна быть способна генерировать неограниченное количество заданий для обучения и контроля.
5. Система должна уметь оценивать и результаты деятельности, и саму деятельность обучаемого как индивидуально, так и в составе группы.

На основании этих принципов была спроектирована архитектура проекта.

Архитектура проекта

В систему входит 2 основных подсистемы: исполняющая и управляющая, которые взаимодействуют как клиент и сервер, — и две вспо-

могательных подсистемы: клиент преподавателя и сервер базы данных.

Исполняющая подсистема представляет собой интегрированную среду, в которой обучаемый работает с кодом на учебном языке программирования.

Управляющая подсистема взаимодействует со всеми подсистемами, и только она взаимодействует с сервером базы данных. В состав управляющей подсистемы входят: подсистема «антиплагиат», подсистема проверки задания, подсистема оценки задания, подсистема генерации заданий, подсистема адаптивного управления обучением на основе оценивания общего уровня усвоения.

Исполняющая подсистема

В качестве редактора интегрированной среды выбран свободно распространяемый редактор VSCode. Он поддерживает множество языков программирования (в частности, Python и C#) и включает разнообразные механизмы расширения и настройки. Важным фактором выбора послужила поддержка Git «из коробки».

Программа на учебном языке проверяется парсером, который реализуется на C# с использованием платформы NET Core. Парсер сгенерирован с использованием генератора парсеров ANTLR, для которого была разработана грамматика учебного языка, уточнённая на основе опыта использования среды Semantic IDE.

Интегрированная среда разрабатывается таким образом, чтобы её можно было использовать как автономно, так и в качестве клиента управляющей подсистемы.

В автономном варианте, если ошибки отсутствуют, код конвертируется в C++, причём код C++ можно видеть в редакторе.

В варианте клиента корректный код на учебном языке отправляется управляющей подсистеме для проверки и оценки, а управляющая подсистема сообщает результаты клиенту и сохраняет все данные в БД.

Управляющая подсистема

Основной подсистемой в составе управляющей подсистемы является подсистема адаптивного управления. Все остальные подсистемы являются подчинёнными и вызываются по мере необходимости.

Для каждого конкретного обучаемого в конкретном сеансе работы система управления формирует запрос на генерацию задания в соответствии с уровнем обученности данного ученика, и вызывает подсистему генерации заданий. Получив от подсистемы генерации вариант, управляющая система отправляет его клиенту.

Получив от клиента решение и лог деятельности обучаемого, подсистема управления запускает подсистемы проверки. Первой вызывается подсистема «антиплагиат», оценивающая вероятность плагиата решения.

Затем код решения передаётся подсистеме проверки выполнения, которая проверяет тестирование программы по схеме олимпиадных систем-контестеров. Результат тестирования передаётся подсистеме оценки, которая оценивает его по одной из моделей оценивания (строгая, нестрогая, со штрафами) в зависимости от настроек.

Затем решение передаётся подсистеме оценивания качества кода, которая оценивает решение на основе набора метрик. Лог деятельности передаётся подсистеме оценивания деятельности.

Отдельные результаты оценивания передаются в подсистему общей оценки уровня обученности.

Все подсистемы управляющей системы разрабатываются на языке C# с использованием платформы NET Core. Модульная структура управляющей системы позволяет проводить эксперименты с различными моделями и алгоритмами оценивания.

Сервер базы данных

База данных хранит следующую информацию:

- пользователи системы;
- множество выполненных заданий для подсистемы «антиплагиат»;
- множество тестовых вопросов разного типа для проверки знаний;
- наборы конкретных заданий по разным темам разной сложности;
- множество типовых шаблонов заданий для генерации конкретных вариантов.

Для каждого пользователя в базе данных сохраняются результаты выполнения, проверки и оценивания каждого задания. Кроме того, сохраняется лог каждого сеанса для оценивания деятельности обучаемого. На основе этой информации управляющая подсистема оценивает общий уровень усвоения и определяет траекторию обучения.

Сервер базы данных построен на основе свободно распространяемой СУБД PostgreSQL.

Текущее состояние проекта

В настоящее время все разработки выполняются группой студентов под управлением автора в рамках курсового проектирования по различным дисциплинам.

Первая версия системы «антиплагиат» была разработана в рамках дисциплины «Объектно-ориентированное программирование» на языке C++ с использованием компилятора GNU C++.

Первая версия исполняющей системы разработана в рамках дисциплины «Командный проект». Для разработки Language Server для VSCode использовался язык программирования C# (Net Core 2.1) и генератор парсеров ANTLR 4, для расширения редактора использовался TypeScript.

Конвертер в C++ реализуется «в лоб» по грамматике методом рекурсивного спуска. Конвертер настоящее время реализуется в 2 вариантах: на языке C++ в среде Code::Blocks с компилятором GNU gcc, и на языке C# на платформе NET Core в самом редакторе VSCode.

Первая версия управляющей системы разработана в рамках дисциплины «Командный проект». Разработана первоначальная структура БД, ядро управляющей системы, интерфейсы для взаимодействия с подсистемами. Начата разработка подсистемы генерации заданий, подсистемы проверки задания и оценки задания. Для отработки взаимодействия клиента и сервера написан веб-клиент преподавателя.

Разработка управляющей системы и веб-клиента осуществлялась на языке C# с использованием ASP.NET Core 2.1, Entity Framework Core 2.1, HTML5, библиотеки Bootstrap 3 и jQuery 1.11. Разработка ведётся с использованием Rider IDE 2018, предоставленной нашим студентам компанией JetBrains по программе JetBrains Product Pack for Students.

Для обеспечения командной работы с исходным кодом используется платформа Github.

Работа системы проверялась в ОС Windows 7 и выше, и в Linux-дистрибутивах, основанных на дистрибутиве Debian — начиная с Ubuntu 14.04.

К началу нового учебного года планируется завершить разработку исполняющей подсистемы в автономном варианте и начать опытную эксплуатацию. Для управляющей системы планируется завершить разработку указанных подсистем и начать стыковку с исполняющей подсистемой как клиентом.

Алиса Пикова, Любовь Бойко, Владимир Симонов

Москва, Федеральное государственное бюджетное образовательное учреждение высшего образования Российский государственный социальный университет (РГСУ), Московский государственный университет имени М. В. Ломоносова (МГУ)

Использование свободного программного обеспечения для проектирования устройств диагностирования психофизического состояния человека

Аннотация

Представлены возможности микроконтроллерных устройств на базе Arduino, IskraJS для проектирования устройств диагностирования психофизического состояния человека. Представлены прототипы, использующие методы электрокардиографии и фотоплетизмографии. В ходе проектирования осуществляется обучение студентов направлений «Информатика и вычислительная техника», «Информационные системы и технологии», «Программная инженерия» в области программирования, электроники и основ робототехники

Проблема диагностирования психофизического состояния человека в настоящее время волнует умы многих учёных. Это связано, прежде всего, с тем, что с человеком каждый день случаются различные события, в результате которых он испытывает разные по характеристикам психофизические состояния.

Указанные психофизические состояния человека являются важными во многих областях. Например, для тренеров разных стран мира, подготавливающих спортсменов для участия в больших спортивных мероприятиях, важно знать, в каком психофизическом состоянии находится спортсмен. В случае, когда спортсмен подавлен или его организм находится в состоянии физического изнеможения, быстрая диагностика поможет разработать быструю и эффективную методику для восстановления оптимального состояния спортсмена.

Следует сказать, что диагностика психофизического состояния также важна и для детей. В зависимости от возрастного периода, дети могут быть подвержены огромному количеству различных психологических состояний, диагностика которых поможет ограничению их негативного влияния на развитие детей. Указанное относится и к студентам вузов, и к школьникам.

Исходя из всего вышеперечисленного, актуальна задача создания небольшого мобильного устройства, обладающего достаточным набором методик диагностирования.

Поскольку проблема является актуальной достаточно длительное время, существует достаточно большое количество разработок в данной области [1-3]. К сожалению, большая часть из них ориентирована на узкоспециализированное использование.

В Российском государственном социальном университете в течение ряда лет в рамках студенческого научного кружка основ электроники, программирования и робототехники [4] проводятся работы по проектированию электронных устройств на базе таких платформ, как Arduino, IskraJS, Raspberry Pi, при этом студентами и магистрантами используются такие языки, как C++ (Wiring), JavaScript, Python. При проектировании электронных микроконтроллерных устройств был накоплен определённый опыт, и членами кружка было решено осуществить разработку прототипа электронного устройства диагностирования психофизического состояния человека. Указанный прототип явился результатом сотрудничества с кафедрой биологического факультета МГУ. Перед авторами стояла задача создания устройства, объединяющего две наиболее сильные методики диагностирования психофизического состояния человека [5], включающие в себя электрокардиографию (ЭКГ) и фотоплетизмографию (ФПГ), которые могут быть взяты за основу.

Для создания устройства используется оборудование (микроконтроллерные платы, датчики сердечного ритма, ФПП) и соответствующее программное обеспечение.

В настоящий момент осуществлено моделирование одной из важных составляющих устройства — пульсометра, где в качестве основы взято устройство [6], для которого переписана часть программного обеспечения. Также были определены основные потребные параметры для измерений (R-R интервал, аритмическая составляющая и др.). Полное устройство в данный момент находится в процессе проектирования. Для целей диагностирования планируется использование искусственной нейросети, обучение которой осуществляется на спортсменах-добровольцах факультета физической культуры РГСУ.

На последующих заключительных этапах планируется создание корпуса для устройства, с использованием технологии 3D-прототипирования и разработка специального ПО для работы с входными данными устройства.

В ходе разработки устройства, авторами предлагается идея создания визуального результата диагностики психофизического состояния человека. Поскольку не всегда будет иметься возможность обработки данных на компьютере, планируется снабдить устройство как небольшим дисплеем, выводящим результаты диагностики, так и светодиодами, индицирующими определённое психофизическое состояние.

В заключение можно отметить, что ещё на этапе разработки, устройство было продиагностировано авторами, где в ходе его тестирования, были получены данные о функциональности сердечной системы испытуемого. Таким образом, уже на этапе разработки устройство может быть использовано как вспомогательное для определения данных о состоянии сердца и сосудистой системы человека.

Литература

- [1] Ильин Е.П. Психофизиология состояний человека. СПб.: Питер, 2005. — 412 с.
- [2] Китаев-Смык Л.А. Психология стресса. Психологическая антропология стресса. М.: Академический Проект, 2009. — 943 с.
- [3] Филиппов М.М. Психофизиология функциональных состояний. К.: МАУП, 2006. — 240 стр.

- [4] Перспективное направление развития научно-технического творчества молодежи — кружок основ электроники и робототехники (тезисы доклада) / Александров П.О., Дмитриева Т.А., Малашенко М.А., Александрова Ю.Ю., Андрианов А.Е., Суханова А.Э., Симонов В.Л. — Материалы XXVII международной конференции «Современные информационные технологии в образовании» 28 июня 2016 г. — ДОГМ, Фонд «Байтик», АНО «Информационные технологии в образовании». — ИТО — Троицк — Москва. — 518с. — С. 454–455.
- [5] ECG vs PPG for Heart Rate Monitoring: Which is Best? [Электронный ресурс] . — URL: <http://neurosky.com/2015/01/ecg-vs-ppg-for-heart-rate-monitoring-which-is-best/> — Яз. англ. — Дата обращения 06.01.2019.
- [6] Пульсометр [Электронный ресурс] . — URL: <https://lesson.iarduino.ru/page/urok-27-pulsometr/>. — Дата обращения 06.01.2019.

Маркушевич Михаил Владимирович

Москва, ГБОУ города Москвы «Школа № 1352»

Проект: Написание диссертации на тему «Методика обучения информатике в основной школе на основе использования свободного программного обеспечения»

Основные методические подходы к построению методики преподавания информатики в основной школе, разработанной на базе свободного программного обеспечения

Аннотация

В докладе детально рассматриваются различные методические подходы к построению методики преподавания информатики в основной школе на базе свободного программного обеспечения. Автор выделяет методические подходы, зависящие от базового типа программного обеспечения и показывает, что данные подходы детерминированы преимуществами применения СПО в учебном процессе общеобразовательной школы, и находятся с ними в тесной причинно-следственной связи.

Анализируя ситуацию в отечественном общем образовании, необходимо отметить наличие серьёзного дефицита в области законченных методик преподавания информатики, основанных на применении свободного программного обеспечения в учебно-воспитательном процессе общеобразовательной школы. Чаще всего авторы методик и соответствующей учебной и учебно-методической литературы предлагают проприетарное программное обеспечение или, в лучшем случае, указывают свободные аналоги в качестве базовых учебных средств, но практически не обращают внимания на тот факт, что методики преподавания информатики, основанные на различных типах ПО будут иметь существенные различия в силу принципиально разных идеологий свободного и проприетарного программного обеспечения.

Отметим, что использование свободного программного обеспечения в общем образовании имеет следующие существенные преимущества по сравнению с проприетарным ПО, к которым можно отнести следующие:

- *Бесплатность лицензий* на использование свободного программного обеспечения.
- *Кроссплатформенность* большей части СПО.
- *Эквивалентность* школьного и домашнего программного обеспечения учащихся и преподавателей.
- *Относительно низкие системные требования*, характерные для большинства программ, относящихся к СПО.
- *Коммюнитарная идеология* СПО.
- *Стимулирование соблюдения авторских прав* учащимися [3].

Занимаясь изучением данной темы, автор обнаружил, что при разработке методики преподавания информатики в основной школе некоторые методические подходы к построению данной методики будут органически обоснованы перечисленными выше преимуществами использования СПО. К таким подходам можно отнести следующие:

- Использование в учебном процессе дополнительно к основным современным персональным компьютерам также и устаревшей, списанной компьютерной техники.
- Использование исключительно практических домашних заданий, выполняемых учащимися на домашних компьютерах.

- Проведение лабораторных работ по установке и настройке операционных систем семейства Linux на реальном физическом компьютере, а не виртуальной машине.
- Обучение по проектной системе, при которой в процессе изучения каждой темы курса информатики основной школы учащимися выполняется учебный проект, результатом которого является некий законченный информационный объект.
- Использование понятия «информационное компьютерное моделирование» в качестве системообразующего в курсе информатики и организация учебных проектов таким образом, что все они посвящены процессам создания и изучения информационных компьютерных моделей различной природы и типа.
- Использование интегрированных методов обучения, построенных по формуле «метод обучения + компьютерное моделирование».
- Формирование личностных универсальных учебных действий с помощью использования коммунотарной идеологии свободного программного обеспечения.

Из представленной на рисунке 1 структурной схемы, отражающей взаимосвязь между преимуществами использования свободного программного обеспечения в учебном процессе и методическими подходами к построению методики преподавания информатики в основной школе можно сделать следующие выводы.

В первую очередь, можно выделить *первичные преимущества* использования СПО, такие как: бесплатность, кроссплатформенность и низкие системные требования к аппаратному обеспечению, коммунотарная идеология СПО и вытекающие из них *вторичные преимущества*, такие как: эквивалентность домашнего и школьного программного обеспечения, стимулирование учащихся к соблюдению авторских прав, отсутствие повреждения в духовно-нравственной сфере, возможность обмена информационными объектами, созданными учащимися в ходе обучения.

Во вторую очередь, необходимо отметить, что применение некоторых методических подходов становится возможным лишь благодаря перечисленным выше преимуществам свободного программного обеспечения. Так, например, использование практических домашних заданий в процессе преподавания информатики в основной и старшей



Рис. 1: Взаимосвязь между преимуществами использования СПО в учебном процессе и методическими подходами.

школе имеет значительный дидактический потенциал в виду высокого процента содержания информационных технологий в содержании образования по данному предмету. Действительно, теоретическое домашнее задание по темам «Технология обработки текстовой информации» или «Технология создания и обработки графической информации (растровая и векторная графика» выглядели бы достаточно парадоксально.

Однако, достаточно часто преподаватели информатики, работающие по методикам, основанным на проприетарном программном обеспечении вынуждены давать учащимся теоретические домашние задания, так как на их домашних компьютерах не установлено программное обеспечение, используемое в классе, такое как Microsoft Office, Adobe Photoshop, Adobe Illustrator и другое аналогичное. Более того, перечисленное ПО и не может быть установлено в виду весьма серьезной стоимости лицензии на его использование, которая состав-

ляет соответственно 4199 рублей за Microsoft Office 2016 [4] и 27163 рубля за Adobe Photoshop CC 2018 (подписка на 1 год)[6].

В свою очередь, невозможность оперировать практическими домашними заданиями делает не реализуемым в полном объеме обучение по системе учебных проектов, выполнение которых учащимися планируется как в школе, так и дома, что негативно влияет на мотивацию школьников к изучению как информатики, так и других учебных предметов, в которых используется метод проектов.

Особое внимание надо уделить формированию личностных универсальных учебных действий с помощью коммунарной [1], по терминологии Бердяева Н.А., идеологии СПО. В основу идеологии данного типа положен принцип свободы, без усвоения которого само формирование личностных УУД вряд ли возможно, так как несвободный, зависимый человек не способен ни к самоопределению, ни к саморазвитию, иными словами, вообще ни к какой форме творческой деятельности [5].

Таким образом, на основании вышесказанного, можно констатировать наличие зависимости между типом используемого при разработке методики преподавания информатики в основной школе программного обеспечения и применяемыми в данной методике методическими подходами [2]. В случае использования свободного программного обеспечения в качестве базового увеличивается многообразие применяемых методических подходов, а также становятся доступны те подходы, которые в принципе невозможно было бы реализовать при использовании проприетарного программного обеспечения, что делает методику преподавания информатики, основанную на СПО, более перспективной для использования в общеобразовательной школе.

Литература

- [1] Коммунарность. Философский словарь / <http://slovari-online.ru/word/философский-словарь/коммунарность.htm>
- [2] Маркушевич М.В. Зависимость между методической системой обучения информатике в школе и типом используемого в ней программного обеспечения // Информатика в школе. № 9 (142). 2018. С. 14–24.
- [3] Маркушевич М.В. Проблема выбора между платным и свободным программным обеспечением для использования его в учебном процессе // Информатика и образование, № 10, 2016. – с. 37–45;

- [4] Офисное приложение MICROSOFT Office для дома и учёбы 2016. Официальный сайт магазина Ситилинк // https://www.citilink.ru/catalog/computers_and_notebooks/soft/soft_office/382626/
- [5] Федосов А.Ю., Маркушевич М.В. Влияние коммонотарной идеологии свободного программного обеспечения на формирование личностных универсальных учебных действий / Актуальные проблемы обучения математике и информатике в школе и вузе: материалы IV Международной научной конференции в двух томах. Т. 1. Москва, ФГБОУ ВО «Московский педагогический государственный университет» (МПГУ), 4–5 декабря 2018 г. / Под ред. М.В. Егуповой, Л.И. Боженковой. — Калуга: Издательство АКФ «Политоп», 2018. — с. 231–238.
- [6] Adobe Photoshop CC 2018. Официальный сайт магазина Allsoft // https://allsoft.ru/software/vendors/adobe/adobe-photoshop/\#ver_9625054

А.Ю. Никитин, Н.Н. Никитина

г. Калуга, Калужский государственный университет им. К.Э. Циолковского
<http://tksu.ru/>

Использование офисных пакетов OpenOffice и Myoffice в учебном процессе в Калужском государственном университете им. К.Э.Циолковского

Аннотация

В докладе обобщается опыт использования офисных пакетов OpenOffice и MyOffice при обучении бакалавров разных направлений подготовки в Калужском государственном университете им. К.Э. Циолковского. Большое внимание уделяется особенностям и сложностям перехода на использование свободного ПО при работе со студентами

В настоящее время в нашей стране уделяется большое внимание процессу перехода на отечественное программное обеспечение (ПО). Данная задача была поставлена в указе Президента Российской Федерации В.В.Путина от 07.05.2018 № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года».¹

¹ Официальный интернет-портал правовой информации. Режим доступа: <http://publication.pravo.gov.ru/Document/View/0001201805070038>

В соответствии с данным указом правительство обязано обеспечить ускоренное внедрение цифровых технологий в экономике и социальной сфере до 2024 года.

Согласно национальной программе «Цифровая экономика Российской Федерации»² правительство должно обеспечить в 2024 году использование преимущественно отечественного программного обеспечения государственными органами, органами местного самоуправления и организациями.

Реализация программы «Цифровая экономика Российской Федерации» требует организовать в системе российского высшего образования подготовку специалистов, владеющих навыками работы с отечественным ПО. В настоящее время в Федеральном государственном бюджетном образовательном учреждении высшего образования «Калужский государственный университет им. К.Э. Циолковского» уделяется необходимое внимание для решения этой задачи. В образовательных целях в качестве альтернативы применения пакета MS Office рассматривается использование в учебном процессе офисного пакета MyOffice (МойОфис), включённого в Единый реестр российских программ для электронных вычислительных машин и баз данных.

Пакет ПО МойОфис Образование поставляется в учебные заведения по бессрочной лицензии «с нулевой стоимостью», что даёт возможность широко использовать его в рамках всех направлений подготовки, осуществляемых в университете. К сожалению, за пределами вуза у студентов возникает проблема: либо покупать пакет МойОфис как частное лицо, что весьма затратно, либо использовать альтернативный бесплатный пакет программ. На данный момент учащиеся пользуются в основном льготным пакетом MS Office для учебных заведений, но данное предложение компании Microsoft может прекратить своё существование в любой момент. В связи с этим остро встает вопрос о поиске альтернативных программ, и таким выбором может стать свободно распространяемое программное обеспечение в виде пакета OpenOffice.

Оба пакета программ (OpenOffice и MyOffice) способны удовлетворить основные потребности студентов в учебном процессе и по функционалу не уступают пакету Microsoft. Они содержат альтернативы программам Word, Exel и PowerPoint.

² Сайт правительства Российской Федерации Режим доступа: <http://government.ru/rugovclassifier/614/events/>

В связи с данными обстоятельствами и переходом на отечественное ПО в нашем вузе разработаны и реализуются учебные курсы по дисциплине ИКТ («Информационные компьютерные технологии») для бакалавров для всех направлений подготовки, позволяющие освоить работу в разных офисных пакетах.

В связи с тем, что большинство учащихся, поступающих в Калужский государственный университет им. К.Э. Циолковского, имеют навыки работы с программами из пакета MS Office, то основные навыки по подготовке и оформлению учебной документации осуществляются с помощью программ Microsoft. После овладения навыками использования данных программ в курсе ИКТ проводятся занятия по знакомству с программами пакета МойОфис и сравнением функционала изученных программ. Для дальнейшего освоения российского программного продукта предлагается все домашние задания выполнять с помощью программ МойОфис или OpenOffice. Все проверочные и самостоятельные работы в курсе ИКТ выполняются в пакете МойОфис, а тематика докладов и сообщений связана с сравнением и демонстрацией возможностей офисных пакетов OpenOffice и MyOffice.

При использовании подобного варианта обучения студенты успешно овладевают навыками работы с пакетами отечественного программного обеспечения и могут убедиться в его конкурентоспособности, т.к. им приходится постоянно сравнивать функционал офисных пакетов и перерабатывать материал из одного пакета в другой.

В целом такой вариант обучения позволяет студентам освоить не один, а сразу несколько офисных пакетов и не испытывать затруднений при использовании отечественного ПО в их будущей трудовой деятельности.

Николай Говорухин

Калуга, Калужский государственный университет имени К. Э. Циолковского
<http://nichost.ddns.net/>

Web хостинг на основе персонального компьютера, создание доменных имён третьего уровня

Аннотация

Основной целью моей работы была компоновка и полная настройка свободного программного обеспечения для создания сервера и Web хостинга на основе персонального компьютера, с подробной инструкцией регистрации доменных имён третьего уровня, без использования паспортных данных и какой-либо оплаты.

Главным связующим звеном является веб сайт в котором уже заранее встроена система обратной связи через электронную почту, для активации данной функции требуется ввести свои данные в файле mail.php.

При запуске всего пакета данный веб сайт будет открываться автоматически в вашем браузере, говоря о том, что сервис работает в нормальном порядке и полностью готов к использованию. Так же веб сайт имеет простую и интуитивно понятную схему, полностью на русском языке.

Для запуска всего сервиса требуется установить основную программу которой является Open server созданной на технологии Apache. Данная программа позволяет запускать как локальный так и глобальный сервер, она имеет графический интерфейс и возможность добавлять большое количество веб ресурсов, единственным ограничением является системные возможности вашего компьютера. Для правильной работы программы требуется ввести ваш ip в настройках. В последствии данный ip будет привязан к доменному имени основного сайта веб хостинга. Так же для проверки работоспособности всей системы необходимо провести трассировку с помощью Cnd команды tracer, после которой любое доменное имя сайта находящегося на вашем сервере будет выдавать ip-адрес вашего компьютера.

Единственной проблемой является динамический ip который меняется каждые 15 минут. Для решения данного вопроса нам необходима технология DDNS предоставляемая сервисом NO-IP, она позволяет привязывать ip данный провайдером к вашему доменному имени. Для активации DDNS требуется установить данную программу.

Для финальной настройки всей системы необходимо создать одно и более доменных имён. В этом на поможет сайт <https://www.noip.com/> на котором на необходимо создать аккаунт и зарегистрировать необходимое количество доменов. Данный пакет позволит проверять работоспособность веб ресурсов в которых применяется технология РНР, централизуя всю вычислительную нагрузку на одном компьютере.

Литература

- [1] Основы сервера Apache. <https://www.hostinger.ru/rukovodstva/web-server-apache/#gref>
- [2] Создание FTP соединения. <https://sonikelf.ru/ftp/>
- [3] Структура доменного имени. <https://blogfreo.ru/vse-o-domennyx-imenax/>

Владимир Суховерхий
Москва, НИУ «МЭИ»

Linux как решение проблемы устаревания компьютеров в учебном заведении

Аннотация

Устаревшее аппаратное обеспечение не находит применения на актуальный момент времени в учебных заведениях и как правило складывается на территории, либо выбрасывается, тем самым загрязняя окружающую среду и нанося финансовые убытки учебному заведению. В работе рассматривается метод решения проблемы на примере решения аналогичной проблемы на территории вычислительного центра НИУ «МЭИ».

В мире, согласно данным ООН по программе о защите окружающей среды, каждый год человечество выбрасывает на свалку от 20 до 50 миллионов тонн электронно-вычислительных устройств. Это число непрерывно увеличивается, нанося тем самым ежегодно ущерб окружающей среде и экономике предприятия или частного лица. [1]

Устаревшее аппаратное обеспечение не находит применения на актуальный момент времени в учебных заведениях и как правило складывается на территории, либо выбрасывается, тем самым загрязняя

окружающую среду. В тоже время складирование устаревшего оборудования на территории приводит к потере пространства, чем вызывает захламление, которое приводит к нарушениям техники пожарной безопасности, а также из-за потери рабочих площадей идёт потеря необходимого пространства, которое могло бы предназначаться для учебных или рабочих целей. [2]

Устаревание аппаратного обеспечения можно определить, как невозможность выполнять поставленные перед ним функции и задачи, и невозможность применения его в рабочем процессе, в связке с программным обеспечением.

В ходе исследования включающего в себя анализ причин возникновения проблемы устаревания аппаратного обеспечения были сделаны следующие выводы:

1. Аппаратное устаревание не сильно влияет на общее устаревание устройства, по сравнению с устареванием программного обеспечения, вызванного использованием закрытого проприетарного программного обеспечения.
2. Невозможное или затруднительное использование вычислительного устройства для новых задач, при использовании в связке с актуальными вычислительными устройствами.

Для решения проблем связанным с устареванием программного обеспечения, согласно сделанным выводам, было принято решение о построении кластера из персональных компьютеров с устаревшим аппаратным и программным обеспечением.

В качестве операционной системы на использовался дистрибутив операционной системы на базе ядра Linux.

Вычислительный кластер состоит из трёх персональных компьютеров в корпусе «тауэр», с характеристиками, перечисленными в Таблице 1.

Компьютеры, представленные в таблице 1, были объединены гигабитной локальной сетью. На каждом компьютере находящейся в сети был предустановлен дистрибутив операционной системы Debian, использующей ядро Linux.

Основным компьютером в вычислительной локальной сети кластера был сделан Компьютер 1, при помощи пакета *Cacher*. Основной компьютер вычислительной сети, так же был назначен консолью кластера.

	Компьютер 1	Компьютер 2	Компьютер 3
CPU	AMD Athlon(tm) 64 X2 Dual Core Processor 5200+	Pentium 4	AMD Athlon(tm) 64 X2 Dual Core Processor 5200+
RAM	2 Gb	2 Gb	2 Gb
HDD	120 Gb	80 Gb	80 Gb

Таблица 1:

Для непосредственного распределения программы на компьютерах входящих в состав вычислительной сети кластера, на каждый компьютер были предустановлены пакеты программного обеспечения: OpenMPI, SSH, NFS.

- **SSH** позволяет организовать доступ к каждому компьютеру входящего в вычислительный кластер пользователю или программе работающей через библиотеку MPI получить доступ к управлению вычислительными мощностями каждого компьютера входящего в кластер, для распределения выполняемых задач между ними.
- **OpenMPI** пакет программного обеспечения для распределённых вычислений, включает необходимый модуль `mpirexec`, включающий в себя диспетчер задач и менеджер очередей, необходимых для построения кластера
- **NFS** модуль поддержки сетевой файловой системы, позволяющий организовать общий для всех узлов кластер каталога, где будут храниться исходники программ для параллельных вычислений и их исполняемые модули и файлы содержащие входные и выходные данные.

В дальнейшем были проведены сравнительные испытания. В качестве сравнения с кластером, построенным из компьютеров с устаревшим аппаратным обеспечением, был использован сервер с актуальным аппаратным обеспечением работающий под управлением закрытого проприетарного обеспечения. В Таблице 2 приведены характеристики сервера с актуальным аппаратным обеспечением.

Для сравнения была запущена расчётная программа, написанная на языке программирования высокого уровня Python.

CPU	Intel Xeon E3 — 1285V4
RAM	8 Gb
HDD	1 Tb

Таблица 2:

	Запуск 1 (мин.:сек)	Запуск 2 (мин.:сек)	Запуск 3 (мин.:сек)	Запуск 4 (мин.:сек)	Запуск 5 (мин.:сек)	Запуск 6 (мин.:сек)	Среднее время (мин.:сек.)
Кластер	16:10	16:05	16:11	16:08	16:12	16:09	16:09
Сервер	16:02	15:07	15:58	16:05	16:03	16:02	16:03

Таблица 3:

Сравнительные результаты, отображающие затраченное время на работу приложения по расчёту математической модели объекта в высокочастотном поле, представлены в таблице 3.

Как можно сделать вывод из Таблицы 3, использование кластера из устаревшего оборудования, позволяет достичь результатов схожих по производительности с современным бюджетным оборудованием.

В качестве вывода стоит отметить что использование кластеров, построенных из устаревших вычислительных устройств, тем самым применив их в качестве оборудования для проведения лабораторных работ и научных исследований.

За счёт применения уже существующего оборудования появилась возможность сэкономить на закупке нового бюджетного оборудования и уменьшить выброс технического мусора, состоящего из работоспособного устаревшего оборудования, тем самым очистив новые рабочие площади.

Литература

- [1] Анастасия Литвинова, Как осуществляется переработка электроники?, 2015, <https://nature-time.ru/2015/03/kak-osushhestvlyaetsya-pererabotka-elektroniki/>
- [2] С. В. Артеменко, Утилизация компьютерной техники, 2012

Андрианов Андрей Михайлович

Москва, НИУ МИЭТ

Проект: Проектное командное обучение

<http://иповс.рф/education/list/obrazovatelnye-proekty/proektnoe-komandnoe-obuchenie-pko.html>

Проектное командное обучение с использованием свободного программного обеспечения

Аннотация

В НИУ МИЭТ (<http://miet.ru>) на кафедре ИПОВС (<http://иповс.рф>) в ряде учебных курсов проходит апробацию проектный командный способ обучения — ПКО (<http://иповс.рф/education/list/obrazovatelnye-proekty/proektnoe-komandnoe-obuchenie-pko.html>) с применением СПО, такого как Java, PHP, JavaScript, MySQL и т.д.

Сочетание ПКО и СПО позволяет улучшить качество подготовки студентов и стимулировать доверительные отношения между вузом и бизнесом в вопросах управления перспективными направлениями обучения.

Принято, что главная задача вузов — подготовка квалифицированных кадров. Конечно для этого вузы должны вести научные и коммерческие разработки но, перефразируя 3-й закон робототехники А. Азимова, в той мере, в которой это не противоречит главной задаче. Унаследованная РФ от СССР система развития и внедрения науки в виде вузов, НИИ и КБ на предприятиях была рассчитана на «большую мощность». С распадом СССР система НИИ и КБ предприятий, как минимум, количественно заметно уменьшилась, а количество вузов, наоборот — возросло. С регрессом НИИ и КБ уменьшилось поступление в вузы информации для актуализации дисциплин обучения. Возникла ситуация, когда студентов — много, а науки для них — мало. Принятая болонская система неявно разделила выпускников вузов на «современных рабочих» — бакалавров и «современных учёных» — магистров. Работа бакалавра — проще диплома вуза СССР, но диссертация магистра — сложнее. По мнению автора, именно подготовка магистров сейчас несёт наибольшие явные и не явные квалификационные риски связанные с вышеуказанным «недостатком количества науки». К явным рискам можно отнести:

1. вузы имеют теперь подушевое финансирование и заинтересованы в том, чтобы не отчислять слабых магистров и бакалавров. Поэтому откровенно слабые и/или не мотивированные студенты не выбывают, как раньше после 1 и 2-го курсов. И далее слабые бакалавры становятся слабыми магистрами.
2. Магистру трудно попасть в фирму с большой долей науки (таких — относительно мало), чтобы иметь достаточную научную практику и доступ к соответствующему оборудованию.
3. Ввиду того, что «много науки» предприятия РФ не могут себе сейчас позволить, если магистр всё-же попал в фирму «с наукой», то ему трудно попасть в КБ, поскольку такое КБ тоже не большое, и в следствии этого, такое КБ уже укомплектовано опытными сотрудниками.

К неявным рискам можно отнести следующее:

1. Практически всегда сообразительные и ответственные студенты устраиваются на 2-4 курсах на работу «обычными» программистами т.е. в терминах информационной эпохи — рабочими и вливаются в «рабочий класс», а не в инженерный, как было при СССР. Поработав 2-3 года «рабочим», получив квалификацию и хорошую зарплату они уже не видят острой необходимости в приложении особых усилий для написания сильной магистерской работы.
2. Работа на рабочих специальностях приучает мозг к решению задач определённого уровня сложности и он (мозг) к этому привыкает и становится инертен к решению задач более высокого, научного уровня сложности.

В итоге, поскольку магистр, как правило, работает на том предприятии, которое даёт ему «крышу» для практики, то выбор между зарплатой и диссертацией, очевидно, делается не в пользу диссертации. В результате страдают все: магистр «вымучивает» содержание своей диссертации и не становится учёным, вуз выпускает слабых магистров, предприятие теряет время работника на задачи, которые не приносят ему выгоды.

Требуется найти способ подготовки кадров, который бы снижал перечисленные риски. В качестве такого способа предлагается к обсуждению проектное командное обучение — ПКО, проходящее апробацию на кафедре ИПОВС НИУ МИЭТ. Пробразом этого подхода

была первая в РФ коллективная защита диплома в МИЭТ в 2005-м году [1] о которой стало известно автору в 2015 году с приходом в НИУ МИЭТ. ПКО удобно проводить с использованием свободного программного т.к. студенты могут вести разработку в домашних условиях, а не только в классе ПК вуза и не ограничены временем доступности класса ПК вуза.

В соответствии с текущей реализацией ПКО поток обучаемых получает на выбор два варианта обучения, позволяющее выделить людей с познавательной мотивацией [2]:

- «Стандартное — лабораторные + лекции, с максимальным баллом — 4.
- ПКО — для коллектива из 2-4 разработчиков с максимальным баллом — 5.

Если курс посвящён конкретному языку, инструменту или технологии, например, «Основы языка Java», то на лабораторных команды ПКО демонстрируют освоение рассматриваемых в лабораторных тем в рамках своего проекта.

Если курс имеет более общий характер, например, «Интернет программирование», то студенты могут выбрать и утвердить у преподавателя свой инструментарий для решения задачи. При этом они должны использовать не менее 50% технологий, из программы курса, а остальные подбирают сами. На лабораторных, в этом случае, демонстрируется прогресс в решении задачи.

Если задача была предложена компанией, партнёром кафедры по ПКО, прошла адаптацию к учебной форме, была выбрана командой студентов и решена, то такое решение рассматривается, как прохождение собеседования; возможно с большей стартовой зарплатой, нежели прохождение «обычного» собеседования.

Таким образом в ПКО:

1. Студенты практически тренируются на задачах «приближённых к боевым».
2. Студенты проходят «боевое слаживание» в ротлируемых командах разработчиков с внутренним самостоятельным разделением ролей по способностям и трудолюбию.
3. В случае успешного решения задачи компании-партнёра, для студенческой команды заранее: за 0,5-1,5 года, открываются перспективы коллективного трудоустройства и практики, как для ВКР, так и в магистратуре.

4. Уравниваются первичные риски знакомства студента и работодателя т.к. есть время «притереться» друг к другу, а студенту подтянуть слабые «места».
5. У студентов с научным складом ума есть время на поиск с практической проверкой таких задач, которые им нравятся и имеют диссертационные магистерские перспективы. Соответственно и выбор для практики фирмы, которая такую задачу предоставила, улучшает шансы на сильную магистерскую работу.
6. Компания, предоставившая задачу получает перспективы её практического (в рамках ПКО и ВКР) и научного (в рамках магистратуры) решения.
7. вуз имеет возможность накапливать базу задач, анализ и обобщение которой потенциально позволяет разделить прикладные и фундаментальные знания, требующиеся студенту на рынке труда и внести соответствующие актуальные коррективы в учебные курсы. Таким образом, системно (см. рис.1.) восстанавливается коммуникация вуза и производства, причём, как наукоёмкого (для квалификации — магистр) так и «трудоемкого» (для квалификации — бакалавр)

На кафедре ИПОВС НИУ МИЭТ первая апробация ПКО была проведена во втором семестре 2016-2017 учебного года для студентов 3-го курса в рамках дисциплины «Интернет-программирование». Всего сформировалось 6 команд. Задачи они выбирали себе сами и согласовывали с преподавателем. В конце семестра была организована защита проектов с приглашением представителей фирм г. Зеленограда. По итогам защиты 3 команды нашли себе фирмы для будущей бакалаврской и магистерской практик. Более подробно защита 2017 года описана по ссылке — <http://иповс.пф/education/list/obrazovatelnye-proekty/proektное-komandное-obuchenie-pko.html>.

Во втором семестре 2017-2018 учебного года было всего 2 проекта и отдельная защита не проводилась. С учётом этого снижения в 2018-2019 учебном году в механизм ПКО были внесены коррективы. Задачи для ПКО были предложены студентам уже в первом семестре в дисциплине «Основы Java». Был сформирован пул из 9-ти задач собранных в 3 группы: логистика, документооборот, визуализация. Разрешалось использовать только язык Java. Дополнительно были сформулированы требования к качеству и составу решения задач. Идея

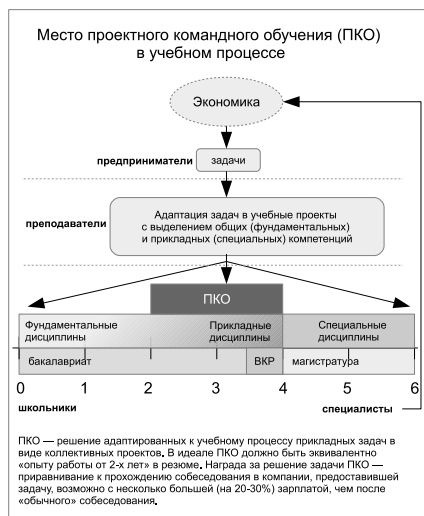


Рис. 1: Место проектного командного обучения в учебном процессе

изменений заключалась в том, чтобы к началу второго семестра, когда студентам будет разрешено самостоятельно предложить задачу и выбирать технологии её решения, они уже «набили шишек», а также имели задачи-заделы, которые они, при отсутствии новых идей, могли бы продолжить разрабатывать. Сформировалось 11 команд. Из них 8 с большим или меньшим качеством завершили свои разработки и 3 сошли «с дистанции».

Внесённые изменения, позволили взглянуть на ПКО более комплексно (см. рис.2.)

Наиболее ответственной частью ПКО является подбор и адаптация задач. Пока задачи выбирались автором и студентами, но в полную силу ПКО может заработать только тогда, когда задачи начнут поступать от фирм — разработчиков ПО. Такой подход хорошо согласуется с показателем оценки деятельности вуза на основе белых зарплат, полученных выпускниками вуза за первые три года их работы [3].



Рис. 2: ПКО в течение 4-х семестров, как эквивалент накопления 2-х летнего опыта разработки

Литература

- [1] *А.В.Капранов*. В МИЭТе прошла первая в Зеленограде коллективная защита диплома // Электронное СМИ «Зеленоград.ру». 2005 г. Ссылка: <https://www.zelenograd.ru/news/7/>. Проверено 16.01.2019.
- [2] *Г. Курячий*. Как я делал проверку копинасты для спецкурса по Python3 и что из этого вышло // Тринадцатая конференция Свободное программное обеспечение в высшей школе. Тезисы докладов, М.: Институт Программных Систем РАН, 2018. — С. 34–38.
- [3] *С.Абрамов*. Ошибки в государственном надзоре за высшим образованием главная проблема высшего образования в России // Тринадцатая конференция Свободное программное обеспечение в высшей школе. Тезисы докладов, М.: Институт Программных Систем РАН, 2018. — С. 56–57.

Татьяна Сундукова, Галина Ванькина

Тула, Тульский государственный педагогический университет им. Л.Н. Толстого

Роль свободного программного обеспечения в преподавании дисциплины «Технологии разработки баз данных» в вузе

Аннотация

В условиях реализации ФГОС высшего образования при подготовке бакалавров ИТ-направлений ключевым вопросом остаётся выбор программного обеспечения для реализации образовательной программы. На основе требований ФГОС и в соответствии с требованиями к выпускникам вузов необходимо выстраивать обучение с применением бесплатного программного обеспечения, отвечающего требованиям актуальности, безопасности, соответствия стандартам.

В связи с процессами модернизации российского образования, внедрением в образовательный процесс новых Федеральных государственных образовательных стандартов, значимости развития компетенций у современных специалистов, формирование профессиональных компетенций у будущих ИТ-специалистов должно рассматриваться как одно из приоритетных направлений деятельности высшей школы [1, 2].

Термин «ИКТ-компетентность» относится к ключевым терминам ФГОС ВО и определяется как способность и умение самостоятельно искать, анализировать, отбирать, обрабатывать и передавать необходимую информацию при помощи коммуникационных и информационных технологий. Техническая и технологическая подготовка будущих специалистов в условиях современного образовательного пространства предполагает наличие базовых универсальных действий по работе с информацией: поиск, аналитическая оценка и эффективное использование результатов на основе известного и осваиваемого ПО.

При реализации образовательной программы бакалавриата по направлению «Математическое обеспечение и администрирование информационных систем» в базовой части учебного плана предусмотрена дисциплина «Технологии разработки баз данных», для преподавания которой необходимо обеспечить требования ФГОС, в частности:

- «Организация должна быть обеспечена необходимым комплектом лицензионного программного обеспечения (состав определяется в рабочих программах дисциплин (модулей) и подлежит обновлению при необходимости)». [3]
- «Обучающимся должен быть обеспечен доступ (удалённый доступ), в том числе в случае применения электронного обучения, дистанционных образовательных технологий, к современным профессиональным базам данных и информационным справочным системам, состав которых определяется в рабочих программах дисциплин (модулей) и подлежит обновлению (при необходимости)» [3]

Дисциплина «Технологии разработки баз данных» направлена на формирование у студентов знаний, умений и владений в области разработки и программирования баз данных. В результате изучения данной дисциплины студенты должны научиться решать прикладные задачи с помощью программного обеспечения и средств вычислительной техники, раскладывать сложные задачи на независимые подзадачи, представлять, как можно использовать возможности компьютера для решения задач конкретной предметной области. Преподавателю необходимо провести систематизацию и выравнивание знаний студентов в области программирования на скриптовых языках, поскольку они могут сильно варьироваться вследствие различного начального уровня подготовки студентов. Содержание лабораторно-практических работ представлено на основе базовых тем курса:

- Введение в предмет.
- Основные понятия. Модели данных.
- Представление данных в информационных системах.
- Реляционная модель данных.
- Языковые средства доступа к БД.
- Реляционная алгебра и исчисление.
- Проектирование баз данных.
- Иерархическая и сетевая модели данных.
- Методы специальной обработки.

Основы работы с технологиями баз данных формируются поэтапно: от реляционных моделей данных к постреляционным и объектно-ориентированным пакетам. Немаловажным аспектом при выборе ПО

является соответствие стандартам, кроссплатформенность, доступность и бесплатность. Ключевым вопросом при организации обучения является обеспечение учебного процесса соответствующим программным обеспечением, к которому можно отнести свободную реляционную СУБД MySQL.

Выделим особенности СУБД MySQL:

- стандартизированное (SQL:2003, SQL:2008) и бесплатное программное обеспечение;
- возможность использования в качестве удалённого сервера, в том числе на виртуальной или локальной машине;
- поддержка большого количества наиболее используемых типов таблиц;
- комплект поставки включает тип EXAMPLE, соответствующий основным принципам создания новых таблиц;
- характеризуется высокой степенью масштабируемости за счёт поддержки большинства популярных платформ;
- открытый исходный код позволяет реализовать модульность и поддержку новых версий;
- разнообразие API обеспечивает поддержку большого числа программных языков;
- поддержка таблиц больших размеров — максимальный размер файла таблицы базы данных ограничивается возможностями операционной системы.

В процессе работы с СУБД MySQL у студентов формируются навыки работы по:

- установке, настройке и администрированию сервера баз данных;
- использованию актуального вида ПО в области разработки структуры баз данных с поддержкой основных типов данных;
- реализации типовых SQL-запросов к данным с возможностью редактирования кодов и тестирования результатов;
- реализации возможности резервного копирования и восстановления данных при поддержке разных форматов данных;
- обеспечению определённого уровня безопасности данных;
- разработке модулей, расширений и обработчиков баз данных.

Дальнейшее изучение технологий баз данных в рамках других дисциплин может быть реализовано на основе соответствующих пакетов, при этом базовые навыки по работе с данными оставляют основу для реализации более сложных и многоаспектных алгоритмов обработки данных.

Литература

- [1] Ванькина, Г.В., Сундукова, Т.О. Использование технологий 1С как средства формирования профессиональных компетенций будущих программистов // Новые информационные технологии в образовании: Сборник научных трудов XIII международной научно-практической конференции. Часть 1. — М.:ООО «1С-Паблицинг», 2013. — С. 117-122.
- [2] Ванькина, Г.В., Сундукова, Т.О. Обучение программистов в вузе на базе технологий «1С» в условиях перехода к новым образовательным стандартам // Университет XXI века: научное измерение: Материалы науч. конф. профессорско-преподавательского состава, аспирантов, магистрантов и соискателей ТГПУ им. Л.Н. Толстого: В 2 т. Т.1. Тула: Изд-во Тул. гос.пед.ун-та им. Л.Н. Толстого, 2010. — С. 195-200.
- [3] Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриат) http://fgosvo.ru/uploadfiles/ProjFGOSV03++/Bak3++/020303_B_3plus_21062017.pdf

Сергей Мартишин, Марина Храпченко

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

Изучение технологий распределённого кэширования данных на базе СПО в высоконагруженных системах

Аннотация

Проанализированы¹ современные технологии распределённого кэширования данных на базе СПО в высоконагруженных системах на платформе Node.js. Подробно рассмотрены принципы построения

¹Работа выполнена при финансовой поддержке РФФИ, проект № 17-07-01006 А

кэш-таблиц, особенности их реализации для различных технологий (memcached, Redis, Cassandra). Даны рекомендации по изучению и использованию на практике систем кэширования в зависимости от задач и условий применения.

Востребованность молодых специалистов определяется не только их теоретическими знаниями, но и практическими навыками. Это достигается только в том случае, если программа подготовки студентов опирается в том числе и на изучение современных технологий. Поскольку сфера информационных технологий является одной из наиболее динамично развивающихся, то необходимо постоянно совершенствовать обучение студентов по направлениям подготовки «Информационные системы и технологии» и «Прикладная информатика».

Одной из наиболее сложных практических задач является разработка и реализация высоконагруженных приложений. В современных условиях всё большее число пользователей используют в своей повседневной практике огромное число on-line приложений, общаются в чатах, читают новостные ленты, запрашивают актуальную информацию с различных сайтов. В этом случае наиболее важным для пользователя является такой показатель как время отклика. Для его уменьшения задействуется такой механизм как кэширование данных.

Кэширование данных — неотъемлемая часть процесса обработки данных. Его основная задача — обеспечить максимальную скорость доступа к требуемой информации, как для приложений, которые выполняются в режиме реального времени, так и для приложений, требующих значительных вычислений (например, при работе с базами данных). Использование кэширования во многих случаях позволяет значительно сократить время отклика. Связано это с тем, что при работе с внешними ресурсами (базами данных, облачными хранилищами и пр.) есть определённый набор данных, обращение к которым происходит значительно чаще, чем к другим (например, при генерации web-страниц). Такие данные помещаются в кэш. Поскольку хранилище кэшированных данных обладает значительно лучшими характеристиками доступа к информации, нежели внешние ресурсы, то, если искомая информация содержится в кэше, скорость доступа к ней многократно возрастает. Конечно, остаются специфические запросы, время обработки которых является значительным, а также проблемы, связанные с небольшим объёмом кэша, но в среднем время отклика значительно уменьшается.

В последнее время одним из наиболее популярных инструментов разработки серверной части высоконагруженных проектов является платформа Node.js — программное обеспечение с открытым кодом [1]. Среди средств кэширования для Node.js на стороне сервера, которые успешно зарекомендовали себя в различных проектах (особенно в web-проектах), можно отметить такие, как: memcached, Redis, Cassandra. Для этих средств кэширования клиентские библиотеки доступны для значительного количества языков и сред: C/C++, Ruby, PHP, Python, Java, Node.js и др.

Однако наиболее популярными являются средства кэширования memcached и Redis, ввиду простоты их настроек и использования. В основе кэширования лежит принцип построения хранилища данных типа «ключ — значение». Оба средства являются свободным программным обеспечением (СПО) с открытым кодом (memcached распространяется под лицензией Revised BSD License, Redis — REmote DIctionary Server — под лицензией BSD).

Хронологически первой была разработана memcached — высокопроизводительная система кэширования с распределённой памятью, позволяющая кэшировать данные в оперативной памяти множества доступных серверов [2]. Система имеет архитектуру клиент-сервер. Серверы поддерживают ассоциативный массив значений ключа, единственный возможный тип данных memcached — строки.

В общем случае работа происходит следующим образом. Клиент при помощи библиотеки обращается с запросом некоторых данных. Приложение сначала обращается к серверу memcached (get-запрос). Если ключ обнаружен, то данные возвращаются клиенту. Если нет, то приложение обращается к внешним ресурсам, полученный результат возвращается клиенту и записывается в memcached (set-запрос). При этом для ключа задаётся определённое время жизни.

Memcached позволяет размещать кэш на различных серверах. В этом случае ключи распределяются по серверам в соответствии с определённой функцией, обеспечивающей наиболее равномерную их загрузку. Кроме того, такое распределение данных по серверам позволяет повысить надёжность хранения информации.

В отличие от memcached Redis не является обычным хранилищем ключей, он фактически является сервером структур данных, поддерживающим разные типы значений, такие как: строки, списки, множества и упорядоченные множества, хеши и даже вероятностные струк-

туры данных, которые используются для оценки мощности множества [3].

Также в Redis имеются различные механизмы вытеснения ключей за счёт более сложной организации управления памятью. Также необходимо отметить возможности репликации Redis, что позволяет противостоять сбоям и отказам.

Таким образом, memcached предпочтительнее при кэшировании относительно небольших и статических данных, таких как фрагменты кода HTML, поскольку речь идёт о хранении строк. В этом случае данные не требуют обработки и подлежат только чтению, а значит использование memcached эффективно, поскольку не требует значительных ресурсов.

В остальных случаях кэширования таких объектов как: изображения, файлы, метаданные, Redis имеет преимущество за счёт более широких возможностей работы с различными типами данных.

Для использования данных систем кэширования под Node.js имеется значительно количество пользовательских библиотек (mc, node-memcached, memjs, redis и пр.), установка которых производится стандартным образом через npm (Node.js Package Manager — менеджер пакетов Node).

Что касается Cassandra, данная распределённая система управления базами данных также может быть включена в программу обучения студентов с одновременным изучением языка SQL [4].

Таким образом сложно переоценить необходимость освоения студентами практических навыков использования кэширования для уменьшения времени отклика высоконагруженных систем.

Литература

- [1] Nodejs [Электронный ресурс] / Режим доступа: <https://nodejs.org/en/>, свободный. (Дата обращения: 16.12.2018 г.).
- [2] Memcached [Электронный ресурс] / Режим доступа: <http://memcached.org/>, свободный. — Дата обращения: 16.12.2018 г..
- [3] Redis [Электронный ресурс] / Режим доступа: <http://redis.io>, свободный. — Дата обращения: 16.12.2018 г.).
- [4] Apache Cassandra [Электронный ресурс] / Режим доступа: <http://cassandra.apache.org/>, свободный. — Дата обращения: 16.12.2018 г..

Крюков Александр

г. Рязань, Рязанское высшее воздушно-десантное командное училище

Выбор ПО для учебного процесса

Аннотация

Разнообразие учебных задач и требования безопасности накладывают ограничения на выбор ПО

В учебном процессе кафедры РРТСиПС используются РС старше 10 лет, унаследованное программное обеспечение: Assistant [1], MicroCAP [2], MManaGAL [3], Packet Tracer [4] и ПО, не доступное обучающимся без преподавателя, например Assistant Builder [1]. Поддержка ряда x32 программ закончилась, а версии x64 на x32 РС класса не устанавливаются. В компьютерном классе проводятся все виды занятий, а также самостоятельная подготовка. Обучающиеся пользуются личными гаджетами, что приводит к заражению носителей свежими вирусами. Обновления антивирусных баз запаздывают от недели до месяца. Поскольку ПО от Microsoft устанавливается и привязывается к номеру процессора, поддержание его в работоспособном состоянии требует регулярной работы техника. Блокировка загрузки внешних носителей (например, Zlock [5]) увеличивает время подготовки класса к занятию, а малая производительность сервера не даёт сделать это дистанционно. Задача выбора ПО для учебного процесса актуальна. Для решения задачи с использованием репозитория операционных систем АЛБТ Рабочая станция 8 [6], Ubuntu 14.4 [7], MX-17 [8] были развёрнуты 3 рабочие станции, включающие все необходимые программы (требующие Windows устанавливались на wine) и зашифрованный раздел, содержащий ПО, недоступное обучающимся без преподавателя. Clonezilla из состава Hiren's BootCD [9] были созданы образы ПО этих рабочих станций для ускорения последующего развёртывания и сохранены на внешний USB-диск. Эти образы поочередно разворачивались на экспериментальный РС, наиболее близкий по составу к РС класса, производилось обновление ПО и сравнивались результаты работы, удобство для обучающегося и быстрота подготовки РС к работе. Экспериментом установлено:

1. АЛБТ Рабочая станция 8.1 с зашифрованным разделом во время загрузки требует ввода пароля root, иначе дальнейшая загрузка невозможна. После выставления в BIOSe даты за 2014 год и ввода

пароля машина загрузилась, но изменила дату на 2 марта 2017 года. Все программы работают, в локальную сеть выйти можно, файлы на сервере читаются. В Интернет выйти нельзя из-за разности дат.

После установки в BIOSе текущей даты машина вышла в Интернет, нашла обновления, обновилась, программы заработали все, кроме той, срок действия которой истёк. При завершении сеанса пользователя с правами root вышла в "чёрный экран после перезагрузки выдала: error: symbol 'grub_term_highlight_color' not found.

АЛБТ Рабочая станция 8 для физлиц бесплатна, для юрлиц 4000 р в softline [10] в Рязани.

Выводы: этот образ ставить на PC класса нельзя, так как техник не успеет подготовить за перерыв класс к занятию, а если обучающиеся подсмотрят пароль root — придётся его менять на всех машинах, а на ряде — переустанавливать ОСь.

2. Ubuntu 14.4 загрузилась с автоматическим вводом пароля пользователя, не допущенного к зашифрованному разделу. После установки в BIOSе даты за 2014 год машина запустилась с этой датой, все программы сработали. Вводом пароля другого пользователя можно открыть доступ к зашифрованному разделу и установленному в нём ПО. При подключении к сети файлы открылись. При подключении к сети Интернет автоматически исправила дату и время на текущие, в том числе в BIOSе, увидела обновления и обновилась, после перезагрузки загрузилась с автоматическим вводом пароля пользователя. Выводы: хороший образ для учебных целей.

3. MX-17 x64 загрузилась с автоматическим вводом пароля пользователя, не допущенного к зашифрованному разделу. Сменить дату в BIOSе можно, программы работают, файлы в сети видны, обновление прошло гладко. Выйти из системы и сменить пароль пользователя на другого можно, всё проходит достаточно быстро для того, чтобы за перерыв поменять пароль. На старом «железе» ОСь и программы «тормозят», Firefox с настройками по умолчанию в сети Интернет занял 18% процессора и 31% памяти. САД-система грузится 3 минуты, загрузка процессора достигает 85%, а памяти — 75%, подкачка при RAM 2 Гб используется на 2%. Выводы: образ можно использовать в учебном процессе после обновления PC класса.

Литература

[1] Assistent. Электронный ресурс <http://asksystem.narod.ru>.

-
- [2] ММНА. Электронный ресурс <http://www.radio.ru/mmana/>
 - [3] Программа моделирования Micro-Cap Электронный ресурс <http://www.spectrum-soft.com/demoform.shtm>
 - [4] Cisco Packet Tracer. Электронный ресурс <https://www.netacad.com/courses/packet-tracer>
 - [5] Zecurion Device Control (Zlock). Электронный ресурс <http://www.zecurion.ru/products/zlock/>
 - [6] Альт Рабочая станция 8. Электронный ресурс <https://www.basealt.ru/products/alt-workstation/>
 - [7] Ubuntu 14.04.5 LTS. Электронный ресурс <http://releases.ubuntu.com/14.04/>
 - [8] MX-17. Электронный ресурс <https://mxlinux.org/version/mx-17>
 - [9] Hiren's BootCD. Электронный ресурс <https://www.hirensbootcd.org/>
 - [10] Интернет-магазин Softline. Электронный ресурс <https://softline.ru/>