

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

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

Переславль-Залесский, 7–9 февраля 2020 года

Сборник тезисов конференции

Москва,
МАКС Пресс,
2020

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

Пятнадцатая конференция «Свободное программное обеспечение в высшей школе»: материалы конференции / Переславль-Залесский, 7–9 февраля 2020 г. / отв. ред. Черный В. Л. — М. : МАКС Пресс, 2020. — 180 с.

ISBN 978-5-317-06345-0

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 № 016

ISBN 978-5-317-06345-0

© Коллектив авторов, 2020

Программа конференции

7 февраля, 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 С. М. Абрамов

Учебный курс «Haskell, как первый язык
программирования» 9

14.40–15.10 В. А. Сухомлин

Концепция навыков информационного века и принципы ее
реализации в системе развития цифровых навыков
ВМК МГУ & Базальт СПО..... 12

15.10–15.30 И. В. Захаров

Подходы к созданию свободного репозитория метаданных
образовательных курсов. Перспективы сотрудничества
образовательных организаций и Базальт СПО. 15

15.30–16.00 Н. Н. Непейвода, И. Н. Григоревский

Нестандартные представления чисел..... 25

16.00–16.15 Р. Лашин

О деятельности Ассоциации АРПП по обеспечению
совместимости отечественных решений и образованию. 31

16.15–16.40 Кофе-пауза

16.40–17.00	Г. В. Грибков	
	Применение статического анализа кода в преподавании и в разработке свободного ПО.....	34
17.00–17.20	Я. Б. Шпунт	
	Свободные наборы офисных приложений, пригодные для повседневного использования в российской образовательной практике. Обзор и сравнение.....	35
17.20–17.40	А. В. Федорчук	
	Linux для пенсионеров.....	40
17.40–18.00	А. С. Черепанов	
	Особенности дистрибутива Альт Образование 9.0.....	42
18.00–18.20	А. В. Бондарев	
	Embox: путь от студенческой забавы до проекта с открытым кодом.....	46
18.20–18.40	И. В. Воронин	
	Создание и использование в образовательном процессе видеороликов по робототехнике. Проект УМКИ.....	48
18.40–19.00	А. С. Проскурнев	
	СПО — инструмент для повышения компьютерной грамотности на уроках информатики.....	54

8 февраля, суббота

09.30 Автобус от гостиницы

Утреннее заседание

10.00–13.00

10.00–10.20	В. В. Лаптев	
	Slang IDE — среда для обучения начинающих программистов	56
10.20–10.40	В. В. Лаптев	
	Среда для проведения контрольных работ по программированию на базе СПО.....	61
10.40–11.00	А. В. Попцов	
	MST: Автоматизация настройки мультисит-системы.....	66

11.00–11.20	Д. А. Костюк, П. Н. Луцюк	
	Практическое изучение средств контейнерной виртуализации и платформы Kubernetes.....	69
11.20–11.40	М. А. Шигорин	
	Альт Образование и Эльбрус: задачи и решения.....	73
11.40–12.00	Кофе-пауза	
12.00–12.20	В. А. Кирсанов, Р. С. Лопухов	
	Апробация типового отечественного модуля изучения интернет-технологий.	76
12.20–12.40	В. С. Малиночкин	
	Лабораторный комплекс по изучению базовых сервисов сети Интернет с использованием свободного программного обеспечения	78
12.40–13.00	М. Р. Магафуров	
	Лабораторный комплекс для изучения протоколов динамической маршрутизации с использованием свободного программного обеспечения.....	80
13:00	Автобус в гостиницу	
13.00–15.00	Перерыв на обед	
14:40	Автобус от гостиницы	

**Вечернее заседание
15.00–19.30**

15.00–15.20	И. А. Хахаев	
	Снова об оценке электронных образовательных ресурсов...	82
15.20–15.40	Р. Г. Ставцев	
	Отечественные аппаратные платформы «Байкал» для СПО	86
15.40–16.00	Д. В. Диденко	
	Использование свободного программного обеспечения на уроках по созданию web-документов в колледже (на основе опыта ГБПОУ МО «Щёлковский колледж» СП8)	86

16.00–16.20	А. Ю. Федосов, М. В. Маркушевич, А. Н. Краснов	
	О концепции комплекта учебников информатики для общеобразовательной школы, базирующегося на свободном программном обеспечении.....	90
16.20–16.40	И. В. Шишунов	
	Использование свободного программного обеспечения в общеобразовательной организации на базе центра формирования цифровых и гуманитарных компетенций «Точка роста»	94
16.40–17.00	А. Г. Михеев, А. М. Андрианов	
	Привлечение студентов к разработке свободной системы RunaWFE Free в рамках проектного командного обучения	99
17.00–17.20	В. Л. Симонов, А. В. Ерпелев	
	Разработка прототипа электронного прибора «Гомеостат» на базе СПО для использования в учебном процессе...	105
17.20–17.40	В. Л. Симонов	
	Разработка прототипа робота-помощника для лиц с ограниченными возможностями здоровья на базе промышленного робота-манипулятора КУКА с электрическим захватом под управлением платформы Arduino и СПО	109
17.40–18.00	В. Л. Симонов	
	Программно-аппаратный комплекс на базе СПО для оценки реакции человека в условиях повышенных нагрузок.....	112
18.00–18.20	Д. Л. Климов, А. Н. Гусин	
	Методические рекомендации по организации рабочих мест для студентов и сотрудников вуза на базе Alt Linux 9.0	115
18.20–18.40	А. В. Иванов	
	Опыт внедрения и использования ОС Альт в образовательном процессе	118

18.40–19.00	С. А. Фомин	
	Udaff — русский пиктографический Python. От элементарных алгоритмов до гомоморфного шифрования	121
19.00–19.20	Д. А. Слинкин	
	Проект RUBIROBOT: управление роботами Lego Mindstorms EV3 с использованием языка программирования Free Pascal	127
19.30	Автобус в гостиницу	

9 февраля, воскресенье

09.30 Автобус от гостиницы

Утреннее заседание

10.00–14.00

10.00–10.20	А. Н. Пустыгин	
	О разработке стандартов анализа исходных текстов программного обеспечения на материалах свободного ПО	131
10.20–10.40	В. Н. Лукин, Л. Н. Чернышов	
	Система онлайн-тестирования с открытым кодом	136
10.40–11.00	С. Г. Бессонов, А. В. Попцов	
	Азбук: как решение для образования оказалось востребовано и в коммерческих компаниях	139
11.00–11.20	К. С. Николаев, А. В. Суворов	
	Разработка программного средства отслеживания субъекта при динамических расстройствах организма .	142
11.20–11.40	С. Г. Зверева	
	Органайзинг в сфере СПО. Привлечение студентов к разработке и использованию СПО	145
11.40–12.00	Кофе-пауза	
12.00–12.20	И. И. Чудов, Е. А. Синельников	
	Приобщение к участию в разработке свободных программ на примере стажировки в компании «Базальт СПО» ..	147

12.20–12.40	В. Л. Симонов, А. В. Ерпелев	
	Использование СПО в учебном процессе на примере разработки портативной метеостанции на Raspberry Pi	151
12.40–13.00	Е. Р. Алексеев, Р. А. Бондаренко	
	Использование программы Maxima при обучении студентов педагогического направления в Кубанском государственном университете	154
13.00–13.20	Е. Р. Алексеев	
	Свободная система математического моделирования Simfor на базе компилятора gfortran (об одном свободном студенческом проекте)	156
13.20–13.40	А. А. Маркина	
	О модернизации подхода к преподаванию основ алгоритмизации и программирования	159
13.40–14.00	Т. О. Сундукова, Г. В. Ваныкина	
	Использование свободного программного обеспечения для учебных курсов в контексте федерального проекта «Кадры для цифровой экономики»	163
14.30	Автобус в Москву	

Вне программы

Е. Лавренова, Л. Павликова, М. Бочаров		
	Организация проектно-практической деятельности студентов вуза по участию в процессе внедрения свободного программного обеспечения в общеобразовательные учреждения	167
Сергей Мартишин, Марина Храпченко		
	Организация облачных вычислений над конфиденциальными данными на СПО	171
Н. А. Зацепин		
	Генератор тестовых заданий для системы Moodle	174
В. Романов, О. Солнцева, А. Трескова		
	Опыт перехода на альтернативное программное обеспечение в учебном процессе УлГАУ	177

Сергей Абрамов

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

Учебный курс «Haskell как первый язык программирования»

В 2005 году в Университете города Переславля имени А. К. Айламазяна был создан учебный курс «Haskell как первый язык программирования». Курс задумывался для чтения первокурсникам, не имеющим устоявшихся навыков в программировании. В течение почти 15 лет курс читался в УГП имени А. К. Айламазяна, ЯрГУ имени П. Г. Демидова, в средней школе 2101 г. Москвы, филиале МГУ имени М. В. Ломоносова в г. Севастополе. Только в двух последних случаях удалось прочитать курс действительно «новобранцам» в программировании, и это был интересный преподавательский опыт.

Целью курса является знакомство студентов с функциональной парадигмой программирования и развития у них алгоритмических навыков. Использование языка Haskell позволяет студентам быстро выйти на написание достаточно серьезных задач в разных областях за счет следующих свойств языка: строгая статическая типизация, чистая функциональность, ленивая семантика, поддержка параметрического полиморфизма и функций высшего порядка, полное освобождение программиста от управления памятью (захват, освобождение, сборка мусора). Все это позволяет концентрироваться на алгоритмической сути задачи, описывать решение в декларативном стиле.

Базой для курса послужила книга [1]. Постепенно была создана оснастка для преподавателя и накоплены материалы для студентов [2], включающие: 370 слайдов, 120 вопросов для формирования билетов письменных контрольных работ, 112 заданий для самостоятельного написания студентами различных программ (от 1 до 20 строк) на языке Haskell.

Курс рассчитан на один семестр — минимум 30 пар (45 астрономических часов) лекций и очень большую самостоятельную работу студентов с консультационной поддержкой преподавателя, конечно.

Слайды для лекций содержат достаточно много включений работы студента у доски: излагается некая порция материала, ставится задача для программирования и звучит команда «Доброволец — к

доске». Один студент у доски, с обсуждением всей остальной аудитории, пишет код решения. После полировки кода всем залом студент садится, лектор переходит к следующему слайду, и все сравнивают то решение, что найдено студентами, с тем, что написано на слайде. В последние годы достаточно много таких фрагментов для работы у доски создано на базе математических и даже шахматных головоломок, заимствованных из Facebook-ленты А. В. Шкрета.

Конечно, лекции и даже работа у доски не могут дать студенту глубокое понимание функциональной парадигмы. А письменные контрольные мероприятия не более чем повод освоить самые начала предмета. Основной образовательный эффект дают 112 задач для самостоятельного программирования.

Задачи имеют разную сложность. Начинаются с проблем, которые даже новичок решит в одну строчку. Десяток несложных однотипных задач вида «найти минимальный корень уравнения. . . » и «найти все (список) корней уравнения. . . » являются этюдом, прививающим навык выделения фрагментов со свойством «reusable code». Этот навык закрепляется затем во многих других задачах. К концу цикла студент решает вполне серьезные задачи. Например, разрабатывает полный комплект функций для работы с полиномами одной переменной (для двух разных представлений полиномов): упрощение полиномов, сложение, вычитание, возведение в степень, деление с остатком, дифференцирование, вычисление для заданного значения переменной.

Практическое программирование поддержано «сервером автопроверки» <http://haskell.pereslavl.ru/>. Написав в файле несколько решений заданий, студент может загрузить этот файл на сервер. Система для каждого решенного задания проверяет его на наборе тестов на некоторое соответствие получаемых результатов эталонным значениям. Если тест провален, то студент получает данные для отладки — сообщение: «на входных данных . . . ожидался результат . . . , а получен . . . ». Успешное прохождение сервера автопроверки всех заданий дает студенту допуск к следующему этапу: проверки кода преподавателем. Здесь речь идет о качестве стиля написания, чистоте и красоте кода — о тех вещах, которые сервер автопроверки проконтролировать не сможет.

В 2019 году был опыт преподавания курса в абсолютно дистанционном режиме — преподаватель ни разу не встречался со студентами филиала в г. Севастополе МГУ имени М.В. Ломоносова. Лекции читались с помощью Skype — в Севастополе к компьютеру со Скайпом

были подключены колонки и проектор, лектор использовал режим «демонстрация экрана» для показа слайдов и для показа процессов написания кусочков кода и запуска их в Hugs. Поддержка работы студента у доски во время лекции обеспечивалась еще одним компьютером в Севастополе: здесь «доброволец» писал Haskell-код в Google-документе, доступ к которому был и у преподавателя. За счет этого преподаватель на своем экране смотрел (и мог вмешаться) на процесс написания кода и (режим «демонстрации экрана»!) все студенты это тоже видели на экране проектора.

Планы и мечты по дальнейшему развитию курса связаны с тремя направлениями:

- развитие сервера автопроверки — здесь ясно, что надо доработать;
- расширение и улучшение набора задач — здесь было бы полезно взаимодействие с любыми заинтересованными партнерами;
- перевод курса в статус открытого проекта — здесь просто нужны партнеры, обладающие технологическими навыками.

Литература

- [1] Роганова Н. А. *Функциональное программирование: учебное пособие для студентов высших учебных заведений*. М.: ГИНФО, 2002 г. 260 с.
- [2] Абрамов С. М., Пармёнова Л. В., Юмагужин Н. В. *Материалы для студентов к курсу лекций «Haskell как первый язык программирования»* — УГП имени А. К. Айламазяна, Переславль-Залесский. Режим доступа: <https://goo.gl/r2IKNz>.

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

Концепция навыков информационного века и принципы ее реализации в системе развития цифровых навыков ВМК МГУ & Базальт СПО

В эру цифровой экономики наиострейшими проблемами становятся проблема подготовки кадров с необходимыми навыками и неразрывно связанная с ней проблема развития образовательных технологий, ориентированных на выявление спроса на востребованные общие и специализированные цифровые навыки и обучение этим навыкам [1]. Общеизвестно, что главным сдерживающим фактором широкомасштабной цифровизации в нашей стране является именно отсутствие подготовленных кадров [2].

Для системного решения задачи подготовки кадров с востребованными навыками важную роль играют стандарты в области классификации и описания цифровых навыков. Среди наиболее широко распространенных и перспективных решений в этой сфере — стандарты навыков информационной эпохи SFIA [3]. С помощью навыков системы SFIA, используемых в качестве строительных блоков, может быть описан широкий класс профессиональных ролей, связанных с областью ИКТ и цифровой трансформацией.

Перспективным практическим решением по организации подготовки кадров в режиме оперативного реагирования для решения задач цифровой экономики представляется создание специализированных платформ, осуществляющих поддержку бизнес-процессов, связанных с выявлением актуальных навыков и подготовкой соответствующих кадров посредством использования интегрированного потенциала как систем образования, так и организаций-работодателей.

Создание такой платформы — системы развития цифровых навыков (СРЦН) осуществляется в рамках совместного проекта факультета вычислительной математики и кибернетики МГУ имени М. В. Ломоносова и ООО «Базальт СПО» [4], который и предназначен для поддержки сотрудничества образовательных учреждений и организаций-работодателей в подготовке кадров.

Основными задачами СРЦН являются:

- селекция, отбор, спецификация востребованных в цифровой экономике цифровых навыков;
- формирование и сопровождение фонда цифровых навыков, благодаря возможности публиковать организациям-работодателям на платформе информацию о имеющихся вакансиях и о требованиях к соискателям на эти вакансии;
- тестирование/сертификация навыков кандидатов на вакансии с помощью тестовых заданий и кейсов, предоставляемых организациями-работодателями, а также проверки у кандидатов необходимых знаний с помощью профессорско-преподавательского состава вузов и использования соответствующих образовательных ресурсов;
- разработка образовательных программ и ресурсов, требуемых для развития и тестирования востребованных цифровых навыков;
- формирование и сопровождение фонда образовательных ресурсов (ФОР) для развития востребованных цифровых навыков;
- организация образовательных процессов, направленных на развитие востребованных цифровых навыков, в том числе с участием организаций-работодателей. Такое участие включает организацию производственных практик обучающихся, проведение сотрудниками организаций практико-ориентированных занятий и самостоятельных проектных работ, онлайн-стажировок и т. п.;
- поддержка сетевого взаимодействия пользователей платформы (представителей отраслевых организаций, образовательных учреждений, обучающихся, кандидатов на занятие вакантных должностей, работников сферы управления персоналом, и т. п.);
- мероприятия, способствующие решению задач кадрового менеджмента и планирования/организации подготовки профессиональных кадров, включая выявление талантов, специализацию обучающихся, управляемое трудоустройство обучающихся и выпускников вузов и др.

В докладе основное внимание уделено рассмотрению метамодели навыка рабочего места и разработанного на ее основе метода описания цифровых навыков, в котором для спецификации навыков используется типовая структура с фиксированным набором элементов и

заданной семантикой, названная SV-вектором (Skill/Vacancy-vector). SV-вектор сконструирован так, что включает в себя модель представления навыка в подходе SFIA, и, таким образом, позволяет использовать весь методический аппарат и справочник навыков SFIA в работе с навыками.

Над множеством SV-векторов введены: операция профилирования SV-векторов для конструирования нового SV-вектора (навыка) посредством агрегации спецификаций однородных элементов навыков-операндов; операция объединения SV-векторов в множество, которое представляется SV-вектором специального вида, называемым **метанавыком**. Используя метанавыки, можно формировать иерархические структуры навыков, определяющие системы навыков организаций, подразделений, проектов и т.п. Также в главе рассматриваются два способа конкретизации навыка дополнением его описанием контекста рабочего места. Еще одна особенность SV-нотации состоит в том, что в структуру SV-вектора введен элемент «Онтологические связи», предназначенный для хранения связей (отношений) между навыками фонда ФЦН и обработки их инструментами онтологических движков.

Литература

- [1] OECD Ministerial Declaration on the Digital Economy: Innovation, Growth and Social Prosperity («Cancun declaration»). Cancun–Mexico, June 21–23, 2016. Режим доступа: <http://www.oecd.org/sti/ieconomy/Digital-Economy-Ministerial-Declaration-2016.pdf>
- [2] Попов М. В., Сухорукова А. М. *Кадровый потенциал в реализации программы «Цифровая экономика Российской Федерации»* // Вестник Саратовского государственного социально-экономического университета. 2018. № 4 (73). С. 15–21. Режим доступа: <https://elibrary.ru/item.asp?id=36413119>
- [3] SFIA Foundation. Режим доступа: <https://www.sfia-online.org/en>
- [4] Сухомлин В. А., Зубарева Е. В., Якушин А. В. *Методические основы создания системы развития цифровых навыков на базе платформы Альт* // Свободное программное обеспечение в высшей школе. Сборник тезисов Четырнадцатой конференции / под ред. В. Л. Чёрного. М.: МАКС Пресс, 2019. С. 8–10. Режим доступа: <https://elibrary.ru/item.asp?id=37016833>

И.В. Захаров
Москва, ООО «Базальт СПО»

Подходы к созданию свободного репозитория метаданных образовательных курсов. Перспективы сотрудничества образовательных организаций и Базальт СПО

Аннотация

Доклад посвящен обсуждению и актуальным перспективам применения методики, предложенной в новой книге «Библиотеки Альт» авторов: В. А. Сухомлина, Е. В. Зубаревой, Д. Е. Намиота и А. В. Якушина «Система развития цифровых навыков ВМК МГУ & Базальт СПО. Методика классификации и описания требований к сотрудникам и содержанию образовательных программ в сфере информационных технологий» [1]. Также рассматривается несколько теоретических вопросов создания свободного репозитория метаданных образовательных курсов: возможный состав этих метаданных и необходимость разделения понятий «предметная область» и «проблемная область».

1. Описание проблемы взаимодействия человека, системы образования и работодателя: рассогласованность и углубление противоречий в понимании подходов к содержанию и направлениям развития образования.

Пример характерного суждения о состоянии и перспективах системы образования из статьи «Проблемные области системы образования в постиндустриальном обществе (по мнению учащейся молодежи)» [2]:

«Удовлетворение образовательных потребностей общества в целом и каждого обучаемого в частности — важнейшая задача сферы образования. Однако проблема постиндустриального периода заключается в том, что эти потребности общества и индивида могут не совпадать. Так, например, сегодня производство не готово предоставить каждому трудящемуся рабочее место, требующее высшего образования. А в перспективе, в связи с ростом таких мест, тем более. По мнению исследователей, «парадокс заключается в том, что выпускники ВУЗов в большинстве своем не хотят идти работать по рабочим профессиям — будет увеличиваться армия безработных, причем в той ее

части, которую наиболее сложно трудоустроить — это лица с высшим образованием, особенно с высшим гуманитарным образованием».

...

Как утверждают специалисты, «наука как бы переключилась больше на непосредственное обслуживание практики, на знание „ситуативное“, т.е. научное знание сегодня сосуществует с другими, ненаучными знаниями». Некоторые авторы предлагают разделять эти понятия на знание (научное знание) и информацию.

«... в системе образования в строящемся постиндустриальном обществе уже наметились проблемные области, требующие принятия решений, как на общественном, так и на личностном уровне. Новое образование должно научить человека дифференцировать информацию, оценивать ее, преобразовывать согласно требованиям общества. Для образования важной задачей должно стать повышение способности адаптироваться к быстрым переменам и реальным изменениям».

Обращает на себя внимание привычное уже постулирование спорных понятий и тезисов:

- «постиндустриальное общество»;
- сокращение рабочих мест, требующих высшего образования;
- разделение понятий «знание (научное знание)» и «информация» — произвольная трактовка, не учитывающая, например, современное состояние теории познания.

При этом формулируются очень значимые эмпирические факты, из которых не делается конкретных выводов, например: «потребности общества и индивида могут не совпадать».

Кроме перечисленных тезисов статья, выбранная для цитирования, обращает на себя внимание тем, что в её названии содержится понятие «проблемная область». На этот факт будет ссылка далее по тексту.

2. Методика организации взаимодействия обучаемых, работодателей и образовательных организаций, предлагаемая ВМК МГУ и Базальт СПО: система развития цифровых навыков (СРЦН).

Книга «Система развития цифровых навыков ВМК МГУ & Базальт СПО. Методика классификации и описания требований к сотрудникам и содержанию образовательных программ в сфере информационных технологий» [1] в качестве основы классификации предлагает онтологию понятий SFIA.

Основным содержанием предлагаемой методики является расширение классификации SFIA. Описание «ИТ-навыков» посредством онтологии SFIA и её расширения сведены в набор параметров, названный авторами «SV-вектор». (Подробному содержанию предлагаемого подхода посвящён доклад В. А. Сухомлина.)

Особой ценностью разработанной методики является обоснованность и хорошая проработка предложенного расширения, а также её универсальность, позволяющая решать, наверное, абсолютное большинство задач по определению требований к работникам и содержанию курсов, а также обеспечению их взаимосвязи.

Ещё одной важнейшей особенностью разработанной методики является учёт и её согласованность со всеми основными существующими документами и трендами в сфере классификации и определения требований к содержанию ИТ-образования, в том числе с ФГОС. Методика не противоречит им и может бесконфликтно использоваться в существующих условиях.

3. Теоретические вопросы построения обсуждаемой методики, определяющие подходы к её применению, развитию и построению свободного репозитория метаданных образовательных курсов.

3.1 Среди значимого достоинства методики хотелось бы отметить её обобщение для описания не только так называемых «навыков» и «вакансий», но и системы навыков организаций, подразделений, проектов и т.п.

Это редкий пример непосредственной корреляции прикладной методики с современной теорией познания (эпистемологией), в рамках которой можно считать доказанной необходимость такого специального направления организационных усилий как формирование коллективного субъекта научной (в т. ч. технической) познавательной деятельности. Современный актуальный «тип научной рациональности... учитывает соотносённость получаемых знаний об объекте не только с особенностью средств и операций деятельности, но и с ценностно-целевыми структурами. Причем эксплицируется связь внутринаучных целей с венаучными, социальными ценностями и целями»[3] — т. е., с некоторым упрощением, можно сказать — с «контекстом рабочего места» в терминах предлагаемой методики.

3.2 Понятия «предметная область» и «проблемная область», их роль при построении репозитория метаданных.

Различение понятий «предметная область» и «проблемная область» является важным аспектом современного конструктивизма.

«Наряду с дисциплинарными исследованиями на передний план все более выдвигаются междисциплинарные и проблемно-ориентированные формы исследовательской деятельности. Если классическая наука была ориентирована на постижение все более сужающегося, изолированного фрагмента действительности, выступавшего в качестве предмета той или иной научной дисциплины, то специфику современной науки конца XX века определяют комплексные исследовательские программы, в которых принимают участие специалисты различных областей знания. Организация таких исследований во многом зависит от определения приоритетных направлений, их финансирования, подготовки кадров и др. В самом же процессе определения научно-исследовательских приоритетов наряду с собственно познавательными целями все большую роль начинают играть цели экономического и социально-политического характера.

... ..

Можно констатировать, что современный синтез достижений различных наук протекает в условиях, когда все большую роль в научном познании начинают играть крупные комплексные программы и проблемно-ориентированные междисциплинарные исследования.

Еще В. И. Вернадский, анализируя тенденции развития науки в первой половине XX столетия, отмечал, что их классификация осуществляется уже не столько по предметам, сколько по проблемам» [3].

Подборка нескольких десятков статей из базы данных РИНЦ (elibrary.ru), в названии которых содержится понятие «проблемная область», позволяет утверждать, что оно в настоящее время употребляется всего лишь как синоним понятия «проблема» или «описание проблемы», а также синоним понятия «предметная область». (Типичные статьи, где эти понятия смешиваются, а их онтологическое значение игнорируется: [4], [5]. Также см. [1]) Во всяком случае, статьи, где понятия «предметная область» и «проблемная область» как-то явно различаются не найдены.

Применительно к обсуждаемой методике важно отметить, что она, фактически, посвящена описанию и классификации именно «проблемных областей».

Для описания «предметных областей» обсуждаемая методика позволяет опираться на различные системы классификаций, например РФФИ, с учётом существующих куррикулумов и так называемых «сводов профессиональных знаний» (СПЗ — Body of Knowledge — ВОК).

В SV-векторе предусмотрены поля для взаимосвязи с «предметными областями»: прежде всего – это поле «Базовые навыки (знания)», но ещё недостаточно проработан вопрос, как оптимально должны быть описаны «предметные области» и в какой форме ссылки на них лучше использовать в SV-векторе. Это вопрос, в том числе, и о взаимосвязи разных образовательных программ, курсов, а также разработке необходимых последовательностей, траекторий, их изучения. Эту проблему можно интерпретировать как проблему взаимосвязи программ дополнительного и основного профессионального образования.

Для решения этого вопроса предлагаю отметить несколько принципиально важных отличий между описаниями «предметных областей» и «проблемных областей». Кратко они сведены в таблицу 1.

Современное пространство «проблемных областей» можно считать бесконечным множеством.

При этом очевидно, что пространство «предметных областей» конечно. Так, текущая версия Универсальной десятичной классификации (УДК) содержит менее 127 тыс. индексов (их количество может быть кратно увеличено посредством расширений, однако речь идёт во многих случаях о предельной детализации, например, до химических формул, что для обсуждаемых целей не требуется).

«Предметные области» в существенно большей степени исторически обоснованы. Формулировка названий и описание «проблемных областей» допускает больше вариаций, практически ничем не ограничена.

Онтологии «предметных областей» представляют собой в основном партономии и таксономии классов. Онтологии «проблемных областей» могут включать все виды связей и определяют отношения самых разных сущностей: классов, экземпляров классов (объектов), ролей и т.д.

3.3 Из-за разнообразия онтологий, особенно «проблемных областей», особую роль приобретает накопленный опыт построения онтологий и распространённые ошибки. Подробно эти вопросы рассмотрены в главах 6-8 книги Н.В. Лукашевич «Тезаурусы в задачах информационного поиска» [6].

Для иллюстрации отметим лишь несколько важных аспектов.

3.3.1 Существует три основных подхода к представлению ролей в компьютерных ресурсах.

Параметр сравнения	«Предметная область»	«Проблемная область»
Количество элементов	В античности: космос, фюсис, тэхе и несколько видовых понятий. В настоящее время: не более 200 тыс. родовых и видовых понятий.	В античности: сотни, возможно тысячи понятий. В настоящее время: бесконечное множество.
Происхождение	Исторически обусловлено текущим уровнем развития науки и техники	Исторически менее обусловлено, более обусловлено социальными условиями, текущими обстоятельствами и субъективным взглядом на проблемы
Характеристики онтологии	Иерархия классов на основе связей «род-вид» и «часть-целое»	Иерархии объектов, ролей и других сущностей на основе всего многообразия связей

Таблица 1: Сравнительная характеристика понятий «предметная область» и «проблемная область».

«Первый подход рассматривает роль только как метку, приписанную к участнику ситуации» [6]. То есть роль — атрибут объекта. Тогда «становится невозможным описать собственные свойства ролей» [6].

«Второй подход представляет роли и сущности отдельными понятийными единицами, однако комбинирует эти два типа понятий в рамках одной иерархии» [6]. Для такого подхода есть несколько формальных приёмов. Однако в любом случае возникают неоднозначности в типизировании связей, например, связи «род-вид».

«В третьем подходе предполагается, что иерархия ролей должна быть представлена отдельно от иерархии типов. Иерархия ролей подчиняется самому верхнему узлу иерархии. В таком представлении

понятия-роли описываются независимо от типов, но каждый пример роли существует только как пример типа, т.е. пример роли не может существовать независимо от примера типа, который может занимать эту роль» [6].

Существуют и более сложные приёмы. Однако проблема различения классов и ролей в построении онтологий существует всегда, и для выбора оптимального варианта их построений требуется тщательный анализ и обоснование выбора. Недостаточное внимание к этому вопросу приводит к непредсказуемым противоречиям при применении онтологий.

3.3.2 Характерные проблемы смешения типов связей в онтологиях (на основе [6]).

Смешение типов и ролей. Характерный пример: работодатель как человек и как организация.

Смешение отношений «род-вид» и «класс-экземпляр класса». Характерный пример: спаниель-собака и спаниель-порода собак.

Смешение отношений «род-вид» и «часть-целое». Характерный пример: река-водоём-водный объект-вода-вещество.

Современная теория построения онтологий позволяет обеспечить обоснованные понятные взаимосвязи между произвольными SV-векторами, а также между ними и описаниями «предметных областей». Соответственно, таким образом определяются взаимосвязи между различными учебными курсами.

4. Репозиторий метаданных как интерфейс между описаниями «предметных областей» и «проблемных областей». Предложения по возможному составу метаданных, обеспечивающих взаимосвязь «предметных областей» и «проблемных областей».

На рисунке 1 представлена схема, предложенная Г. Маргенау. Здесь С — теоретические конструкты; N — непосредственно данная в наблюдении и эксперименте изучаемая реальность; ---- — внутри-теоретические связи между конструктами; — — связи конструктов с эмпирическим уровнем (эмпирические связи).

Важен следующий пример, характеризующий роль, значение и последствия манипуляций с теоретическими схемами и ее элементами. Если из классической механики исключить «такой объект, как „материальная точка“, то механика будет разрушена, если же вместо абстрактного объекта „сила“ ввести новый фундаментальный объект, например „энергию“, то вместо ньютоновской механики можно получить другую теоретическую конструкцию — механику Гамильто-

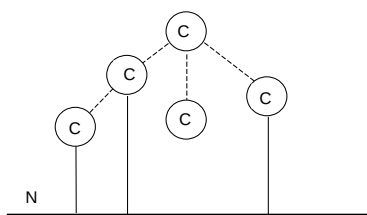


Рис. 1: схема Г. Маргенау.

на, а исключая „энергию“ и „силу“ из состава фундаментальных абстрактных объектов, можно прийти к основным принципам механики Г. Герца, которая также является иной, чем ньютоновская механика, теоретической конструкцией, описывающей механическое движение.

Таким образом, в основании сложившейся теории всегда можно обнаружить взаимосогласованную сеть абстрактных объектов, определяющую специфику данной теории. Эту сеть объектов мы будем называть *фундаментальной теоретической схемой* [3].

Схема Г. Маргенау достаточно наглядна и может служить ориентиром при создании теории, однако необходимо помнить, что в реальной «теории нет линейной цепочки абстрактных объектов, последовательно конструируемых один из другого ... Скорее, следует говорить о некоторых узловых системах таких объектов, вокруг которых формируются непосредственно относящиеся к ним „дочерние“ конструкторы. Своеобразным каркасом, сцепляющим все эти элементы в единую организацию, служат *фундаментальная теоретическая схема и частные теоретические схемы*, которые формируются на основе фундаментальной и вместе с ней включаются в состав научной теории» [3].

Математические модели являются желательной формой развития теоретической схемы. Благодаря им не только создается «богатство связей и отношений абстрактных объектов» и раскрывается «содержание теоретической схемы наиболее простым способом и в наиболее полной форме», но и **сами математические модели могут быть источниками иным способом не формируемых «абстрактных объектов теоретической схемы»** [3].

«Уравнения и абстрактные объекты теоретической схемы можно рассматривать как относительно самостоятельные компоненты теоретического знания. Такой подход оправдан, по меньшей мере, двумя обстоятельствами. Во-первых, одни и те же уравнения могут быть связаны с различными теоретическими схемами и, если последние обоснованы как отображение соответствующих фрагментов физической реальности, могут предстать как описание различных физических взаимодействий (хрестоматийными примерами здесь могут служить использование уравнений колебания для теоретического описания и механических, и электромагнитных колебаний, применение Максвеллом уравнений гидродинамики к описанию электромагнитных взаимодействий и т. д.). Во-вторых, теоретическая схема, если зафиксировать ее в языке содержательного описания, может существовать относительно независимо от уравнений. Так, описывая фундаментальную теоретическую схему механики (движение материальной точки в пространстве системы отсчета под действием силы), можно ввести абстрактную модель реальных механических движений, не прибегая к уравнениям. Опираясь на эту модель, можно получить и качественную характеристику законов механики (например, в „Математических началах натуральной философии“ Ньютона три основных закона механики излагались вначале без применения формул, в качественном виде)» [3].

Таким образом, основными метаданными, обеспечивающими связь между «предметными областями» и «проблемными областями», могут являться описания:

- самих теоретических схем;
- абстрактных объектов, имеющих наиболее важное прикладное значение.

Эти метаданные могут быть описаны с разной степенью глубины для разных уровней образовательных программ. Перечень этих метаданных существенно меньше, чем развёрнутая онтология современных «предметных областей».

Кроме того, возможно, при акцентировании внимания обучаемых на небольшом, как правило, количестве абстрактных объектов, имеющих наиболее важное прикладное значение, удастся решить актуальную проблему повышения заинтересованности обучаемых, о которой речь шла в процитированной в начале доклада статье [2].

Заключение

Краткие итоги двух лет сотрудничества образовательных организаций и Базальт СПО в рамках договора о сотрудничестве и договора о создании авторизованных учебных центров.

Перспективные направления сотрудничества, в том числе в связи с рассматриваемой методикой.

Литература

- [1] Сухомлин В. А., Зубарева Е. В., Намиот Д. Е., Якушин А. В. Система развития цифровых навыков ВМК МГУ & Базальт СПО. Методика классификации и описания требований к сотрудникам и содержанию образовательных программ в сфере информационных технологий./ответственный редактор: В.Л.Черный. М. Базальт СПО; МАКС Пресс, Библиотека АЛТ. 2020.
- [2] Гудина М. А., Никонова Э.И. Проблемные области системы образования в постиндустриальном обществе (по мнению учащейся молодежи) // Качество высшего и профессионального образования в постиндустриальную эпоху: сущность, обеспечение, проблемы. Материалы 10-й Международной научно-практической конференции: в 2х частях. 2016, С. 153–157.
- [3] Степин В. С. Теоретическое знание. М. Прогресс-Традиция, 2000. 744 с.
- [4] Барков И. А. Описание проблемной области в интеллектуальных информационных технологиях// Современные информационные технологии и письменное наследие: от древних рукописей к электронным текстам. Материалы международной научной конференции, Ижевск, 13–17 июля 2006. С. 10–21.
- [5] Сагайда П. И. Онтологическое моделирование проблемной области «интеллектуальный анализ данных для поддержки принятия решений»// Наукові праці Донецького національного технічного університету. Серія: Обчислювальна техніка та автоматизація. 2010. № 19 (171). С. 27–33.
- [6] Лукашевич Н. В. Тезаурусы в задачах информационного поиска. — М.: Издательство Московского государственного университета, 2011. — 512 с.

Григоревский И. Н., Непейвода Н. Н.
Переславль-Залесский, ИПС РАН

Нестандартные представления чисел

Аннотация

В связи с просьбой организаторов, мы сосредоточились на описании задач, требующих высокой точности, программ, работающих с точными вычислениями, и выделили среди программ свободные¹.

Достижения в области суперкомпьютерных информационно-вычислительных технологий позволяют ставить и решать ранее «невозможные задачи», которые крайне важны уже сегодня не только в научной сфере, но и на практике. Отмечается, что в современном мире невозможно победить, не победив в вычислениях. В частности, возникают задачи, в которых по самой сути необходимы точные вычисления с большими разрядностями операндов или же вычисления с очень большой гарантированной точностью. Они принадлежат большей частью к одному из двух классов: экспериментальная математика и компьютерная криптография, прежде всего, гомоморфная.

Второй класс задач, в которых требуются сверхточные и надёжные вычисления: практические и теоретические задачи, в которых существующие математические модели систем и алгоритмы численного моделирования оказываются почти неприемлемыми из-за недостатков в стандартном представлении чисел.

Задачи, в которых сверхточные вычисления нужны по их сути

1. прежде всего, это задачи экспериментальной математики, требующие для своего выполнения от нескольких десятков до нескольких тысяч и даже десятков тысяч десятичных цифр. Методологические и обзорные материалы по экспериментальной математике даны в серии работ Bailey D. H., Borwein J. M. ([1] и др.), где показано, что они приводят и к философским, в частности, онтологически значимым выводам;

¹Работа выполнялась при финансовой поддержке Российской Федерации в лице Минобрнауки России (идентификатор RFMEFI61319X0092)

2. область экспериментальной статистики распределения результатов точных математических задач;
3. проверка предположений, связанных с гипотезой Римана;
4. математическое моделирование, в частности, изучение неустойчивых моделей с трением и мелкой воды при помощи точных рациональных вычислений;
5. задачи целочисленной линейной оптимизации;
6. символьные и рациональные вычисления, в частности, задачи символьного интегрирования,
7. подсчёт числа гамильтоновых циклов на двумерных решетках, трёхмерных решётках, кубах, параллелепипедах и других геометрических объектах: практическое применение в физике полимеров.

В целом, за последние два десятилетия, этим вопросам посвящено более сотни разнообразных работ.

Для решения перечисленных и задач аналогичного типа необходимо повышение точности до 100–1000 десятичных цифр (приблизительно 512–4096 бит), что требует использования методов вычислений с применением «длинной арифметики».

Операции с многоразрядными числами являются базовым компонентом современных криптографических систем. В связи с этим в сентябре 2019 года прошла конференция IEEE по сверхточным вычислениям, чьи материалы не выложены в открытый доступ.

Задачи, в которых дефекты стандартных представлений приводят к переходу к сверхточным представлениям

Современные компьютеры и вычислительные системы позволяют сегодня решать достаточно сложные вычислительные задачи. Одной из актуальных проблем стала проблема контроля ошибок вычислений, особенно массово возникающих при выполнении вычислений над данными большого объема или длительных итерационных вычислениях, что характерно при решении современных прикладных задач.

Важной проблемой современных вычислительных технологий, в особенности при использовании высокопроизводительных (HPC) платформ, является и проблема «численной» воспроизводимости.

Под численной воспроизводимостью понимается гарантированное получение тождественных результатов при повторном решении прикладной задачи, причём с использованием одной и той же вычислительной платформы, или, более сильное условие, других платформ.

На самом деле, более важным и реалистичным был бы подход, связанный с получением результатов, совпадающих по модулю точности исходных данных и требуемой точности результатов.

В противном случае, возникает абсурдная, с точки зрения системного и логического анализа, ситуация: исходные данные, известные с точностью до 2–3 десятичных разрядов и результат, который требуется для оценки порядка практически нужной величины с точностью до 2 разрядов (а то и до одного), дополняются не имеющими никакого физического и математического обоснования разрядами, и затем эти разряды контролируются на воспроизводимость вычислений.

Стоит привести здесь некоторые из таких задач (список, разумеется, неполный, но достаточный для общего представления):

1. задачи волнового рассеяния из-за накопления неточностей и биения результатов сейчас считаются с использованием 32–64 десятичных цифр;
2. моделирование «поведения» сверхновых звезд и чёрных дыр — по тем же причинам нужна такая же точность;
3. моделирование метаболизма и конфигурации макромолекул приводит к той же ситуации, из-за чего создана система с открытым кодом SolveME;
4. в задачах оптимального управления иногда требуется 60–90 десятичных цифр; аналогичная ситуация возникает при вычислениях прямого и обратного преобразования Лапласа;
5. в астрономической одновременно теоретической и практической задаче многих тел требуется от 32 до 120 десятичных цифр;
6. матричные логарифмы, интегралы Изинга и Фейнмана, встречающиеся во многих практических и теоретических задачах, требуют от 100 до 1000 десятичных цифр;
7. изучение атомного уровня кулоновских систем требует использования более 100 десятичных цифр для получения корректных результатов;
8. ангармонические осцилляторы играют ключевую роль в исследованиях молекулярных колебаний, квантовых колебаний и по-

лупроводниковой технике; корректное нахождение собственных значений для ангармонических осцилляторов требует вычислений с 80 и более десятичными цифрами;

9. многочлены Пуассона.

В целом, проблемы с числами вызвали общую проблему проверки адекватности результатов вычислений.

В частности, министерство энергетики США назвало проблему высокоточных вычислений в числе девяти областей вычислительной математики, наряду с комбинаторными задачами, решением систем линейных алгебраических уравнений, приемлемое решение которых необходимо для вычислений эксамасштаба (ExaScale).

Работа со стандартными представлениями

Основным средством повышения точности вычисления сегодня является выполнение вычислений с применением методов и средств «длинной» арифметики с динамически изменяемым диапазоном представления целых и вещественных чисел. Частным случаем длинной арифметики является арифметика произвольной точности, в которой длина чисел ограничена только объемом доступной памяти. Арифметика произвольной точности находит широкое применение при решении вышеперечисленных и аналогичных по масштабу вычислений прикладных задач. Для таких вычислений применяются специализированные программные библиотеки и, в ряде случаев, специальное аппаратное обеспечение.

Всё множество предлагаемых библиотек для арифметики произвольной точности можно разделить на коммерческие (IMSL), бесплатные для некоммерческого использования (LiDIA, MIRACL) и открытые (GMP, NTL, CLN, MPI, Imath). Вторым важным моментом, определяющим область применения библиотек, является перечень поддерживаемых типов и структур данных. Это: целые числа произвольной длины (знаковые и беззнаковые); рациональные дроби; числа с плавающей точкой произвольной точности; комплексные числа; векторы; матрицы; полиномы. В частности, в задачах современной криптографии наиболее востребованными являются расчеты целочисленных значений произвольной длины и полиномиальная арифметика. Например, GMP — это открытая библиотека длинной арифметики,

поддерживающая работу со знаковыми целыми числами, рациональными числами и числами с плавающей запятой.

CLN — это библиотека для расчетов с использованием всех существующих числовых типов. В ней реализованы классы таких типов данных, как целые числа, рациональные дроби, числа с плавающей точкой, комплексные числа, инвариантные полиномы, вычисления по модулю. Библиотека имеет удобный интерфейс, поддерживает механизмы взаимодействия и преобразования друг в друга различных структур. CLN использует GMP в качестве вычислительного ядра. CLN лежит в основе многих программных продуктов, связанных с научными исследованиями и математическими расчетами: Scilab, Octave (свободные аналоги MatLAB), maxima (пакет для алгебраических вычислений, свободный аналог MAPLE) и др.

NTL — это высокопроизводительная библиотека C++, предоставляющая структуры данных и алгоритмы для работы с целыми числами произвольной длины, векторами, матрицами, полиномами, числами с плавающей точкой произвольной точности. Все алгоритмы NTL реализованы на C++ с открытым кодом, что обеспечивает кросс-платформенность и позволяет использовать библиотеку GMP.

В качестве примера можно перечислить также следующий набор приложений и прикладных областей, в которых используется длинная арифметика:

- пакет KDE Abacus поддерживает вычисления произвольной точности; библиотека APRON обеспечивает решение задач линейной алгебры и статического анализа данных;
- Arb-библиотека применяется для поддержки интервальной арифметики с произвольной точностью и с плавающей точкой;
- библиотека CGAL применяется для решения задач вычислительной геометрии;
- FLINT библиотека ориентирована на решение задач из области теории чисел.

Вычисления произвольной точности сегодня поддерживаются такими языками программирования, как Julia и C#.

Для других языков разработаны специальные библиотеки длинной арифметики: mpmath и SymPy для Python; GNU MP и GNU MPFR для C; BOOST, CLN и NTL для C++; ARPREC, MPFUN2015 и QD для языков C++ и Fortran.

Таблица 1: Таблица программ длинной арифметики, составленная в работе [2]

Пакет/название библиотеки	Поддерживаемые типы данных	Языки программирования	Лицензия
apfloat	Decimal floats, integers, rationals, and complex	Java and C++	LGPL and Freeware
ARPREC and MPFUN	Integers, binary floats, complex binary floats	C++ with C++ and Fortran bindings	BSD
Base One Number Class	Decimal floats	C++	Proprietary
Bbnum library	Integers and floats	Assembler and C++	New BSD
phseclib	Decimal floats	PHP	LGPL
BigDigits	Naturals	C	Freeware[1]
BigFloat	Binary Floats	C++	GPL
BigNum	Binary Integers, Floats (with math functions)	C#/.NET	Freeware
C++ Big Integer Library	Integers	C++	Public domain
CLN, a Class Library for Numbers	Integers, rationals, and complex	C and C++	GPL
Computable Real Numbers	Reals	Common Lisp	
IMSL		C	Commercial
DecNumber	Decimals	C	ICU licence (MIT licence) [2]
FMLIB	Floats	Fortran	
GNU Multi-Precision Library (and MPFR)	Integers, rationals and floats	C and C++ with bindings (GMPY...)	LGPL
GNU Multi-Precision Library for NET	Integers	C#/.NET	LGPL
Eiffel Arbitrary Precision Mathematics Library	Integers	Eiffel	LGPL
HugeCalc	Integers	C++ and Assembler	Proprietary
Imath	Integers and rationals	C	Freeware
IntX	Integers	C#/.NET	New BSD
Jscience LargeInteger	Integers	Java	
libmpdec	Decimals	C and C++	Simplified BSD
LibTomMath	Integers	C and C++	Public domain
LiDIA	Integers, floats, complex floats, and rationals	C and C++	Free for non-commercial use
MAPM	Integers and decimal floats	C (bindings for C++ and Lua)	Freeware
MIRACL	Integers and rationals	C and C++	Free for non-commercial use
MPI	Integers	C	LGPL
MPArith	Integers, floats, and rationals	Pascal/Delphi	zlib
mpmath	Floats, complex floats	Python	New BSD
NTL	Integers, floats	C and C++	GPL
BigInteger (and bigRational)	Integers and rationals	C and Seed7	LGPL
TTMath library	Integers and binary floats	Assembler and C++	New BSD
vecLib. framework	Integers	C	Proprietary
W3b. Sine	Decimal floats	C#/.NET	New BSD

Основные характеристики современных пакетов и библиотек, поддерживающих вычисления в длинной арифметике, приведены в таблице 1.

Литература

- [1] *Bailey D. H., Borwein J. M.* Experimental applied mathematics // In Higham, N. J. (Ed.), Princeton Companion for Applied Mathematics. Princeton University Press, 2015. 1016 p.
- [2] *Неласая А. В., Верещак М. И.* Оценка эффективности использования библиотек длинной арифметики в криптографических приложениях. Радиоэлектроника, информатика, управління. 2010. № 2. С. 67–73.

Р.Л. Лашин

Москва, Ассоциация Разработчиков Программных Продуктов
«Отечественный софт»

О деятельности АРПП по обеспечению совместимости отечественных решений и организации взаимодействия с образовательным сообществом

Ассоциация Разработчиков Программных Продуктов «Отечественный софт» учреждена в 2009 году крупнейшими российскими разработчиками тиражируемого программного обеспечения. В настоящее время в Ассоциацию входит более 170 российских ИТ-компаний (АВВУУ, Аскон, 1С, Лаборатория Касперского, Диасофт, InfoWatch, Docsvision, СКБ Контур, ЦФТ, Базальт СПО, Новые Облачные Технологии, ЦРТ, Галактика и др.).

Членами Правления Ассоциации являются: Наталья Касперская (председатель), Александр Голиков, Евгений Михалицын, Дмитрий Комиссаров, Алексей Смирнов, Андрей Ярных, Дмитрий Шушкин и Дмитрий Дырмовский¹.

Среди наиболее важных достижений Ассоциации стоит выделить работу над проектом федерального закона № 188-ФЗ от 29.06.2015 г., который утвердил разработанные Ассоциацией критерии отечественного ПО и обосновал создание Единого реестра российских программ для электронных вычислительных машин и баз данных².

¹<https://www.arppsoft.ru/association/board-of-directors/>

²<https://www.arppsoft.ru/10years/>

Приоритетными направлениями работы Ассоциации являются льготное налогообложение отрасли, импортозамещение и поддержка экспорта отечественного ПО, разработка совместных решений и продвижение их в федеральные органы законодательной и исполнительной власти, государственные корпорации.

В составе Ассоциации образованы и на регулярной основе действуют следующие комитеты:

- Комитет по интеграции российского ПО;
- Комитет по экспорту российского ПО;
- Комитет по информатизации образования;
- Комитет по развитию финансирования ИТ-отрасли;
- Комитет по развитию и продажам на российском коммерческом рынке;
- Правовой комитет.

Программные продукты подавляющего большинства ИТ-компаний входят в Единый реестр ПО Минкомсвязи, и многие из членов Ассоциации активно участвуют в процессе импортозамещения. Спектр компаний широк — тематически они закрывают практически весь Реестр по 24 классам программных продуктов. У членов Ассоциации имеется экспертиза по российским разработкам от системного программного обеспечения, систем серверной виртуализации, систем хранения данных, операционных систем, баз данных, систем информационной безопасности до различного прикладного ПО.

В рамках деятельности Комитета по интеграции российского ПО сформировано более 20 стеков совместимых отечественных решений, готовых для закупки госсектором и госкомпаниями: «Рабочее место госслужащего», «Рабочее место бухгалтера», «Оказание государственных услуг», «Ядро корпоративных коммуникаций», «Офисный пакет в частном облаке», «Обнаружение утечек информации», «Серверная ИТ-инфраструктура для федеральных и региональных ИС (ЗАГС)», «Интегрированная система управления предприятием», «Единая платформа для работы с документами», «Интегрированная защищенная система унифицированных коммуникаций, хранения и виртуализации», «Импортонезависимая полнофункциональная система электронного документооборота» и др. Одновременно силами членов Ассоциации создан и наполняется Каталог совместимости российского ПО, содержащий сведения из Единого реестра и позволяющий

формировать вариативные стеки совместимых отечественных решений членов Ассоциации. Данный каталог будет доступен для широкого использования уже в феврале этого года.

2 сентября 2019 года в 34 пилотных школах столицы начали свою работу ИТ-классы. Партнером данного проекта выступает АРПП «Отечественный софт». Учащимся предложены образовательные траектории по самым актуальным направлениям ИТ-сферы: Программирование, Робототехника, Моделирование и прототипирование, Информационная безопасность, Технологии связи и Большие данные.

Российские ИТ-компании, являясь будущими работодателями, заинтересованы в развитии у учеников востребованных на рынке труда компетенций в области ИТ. Для этого в рамках учебной программы при активном участии ИТ-компаний предусмотрены стажировки и хакатоны, разбор реальных кейсов, проведение мастер-классов и многое другое. Таким образом, создание ИТ-классов позволит московским школьникам освоить различные отечественные программные решения и практически применять знания по основам моделирования бизнес-процессов, архитектуре информационных систем, созданию искусственного интеллекта, средствам защиты информационных систем, методам криптографии и другим актуальным направлениям. В заключении хотелось бы отметить, что Ассоциация, являясь крупнейшей независимой коммуникационной площадкой, способствует выявлению проблем и формированию консолидированной позиции российских разработчиков тиражируемого ПО. Подготовленные предложения выносятся на рассмотрение в различные профильные экспертные группы Государственной Думы, Совета Федерации, Минкомсвязи России, Минпромторга России, АНО «Цифровая экономика», Центра компетенций по импортозамещению в сфере ИКТ и др.

Георгий Грибков

Тула, Тульский государственный университет

Применение статического анализа кода в преподавании и в разработке свободного ПО

Аннотация

В своём докладе я расскажу, как использование статического анализа может помочь преподавателям и студентам обучать и обучаться. Также я расскажу, как именно стоит наладить процесс анализа для использования в студенческих проектах разработки СПО, чтобы удовлетворить специфике работы в рамках учебного заведения.

Статический анализ — технология, в последнее время быстро набирающая популярность. Современные анализаторы представляют собой сложные программы, давно вышедшие за рамки `lint`'еров и регулярных выражений. В них используются такие технологии, как сопоставление с шаблоном, анализ потока данных, символьное выполнение и многие другие, поэтому статический анализ способен покрывать практически все области современных языков программирования.

Благодаря этому использование статического анализа является полезным инструментом для использования в учебном процессе. Например, преподаватель может проверять анализатором лабораторные и курсовые работы студентов, что позволяет легче проверить их на наличие ошибок. Студенты, которые только учатся программировать (и, что естественно, допускают большее количество ошибок), могут проверять анализатором свои работы, тем самым подробнее знакомясь с тонкими местами языка.

Преподаватель может дать своим студентам задание использовать статический анализатор при выполнении своей работы. Такие студенты будут уметь использовать статический анализ еще до окончания учебного заведения. Так как эта методология активно набирает популярность, сейчас всё чаще и чаще можно увидеть в предложениях о работе пункт «опыт работы с анализаторами кода». Следовательно, использование статического анализа при обучении может повысить стоимость выпускников на рынке труда.

Польза статического анализа для разработчиков открытого или свободного программного обеспечения соразмерна пользе для разработчиков коммерческих проектов. Программисты — люди, и им свойственно допускать ошибки, независимо от разрабатываемого проекта.

Тем не менее, разработка СПО имеет свою специфику. В своём докладе я также расскажу, как стоит использовать статические анализаторы кода при разработке свободного программного обеспечения.

Яков Шпунт

Москва, Comnews Group, журнал «Стандарт»

Свободные наборы офисных приложений, пригодные для повседневного использования в российской образовательной практике. Обзор и сравнение

Аннотация

OpenOffice.ORG и LibreOffice наиболее известны среди свободных офисных приложений. Однако их возможности избыточны для повседневных нужд большинства студентов, не говоря уже о школьниках. Тем более, что существуют более простые и удобные альтернативные варианты.

Пока наиболее распространенным из офисных приложений является Microsoft Office, но политика импортозамещения будет способствовать тому, что от ПО Microsoft будет отказываться все больше учреждений. Будет сказываться и то, что Office 365 громоздок и имеет довольно высокие системные требования, в итоге его функционирование на устаревших ПК начального уровня, которые преобладают в сфере образования, будет далеко от комфортного.

Другой крайностью является использование издательских систем вроде LaTeX. Однако работа в них ближе к программированию, что очень часто отвлекает от создания документов. Классические редакторы текстов имеют крайне ограниченные возможности по оформлению документов.

Тем не менее, у MS Office есть немало альтернатив. Среди них наиболее известны отечественный «Мой Офис» компании «Новые облачные технологии» и свободный LibreOffice, однако только ими перечень возможных альтернатив не исчерпывается.

LibreOffice и OpenOffice.ORG

Оба набора по факту являются форками одного проекта. LibreOffice, однако, развивается несколько более динамично и более функ-

ционален (в частности, намного лучше реализована поддержка макросов на VBA из MS Office), а также поддерживает разные виды пользовательского интерфейса. Однако в целом состав и функциональные возможности близки.

Поддерживается широкий выбор ОС и платформ, включая Windows, Linux, macOS, Android (неофициально). У OpenOffice.ORG также имеется онлайн-версия. Системные требования близки к минимальным: возможна продуктивная работа на системе с одноядерным процессором и 1 Гбайт оперативной памяти. Для пользователей отечественных платформ на базе Linux, входящих в Реестр отечественного ПО, установка происходит уже на стадии инсталляции ОС. В Windows, macOS и Android набор можно найти в штатном Магазине приложений.

Содержит целый комплекс расширенных функций, включая редактор формул и модуль работы с индексами. Есть СУБД и довольно мощный редактор векторной графики. В целом данные продукты являются универсальными офисными редакторами, пригодными для любых задач.

Главный недостаток — проблемы при импорте документов в формате приложений MS Office. Особенно критичны они в Impress, который очень существенно искажает внешний вид слайдов PowerPoint.

GNOME Office

По факту данный набор является разрозненными программами, которые базируются на общем тулките GTK. Официальные версии приложений есть только для Linux, хотя раньше поддерживался весьма широкий спектр платформ, включая мобильные.

Главное достоинство ключевых приложений, за исключением Evolution, — низкие системные требования. AbiWord вполне работоспособен даже на компьютере со 128 Мбайт оперативной памяти. При этом функциональные возможности включают все необходимые функции. Возможно использование подключаемых модулей, применение которых расширяет штатную функциональность. Возможности Gnumeric и вовсе практически те же, что и у Calc или Excel.

Недостатки набора связаны с издержками слабой координации разработчиков. В итоге если Gnumeric или Evolution вполне функциональные и зрелые приложения, то AbiWord им серьезно уступает. Нет редактора презентаций.

Calligra Suite

Данный набор изначально назывался Koffice и создавался для среды KDE. Основу составляют текстовый процессор Words, электронная таблица Sheets, редактор презентаций Stages, а также векторный редактор Karbon. Раньше в состав входил также фоторедактор Krita, который в настоящее время развивается отдельно. СУБД Krita находится в стадии бета-тестирования.

Разработчики Calligra ориентировались не на продукты Microsoft или Google, как остальные, а на Apple iWork. В итоге Calligra довольно популярен у мигрантов с macOS. Есть неофициальные версии для Windows, в том числе планшетной, и macOS.

Продукты из состава набора изначально ориентированы на академическую среду и сферу образования. Используется единый интерфейс, и в целом уровень интеграции высочайший. Поддерживаются дополнения, применение которых позволяет реализовать отсутствующие функции. Возможен импорт и экспорт как документов в формате ODF, так и MS OpenXML.

Главный недостаток — крайне низкое качество файловых конвертеров в сторонних форматах.

OnlyOffice/P7 Офис

Продукт от российской компании «Новые коммуникационные технологии», с 2016 года распространяется по лицензии GNU GPL 3. Есть улучшенный вариант P7, входящий в Реестр российского ПО.

Изначально данный продукт создавался для нужд небольших компаний. При этом за образец были взяты Gapps. Продукт изначально был предназначен для развертывания онлайн, в том числе в корпоративной сети, и функция редакторов документов появилась не сразу.

Функциональность приложений достаточная, но при этом «лишних» функций, которые будут заведомо не востребованными, там нет. Но стоит иметь в виду, что не включен целый ряд возможностей, которые необходимы в образовательной и научной практике. Впрочем, продукт довольно активно развивается, и функциональность растет.

Главное достоинство — полноценная и при этом корректная поддержка как ODF, так и OpenXML форматов. Практически идеальное решение для тех, у кого большая библиотека унаследованных документов или кому необходимо обмениваться ими с внешним миром.

Недостаток — довольно высокие системные требования. Впрочем, работа с десктопным редактором будет вполне приемлемой даже на системе с Intel Atom 28xx, имеющей 2 Гбайта оперативной памяти.

Мой Офис

Данный продукт включен в обзор вне конкурса, поскольку не является свободным ПО. Тем не менее, он, во-первых, бесплатен для образовательных учреждений, и, во-вторых, именно его наиболее часто выбирают для замены приложений Microsoft при реализации программ импортозамещения в госучреждениях и госкомпаниях. Входит в Реестр российского ПО.

Текстовый и табличный процессоры имеют достаточную для работы с офисной документацией функциональность. За образец при этом был взят не MS Office, а Gapps, тем более, что значительная часть разработчиков участвовали в создании его «предка» QuickOffice. Возможно развертывание в локальной сети или частном облаке. Есть версии для Windows, Linux, Android, iOS, «Аврора», в ближайших планах — портирование на macOS.

Полноценное средство для редактирования презентаций пока находится в стадии разработки, в состав релизной версии входит только просмотрщик, созданный на базе LibreOffice Impress.

Чаще всего объектом критики является интерфейс приложений. Кроме того, Impress часто не вполне корректно отображает содержимое презентаций.

Softmaker FreeOffice

Данный набор также включен вне конкурса, поскольку не является свободным ПО. Однако он полностью бесплатен. Включается в состав некоторых дистрибутивов РОСА.

Главное достоинство — компактность. Версия для Linux или Windows занимает на диске не более 250 Мбайт, меньше, чем объем дистрибутива LibreOffice. Существует 4 вида сменных интерфейсов. Имеется официальная версия для Android, в том числе рассчитанная на экраны высокого и низкого разрешения.

Хорошая функциональность, уступает только Open/LibreOffice и превосходит все остальные. Например, текстовый процессор поддерживает работу с указателем.

Недостатки — невысокое качество системы проверки правописания, находили ошибки в словарях; проблемы с импортом и особенно экспортом документов ODF; проблемы с масштабированием на экранах высокого разрешения в Linux; сложности с установкой на дистрибутивы Linux, где используются менеджеры пакетов, отличные от RPM и DEB.

Выводы и рекомендации

Наиболее универсальным вариантом является LibreOffice. К тому же, скорее всего, нет необходимости загружать его и дополнительно устанавливать, он является частью базовой поставки ОС. Однако его применение целесообразно лишь тогда, когда востребованы его дополнительные функции (работа с индексами, формулами, сводными таблицами, базой библиографии), например, для магистрантов, аспирантов, сотрудников.

При большом количестве унаследованных документов в форматах MS Office оптимальным выбором будет использование OnlyOffice/P7. Кроме того, его можно развернуть в сети образовательного учреждения и использовать как облачное приложение. При таком сценарии нет необходимости устанавливать его на компьютеры учеников.

Мой Офис имеет смысл использовать в вузах, готовящих кадры для госсектора и госкомпаний. Данный продукт чаще всего выбирают в ходе программ по импортозамещению, и он может стать фактическим стандартом офисных приложений в российском госсекторе.

GNOME Office — оптимальный вариант для эксплуатации на устаревших ПК с недостаточными ресурсами.

Calligra Suite имеет целый ряд критичных недостатков, его применение может быть оправдано только в случае миграции с mac OS, что для наших условий не вполне типично.

Алексей Федорчук, Ольга Ломова

Москва, Сіпіа

<https://www.cinia.ru/>

Linux для пенсионеров

Аннотация

Проект посвящён созданию сборника материалов, которые должны служить руководством к действию для совсем начинающих пользователей, даже тех, кто не затронут тлетворным влиянием Windows. В том числе отошедших от активной деятельности до широкого распространения персоналок. Однако они, как правило, имеют большой жизненный опыт и вескую мотивацию для освоения РС. Что создаёт благоприятные условия для того, чтобы освоение это начиналось с Linux'a и СПО.

Практически все известные нам руководства для начинающих подразумевают некоторое их знакомство с компьютерами и софтом для них, хотя бы «выньдовым». И потому в них всегда имеется немало «выньдовых» аллюзий. Ибо, как говорил Вилли Старк, добро можно делать только из зла, потому что больше его просто не из чего делать.

Целевая же аудитория представляемого проекта — не просто те, кто впервые знакомится с операционной системой Linux, её дистрибутивами и прикладными программами, но, возможно, даже те, кто не имел ранее дела с настольными персоналками и работающими на них системами, ни в ипостаси «чёрного DOS'a», ни в виде «окон» разных версий и поколений.

И потому тут очень важна обратная связь между теми, кто сочиняет, и теми, для кого сочиняется. А это требует редкого, можно сказать, уникального, стечения обстоятельств. Иначе написанное будет казаться потенциальному читателю из ЦО занудной заумью. А сочинителя будет раздражать незнание читателем элементарных вещей — например, что такое ОС вообще и ОС Linux в частности. Вот с этого-то и начинается наш проект.

Останавливаться на деталях не будем. Ибо единственное различие здесь — операционная ли система Linux, или только ядро её, а система должна называться GNU/Linux. Авторы считают её правильной, однако вдаваться в дискуссию и обоснование своей точки зрения здесь не будем. Потому что целевую аудиторию наших заметок этот вопрос нисколько не волнует, по крайней мере поначалу. Так

что пока просто заметим: одной из ОС и является Linux, который начал разрабатываться в 1991 году Линусом Торвалдсом. История эта описана её автором в книге «Just for Fun», изданной на английском в 2001 году (русский перевод — 2002 год).

Книга Линуса и Дэвида написана очень простым языком, содержит минимум «умных» слов, которые к тому же объясняются более чем понятно. И потому вполне по силам даже тем, кто не имеет никакой спецподготовки.

Разумеется, Linux не ограничивается ядром. Очень важная для пользователя его часть — драйверы устройств (например, накопителей) и файловых систем. А также утилиты оформления, обеспечивающие работу с ними.

Большинство утилит оформления разрабатывалось в рамках проекта GNU, почему вся ОС часто называется GNU/Linux. Мы обещали не вдаваться в дискуссию по этому вопросу, но не можем не отметить пару моментов. Во-первых, ряд очень важных утилит оформления не имеет к проекту GNU ни малейшего отношения. А во-вторых, в принципе Linux может быть собран вообще без компонентов GNU'ого происхождения.

ОС Linux может комплектоваться очень по-разному. Варианты сборок этой системы называются дистрибутивами, и состав их определяется их сопровождающими (далее — майнтейнер, от англ. maintainer).

Разговор о дистрибутивах логично начать с определения — что такое дистрибутив Linux'а. Тут всё просто: это варианты комплектации ядра этой ОС прикладными программами, предназначенные для распространения среди конечных пользователей самого разного профиля. Важно понимать только, что дистрибутив — это не просто ядро с набором утилит и приложений, а системная целостность, объединяемая репозиторием.

В связи с разнообразием целевого назначения дистрибутивов существует огромное количество. Но для нас сейчас интересны только те, которые именуются десктопными. Они в обязательном порядке включают в себя как минимум одну рабочую среду, которая устанавливается по умолчанию.

Рабочие среды условно можно разделить на три группы по требовательности к ресурсам компьютера. «Тяжёлая», то есть наиболее ресурсоёмкая группа, включает в себя KDE и GNOME. В «промежу-

точную» группу входят Cinnamon, MATE, Xfce, Budgie, в «лёгкую», наименее требовательную — LXQt.

Изобилие десктопных дистрибутивов нуждается в каком-никаком упорядочивании. И некогда вопросам их классификации уделялось немало внимания. Однако нынче все они сохранили только исторический интерес.

Последующие части этого сочинения, посвящённые некоторым теоретическим и практическим вопросам, будут иллюстрироваться конкретными примерами. Мы будем подбирать их по возможности наиболее универсальными для большинства десктопных дистрибутивов. Однако это не всегда реально. И потому в качестве иллюстративного материала мы используем в основном дистрибутивы Linux Mint Cinnamon Edition и Alt Linux Starter Kit также со средой Cinnamon. И, конечно, не пройдём мы и мимо нашей системы Cuntu.

Андрей Черепанов

Москва, ООО «Базальт СПО»

Проект: [Sisyphus packages.altlinux.org](https://sisyphus.packages.altlinux.org)

Особенности дистрибутива Альт Образование 9.0

Аннотация

В докладе рассматриваются ключевые особенности, которые реализованы в дистрибутиве Альт Образование 9.0, предназначенном для образовательных учреждений.

25 октября 2019 года на базе Девятой платформы выпущен дистрибутив Альт Образование 9.0, предназначенный для образовательных учреждений.

Поддержка новых аппаратных архитектур

Выпуск Альт 9.0 интересен тем, что впервые планируются дистрибутивы на разных аппаратных архитектурах. Помимо образов под 32-битной и 64-битной архитектуры x86, Альт Образование 9.0 доступна и под архитектуру AArch64: в виде образов ISO (например, для процессора Kunpeng на серверах Taishan производства компании Huawei) и в виде архивов файловых систем для Raspberry Pi 3 и NVIDIA Jetson Nano, а также и под Эльбрус v3 и v4.

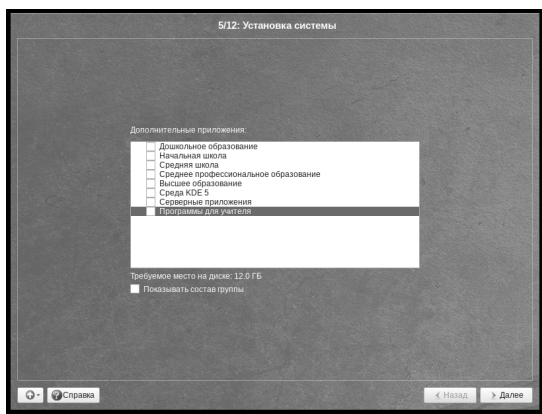
Использование доступных одноплатных компьютеров на архитектуре AArch64 может быть весьма интересно как с образовательной, так и с финансовой точки зрения.

Доступность версии под отечественную процессорную архитектуру Эльбрус позволяет уже сейчас полноценно использовать Эльбрусы в образовательном процессе.

Профили установки для различных уровней обучения

В дистрибутиве впервые появилась возможность выбора (в том числе нескольких пунктов) различных профилей уровня обучения.

Не выбирая ни одного пункта вы получите среду по умолчанию: полноценный графический стол на базе XFCE, офисный пакет LibreOffice и набор приложений, соответствующий Распоряжению Правительства РФ от 18 октября 2007 г. 1447-р., в том числе средства для редактирования растровых и векторных изображений, звука и видео, распознавания текста, а также клиента ITalc3 для управления классом.



Дополнительные профили включают в себя следующие компоненты:

Дошкольное образование

- Пиктомир

- Gcompris-qt
- Childsplay
- Tuxpaint
- KHangman
- Kanagram

Начальная школа

- Пиктомир
- Gcompris-qt
- Scratch
- trikStudio
- Kolourpaint
- Ktouch, Kbruch
- Abiword

Средняя школа

- Кумир 2, Scratch
- Lazarus и Code::Blocks
- Synfig Studio
- QCAD, FreeCAD, OpenSCAD
- wxMaxima и Scilab
- Kolourpaint
- trikStudio, Arduino

Среднее профессиональное образование

- Lazarus, Code::Blocks и Qt Creator
- QCAD, FreeCAD
- wxMaxima и Octave
- Kino
- Arduino
- Logisim и Fritzing

Высшее образование

- Lazarus, Code::Blocks и Qt Creator
- swi-prolog, Gambas и Monodevelop
- wxMaxima и Octave
- QCAD, FreeCAD, OpenSCAD
- Projectlibre
- QGis
- Fritzing

Среда KDE 5

Полноценная среда KDE5, включая полный набор образовательных приложений KDEEdu и KDE Connect для мобильных устройств.

Серверные приложения

На базе дистрибутива можно организовать прямо на компьютере учителя сервер для онлайн-совместной работы и обучения на базе веб-сервисов Nextcloud, Moodle, MediaWiki, а также терминальный сервер на базе XRDP и сервер коммуникаций на Openfire. При этом веб-сервисы автоматически разворачиваются при первом запуске, а смена пароля root в системе устанавливает его как пароль администратора этих сервисов.

Программы для учителя

Компоненты включают в себя управление классом ITalc3 и Semaphore (для управления через Ansible), а также сервер iTest для проведения тестирования.

Антон Бондарев

Санкт-Петербург

Проект: Embox <http://embox.github.io>

Embox: путь от студенческой забавы до проекта с открытым кодом

Аннотация

Доклад посвящен истории становления проекта Embox. В докладе описываются преимущества участия в open-source проектах и проблемы, которые с этим связаны.

Embox — свободная операционная система реального времени. Идея создания проекта появилась на кафедре системного программирования СПбГУ. Причём изначально основная идея заключалась в том чтобы улучшить навыки системного программирования у студентов, вовлекая их в реальный проект. Как преподаватели так и студенты являлись участниками свободного распределенного проекта. В котором более опытные участники являются наставниками менее опытных. Участники из преподавателей являлись не руководителями, а скорее кураторами проекта. Например они искали задачи которые могли бы быть решены в рамках проекта.

Первыми пользователями стали сотрудники кафедры теоритической кибернетики с Мат-Меха (того же факультета) СПбГУ. Проблема, которую предстояло решить группе проекта, заключалось в разработке альтернативной прошивки для Lego MindStorm 2. С поставленной задачей группа успешно справилась. Работа велась в рамках студенческого проекта и летней школы на кафедре системного программирования.

Кураторы совместно с опытными участниками из студентов параллельно решали организационные вопросы: выбор лицензии, поиск задач и направлений для развития, продвижение и создание сообщества.

В результате выбора лицензии выяснилось, что, поскольку проект ориентирован на встраиваемые системы, для него важна возможность оставлять некоторые части закрытыми, следовательно лицензия должна была отвечать этому условию. В итоге проект имеет BSD лицензию.

В результате поиска направлений развития был сформирован ряд задач, в которых применение Linux затруднено или невозможно в принципе. К таким задачам относятся: жесткое реальное время, различные виды сертификации и верификации кода, аппаратные ограничения по ресурсам и так далее. С другой стороны, появилось понимание, что пользователи хотели бы использовать всю мощь и удобство Linux, а не разрабатывать прикладное программное обеспечение с нуля, как это принято в небольших RTOS. В результате в Embox появился достаточно хороший слой совместимости с POSIX. Кроме стандарта POSIX разработчики в Embox могут использовать C++ и ряд других языков программирования, включая: python, lisp, lua, tcl и так далее.

Желание использовать аппаратные платформы с ограниченными ресурсами привело к созданию собственной системы сборки на основе специального языка описания модулей. Данный подход позволил включать в систему только требуемые части, детально их настраивать и затем генерировать различные артефакты. Все это позволяет создать систему с детально описанной функциональностью.

Кроме экономии ресурсов специальный язык описания модулей позволяет создать артефакты, упрощающие верификацию и сертификацию конечных систем, ведь в процессе конфигурирования и сборки строится полная внутренняя модель системы.

Сообщество наряду с направлениями (идеями) проекта является определяющим для любого проекта с открытым кодом. Без сообщества проект не существует, поэтому вопросам построения сообщества было также уделено большое внимание.

На старте проекта сообщество было создано из подмножества студентов Мат-Меха, которые хотели бы заниматься подобными вещами. Причем уже упомянутые сотрудники кафедры теоритической кибернетики также являются членами этого сообщества, как и любой пользователь проекта. Основным правилом развития проекта Embox, как, впрочем, и других СПО проектов, стало то, что любой участник что-то получает в замен своего участия, и то, что он получает, должно по его оценке быть адекватно его вкладу в проект.

Мотивацией для студентов, как первых участников проекта, была рекомендация на кафедру системного программирования, но очень быстро этой мотивации стало недостаточно для развития проекта. На смену этой мотивации пришли: желание улучшить свои качества как специалиста, а так же улучшить свое резюме, ведь участие в проекте с

открытым кодом может являться визитной карточкой при поступлении на работу. Еще большую мотивацию давали: желание заниматься интересными вещами и возможность реализовать себя.

При этом мотивацией безусловно служили и такие вещи, как возможность оформить курсовую, дипломную или научную работы, но они вряд ли могли являться основными, поскольку тематика и принятые процессы выводят сложность работы в проекте на более высокий уровень, по сравнению с учебными заданиями.

Необходимо также отметить недостатки или сложности использования СПО проекта в качестве средства обучения. Во-первых, подобный подход является более затратным по сравнению с обычными лекциями и практическими занятиями. Во-вторых, у студентов происходит очень сложный переход в свободное обучение, по сравнению с обычным подходом, когда сначала начитывают материал, а потом проверяют его усвоение. В-третьих, данный метод не может быть использован в качестве обязательного предмета.

Применение Embox как свободного проекта для обучения студентов в данной области показало себя эффективным. Участники проекта существенно улучшали свои навыки как в системном программировании, так и в промышленной разработке программного обеспечения. Подобный результат не мог быть достигнут без вовлечения студентов в качестве полноценных участников в проект.

Воронин Игорь Вадимович, Воронина Вероника Вадимовна
Шатура, Институт Проблем Лазерных и Информационных технологий
Российской Академии Наук, и СОШ, г. Павлово, Нижегородской области
Проект: Образовательный проект УМКИ <http://umki-kit.ru/>

Создание и использование в образовательном процессе видеороликов по робототехнике. Проект УМКИ

Аннотация

В статье обсуждаются способы создания и использования в учебном процессе видеопособий. Рассмотрены необходимые ресурсы для создания видеороликов. Предлагается решение в виде комплекта видеопособий по образовательной робототехнике УМКИ, как пакет в школьном дистрибутиве УМКИ.

Нынешние школьники очень много времени проводят в социальных сетях: общаются при помощи мобильных устройств, интересующую их информацию получают в мультимедийном виде, как правило, в форме видеороликов. Такое клиповое мышление современных подростков, затрудняет понимание информации, представленной в традиционной текстовой форме. Как показывает опыт, зачастую им проще понять необходимый материал через просмотр ролика в Ютубе, чем через прочтение нескольких страниц текста.

Видеоплощадки становятся традиционными для преподавания технических дисциплин, набирают массовую популярность, например, такие как Coursera, edX, Udacity. Публикуемые типы роликов могут быть в виде записи лекции или доклада на конференции. Но возможен и другой вариант: запись обучающих видеоматериалов в форме пособия, к которому обращаются постоянно, чтобы понять и разобраться, как достигнуть того или иного результата.

В современном образовательном процессе ведущих мировых университетов — MIT, Harvard и UC Berkeley — видеоконтент образовательных ресурсов уже давно занимает важное место. Университеты тратят много времени и денег на создание таких видеороликов, решая при этом связанный вопрос, как улучшить обучение учащихся, сохраняя высокие темпы производства учебного видео и придерживаясь разумных финансовых и трудовых затрат. Профессора ведущих университетов мира проводят различные эмпирические исследования взаимодействия студентов с видеоконтентом. Эти исследования обращают внимание на качество воспринимаемой информации — можно выделить два основных параметра: время просмотра ролика и качество усвоенного материала. Качество восприятия проверяется итоговым тестированием ученика, время просмотра фиксируется площадкой видеохостинга.

Учебные видеоролики могут быть представлены в различных стилях, которые в основном можно сгруппировать как:

- набор слайдов презентации с пояснениями преподавателя;
- код набираемый преподавателем в видеоредакторе — скринкаст (грабилка) экрана;
- полноэкранное видео рисунков от руки с цифрового планшета;
- учебная аудитория или студия для записи выступления, оформленная под обсуждаемую тему;

- запись выступления одного или интервью разных преподавателей.

Вместе с тем, хорошим решением для повышения усвоения материала обучающимися является связка печатного издания с набором видеоматериалов по общей теме. В 2018 году издательством Питер была выпущена книга за нашим авторством «Программирование для детей. От основ к созданию роботов» как учебное пособие для проведения занятий, и в 2019 году в издательстве «Дрофа» вышел единый учебник технологии, где авторами раздела по робототехнике также являлись мы. Эти издания входят в учебно-методический комплект образовательной робототехники УМКИ (пакет в школьном дистрибутиве УМКИ с использованием сред визуального программирования SNAP и КУМИР, которые включены в базовую поставку Альт Образование). В связи с изложенным в начале статьи, для повышения вовлеченности обучающихся в образовательный процесс, нами были разработаны наборы мультимедийных материалов, включая и наборы видеороликов, как дополнительное пособие по технологии программирования. Использование видеоматериалов позволяет педагогу на занятиях в полной мере реализовать базовые педагогические принципы: наглядности, научности и практикоориентированности, позволяя ученику самостоятельно заниматься с материалом, переходя от простого к сложному, повторяя при необходимости нужные фрагменты и продвигаясь вперед в собственном темпе.

На сайте ИПЛИТ РАН мы опубликовали ссылку на вики-ресурс, где собраны ролики по различным категориям, сами ролики при этом загружены на Ютуб. В этом курсе предлагается два направления обучения программированию микропроцессоров в среде Ардуино. Первый раздел ориентирован на учеников начальной школы: обучение из среды визуального программирования Snap! (аналог Scratch). Второй — для учеников основной школы: обучение программированию в среде Ардуино IDE. В двух различных средах ученики знакомятся с базовыми функциями обращения к портам процессора, начиная от самых простых, управляя встроенным светодиодом на тринадцатом контакте Ардуино, до достаточно интересных проектов, помогающих юному пользователю самостоятельно создать работу, которая в будущем сможет оказаться базой для собственного стартапа.

Используемое в обсуждаемом учебно-методическом комплекте оборудование построено на базе AVR микропроцессоров ATmega —

контроллеров Xbee, Arduino с наборами датчиков и электронными компонентами, позволяющими создавать программируемые модели роботов.

Для записи видео роликов использовались инструменты, доступные в среде: «Альт Образование 9» это:

- SimpleScreenRecorder — программа для записи игр с экрана использовалась для записи действий на экране;
- Inkscape — профессиональное программное обеспечение для работы с векторной графикой использовалась для записи титров;
- MEncoder — программа для конвертирования аудио и видео файлов. Использовалась для нарезки видео роликов, склейки по сценарию, конвертации видео в конечный формат.

Естественно, использовались программы самой операционной системы Linux — FTP, SCP для переноса видеороликов с мобильного телефона на сервер хостинга.

Поскольку обучающих видеороликов было создано достаточно большое количество, то для удобной навигации был организован и использован WIKI-сервер, чтобы всю информацию упорядочить организовать структурную навигацию для ориентации по группам роликов.

Порядок создания был следующий.

1. Составление сценария ролика для определения того, что за чем должно быть представлено и к чему необходимо подойти в финале ролика.
2. Составление схемы на базе набора оборудования, задействованного в ролике. Набор оборудования, используемого для этого был следующим:
 - ноутбук с ОС ALT Linux для сборки готового видеоролика;
 - смартфон, на который снимались действия на столе;
 - штатив — для крепления смартфона;
 - плата УМКИ-К6, включающая контроллер Arduino Nano;
 - набор электронных компонентов подключаемых к плате УМКИ-К6 и комплект проводов;
 - стол, свет, фон — хромакей.

3. Составление программного кода, который будет разбираться в ролике, проверка его работоспособности и соответствия портам, собранным на схеме.
4. Съемка нескольких частей видеозаписи, на смартфоне и на экране, с сохранением результатов в папке конкретного видеоролика.
5. Обрезка ненужных моментов из роликов, например, командой:
`$mencoder sh924_2.mp4 -ovc copy -oac mp3lame -of lavf -ss 1:38 -endpos 6 -o 116.mp4,`
где **sh924_2.mp4** — имя входного файла, **-o 116.mp4**, — имя выходного файла, **-ss 1:38** обрезаем с минуты 38 сек. в течение 6 секунд.
6. Переноска нарезанного набора роликов в папку с файлами титров, заранее подготовленных в едином стиле в виде картинок, и запуск склейки финального ролика, например, командой:
`$mencoder 1*.mp4 -ovc copy -oac mp3lame -of lavf -o 222.mp4,`
где **1*.mp4** — имена всех файлов в нужной последовательности, **-o 222.mp4** — имя выходного файла.
7. Публикация полученного файла на сервере видеохостинга youtube.com.
8. Создание на WIKI сервере раздела со ссылкой на ролик в Ютубе и, оформление его изображением схемы эксперимента и программного кода для этого занятия.

Опыт преподавания программирования в школе показывает, что публикацию подобных видеоматериалов можно организовать на потоке для достаточно большого числа учащихся, причем быстро, с высоким качеством и доступностью для преподавателя, даже в том случае, когда преподаватель может не владеть в полной мере новейшими информационными технологиями, необходимыми для организации обучения в рамках ФГОС последнего поколения. Мы уверены, что использование подобных видеоматериалов позволит организовать современный учебный процесс с минимальными затратами на переобучение преподавателей и с максимальным положительным и качественным результатом.

Нужно отметить, что все видеоролики, приведенные на сайте, а также 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 Девятой платформы (p9) .

Литература

- [1] Фомин Стас. Инструмент для эффективного монтажа семинаров, докладов и прочего образовательного видео. Режим доступа: <http://wiki.4intra.net/SeminarAssembler>
- [2] Воронин И. В., Воронина В. В. Программирование для детей. От основ к созданию роботов. СПб.: Издательство Питер, 2018.
- [3] Фомин Стас. Крупнейший каталог российских докладов и семинаров ПО IT-тематике! Режим доступа: <http://0x1.tv/Медиаотека>
- [4] Воронин Игорь. Использование программируемого робота конструктора в образовательных целях, Ярославский педагогический вестник. 2013.
- [5] Басан Е. С., Басан А. С., г. Таганрог, РФ. Обзор аппаратных платформ для реализации беспроводных сенсорных сетей.
- [6] Астапова О. В., Использование видеороликов в процессе обучения. // Успехи современного естествознания. 2011. № 8. С.152а. Пенза: Академия Естествознания.
- [7] Бэрт Мартен, автор SimpleScreenRecorder — программы в Linux для записи программ и игр Режим доступа: <https://www.maartenbaert.be/simplescreenrecorder/>
- [8] Страница сайта образовательных видео роликов УМКИ <http://meteor.laser.ru/umki/index.php/Видео>

Проскурнев Артем

Москва, Школа № 830

Проект: ОС «Комета» <http://oskometa.ru>

СПО — инструмент для повышения компьютерной грамотности на уроках информатики.

Аннотация

В современной московской школе есть возможность использовать самые новые аппаратные и программные средства на уроках информатики. Потенциально способные ученики попадают в «ловушку прогресса». Лёгкие и примитивные интерфейсы, решения, спрятанные настройки, ну и, конечно, закрытый код «балуют» способных учеников и не побуждают желания разбираться в задачах глубже среднестатистического пользователя. СПО способно решить часть проблем и успешно решает их в одной из московских школ.

В современной московской школе есть возможность использовать самые новые аппаратные и программные средства на уроках информатики. Производители программного обеспечения сейчас делают всё возможное, чтобы удовлетворить потребности даже самых низкоквалифицированных пользователей. Это необходимо делать, так как компьютеры плотно вошли во все сферы жизни и большинство пользователей просто не справляется со сложными задачами или сложным интерфейсом. Однако обратной стороной этого процесса упрощения является подавление желания хорошо разобраться в задаче и тем, кто на это потенциально способен.

В качестве примера можно привести опыт установки операционных систем на компьютер. Способные ученики пробовали установить Windows и Linux. Современная установка Linux позволяет также установить систему, не углубляясь в настройки. Достаточно просто нажимать «Далее», соглашаясь со всем на всех этапах. Система Комета[2] специально была подготовлена так, чтобы вообще ничего не нужно было настраивать, только ввести пароль администратора в конце установки и ввести имя и пароль пользователя. Учащиеся не справились с задачей, потому что начали читать, что написано на этапах установки и пытались в неё вмешаться. По их словам, при установке Windows они почти никогда не читают, что написано на экране, а

просто пытаются найти кнопку «Ок» или «Далее». Уже данный пример показывает полезность изменения «стандартного и привычного». Ученик начал задумываться над тем, что он делает.

Свободное программное обеспечение для полноценного использования требует определённых навыков, в которых так нуждаются учащиеся на уроках информатики в школе. Хотя интерфейс свободных программ также стремится к упрощению для конечного пользователя, но, в связи с открытостью кода, позволяет, в отличие от проприетарных программ, использовать недокументированные способы управления, что развивает мышление учащихся.

Очень популярное сейчас в школьной среде самовыражение детей можно направить в глубокую и полную настройку компьютера с Linux «под себя». Кому-то хватает только цвета и картинок, а кто-то хочет сделать такое, что возможно только при вмешательстве в код программы.

Конечно, нельзя обойти стороной и игры. Свободные игры позволяют реально, а не на словах, изменить то, чего хочет ученик. Открыть код, найти нужное, изменить по своему желанию и скомпилировать — вот то, что также очень помогает развитию даже самого предмета «Информатика».

Не забудем, конечно, и про робототехнику. Тут в качестве примера, я хочу привести проект «УМКИ» Игоря Воронина[1], так как эти наборы были поставлены в нашу школу. Набор использует платформу `arduino`, чтобы позволяет ученикам пройти, увидеть и изменить всё — от кода программного обеспечения УМКИ до схемы самого аппарата.

Выводом напрашивается предложение сделать обязательным изучение свободного программного обеспечения наряду с обязательным изучением офисных систем. Умение пользоваться последними проверяется на Государственной итоговой аттестации (ГИА, ОГЭ). Было бы замечательно дать информацию в школе, что для различных задач существуют свободные программы, которые способны решить почти любые возникающие задачи: редактирование фото — `gimp`, векторные рисунки — `inkscape`, 3D моделирование — `blender`, и так далее.

В робототехнике отказаться от проприетарных платформ в пользу открытых по всем тем же причинам, которые изложены выше.

В качестве итога можно указать на тот факт, что всё вышеизложенное успешно применяется, и учащиеся демонстрируют изменение подхода, а, главное, у многих возникает желание учиться.

Литература

- [1] Воронин Игорь, УМКИ
- [2] Проскурнев Артем, ОС «Комета», 2015

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

Slang IDE — среда для обучения начинающих программистов

Аннотация

Описывается среда Slang IDE, предназначенная для продолжения обучения программированию после начального обучения в среде Semantic IDE. Среда Slang IDE построена на базе редактора Visual Studio Code. Учебный язык Slang является англоязычным и представляет собой модифицированную версию языка Semantic Language. Создание программы осуществляется с помощью набора сниппетов, причем допускаются вставки кода на C++. Среда позволяет создавать одномодульные и многомодульные проекты и транслирует программы на языке Slang в программы на языке C++14 и выше. Создание исполняемого кода осуществляется компилятором g++. Стандартная библиотека написана на C++ и может быть расширена.

В 2012 году на кафедре АСОИУ Астраханского технического университета была разработана дружественная для начинающих среда программирования Semantic IDE на основе учебного языка Semantic Language, которая стала использоваться в учебном процессе. 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 переходную среду программирования с тем же учебным языком программирования, но более приближенную к реальным профессиональным средам типа Visual Studio.NET. Эта интерпретированная среда должна в дальнейшем войти в систему обучения программированию в качестве клиента.

В качестве редактора новой интегрированной среды выбран свободно распространяемый редактор Visual Studio Code (VSCode). Редактор работает на любой платформе, что важно в связи с предстоящей установкой в учебных классах операционной системы Альт Образование. 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».

Разработка осуществляется в трех направлениях: учебный язык, компилятор и среда.

Учебный язык

Разработанный в 2012 году учебный язык программирования Semantic Language включал типовые для императивных языков средства:

- небольшое множество элементарных типов данных;
- множество типичных операций с элементарными типами;
- минимальный набор управляющих операторов;
- небольшой набор типичных агрегатов данных;
- средства определения новых процедур и функций;
- средства для работы с динамической памятью;
- механизмы определения новых типов данных (классы);
- конструкция определения модулей.

Важной особенностью языка является запрет умолчаний — все намерения программиста должны быть указаны явно (например, указание класса как базового). Отсутствие умолчаний означает, что все ошибки, совершенные при написании программы, совершены программистом, а не вследствие неучтенных/неизвестных действий системы.

Была разработана и библиотека стандартных модулей, которая включала более 10 модулей как для английской, так и для русской нотации.

При разработке новой IDE учебный язык Semantic Language был проанализирован и доработан с целью приблизить его лексику и синтаксис к промышленным языкам программирования C++, Java и C#. В частности:

1. лексика оставлена только англоязычная;
2. оператор `import` вынесен за пределы модуля;

3. исключены переменные процедурного типа;
4. тип `string` включен в систему базовых типов языка;
5. полностью определены синтаксис и семантика объектно-ориентированной части языка на основе встроеного в язык базового класса `Object`;
6. поля в классе могут иметь атрибут `readonly`;
7. уточнен минимально необходимый состав стандартных модулей: `system`, `files`, `math`, `random`, `string`.

Пересмотренный учебный язык программирования назван `Slang`.

Компилятор

Вместо интерпретатора, который был реализован в `Semantic IDE`, новая интегрированная среда включает компилятор, который реализуется как консольное приложение на языке `C#` на платформе `Net Core`. Компилятор имеет в своем составе четыре подсистемы:

1. Синтаксический анализатор;
2. Семантический анализатор;
3. Конвертор в язык `C++`;
4. Компилятор `C++`.

Синтаксический анализатор (парсер) сгенерирован с использованием генератора пасеров `ANTLR`, для которого была разработана грамматика учебного языка `Slang`. Семантический анализатор пишется вручную и осуществляет семантические проверки синтаксически правильного кода. Эти две подсистемы совместно представляют `front-end-compiler`, который работает до первой ошибки (синтаксической или семантической). Работа до первой ошибки для начинающих программистов более удобна, чем получение всех обнаруженных ошибок, многие из которых являются следствием нескольких первых. Текст сообщения об ошибке выдается на русском языке.

Правильная программа конвертируется в `C++`, и компилятор создает исполняемый файл. В настоящее время используется компилятор `g++` из пакета `MinGW`, однако имеется возможность подключить любой доступный компилятор `C++`. Конвертер и компилятор совместно представляют собой `back-end-compiler`.

Редактор кода

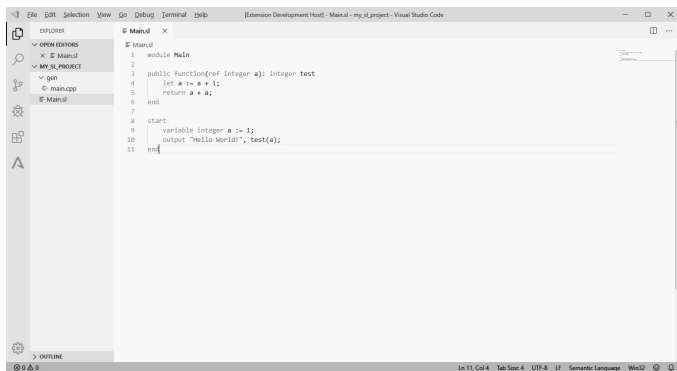
Интегрированная среда реализуется на основе свободного редактора VSCode. Для поддержки обучения программированию на учебном языке Slang в редакторе реализована типичная подсветка синтаксиса (ключевые слова выводятся голубым цветом).

В целях минимизации ошибок при наборе кода реализован большой набор языковых сниппетов, который будет дополняться и модифицироваться по результатам опытной эксплуатации.

В редактор добавлен механизм создания проектов и интеграции созданных проектов с рабочими пространствами (workspace) VSCode. Реализованы возможности создания как одномодульных, так и многомодульных проектов. При создании любого проекта сразу создается главный модуль Main.sl, который является точкой входа в программу.

Внешний вид редактора с кодом простой программы на языке Slang показан на рисунке 1.

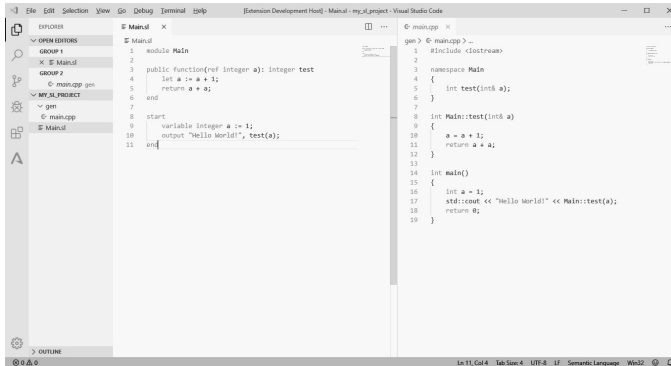
Рисунок 1 — Внешний вид окна редактора с главным модулем Main.sl



Помимо традиционных средств разработки программ среда позволяет вставлять в код на языке Slang фрагменты кода на C++, используя специальные атрибуты в коде.

Для целей обучения реализована возможность наблюдать в редакторе одновременно код на языке Slang и соответствующий ему код на C++, который сгенерировал конвертер. Вид окна редактора с кодами на двух языках показан на рисунке 2.

Рисунок 2 — Окно редактора с исходным и транслированным кодом



Текущее состояние проекта

В настоящее время все разработки выполняются авторами в рамках курсового и дипломного проектирования по различным дисциплинам.

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

Проекты текущей рабочей версии проекта размещены на github: <https://github.com/NothingIsGood/SlangCompiler>;
<https://github.com/NothingIsGood/Semantic-Language-Extension>.

В настоящее время планируется установка Slang IDE в учебные классы с операционной системой Windows для апробации в учебном процессе. После установки в учебные классы операционной системы Альт Образование интегрированная среда будет установлена и в этой ОС.

Валерий Лаптев, Виктория Толасова
Астрахань, Астраханский государственный технический университет

Среда для проведения контрольных работ по программированию на базе СПО

Аннотация

Описывается среда Control IDE, которая на кафедре АСОИУ используется на экзаменах по программированию с 2013 года. Среда ограничивает доступ экзаменуемого к информации вне среды. Ограничена работа с буфером обмена в рамках среды. В среде разрешаются все типовые операции для создания и проверки программ на C++. Используется свободный компилятор g++ из пакета MinGW. Все действия студента записываются в журнал и могут быть просмотрены преподавателем в браузере. Среда позволяет ограничить время написания программы в течение экзамена.

При проведении контрольных мероприятий по программированию помимо проверки теоретических знаний требуется проверять умение писать программы. Поэтому в контрольной работе (на экзамене) обязательными являются задания по написанию программы для решения некоторой типовой задачи. Такие задачи желательно выполнять с использованием некоторой интегрированной среды.

При проведении контрольного мероприятия требуется снизить риски академической нечестности экзаменуемых. Использование компьютера на экзамене не исключает возможность несанкционированного доступа студента к информации по теме экзамена. Промышленные среды, которые обычно используются для написания программ, не позволяют снизить эти риски. Поэтому необходима среда, которая ограничивает доступ экзаменуемого к любым документам и ресурсам.

На кафедре АСОИУ Астраханского государственного технического университета разработана и с 2013 года эксплуатируется интегрированная среда, предназначенная для контроля знаний студентов по программированию на языке C++.

Студент при работе в среде может выполнять следующие операции:

- создавать проекты как из одного модуля, так и многомодульные;
- редактировать и форматировать код программы на языке C++;
- компилировать программу и выявлять синтаксические ошибки;

- запускать откомпилированную программу на выполнение;
- вводить тестовые данные для проверки программы;
- эмулировать работу с файлами (как текстовыми, так и бинарными).

Среда позволяет пользоваться буфером обмена при редактировании кода, но только внутри самой среды — скопировать заранее подготовленный текст из другой программы невозможно.

При решении задач на обработку файлов требуется предотвращение открытия произвольного файла, поэтому среда разрешает использовать в программе только два имени файла — `input.txt` и `output.txt`, без указания каталога. Допускается работать с этими файлами и как с текстовыми, и как с бинарными. Среда отображает содержимое этих файлов в своих окнах, не позволяя просматривать их другими средствами.

Среда не позволяет студенту использовать конструкции языка и возможности операционной системы, с помощью которых можно получить доступ к посторонней информации, так как во время работы среда подавляет все посторонние окна. Более того, при работе в среде невозможно покинуть окно редактора, причем все такие попытки средой фиксируются, и осуществляется визуальная сигнализация для экзаменатора.

Помимо запрета действий среда выполняет следующие полезные действия:

- сохраняет код программы перед каждой компиляцией;
- записывает в журнал информацию о действиях студента при работе над заданием;
- отслеживает общее время работы и сохраняет время начала каждой попытки компиляции;
- подсчитывает общее количество попыток компиляции и количество удачных.

Также периодически осуществляется автоматическое сохранение кода на случай внезапного отключения электроэнергии. В программе предусмотрена также защита от использования слишком длинных строк (свыше 200 символов) и заикливания при записи в файл.

Внешний вид окна среды с запущенной программой студента показан на рисунке 1. Все кнопки имеют всплывающие подсказки.



Рис. 1: Окно среды с программой студента, запущенной на выполнение

Для преподавателя, проводящего контрольную работу или экзамен, среда обеспечивает следующие возможности:

- запустить и отключить среду (по паролю);
- ограничивать время работы студента над программой;
- просматривать каждый вариант программы, отправленной на компиляцию;
- копировать тексты сохранённых вариантов программы в буфер обмена;
- посмотреть статистику действий студента, собранную программой.

После запуска среда перехватывает все несанкционированные команды, однако позволяет загрузить текстовый файл с заданием (формат `txt` или `rtf`), указанный преподавателем.

После завершения работы программы журнал действий пользователя может быть открыт в браузере. В журнале содержатся гиперссылки на версии текстов программ, сохраняемых перед компиляцией проекта, а также на список и общее количество обработанных процессов. Пример открытого журнала показан на рисунке 2.

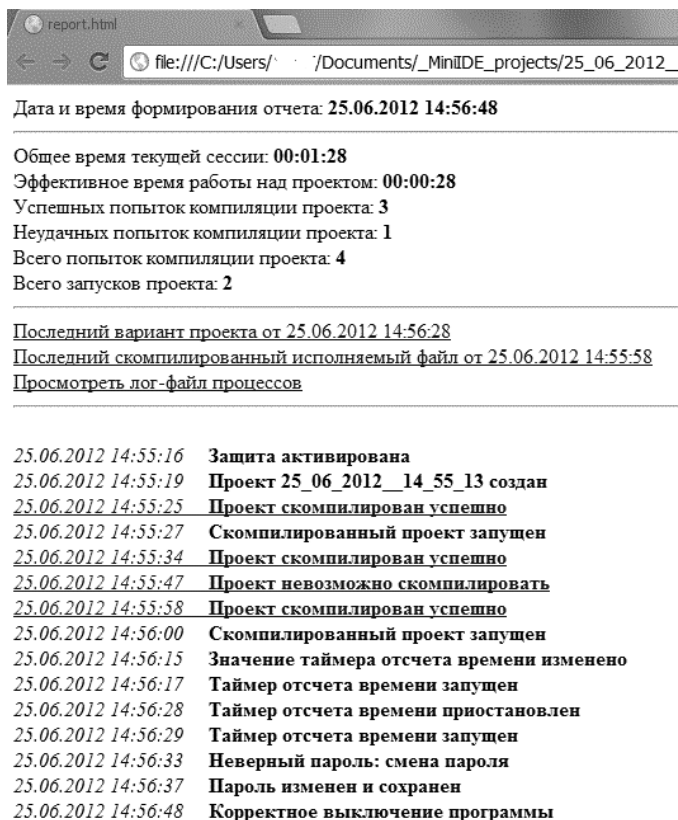


Рис. 2: Журнал действий студента

Все ограничения среда обеспечивает в контрольном режиме. Помимо этого среда может использоваться как обычная интегрированная среда для подготовки студентами домашних и лабораторных работ.

Разработка велась силами студентов направления «Программная инженерия» под руководством авторов. Основной разработчик был один, но иногда привлекались другие студенты. Язык реализации C# в среде Visual Studio.NET 2010. Во время работы среда взаимодей-

ствуется с компилятором GCC, входящим в свободно распространяемый пакет MinGW.

Эксплуатация проекта в течение 6 лет показала его практическую полезность. Однако в настоящее время среда устарела и имеет ряд серьезных недостатков.

1. Проектная документация отсутствует, поэтому проект сложно развивать.
2. Проект был реализован с использованием бесплатных средств от Microsoft, поэтому работает только на платформе Windows, и перенос его на Linux или в Интернет по трудоемкости сравним с новой разработкой.
3. Среда является локальной, предназначена для одного пользователя и не содержит никаких средств для коллективной работы.

Поэтому в 2019 году принято решение модифицировать данный проект.

Текущее состояние проекта

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

Для работы в новых версиях операционной системы Windows система переписывается с использованием свободного пакета NET Core с целью последующего включения его в общую обучающую среду кафедры.

Кроме того, в связи с установкой в ближайшем будущем в учебных классах операционной системы Альт Образование разрабатывается версия контрольной среды для работы в Linux.

В обеих версиях используется одна и та же версия компилятора GCC из пакета MinGW-64.

Проект текущей рабочей версии размещен на github:
<https://github.com/iwbas/control-environment>

К началу нового учебного года планируется завершить рефакторинг версии среды для Windows и разработать первую версию для работы в Linux.

Попцов Артём Вячеславович, Плеханов-Хоршев Антон
Сергеевич

Нижний Новгород, ООО «Группа компаний Азъ»

Проект: MST <https://gitlab.com/gkaz/mst>

MST: Автоматизация настройки мультисит-системы

Аннотация

Конфигурация компьютера в формате «Мультисит», или в режиме многоместной системы, когда за одним системным блоком могут одновременно работать несколько пользователей, известна давно. Однако до недавнего времени не существовало свободного инструмента для полноценной настройки графического мультисита на GNU/Linux. Разработанное нами свободное программное обеспечение «MST» позволяет решить эту проблему, полностью автоматизируя настройку многоместной системы (multi-seat) на ОС Альт.

Unix была одной из первых систем с разделением времени, иными словами, позволяющей работать за одним компьютером нескольким людям одновременно. Эти возможности являются частью Unix-подобных систем и по сей день. Однако процесс настройки подобной конфигурации требует обширных знаний в системном администрировании и написании скриптов, либо же требует наличия дополнительных видеокарт на каждое рабочее место. До недавнего времени не существовало полноценной свободной утилиты для полной автоматизации процесса настройки многоместной системы на базе компьютера с одной видеокартой. Разработанное ООО «Группа компаний Азъ» свободное (GNU GPL v3.0+) программное обеспечение под названием «MST» для ОС Альт позволяет решить данную проблему.

За счёт использования общедоступного свободного программного обеспечения «MST» является легко переносимым на другие дистрибутивы GNU/Linux. На данный момент стабильная работа системы обеспечена на ОС Альт, и ведутся работы по переносу «MST» на Ubuntu GNU/Linux.

Возможности системы

- «MST» позволяет полностью автоматизировать процесс настройки мультисит-системы на персональном компьютере или ноутбуке.

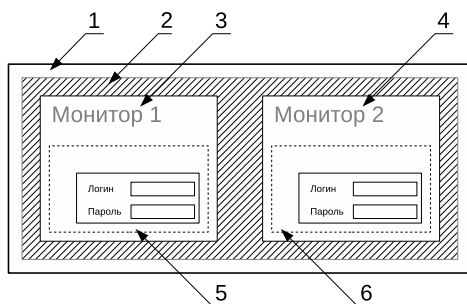


Рис. 1: Архитектура MST: сначала запускается основной X-сервер (1); затем поверх него запускается тайловый оконный менеджер Awesome (2); в Awesome на каждом мониторе разворачивается окно процесса Xephyr, которое порождает вложенный X-сервер (3, 4); внутри каждой Xephyr-сессии запускается менеджер входа в систему LightDM (5, 6).

- Система позволяет сделать два или более рабочих мест из одного компьютера.
- Система позволяет задействовать аппаратное ускорение графики на рабочих местах.
- Не требуется никакого специального оборудования – «MST» может задействовать встроенную видеокарту с несколькими выходами.
- Система позволяет откатить сделанные изменения и вернуть изначальную настройку операционной системы.

Принцип работы

«MST» использует свободно доступные компоненты операционной системы Альт, которые могут быть установлены из репозитория. Архитектура «MST» представлена на рис. 1.

При запуске «MST» показывает графическое окно, которое позволяет администратору системы настроить рабочие места, начиная от



Рис. 2: Главное окно программы

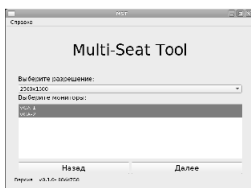


Рис. 3: Выбор мониторов и настройка разрешений



Рис. 4: Настройка рабочих мест



Рис. 5: Привязка устройств к рабочему месту

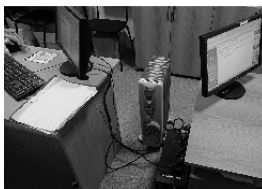


Рис. 6: Два рабочих места на «MST» в НРПК

разрешения экрана и заканчивая привязкой USB-устройств к каждому рабочему месту.

Заключение

На данный момент «MST» успешно внедрён на рабочих местах сотрудников на заочном отделении Нижегородского радиотехнического колледжа (НРПК), в отделе кадров и у заведующего учебным отделением.

Д.А. Костюк, П.Н. Луцюк

Брест, Брестский государственный технический университет

Практическое изучение средств контейнерной виртуализации и платформы Kubernetes

Аннотация

Приводится опыт разработки учебного курса по основам использования оркестратора контейнеров Kubernetes, включая принципы контейнерной виртуализации, архитектурные особенности и ключевые компоненты платформы Kubernetes, развертывание и масштабирование приложений на ее основе. Приведена структура практической части курса, меры по обеспечению работоспособности изучаемого программного обеспечения в условиях изолированного сегмента сети и ограниченного Интернет-доступа, обсуждаются способы преодоления высокого входного порога Kubernetes.

Введение

Контейнеризация приложений — существенная часть актуального подхода к системной инженерии, предполагающего интеграцию разработки и эксплуатации программных систем при построении сложных приложений. В основе лежит использование комплекса таких современных программных технологий, как виртуализация, автоматический аудит и контроль производительности, решение задач с применением гетерогенных систем, включающих несколько различных платформ.

Одним из характерных средств реализации данного подхода является Kubernetes — развиваемая Google сложнокomпонентная система оркестрации контейнеров с открытым исходным кодом.

Объекты Kubernetes разворачивают и масштабируют приложения на основе требований к памяти, ЦП и др. При этом поддерживается любая система контейнерной виртуализации, удовлетворяющая Open Container Initiative (OCI). Kubernetes обеспечивает масштабирование и балансировку нагрузки, автоматическое обнаружение сервисов и управление секретами. При этом контейнеры, являющиеся компонентами одного или нескольких приложений, изолированы друг от друга, если только разработчику/devops-инженеру не требуется обратного. Платформа берет на себя технические сложности по обеспечению прозрачного взаимодействия контейнеров и автообнаружения сервисов, предоставляя соответствующие возможности с помощью API.

Специфика изучаемого материала

Проблема, с которой регулярно сталкиваются при освоении Kubernetes — высокий порог вхождения. С одной стороны причина в достаточно большом наборе сущностей данной платформы, благодаря которым кластер на базе Kubernetes действует как единое целое, а с другой — в значительном числе входящих в ее состав компонентов, некоторые из которых к тому же являются заменяемыми. Универсальный кластер, автоматически выполняющий обновления, масштабирование и самовосстановление, демонстрирует достаточно высокую концептуальную сложность своего устройства. По этой причине, несмотря на высокую востребованность, преподавание данной платформы студентам часто оказывается проблематичным.

В нашем случае разрабатывался курс по изучению Kubernetes студентами старших курсов, уже знакомыми с рядом необходимых технологий, включая компьютерные сети, элементы системного администрирования GNU/Linux, архитектуры клиент-серверных приложений и др.

Разработанный учебный курс рассчитан на первую ступень высшего образования и предполагает дальнейшее развитие в рамках второй ступени. Теоретический материал основан на официальной документации Kubernetes [1], а лабораторный практикум охватывает лишь часть теоретического материала, в результате чего студенты имеют дело с меньшим количеством объектов платформы.

Конфигурирование программного обеспечения и сетевой доступ

Практическая часть курса построена на основе Minikube — специализированного дистрибутива Kubernetes, предназначенного для развёртывания на локальной машине с совмещением аппаратной и контейнерной виртуализации [2]. Minikube имеет ряд ограничений, связанных в первую очередь с его локальной природой. Это не является проблемой, поскольку практическая часть курса в силу ограничений по времени не затрагивает ряд несовместимых аспектов платформы: такие темы, как, например, реализация аффинитета и антиаффинитета, рассматриваются только в теоретической части.

Из внешних программных зависимостей требуется только VirtualBox, который доступен в стандартных репозиториях Linux; остальное скачивается из сети при установке и развёртывании:

- Minikube скачивает готовый установочный образ с Docker и компонентами Kubernetes, а также актуальные версии kubelet и kubeadm;
- далее при развёртывании приложений в кластере Kubernetes контейнеры приложения скачивает уже Docker.

В связи с этим генерируется сравнительно большой сетевой трафик (бинарные файлы minikube-linux-amd64 и kubect1, образ виртуальной машины minikube-v*.iso, а также образы контейнеров имеют общий объём более 600 Мб), и потому одной из задач, которые потребовалось решить для входящего в курс лабораторного практикума, было функционирование Minikube в изолированном сегмен-

те локальной сети, в условиях ограниченного доступа к внешним интернет-ресурсам (помимо уменьшения трафика и сопутствующих временных задержек к такой схеме подталкивала внутренняя политика безопасности локальной сети, предполагающая персонализированный интернет-доступ через VPN).

Было рассмотрено и опробовано два варианта локального использования Kuberbnetes:

1. создание приватного реестра Docker и перенастройка системы на его эксклюзивное использование, а также развёртывание приватного репозитория образов для Minikube;
2. запуск Minikube и пробное развёртывание учебных приложений при включённом доступе к Интернету с последующим клонированием полученного профиля на рабочие станции учебного класса (решаются как вопросы развёртывания образов для Minikube, так и импорт образов Docker через файловую систему вместо их получения из внешней сети).

После экспериментов с обоими вариантами мы остановились на втором как наименее трудоёмком, с учетом имеющейся действующей системы тиражирования образов для рабочих станций [3].

При этом следует отметить, что несмотря на стандартную политику обновления контейнеров Kubernetes `IfNotPresent` (скачивать образ контейнера в случае его отсутствия в локальном кэше), на практике потребовалось устанавливать ее значение `Never`, чтобы обеспечить полную воспроизводимость использования именно локальных образов без обновления.

Структура практикума

Структура разработанного практикума для студентов первой ступени включает 4 лабораторные работы. Первая работа носит ознакомительный характер: в ней рассказывается о Minikube и системах виртуализации, которые он может использовать на различных аппаратных платформах, а также об особенностях установки в случае использования на собственных устройствах. Во второй работе рассматриваются особенности доступа к кластеру Kubernetes с помощью командной строки (`kubect1`), включая минимальное конфигурирование доступа (`kubect1 proxy`), и веб-интерфейса (Kubernetes Dashboard). В третьей работе выполняется развёртывание минимального веб-

приложения в кластере, включая проверку созданных Deployment, ReplicaSet и Подов, а также Service с доступом через NodePort. Четвёртая работа позволяет изучить развертывание многокомпонентного приложения с заготовленной клиентской частью на базе веб-технологий и серверной подсистемой на основе СУБД MongoDB.

Литература

- [1] Kubernetes Documentation. Режим доступа: <https://kubernetes.io/docs/home/>
- [2] Hello Minikube. Режим доступа: <https://kubernetes.io/docs/tutorials/hello-minikube/>
- [3] *Пойта П. С., Костюк Д. А., Дереченник С. С., Луцюз П. Н.* Повышение сетевой безопасности в компьютерном парке вуза за счет буферизации и изоляции ресурсов // *Электроника инфо*. 2013. №6(96). С. 111–113.

Михаил Шигорин

Москва, ООО «Базальт СПО»

Альт Образование и Эльбрус: задачи и решения

Аннотация

Следом за переносом наших дистрибутивов для сервера и рабочей станции на платформу «Эльбрус» настал черёд и образовательного. С одной стороны, эта задача во многом близка к уже решённым, так как примерно половина компонент пересекается с требуемыми для обычного десктопа и немножко сервера; но с другой – всю соль варианта Альт Образование составляют именно пакеты для обучения, некоторые из которых написаны на языках или библиотеках, реализации которых на e2k ещё не портированы.

Задачи

Что требуется от образовательной среды? В целом – соответствие Распоряжению Правительства РФ от 18 октября 2007 г. 1447-р, а также здравому смыслу (включая разумную цену отдельного места).

И если с первой частью текущий состав дистрибутива Альт Образование на платформе x86 предоставляет адекватное решение, то

стоимость вычислительных комплексов «Эльбрус» легко может оказаться запредельной для образовательных учреждений (на данный момент она примерно на порядок превышает типовую стоимость достаточных ПК).

Решения

Мультитерминальность

Как уже упоминал в докладах про многоместные системы на «Эльбрус» [4][5], нынешнюю цену этих комплексов вполне получается делить как минимум надвое при помощи хорошо известного в некоторых вузах подхода, получившего в России название «Горыныч» [6]: к одному системному блоку подключается несколько рабочих мест (обычно через несколько видеокарт, хотя и тут возможны варианты).

Очевидной выгодой такого подхода является экономия на стоимости необходимых для организации учебного класса системных блоков: вместо, скажем, тридцати может оказаться достаточно пятнадцати или даже десяти. Разумеется, есть и обратная сторона медали: места оказываются взаимосвязанными по вычислительным ресурсам и администрированию, а их настройка требует некоторых дополнительных усилий.

В нашем случае рассматриваются современные отечественные компьютеры «Эльбрус 801-РС», штатно оснащённые одним восьмиядерным процессором «Эльбрус-8С», 32 Гб ОЗУ и одной видеокартой Radeon R5 230; даже беглый взгляд на эту конфигурацию подсказывает её явную избыточность для одного-единственного пользователя, а более внимательный — то, что имеющиеся ресурсы можно разделить между двумя (как минимум) пользователями без какого-либо практического ущерба для работы.

Поскольку конфигурация аппаратуры в целом известна и более-менее предсказуема, в нашем конкретном случае получается заготовить программную конфигурацию таким образом, что при установке дистрибутива достаточно просто согласиться с предложенным инсталлятором количеством рабочих мест в зависимости от количества обнаруженных видеокарт (разница состоит по большей части в конкретике распределения USB-портов между рабочими местами).

На сегодня реализованы и поставлены в несколько детских технопарков «Кванториум» двух- и трёхместные системы на основе ра-

бочей станции «Эльбрус 801-РС» с 16 Гб ОЗУ, двумя видеокартами Radeon RX570 или тремя — RX470, многоместный режим работы которых обеспечивается штатными средствами дистрибутива Альт Общественное 9.0 для Эльбрус.

Пакетная база

Исходный x86-вариант дистрибутива состоит из базовой части, дающей львиную долю объёма установки, и дополнительно устанавливаемых групп пакетов сообразно уровням ОУ (от детского до вуза), а также для учительского рабочего места и комплекта серверных приложений.

Не всё из этого получается собрать для архитектуры e2k на данном этапе — где-то упёрлись в архитектурнозависимый код, который надо портировать, где-то (как с некоторыми ruby- или python-пакетами) — в необходимость просто приложить больше усилий для синхронизации репозитория р9 на основных архитектурах и на интересующей нас вторичной.

В целом на сейчас ситуация со степенью воспроизведения оригинала выглядит вполне удовлетворительной: различия описываются скорее списком исключений, а основной комплект на месте и работает.

Например, не включены:

1. chromium (для переноса требуется ориентировочно год);
2. зависящие от webkit/webengine части KDE Edu (отчасти перекликается с предыдущим пунктом);
3. puppet (требуется подтягивание ruby-стека в части актуальности и полноты);
4. LibreOffice-still (работы по переносу шестой версии уже начались, а пока пятая);
5. freecad/qcad, qgis, openscad, scilab, synfigstudio, gambas (надо портировать);
6. fpc (и lazarus), freebasic (кодогенераторы), swi-prolog;
7. fritzing (собирается, сбоит при запуске);
8. wine (не имеет физического смысла);
9. nvidia_glx (проприетарные драйверы), bumblebee;
10. monodevelop (требуется работы по нативной сборке портированного mono).

По некоторым программам пришлось довольствоваться предыдущими версиями.

При этом есть и другие прикладные пакеты, которые не входят в текущий выпуск образовательного дистрибутива и на x86, но тем не менее уже востребованы — например, `arduino`, где одна из нужных программ (`arduino-builder`) зачем-то написана на Go; по ним также ведётся работа.

Литература

- [1] <http://altlinux.org/education>
- [2] <http://altlinux.org/elbrus>
- [3] <http://altlinux.org/X11/DualSeat>
- [4] <http://0x1.tv/20170128J>
- [5] <http://0x1.tv/20180930D>
- [6] <http://old.computerra.ru/197895>
- [7] <http://sdelanounas.ru/blogs/119017>

Александр Иванюк, Виктор Кирсанов, Роман Лопухов,
Тимофей Черствов

Москва, Московский технический университет связи и информатики (МТУСИ), Общественно-государственное объединение «Ассоциация документальной электросвязи» (АДЭ), кафедра «Технологии электронного обмена данными» (ТЭОД).
<http://teod.rans.ru/>

Апробация типового отечественного модуля изучения интернет-технологий

Аннотация

В докладе представлена архитектура типового отечественного модуля изучения интернет-технологий (ТОМИИТ) и результаты его апробации на базовой кафедре Ассоциации документальной электросвязи в Московском техническом университете связи и информатики. В состав ТОМИИТ входит программно-аппаратный комплекс, разработанный и

произведенный в России, комплект учебно-методических материалов. Рассматриваются возможности ТОМИИТ для изучения межсетевого экранирования с использованием свободного программного обеспечения.

Типовой отечественный модуль изучения интернет-технологий (ТОМИИТ) разработан и внедрен для подготовки ИТ-специалистов, способных и готовых использовать интернет-технологии, реализованные на отечественных программно-аппаратных средствах с использованием свободного программного обеспечения.

Программно-аппаратный комплекс разработан с использованием вычислительных комплексов Эльбрус и операционной системы Альт.

Специально подготовленный комплект учебно-методических материалов для использования в учебных курсах по изучению интернет-технологий содержит теоретическую и практическую части.

В теоретической части курса рассматриваются:

- технологии сети Интернет (стек сетевых протоколов, адресация и маршрутизация, принципы взаимодействия сетей связи и обмена трафиком, DNS и др.);
- основы регулирования сети Интернет на национальном и международном уровне;
- общие сведения об архитектуре процессоров «Эльбрус», открытой модели разработки ПО, свободных лицензиях, процессах разработки и внедрения новых сервисов;
- гуманитарные аспекты работы в сети Интернет.

Практическая часть курса содержит методические указания по выполнению 24 лабораторных практикумов, которые позволяют студентам получить навыки работы на отечественных программно-аппаратных средствах.

В докладе рассматривается общая структура лабораторного стенда, а в качестве примера, — реализация практикума по межсетевому экранированию.

ТОМИИТ является первым в России типовым решением для образования, полностью реализованным на отечественных технологиях, прошедшим апробацию и готовым к тиражированию.

Литература

- [1] Кремер А.С., Мальянов С.А., Малюк А.А. Обеспечение доверия и безопасности при использовании ИКТ учебное пособие. М. ОГО АДЭ, 2017.
- [2] Основы технологии сети Интернет: учебно-методическое пособие / МТУСИ; Кремер А.С., Иванюк А.В. М.: ФОП, 2019. 200 с.
- [3] Основы технологии сети Интернет: лабораторный практикум / МТУСИ; Кремер А.С., Иванюк А.В. М.: ФОП, 2019. 376 с.

Иванюк Александр, Малиночкин Вячеслав

Москва, Московский технический университет связи и информатики (МТУСИ), общественно-государственное объединение «Ассоциация документальной электросвязи (АДЭ), кафедра «Технологии электронного обмена данными» (ТЭОД)

<http://www.teod.rans.ru>

Лабораторный комплекс по изучению базовых сервисов сети Интернет с использованием свободного программного обеспечения

Аннотация

В докладе представлены этапы реализации лабораторного комплекса по изучению базовых сервисов сети Интернет с использованием свободного программного обеспечения: от постановки задачи до апробации в высшем учебном заведении. Обсуждается опыт внедрения и рассмотрены пути дальнейшего развития и расширения лабораторного комплекса.

В состав типового отечественного модуля изучения интернет-технологий (ТОМИИТ) для преподавания дисциплины «Основы технологий сети Интернет» входит лабораторный практикум, включающий 24 практических занятия.

В докладе рассмотрены лабораторные работы, посвящённые базовым сервисам сети Интернет, включенные в практикум, а также работы, которые планируются включить в дальнейшем. В качестве основных базовых сервисов в лабораторный практикум включены: dns-инфраструктура, веб-сервера и электронная почта.

Свободное программное обеспечение и отечественная аппаратная платформа

ТОМИИТ работает на базе отечественной аппаратной платформы Эльбрус, а все лабораторные работы выполняются с использованием свободного программного обеспечения на ОС Альт.

На лекционных занятиях по дисциплине «Основы технологий сети Интернет» особое внимание уделено свободному программному обеспечению: преимущества и недостатки, свободные лицензии, репозиторий. На практических и лабораторных занятиях студентов знакомят с конкретными приложениями (утилитами), учат работать с репозиторием.

Исследование dns-инфраструктуры

В работе по исследованию dns-инфраструктуры с использованием свободного ПО обучающиеся, применяя утилиты dig, nslookup и анализатор трафика Wireshark, изучают структуру dns, типы записей и формат запросов/ответов.

Настройка веб-сервера Apache

В работе по настройке веб-сервера Apache студенты настраивают доступ к серверу по HTTP и HTTPS.

Для подключения по HTTPS серверу создаётся самоподписанный сертификат с помощью утилиты openssl.

Дальнейшее развитие

В настоящее время разрабатывается цикл лабораторных работ, посвященный электронной почте, национальным доменам и универсальному принятию.

Студенты с помощью почтового приложения ThunderBird и анализатора трафика Wireshark, будут изучать работу электронной почты и почтовых протоколов SMTP, POP и IMAP.

В работах, посвященных национальным доменам и универсальному принятию, обучающиеся будут работать с преобразованием текста в рунусоде, осваивать особенности использования символов национальных алфавитов в названии сайтов и электронной почты.

Выводы

1. ТОМИИТ — первое в стране решение, построенное на базе отечественных программно-аппаратных средств, готовое к тиражированию в учебные заведения.
2. Разработан лабораторный комплекс по изучению базовых сервисов сети Интернет с использованием свободного ПО.
3. Планируется дальнейшее развитие и доработка лабораторного комплекса.
4. ТОМИИТ прошёл апробацию с февраля по май 2019 года в Московском техническом университете связи и информатики на базовой кафедре АДЭ «Технологии электронного обмена данными».

Литература

- [1] Кремер А. С., Мальянов С. А., Малюк А. А. Обеспечение доверия и безопасности при использовании ИКТ: учебное пособие. М. : ОГО АДЭ, 2017.
- [2] Основы технологии сети Интернет: учебно-методическое пособие / МТУСИ; Кремер А.С., Иванюк А.В. М.: ФОП, 2019. 200 с.
- [3] Основы технологии сети Интернет: лабораторный практикум / МТУСИ; Кремер А.С., Иванюк А.В. М.: ФОП, 2019. 376 с.

Иванюк Александр Викторович, Магафуров Марс Рафикович
Москва, Московский Технический университет связи и информатики (МТУСИ), общественно-государственное объединение «Ассоциация документальной электросвязи (АДЭ), кафедра «Технологии электронного обмена данными» (ТЭОД)
<http://www.teod.rans.ru/>

Лабораторный комплекс для изучения протоколов динамической маршрутизации с использованием свободного программного обеспечения

В докладе рассматривается лабораторный комплекс для изучения основных протоколов динамической маршрутизации с использованием свободного программного обеспечения. Комплекс апробирован в рамках учебной дисциплины «Основы технологий сети интернет» на базовой кафедре АДЭ в МТУСИ. Цель создания комплекса — знакомство и получение необходимого набора знаний и практических навыков при работе с настройкой сети, используя протоколы маршрутизации RIPv2, OSPF, BGP, OSPFv3, RIPng, на отечественных программно-аппаратных средствах: ВК Эльбрус и ОС Альт.

В докладе рассматривается лабораторный комплекс для изучения протоколов динамической маршрутизации, созданный на свободном программном обеспечении (СПО) в рамках реализации проекта «Типовой отечественный модуль изучения интернет-технологий» (ТО-МИИТ).

Лабораторный комплекс функционирует на отечественных программно-аппаратных средствах: операционной системе Альт и рабочих станциях Эльбрус 101-РС. В докладе представлены пять работ лабораторного практикума, которые позволяют студентам изучить основные протоколы динамической маршрутизации для сетей на основе IPv4 и IPv6: RIPv2, OSPF, BGP, OSPFv3, RIPng. Все лабораторные работы реализованы на основе пакета свободного программного обеспечения quagga.

Для выполнения каждой лабораторной работы создается кольцевая сетевая топология путем соединения четырех рабочих станций Эльбрус между собой с помощью виртуальных локальных сетей на коммутаторе.

С помощью утилит WireShark, ping, traceroute студенты изучают функционирование протоколов динамической маршрутизации.

Выводы

1. Разработан лабораторный комплекс для изучения протоколов динамической маршрутизации с использованием СПО.

2. Лабораторный комплекс для изучения протоколов динамической маршрутизации прошел апробацию на базовой кафедре АДЭ «Технологии электронного обмена данными» в МТУСИ в рамках учебной дисциплины «Основы технологий сети интернет» и готов к тиражированию.

Литература

- [1] Кремер А. С., Мальянов С. А., Малюк А. А. Обеспечение доверия и безопасности при использовании ИКТ: Учебное пособие. М. : ОГО АДЭ, 2017.
- [2] Основы технологии сети Интернет: учебно-методическое пособие / МТУСИ; Кремер А. С., Иванюк А. В. М. : ФОП, 2019. 200 с.
- [3] Основы технологии сети Интернет: лабораторный практикум / МТУСИ; Кремер А. С., Иванюк А. В. М. : ФОП, 2019. 376 с.

И.А.Хахаев

Санкт-Петербург, СПбГЭТУ «ЛЭТИ»

<https://etu.ru/>

Снова об оценке электронных образовательных ресурсов

Аннотация

Обсуждаются характеристики систем дистанционного обучения (СДО) со свободными лицензиями с позиций решения задачи организации межфакультетских и межвузовских курсов с применением систем web-конференций. Исходя из задачи формируются требования и вводится система показателей, имеющих бинарную оценку (0/1).

Варианты оценки электронных образовательных ресурсов (ЭОР) и систем дистанционного обучения (СДО, LMS) уже обсуждались ранее [1], в продолжение этой работы был предложен вариант обработки экспертных оценок [2].

Однако поскольку СДО является инструментом решения задач организации он-лайн и дистанционных курсов, то имеет смысл рассматривать различные системы именно в аспекте пригодности для решения поставленных задач.

В рейтингах СДО, отобранных по назначению (academic/education) и ценовой политике (free), приведенных на ресурсах [3] и [4], можно получить более 50 вариантов. Такое количество вариантов трудно изучить в разумное время.

Можно воспользоваться результатами работы по отбору лучших СДО со свободной лицензией. Ведется и регулярно обновляется рейтинг 20 лучших свободных LMS [5].

В качестве дополнительного фильтра можно исключить из рейтинга [5] все варианты, в которых предоставляется только сервис, а также производные системы (пересборки Moodle или Open edX). Для оставшихся систем основные (маркетинговые) характеристики приведены в таблице 1.

Таблица 1: Основные характеристики СДО со свободными лицензиями

СДО	Платформа	Кол-во плагинов	Зарег. установок	Зарег. РФ	Партнёров в мире	Партнёров РФ
Moodle	Apache/ Nginx, PHP, SQL	1600+	102000	3000+	86	1
Chamilo	Apache, PHP, MySQL/ MariaDB	N/A	N/A	N/A	8	0
Open edX	Nginx, Python/ Django, MongoDB+ MySQL	2	2000+	6 (5+(1in8))	57	0
Canvas	Ruby, PostgreSQL	0	17	1 (МИСиС)	678	0
Forma	Apache/ IIS, PHP, MySQL/ MariaDB	non-free	N/A	N/A	15	0
ILIAS	Apache, PHP, MySQL	N/A	215	2	77	0
OpenOLAT	Java (Tomcat), PostgreSQL/ MySQL/ MariaDB	0	44	0	12	0
Opigno (Drupal LMS)	Apache/ Nginx, PHP, SQL	185	8000	N/A	N/A	N/A
ATutor	Apache, PHP, MySQL/ MariaDB	50+	N/A	N/A	N/A	N/A
Percolate	Apache, PHP +JS, PostgreSQL	0	N/A	N/A	N/A	N/A
Sakai	Java (Tomcat), Oracle/ MySQL	0	230+	10+ (2012)	N/A	N/A

СДО Opigno не является самостоятельным проектом, а представляет собой CMS Drupal со специализированными модулями, поэтому из дальнейшего рассмотрения исключается.

Таблица 2– Продолжение

Требование	Moodle	Chamilo	Open edX	Forma	ILIAS	ATutor	Percolate	Sakai
Интеграция с ВКС	1	1	1	1	1	1	0	0
Выбор ВКС	1	0	0	1	1	1	0	0
Ролевая модель	1	1	1	1	1	1	1	1
Обратная связь	1	1	1	1	1	1	1	1
Возможности wiki	1	1	1	1	1	1	0	1
Геймификация	1	0	0	0	0	1	0	0
Прозрачный API	1	1	1	1	1	1	1	1
Поддержка в РФ	1	0	0	0	0	0	0	0
Итого баллов	9	6	6	7	7	8	4	5

Таким образом, максимальная оценка получается для СДО Moodle, которая оказывается наиболее пригодной для решения поставленной задачи. Если снять требование наличия официальной поддержки в Российской Федерации, то можно на равных рассматривать СДО Moodle и ATutor.

Литература

- [1] Хахаев И.А., Зарубин В.С., Шаповалов Е.Н. Возможность сравнительной оценки электронных образовательных ресурсов. XI конференция «Свободное программное обеспечение в высшей школе»: Материалы конференции / Переславль, 30–31 января 2016 года. М.: Альт Линукс, 2016. С. 23–25.
- [2] Хахаев И.А., Шаповалов Е.Н., Вайнтрауб А.И., Зиновьев К.Г. Сравнительная оценка электронных образовательных ресурсов. Известия СПбГЭТУ «ЛЭТИ», № 1-2017, С. 12–15. СПб: СПбГЭТУ «ЛЭТИ», 2017.
- [3] Best LMS Software. Электронный ресурс. Режим доступа: <https://www.capterra.com/learning-management-system-software/>.
- [4] Learning Management Systems. Software Advice, Inc. Электронный ресурс. Режим доступа: <https://www.softwareadvice.com/lms/>.
- [5] Christopher Pappas The 20 Best Learning Management Systems (2019 Update). Электронный ресурс. Режим доступа: <https://elearningindustry.com/the-20-best-learning-management-systems>.

Ставцев Роман Геннадьевич
Москва, АО «БАЙКАЛ ЭЛЕКТРОНИКС»

Отечественные аппаратные платформы «Байкал» для СПО

Аннотация

Обзор аппаратных платформ «Байкал» и опыт использования с СПО

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

Использование свободного программного обеспечения на уроках по созданию WEB-документов в колледже (на основе опыта ГБПОУ МО «Щёлковский колледж» СП8)

Аннотация

В докладе рассмотрен собственный 9-летний успешный опыт работы с системой по созданию WEB-документов в ГБПОУ МО «Щёлковский колледж». Приводятся причины, обуславливающие необходимость, если не полного, то хотя бы частичного, перехода на свободное программное обеспечение.

Общие сведения

Для создания Web-страниц и сайтов в целом в ГБПОУ МО «Щёлковский Колледж» СП-8 используется следующее (только свободное) программное обеспечение:

1. Редактор для работы с HTML кодом Seamonkey.
2. Bluefish editor — это основная программа в нашей учебной деятельности и достаточно мощный редактор HTML кода.

Отличительные черты Bluefish:

- HTML-редактор Bluefish стремится формировать чистый и не перегруженный лишними тегами web-документ;
 - Быстрота. Bluefish работает весьма быстро (в том числе и на нетбуке);
 - Многопоточная работа с удалёнными файлами посредством GVFS (своеобразная виртуальная ФС — альтернативная, это виртуальная файловая система, созданная как альтернатива для GnomeVFS). Также работает с FTP, SFTP, HTTP, HTTPS, WebDAV, CIFS и т. д.
 - 2.1. Мощнейшая функция поиска и замены, а также поиск и замена в файлах не только в корневом каталоге проекта, но и на диске компьютера в целом.
 - 2.2. Автодополнение тегов и их автозакрывание, что значительно ускоряет работу над проектом.
 - 2.3. Поддержка множества кодировок. Bluefish работает по умолчанию с UTF8, однако способен сохранить проект в любой другой кодировке.
3. Локальный WEB-сервер LAMP. LAMP — аббревиатура, которая обозначает набор (комплекс) серверного программного обеспечения, широко используемый во Всемирной паутине. LAMP назван по первым буквам входящих в его состав компонентов:
- Linux — операционная система на базе ядра Linux (имеется опыт построения учебного процесса на базе дистрибутивов Ubuntu, Альт Образование 8, в настоящее время используется Альт Образование 9);
 - Apache — веб-сервер;
 - MySQL — СУБД;
 - PHP — язык программирования, используемый для создания веб-приложений.
4. Для разработки дизайна веб-страниц используется замечательный редактор изображений Gimp.
5. Filezilla — очень хороший FTP-клиент и двухпанельный файловый менеджер.
6. Italc (Veyon) — ПО для управления классом.

Организация работы с ОС на базе ядра Линукс в Щёлковском филиале ГБПОУ МО «Щёлковский Колледж»

В ГБПОУ МО «Щёлковский Колледж» уже около 9 лет довольно успешно на компьютерах студентов используется операционная система на базе ядра Линукс. За это время были апробированы несколько видов таких ОС, но окончательный выбор был сделан в пользу ОС Альт Образование (<https://www.basealt.ru/products/alt-education/>). Он обусловлен рядом причин:

1. Более высокая стабильность в работе, особенно при использовании локальной сети, а также в сети Интернет.
2. Альт Образование — это дистрибутив, «заточенный» под нужды именно образовательных учебных заведений и содержит набор всего необходимого ПО.
3. Входит в Единый реестр российских программ для электронных вычислительных машин и баз данных (<https://reestr.minsvyaz.ru/>).
4. Стоимость проприетарного программного обеспечения настолько высока, что может сильно превышать стоимость техники как таковой.
5. Отсутствие необходимости постоянной борьбы с вирусным программным обеспечением по причине почти полного отсутствия вредоносного программного обеспечения для операционной системы на базе ядра Линукс и гораздо более высокой организации системы безопасности внутри ОС.
6. Легальное использование программ без дополнительных финансовых затрат на острие новейших разработок в области программного обеспечения.
7. В силу того, что на занятиях мы занимаемся разработкой WEB-документов, а удаленные серверы, как правило, работают под управлением Линукс-систем, то знакомство с файловой структурой операционной системы и с принципами её работы позволяет лучше понять работу удалённого сервера.

Все компьютеры образовательного учреждения объединены в локальную сеть, в состав которой входят как компьютеры под управлением

различных версий Windows, так и под управлением Linux, при этом не имеют трудностей при сетевом обнаружении друг друга.

Следует заметить, что операционная система включает в себя все необходимые приложения не только для организации обучения в рамках WEB-дизайна и WEB-программирования, но и при этом исключена возможность установки студентами чего-то нежелательного, так как подавляющее большинство программ устанавливается из репозитория.

Выводы

По результатам апробации операционной системы Альт Образование был сделан однозначный вывод, что процесс обучения в колледже довольно успешно реализуется на основе операционной системы на базе ядра Линукс, с использованием свободного программного обеспечения, включенного в состав дистрибутива (или доустановленного из репозитория).

Большинство используемых проприетарных программ имеют свободные или бесплатные аналоги, которые имеют Линукс-версии и позволяют их использовать в работе для решения каждодневных задач пользователей, что для небольших предприятий с ограниченным бюджетом очень большое подспорье.

Кроме того, в условиях государственных образовательных организаций, которые, с одной стороны, обладают достаточно скудным бюджетом, а с другой — обязаны соблюдать лицензионное законодательство, все это позволяет эффективно решать вопросы не только финансового, но и учебного характеров.

Федосов А.Ю., Маркушевич М.В., Краснов А.Н.

Москва, Российский государственный социальный университет, ГБОУ школа № 1352, ГБОУ школа № 1352

О концепции комплекта учебников информатики для общеобразовательной школы, базирующегося на свободном программном обеспечении

Аннотация

В настоящей работе авторы отмечают насущную необходимость и предлагают концепцию разработки современного комплекта учебников информатики для российской общеобразовательной школы, базирующегося на исключительно свободном программном обеспечении.

В настоящее время обучение информатике в отечественной общеобразовательной школе опирается на использование современных учебно – методических комплектов, разработанных различными авторскими коллективами (Босова Л.Л., Босова А.Ю., Поляков К.Ю., Еремин Е.А., Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В) [1].

При всех существующих преимуществах рассматриваемых учебных пособий необходимо отметить тот факт, что их авторы в большинстве своем не акцентируют внимания педагога на том, к какому типу относится используемое при реализации соответствующей методики программное обеспечение, что может привести к серьезным материальным затратам на приобретение необходимой лицензии (в случае применения проприетарного программного обеспечения), как со стороны государства, так и со стороны родителей учащихся при использовании программного обеспечения для выполнения практических домашних заданий.

Кроме того, большинство авторов существующих учебников по информатике исходят из принципа инвариантности методики преподавания относительно программного обеспечения, т. е. утверждают, что сама методика и качество обученности информатике не зависит от типа выбранного преподавателем программного обеспечения.

Также в педагогической литературе отмечается, что использование в учебном процессе общеобразовательной школы программного обеспечения того или иного типа может оказывать влияние на отдельные аспекты духовно-нравственного воспитания учащихся, так

как данный выбор затрагивает категорию свободы, имеющую отношение к высшим духовным ценностям [5].

Авторы настоящей статьи в свою очередь исходят из гипотезы о том, что достижение образовательных результатов по информатике зависит от типа выбранного для построения методики обучения программного обеспечения (свободное или проприетарное) и формулируют в качестве системообразующего принцип выбора свободного программного обеспечения, применение которого позволит достичь более высоких образовательных результатов (личностных, метапредметных и предметных) учащихся. Данная точка зрения базируется на следующих преимуществах использования свободного программного обеспечения в учебном процессе общеобразовательной школы, сформулированных авторами ранее в [2]:

1. бесплатность лицензий на использование свободного программного обеспечения;
2. кроссплатформенность большей части свободного программного обеспечения;
3. эквивалентность школьного и домашнего программного обеспечения учащихся и преподавателей;
4. относительно низкие системные требования, характерные для большинства программ, относящихся к свободному программному обеспечению;
5. комьюнотарная идеология свободного программного обеспечения;
6. открытый код свободного программного обеспечения;
7. стимулирование соблюдения авторских прав учащимися;
8. отсутствие негативного влияния на духовно – нравственную сферу учащихся.

Таким образом, ввиду серьезных различий между идеологиями свободного и проприетарного программного обеспечения, приводящих в свою очередь к необходимости разработки специальной методики обучения информатике, построенной на основе применения свободного программного обеспечения, можно говорить о необходимости разработки соответствующего современного комплекта учебников по информатике, в котором в качестве программных инструментов применяется исключительно свободное программное обеспечение

Целью настоящей статьи, кроме констатации имеющейся в настоящее время потребности отечественной педагогике в учебнике по информатике, базирующемся на СПО, является также намерение авторов предложить новые подходы к проектированию современного учебника с точки зрения формирующейся информационно-коммуникационной образовательной среды, ориентированной на достижение новых образовательных результатов.

Как отмечает ряд авторитетных ученых, роль и место учебника в новой образовательной среде в ближайшие годы не должна принципиально измениться — он останется ключевым объектом этой среды. Такая убежденность исследователей вытекает из функций учебной книги как носителя содержания образования и средства организации учебной деятельности обучающихся [3]. Но, тем не менее, та же группа исследователей указывает на необходимость корректировки существующих в настоящее время подходов к созданию и использованию школьных учебников, а в ряде случаев — формирования новых принципов их разработки.

Разделяя приведенную выше позицию, авторы настоящей статьи считают возможным предложить следующие принципы, которые могут быть использованы при разработке нового комплекта учебников по информатике для общеобразовательной школы, базирующегося на свободном программном обеспечении, вытекающие в том числе из результатов современных исследований в области когнитивных технологий в области образования [4]. К таковым можно отнести:

1. Популяризация создания и использования различных сообществ учащихся, направленных на организацию взаимопомощи и оказания поддержки школьниками друг друга в ходе осуществления ими учебной деятельности;
2. Акцент на максимально возможной визуализации предлагаемого к изучению материала;
3. Широкое применение разговорного стиля в тексте учебника;
4. Глубокое погружение в изучаемый материал, рассмотрение изучаемого объекта с различных сторон;
5. Захват внимания учащихся;
6. Наличие эмоционального содержания в изложении учебного материала;

7. Большое количество практических заданий, реализация принципа обучения в деятельности;
8. Избыточность наиболее важной учебной информации, заключающаяся в многократном повторении в различных внешних формах представления информации.

Авторы были бы благодарны всем заинтересованным в перспективе разработки и издания отечественного комплекта учебников для общеобразовательной школы, базирующегося на свободном программном обеспечении, концепция которого описана в настоящей статье, за любые конструктивные комментарии и рекомендации, и предлагают направлять их на электронную почту mihael1171@mail.ru.

Литература

- [1] Федеральный перечень учебников, рекомендованных к использованию при реализации программ общего образования [Электронный ресурс] – Режим доступа: <http://fpu.edu.ru/fpu/> (дата обращения: 22.12.2019 г.)
- [2] Федосов А.Ю., Маркушевич М.В. Актуальные вопросы построения методики преподавания темы «векторная и растровая графика» в основной школе на основе свободных графических редакторов OpenOffice.org Draw и GIMP // Информатика в школе. № 8. 2019. С. 9 – 18;
- [3] Абдуразаков М. М., Зенкина С. В., Ниматулаев М. М. Как реализовать основную образовательную программу на основе трех требований федерального государственного образовательного стандарта // Информатика и образование. 2019. №10. С. 5–12;
- [4] Бэрри П. Изучаем программирование на Python / Пол Бэрри ; [пер. с англ. М.А. Райтман]. – Москва : Эксмо, 2020. – 624 с;
- [5] Федосов А.Ю., Маркушевич М.В. Применение свободного программного обеспечения в учебном процессе как фактор духовно-нравственного воспитания учащихся // Информатика в школе. № 4, 2018. С. 40-43.

Иван Шишунов

Талдом (Московская область, МОУ Квашёнковская СОШ)

Проект: Творческое объединение «Квашёнки Пикчерс»: создание компьютерной анимации с помощью свободного программного обеспечения
<http://kvashenki.ru/pictures/>

Использование свободного программного обеспечения в общеобразовательной организации на базе центра формирования цифровых и гуманитарных компетенций «Точка роста»

Аннотация

В докладе излагается опыт установки операционной системы Альт Образование 9 на оборудование, поставленное в школу в рамках федерального проекта «Точка роста», а также анализируются возможности использования на этом оборудовании свободного программного обеспечения в урочной и внеурочной деятельности.

Введение

В этом учебном году наша школа стала участником федерального проекта «Точка роста» [6], в рамках которого мы получили новое современное оборудование с предустановленным комплектом программного обеспечения на базе операционной системы *Windows 10*. Было принято решение о проведении работ по выявлению возможности организации образовательного процесса с использованием этого оборудования на базе операционной системы из Единого реестра российских программ для электронных вычислительных машин и баз данных [3] на основе ядра *Linux*. В качестве такой операционной системы была выбрана *Альт Образование 9* [5], включающая в себя широкий набор образовательного программного обеспечения.

Установка Linux

В рамках решаемой задачи необходимо было сохранить на компьютерах из поставки предустановленную операционную систему *Windows 10*.

Общая последовательность действий для каждого из тестируемых компьютерных устройств была такой:

— В самом начале работы необходимо было сжать том (самый большой раздел жёсткого диска) для выделения свободного дискового пространства, что было успешно проделано штатными средствами системы *Windows* (*Управление* → *Управление дисками*).

— После этого выполнялась загрузка с установочной флэшки *Альт Образование 9* в режиме *UEFI* и создавались в неразмеченном пространстве жёсткого диска 2 раздела: *efi* с файловой системой *FAT32* объёмом 500 Мб для установки загрузчика *grub*, а также стандартный раздел *Linux* с файловой системой *ext3/4*.

Результаты установки для каждого устройства получились разными.

Учительский ноутбук-трансформер HP Pavilion x360: в *BIOS* компьютер несколько раз предупреждал о попытке загрузиться с флэшки и даже попросил ввести контрольный код с экрана, но после этого установка прошла успешно.

Ученический ноутбук-трансформер 3Logic Lime ED20PA2 (10 штук): установка прошла успешно, однако сенсор экрана и тачпад не заработали.

На игровой ноутбук *Asus FX505G* (для VR-очков) *Альт Образование 9* установить не удалось, программа установки зависла на этапе сохранения настроек (перед установкой загрузчика). На данном ноутбуке два физических жёстких диска, и диск *WDC* 256 Гб, на котором установлена система *Windows*, на этапе установки не определяется. Попытка установить *Linux* на другой жёсткий диск *HGST* 1 Тб приводит к вышеописанным результатам. Таким образом, проверить работу VR-очков на операционной системе *Альт Образование 9* не удалось.

Интерактивная панель Promethean ActiveBoard оснащена съёмным вычислительным блоком *Giada*, который превращает устройство в огромный моноблок. Здесь при установке *Linux* возникли большие проблемы. Жёсткий диск этого устройства имеет разметку в формате *MBR*, но при этом загрузка операционной системы *Windows* проходит в режиме *UEFI*. После нескольких неудачных попыток установка всё же прошла успешно, помогло создание раздела *efi* в файловой системе *FAT16* (вместо *FAT32*). Далее выяснилось, что *grub* не нашёл загрузчик *Windows*, так что единственным вариантом выбора операционной системы на этом устройстве является смена порядка загрузки в

BIOS. Кроме того, после установки пришлось восстанавливать загрузку *Windows* с помощью загрузочного диска [2]. При работе в *Linux* сенсор экрана работает, но не реагирует на двойное нажатие, то есть, например, файл с помощью сенсора можно выделить, но нельзя открыть. На официальном сайте *Promethean* [4] можно скачать драйвер устройства *Active Driver* и программное обеспечение *Active Inspire* под *Linux*, но только для *Ubuntu*. На *Alt Linux Wiki* [1] есть статья о том, как собрать пакеты под *Alt Linux* для 7-ой и 8-ой платформ. Следуя данной инструкции, у меня получилось установить только программу *Active Inspire*, но без драйвера устройства *Active Driver*. Ссылки на готовые файлы в конце статьи, к сожалению, оказались битыми (ошибка 404).

Подводя итог, можно сказать, что операционную систему *Альт Образование 9* удалось установить на все компьютерные устройства, кроме одного. Принято решение обратиться в службу технической поддержки за помощью в решении возникших проблем.

Использование свободного программного обеспечения в работе центра

В нашем центре «Точка роста» компьютеры используются в основном для преподавания школьного курса информатики, а также для организации внеурочной деятельности обучающихся по программированию, созданию компьютерной анимации, 3D моделированию и инженерному дизайну.

В Таблице 1 приведен перечень программного обеспечения, используемого на занятиях с обучающимися.

Получены первые результаты апробации

Для использования 3D-принтера *Hercules* достаточно программы, умеющей генерировать G-код 3D-моделей, например, *Slic3r*, которая была установлена с помощью менеджера пакетов *Synaptic*. Работа *Slic3r* была успешно протестирована и ничем не отличается от работы этой же программы в *Windows*. Для создания самих 3D-моделей в операционной системе *Альт Образование 9* есть *Blender*, а для проектирования по чертежам – *FreeCAD*. Сможет ли *FreeCAD* адекватно заменить *Компас 3D*, который мы на данный момент используем для

Программное обеспечение на базе Windows	Программное обеспечение в составе Альт Образование 9	Предмет / внеурочная деятельность
Мой офис, Libre Office, Gimp, Inkscape, Кумир, <i>Pascal ABC NET, Google Chrome, Scratch 3, Python IDLE</i>	<i>Libre Office, Gimp, Inkscape, Кумир, Geany (с компилятором языка Паскаль), Chrome, Scratch 1, Python IDLE</i>	Информатика, кружок внеурочной деятельности «Эрудит» (программирование и подготовка к экзамену по информатике)
Synfig Studio, Audacity, Kdenlive	Synfig Studio, Audacity, Kdenlive	Творческое объединение «Квашёнки Пикчерс»
3D Builder, Компас 3D, Blender, Slic3R	<i>Blender, Slic3R, FreeCAD</i>	Кружок 3D-моделирования и инженерного дизайна

Таблица 1: Перечень используемого ПО на занятиях в центре «Точка роста»

подготовки к участию в соревнованиях *Junior Skills* (компетенция *Инженерный дизайн*) – пока непонятно, будем пробовать.

Протестировав программу создания 2D компьютерной анимации *Synfig Studio* в составе *Альт Образование 9*, которая является базовой в работе творческого объединения «Квашёнки Пикчерс», можно сказать, что она стала работать стабильнее по сравнению с предыдущими платформами и версией для *Windows*. Но вышеупомянутый минус — нерабочий сенсор экрана на ученических ноутбуках, и, соответственно, невозможность рисовать стилусом, — не позволяет в данном виде внеурочной деятельности полностью перейти на свободное программное обеспечение.

Однако стоит отметить, что для освоения школьной программы по предмету «Информатика» свободного программного обеспечения, входящего в состав *Альт Образование 9*, вполне достаточно.

Литература

- [1] AltLinux Wiki [Электронный ресурс] / Интерактивная доска Promethean ActivBoard. — Режим доступа: https://www.altlinux.org/Интерактивная_доска_Promethean_ActivBoard, свободный.
- [2] Remontka.pro [Электронный ресурс] / Восстановление загрузчика Windows 10. — Режим доступа: <https://remontka.pro/windows-10-bootloader-fix/>, свободный.
- [3] Единый реестр российских программ для электронных вычислительных машин и баз данных [Электронный ресурс]. — Режим доступа: <https://reestr.minsvyaz.ru/reestr/>, свободный.
- [4] Официальный портал поддержки компании Promethean [Электронный ресурс] / Страница загрузки программного обеспечения Active Inspire на официальном сайте Promethean. — Режим доступа: <https://www1.support.prometheanworld.com/ru/download/activinspire.html>, свободный.
- [5] Официальный сайт компании Базальт СПО. Страница «Операционное система Альт Образование» [Электронный ресурс]. — Режим доступа: <https://www.basealt.ru/education/>, свободный.
- [6] Распоряжение Министерства просвещения Российской Федерации от 01.03.2019 года № Р-23 [Электронный ресурс]. — Режим доступа: <https://legalacts.ru/doc/rasporjazhenie-minprosveshchenija-rossii-ot-01032019-n-r-23-ob-utverzhdenii/>, свободный.

Андрианов Андрей Михайлович, Михеев Андрей Геннадьевич
Москва, НИУ МИЭТ; НИТУ МИСиС, ООО «Процессные технологии»

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

Привлечение студентов к разработке свободной системы RunaWFE Free в рамках проектного командного обучения

Аннотация

В настоящее время существует много препятствий, мешающих студенту выйти на уровень своих способностей: Сложно выбрать для изучения современные технологии, на которых можно эффективно учиться, т.к. количество IT-технологий очень велико. Кроме учебы студенту требуется работать, чтобы получать средства к существованию. При этом у него есть сильный соблазн освоения простых технологий, позволяющих получать быстрый доход, значительный с точки зрения студента, но небольшой по сравнению с тем, что позволяют его потенциальные способности. Что приводит к отсутствию у студента времени на получение фундаментального образования.

В этих условиях мы предложили студентам в рамках входящего в учебный процесс проектного командного обучения (ПКО) принять участие в проекте разработки свободного ПО.

Существующая проблема

Современному студенту сложно выбрать изучаемые технологии, т.к. количество существующих IT-технологий очень велико, а количество реализующих их программ - в несколько раз больше. Большинство современных технологий быстро возникают и так же быстро теряют популярность. Студенты выбирают их для изучения случайно, без какой-либо системности. Таким образом, студенту оказывается сложно получить хорошее базовое IT-образование.

Кроме того, студенту необходимо получать средства к существованию, т.к. сейчас на стипендию прожить нельзя. При этом, начиная со второго — третьего курса, у него есть возможность быстро освоить относительно простую технологию и, используя ее, начать работать в одной из IT-компаний. Это дает студенту какие-то деньги, но приводит к отсутствию у него времени на получение фундаментального образования. А отсутствие фундаментального образования не даст

ему возможности в будущем разрабатывать по-настоящему сложные системы.

Если студент после бакалавриата продолжил обучение в магистратуре, то возникает еще одна проблема: Магистратура предполагает научную деятельность. При этом сложно найти IT-компанию, работа в которой требовала бы решения научных задач. Поэтому в магистратуре трудно совмещать с работой написание магистерской диссертации с серьезной наукой. Что приводит к очень высокому отсеву (до 50% студентов отчисляются из магистратуры). Попытки использовать двухлетний «безотзывный» договор на магистерскую практику, встречают понятное сопротивление бизнеса т. к. такая форма сотрудничества имеет высокие риски для бизнеса (IT-компаний) — компании опасно связывать себя на два года со студентом, которого она плохо знает.

Поэтому предлагается искать темы для ВКР и магистерских диссертаций раньше, чем это традиционно делается — уже в конце 2-го или в начале 3-го курса. Надо искать темы работ у компаний, имеющих задачи, как уровня ВКР, так и уровня магистерских работ. Желательно, чтобы в рамках этих задач ВКР постепенно трансформировались бы в магистерские работы. Лучшими задачами являются идеи ведущего специалиста компании, до решения которых, у него не доходят руки из-за занятости.

Проектное командное обучение

Сдача экзаменов по материалам лекций и выполнение лабораторных работ не дает студенту, уже имеющему практику по изучаемой дисциплине (частое явление для IT-сферы) получить опыт создания законченного продукта, опыт коллективной работы над одной задачей, а также включить результаты в портфолио проектов для будущего работодателя. В этом смысле часто самостоятельная работа над ВКР и Магистерской работой оказывается для студента неожиданно сложной. Поэтому предлагается при изучении нескольких дисциплин разрешить студентам по желанию заменить выполнение лабораторных работ по дисциплине на прохождение ПКО (проектного командного обучения).

В рамках ПКО студенты 3-4 курсов организуются в команды из 2-4 студентов, для каждой команды создается проект для решения сложной задачи, который включает все темы, рассматриваемые в дис-

циплине, а также, возможно, некоторые не рассматриваемые темы. Результат ПКО представляется на открытой защите, куда приглашаются представители IT-предприятий.

ПКО — решение адаптированных к учебному процессу реальных задач в виде коллективных проектов. В идеале ПКО должно быть эквивалентно «опыту работы от 2-х лет» в резюме. Награда за решение задачи ПКО — приравнивание к прохождению собеседования в компании, предоставившей задачу, возможно с несколько большей (на 20-30%) зарплатой, чем после «обычного» собеседования

Требование к задачам:

1. Освоение технологий, предусмотренных программой курса. - Не менее $\frac{2}{3}$ технологий, перечисленных в программе курса (из них, по согласованию с преподавателем, до $\frac{1}{2}$ могут быть заменены аналогами).
2. Наличие исследовательской составляющей
3. Полезность темы работы для ВУЗа и/или внешнего заинтересованного партнера. Тему могут предложить: студент, преподаватель, внешний заинтересованный партнер. Преподаватель осуществляет контроль и модификацию задач с целью их соответствия программе дисциплины. Приоритет получают задачи имеющие в дальнейшем потенциал ВКР/Магистерской работы.
4. Должно быть пройдено четыре обязательных этапа представления работы за семестр и промежуточный контроль активности (раз в неделю или раз в две недели). Команда, не представившая результаты этапа в срок, отправляется выполнять лабораторные с самого начала.
5. Должна быть возможность публичной защиты выполненной задачи (последний этап) с приглашением представителей фирм, которых может заинтересовать тема или команда разработчиков

Ожидаемые результаты:

- Студент часто не знает своих интересов и возможностей, а пробовать на бакалаврской или магистерской работе, обычно уже поздно. ПКО дает ему такую возможность. Студенты находят себе темы для бакалаврской работы и (или) организацию для практики заранее (т.е. до 4-го курса) и, поэтому имеют время для «маневра темой»

- Студенты имеют возможность попробовать проверить/реализовать «свою» тему.
- Студент лучше понимает что ему важнее: результат или процесс, кто он в команде: координатор (лидер), разработчик, демонстратор (продвиженец), тестировщик.
- Есть студенты, которым помощь мешает, а есть - которым помощь нужна в «повышенном объеме». ПКО позволяет гибко работать с этим. Преподаватель имеет больше возможностей (времени) для занятий с теми, кому требуется его помощь, и не «тормозит» усредненными заданиями тех, кто уже достаточно знает и умеет.
- Тема работы может стать стартапом команды, который они совместно затем «допиливают» на базе бакалаврских и магистерских диссертаций. Если тема предоставлена внешним заинтересованным постановщиком из IT отрасли, то компания имеет возможность начать готовить специалиста «под себя»

Для лучшей эффективности ПКО при отборе предприятиями тем для ПКО предлагается исходить из следующих критериев:

1. Идеи/потребности ведущих специалистов до проверки/реализации которых не находилось человеческого ресурса

2. То, что давно хотелось сделать, но не «доходили руки»

У ведущих специалистов всегда много идей и они всегда слишком заняты текущей работой, чтобы их проверять/реализовывать. На проверку таких идей часто бывает недостаточно ресурсов в виде отвлечения других сотрудников.

Но при этом эти идеи имеют нужный для развития хорошего студента потенциал. ПКО позволяет проверить такую идею с разумным риском и наградой по результату. Такие задачи могут сразу иметь потенциал ВКР и магистерской работы.

Нужные «вещи», которые обязательно сделали бы если бы появился хотя-бы маленький свободный ресурс. Такие задачи, это скорее, ВКР и в меньшей степени магистерская работа.

Чем шире список тем/задач ПКО, тем выше шанс у участников процесса: студента, ВУЗа, предприятия найти друг-друга, чтобы достигнуть следующих целей:

1. Получение квалификации (личной)
2. Получение специалиста (проф. и деловые навыки)

3. Решение задачи (своей или «заказной»)
4. Получение команды (под задачу)
5. Создание бизнеса (стартап)

ПКО по задачам свободного проекта RunaWFE Free

В конце января 2019 г. на 14 конференции «Свободное программное обеспечение в высшей школе» авторы настоящего доклада договорились включить задачи проекта RunaWFE Free в число задач для ПКО студентов третьего курса института СПИНТех МИЭТа.

Свободное программное обеспечение удобно для проведения ПКО по следующим причинам:

1. Студентам не требуется получать доступ к документации и коду ПО, которое они модифицируют. - Все выложено в интернет. Любая команда студентов может выполнить задачу и направить пул-реквест в репозиторий на github.com. Далее опытные разработчики проекта знакомятся с пул-реквестом и начинают работать с командой: писать замечания, предлагать решения, проверять требования к качеству кода, тестировать и т.д.
2. Репозиторий является открытым, любой желающий может удостовериться, что данный студент является участником открытого проекта, посмотреть его коммиты.
3. В частности, студенты имеют возможность показать свой код будущему работодателю при устройстве на работу

Для ПКО студентам МИЭТа были предложены следующие задачи:

1. Реализовать проверку наличия возможных бесконечных циклов на схеме бизнес-процесса.
2. Реализовать проверку схем бизнес-процессов на корректность. Схема бизнес-процесса называется корректной, если экземпляр бизнес-процесса, соответствующего данной схеме, обязательно завершится в случае корректного задания всех условий (исключающих шлюзов, обработчиков, посылаемых сообщений и т.п.) и корректного выполнения заданий узлов-действий.
3. Чат участников экземпляра бизнес-процесса.
4. Пилот для web-дизайнера.

При реализации задач предполагалось использовать следующие технологии: Язык — Java + Web (HTML, CSS, JavaScript).

За выполнение задач взялись две команды студентов, но одна из команд не справилась с ПКО и в конечном счете предпочла вместо ПКО выполнить лабораторные работы. Вторая команда, состоящая из двух студентов, взялась за третью задачу «Чат участников экземпляра бизнес-процесса», которую в конце концов выполнила и разместила пул-реквест в репозиторий проекта RunaWFE Free — <https://github.com/processtech>.

В дальнейшем доработка чата трансформировалась из ПКО в официальную производственную практику ВКР с оформлением необходимых документов.

В процессе выполнения ПКО у студентов возникли следующие проблемы: Сложно было прийти к одинаковому пониманию поставленной задачи и распределить между собой обязанности. Потребовалось разобраться в большом объеме кода системы RunaWFE Free. Оказалось сложно работать с документацией. Разработанные функции сначала неудачно называли, поэтому их трудно было обсуждать с основными разработчиками системы. Были проблемы в понимании некоторых моментов работы IDE Eclipse. Несколько разделов первоначальной архитектуры решения оказались плохо продуманы, что привело к проблемам при реализации. Некоторые технологии, которые потребовалось применить (многопоточность, асинхронность, технология Web-Socket'ов, некоторые особенности реляционных баз данных), на момент реализации задачи выходили за рамки учебной программы.

Со стороны компании «Процессные технологии» сложности были следующими: на работу со студентами у сотрудников компании ушло больше усилий, чем изначально планировалось. В частности, трудно оказалось объяснить требования к архитектуре решения, к качеству кода и структурам данных. Несколько раз студентам потребовалась помощь преподавателей. Сроки полностью выдержаны не были — что-то студентам пришлось доделывать в сессию и даже уже после сессии.

Но, в конце концов поставленная цель была достигнута.

Литература

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

- [2] Михеев А. Г., Орлов М. В. Система управления бизнес-процессами и административными регламентами. Программные продукты и системы, № 3, 2011, с. 126–130.
- [3] Отчет о выступлении команды студентов, разрабатывающих «Чат» на XVI конференции разработчиков СПО в г. Калуга: <http://institut-spintex.ru/news/education/my-na-xvi-konferentsii-razrabotchikov-spo.html>

Алексей Ерпелев*, Ольга Позднякова*, Александра Савинкина**, Владимир Симонов*

Москва, * ФГБОУ ВО «Российский государственный социальный университет», ** ФГБУН ГНЦ РФ - Институт медико-биологических проблем РАН

https://github.com/Vladimir28091959/gomeostat_prototip_V4.git

Разработка прототипа электронного прибора «Гомеостат» на базе СПО для использования в учебном процессе

Аннотация

Рассмотрена реализация на современной программно-аппаратной базе с использованием свободного программного обеспечения прототипа разработанного в 60-70-х годах электронного прибора «Гомеостат Горбова Д.Ф.». Гомеостат применяется для оценки сплоченности членов малых групп в критически важных сферах - космонавтика, медицина, спорт, военная сфера и др. Представленный прототип помимо своего прямого назначения (оценки сплоченности членов малых групп), используется в учебном процессе для обучения студентов направлений подготовки «Информатика и вычислительная техника», «Информационные системы и технологии» и ряда других.

На сегодняшний день важным остается вопрос формирования команд, что связано с повышенным вниманием в системе комплексного управления качеством, с усложнением принимаемых стратегических управленческих решений и усложнением задач корпоративного обучения.

Исследование взаимоотношений в малых группах, выделение лидеров, является важной задачей социальной оценки коллектива.

Главным критерием малой группы является ее малая численность. В большинстве исследований число членов малой группы колеблется между 2 и 7. [1]

Совместимость партнеров в команде является одним из ведущих факторов групповой удовлетворенности членов команды.

Такие команды, которые часто взаимодействуют путем экспериментов или групповой деятельности, подвержены проблемам потенциала группового взаимодействия и психологической совместимости людей.

Особые процедуры исследования и измерения, предполагающие использование неких технических устройств — это аппаратные методы, применяющиеся в разных направлениях психологической науки. Аппарат «гомеостат», разработанный Федором Дмитриевичем Горбовым — одна из самых интересных аппаратных методик (рис.1). [3]

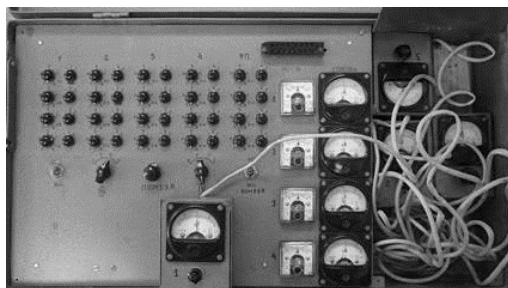


Рис. 1: Внешний вид оригинального прибора «Гомеостат»

Принцип работы аппарата основан на изменении электрического потенциала, что отражалось на шкалах прибора. Персонал, проходящий исследование на «Гомеостате» должен решить общую проблему, в данном случае - установить стрелки всех приборов в определенное (нулевое, т. е. центральное) положение, но при этом участники не могут общаться, и их действия влияют на действия товарищей по группе [2].

Позднее в социальной психологии было создано много подобных аппаратов, и они стали применяться в практике отбора членов группы для выполнения задач в экстремальных условиях, когда нет возможности у группы совместно обсуждать ход выполнения операции [5].

Недостатки прибора:

- прибор изготавливался вручную в нескольких экземплярах;
- для его использования необходимо большое количество проводов;
- габариты прибора. Не предназначен для частого перемещения;
- устарел.

На данный момент существуют разработки, которые взяли за основу принцип работы «Гомеостата».

В компании «ООО НПКФ «Медиком МТД»» был создан тренинг для изучения социально-психологического климата в малой группе. В отличие от «Гомеостата» измерялись физические параметры испытуемых во время решения задачи. Каждый испытуемый должен управлять вращением своего кольца путем нажатия клавиш на клавиатуре. Поворачивая свое кольцо, испытуемый влияет на вращение кольца своих коллег. Задача: построить тоннель напротив шарика так, чтобы он смог закатиться в середину (рис. 2) [4].

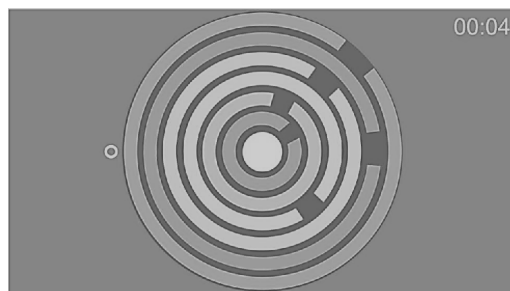


Рис. 2: Интерфейс устройства «ООО НПКФ «Медиком МТД»»

В программе заложено большое количество функций, что делает её дорожке. Для моделирования отдельных функций гомеостата, авторами были разработаны макеты, один из таких макетов представлен на рис. 3.

Указанные макеты содержат ряд органов управления для оценки взаимодействия партнеров, настройки коэффициентов преобразований различных величин и пр. Программный код (СПО) одного из вариантов макета представлен в [6].

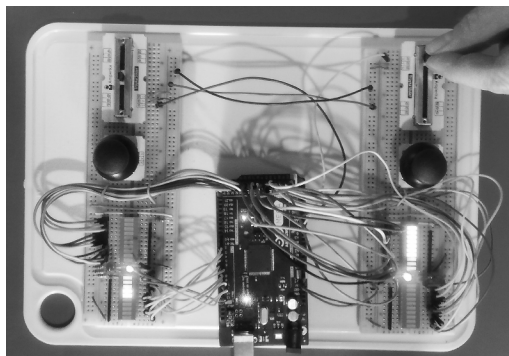


Рис. 3: Макет для отработки отдельных функций гомеостата

Таким образом, можно сделать вывод, что для определения совместимости членов команды актуальна модернизация именно первоначального прибора «Гомеостат», с условием обеспечения в нем только необходимого функционала, во избежание излишней дороговизны прибора.

Литература

- [1] Анцупов А. Я., Ковалёв В. В. Социально-психологическая оценка персонала. Учебное пособие. — 3-е изд. Юнити-Дана. 2013.
- [2] Горбов Ф.Д., Новиков М.Л. Вопросы интегративной оценки групповой активности // Тезисы докладов на II съезде психологов СССР. АПН РСФСР, 1965.
- [3] Ельникова Н.Г, Метод «Гомеостат» в задаче социально-психологического исследования малых групп // Научно-практический электронный журнал Аллея Науки, №10, 2017.
- [4] Журавлев А.Л. Психология совместной деятельности. «Институт психологии РАН». 2015.
- [5] Поляков С. А. Методика социально-психологической оценки малых групп // Молодой ученый. — 2013. [Электронный ресурс]. Режим доступа: <https://moluch.ru/archive/53/7187/> (дата обращения: 18.01.2020).
- [6] Программный код макета отработки отдельных функций электронно-го прибора «Гомеостат» (СПО) [Электронный ресурс]. Режим досту-

па: https://github.com/Vladimir28091959/gomeostat_prototyp_V4.git
(дата обращения: 30.01.2020).

Максим Каторгин*, Никита Михайлов **, Алексей Ерпелев*,
Иван Бакалым*, Вячеслав Улатов*, Владимир Симонов*

Москва, * ФГБОУ ВО «Российский государственный социальный университет», ** Государственное бюджетное профессиональное образовательное учреждение города Москвы «Московский государственный образовательный комплекс» (ГБПОУ МГОК)

https://github.com/Vladimir28091959/Robot_zahvat.git

Разработка прототипа робота-помощника для лиц с ограниченными возможностями здоровья на базе промышленного робота-манипулятора KUKA с электрическим захватом под управлением платформы Arduino и СПО

Аннотация

В последние годы значительное внимание уделяется лицам с ограниченными возможностями здоровья (ОВЗ), существенным этапом здесь является введение ФГОС ОВЗ [1]. Для некоторых из таких лиц (например, для лиц, имеющих диагноз ДЦП, со спастическим тетрапарезом, с сильными гиперкинезами, а также смешанной формой) является проблемой налить стакан воды, а также взять его в руки и выпить, из-за опасения расплескивания. Поэтому указанным лицам необходимо оказание соответствующей помощи.

Настоящее исследование посвящено автоматизации процесса помощи указанным лицам с ОВЗ, а именно, разработке действующего прототипа робота-помощника для наливания стакана воды или иного напитка, перемещения стакана в зону досягаемости лица и пр. Разработка выполнена на базе промышленного робота-манипулятора KUKA [2], на фланце которого размещен электрический захват под управлением одной из микроконтроллерных платформ, например платформы Arduino (функционирует под управлением свободного программного обеспечения (СПО)).

Принцип функционирования разработки состоит в следующем. Заказчик (лицо с ОВЗ) через мобильное приложение заказывает требуемый напиток, а также определяет момент времени, к которому напиток должен быть приготовлен. Манипулятор захватывает бутылку с требуемым напитком, наполняет стакан на подносе и пододвигает поднос в сторону заказчика.

Одной из важных задач такого робота-помощника – не подвергать опасности находящегося рядом человека (не нанести ему механических повреждений, ушибов). Используемый робот-манипулятор КУКА не является коллаборативным, поэтому в представляемом проекте обеспечивается физическая недосыгаемость частей тела заказчика движущимися частями робота КУКА с захватом. С этой целью используется специальный поднос, который пододвигается в сторону заказчика с готовым продуктом.

В разработку заложен один из принципов взаимного позиционирования объектов, прототипом которого является разработка авторов, представленная в [3].

Следует отметить, что представляемая модель является учебным проектом (преследует в первую очередь учебные цели), поэтому не претендует на выполнение всех функций робота-бармена.

Алгоритм работы устройства состоит в следующем.

- получить заказ на тип напитка и момент приготовления;
- в нужное время захватить стакан, перенести его на поднос ближе к краю, со стороны заказчика;
- захватить требуемую бутылку с напитком, наполнить стаканчик, переместить бутылку на свое место;
- пододвинуть поднос со стаканчиком в сторону заказчика;
- дождаться, когда стаканчик будет взят с подноса лицом с ОВЗ;
- передвинуть поднос в исходное положение;
- перевести робот КУКА в исходное положение и привести систему в состояние готовности к приему следующего заказа.

На данном этапе отрабатывались операции по перемещению бутылки, стаканчика и подноса.

В проекте использовалась следующая программная и аппаратная база:

- аппаратная база: Arduino Mega 2560, электрический захват, аналогичный представленному в [4];

- программная база: среда разработки Arduino IDE, язык программирования C++ (Wiring).

Проект (его первоначальный тестовый вариант) успешно был испытан при взаимодействии с реальными лицами с ОВЗ.

Программный код разработки является СПО и доступен на ресурсе [5].

Таким образом, было спроектировано и реализовано устройство оказания помощи лицам с ОВЗ, страдающим ДЦП (спастический тетрапарез, гиперкинетическая форма) - прототип робота-помощника для наливания стакана воды или иного напитка и перемещения стакана в зону досягаемости лица. Указанное устройство также является студенческим проектом и используется в учебном процессе при обучении студентов направлений подготовки, таких как «Информатика и вычислительная техника», «Информационные системы и технологии», «Управление в технических системах».

Литература

- [1] Письмо Министерства образования и науки РФ от 11 марта 2016 г. № ВК-452/07 «О введении ФГОС ОВЗ» [Электронный ресурс]. Режим доступа: <https://www.garant.ru/products/ipo/prime/doc/71254376/> (дата обращения: 18.01.2020).
- [2] Промышленные роботы KUKA [Электронный ресурс]. Режим доступа: <https://www.kuka.com/ru-ru> (дата обращения: 18.01.2020).
- [3] Simonov V.L., Kuzin A.V. Intelligent system for optoelectronic surface vibration testing / Report on the conference ICIAF'03. Proceedings of the 20-th International Congress on Instrumentation in Aerospace Simulation Facilities. Deutsches Zentrum fur Luft- und Raumfahrt e.V. (DLR), Goettingen, Germany. IEEE Catalog Number 03CH37501. ISBN 0-7803-8149-1. No 4.3 (pp.115–117). August 2003. Интеллектуальная оптоэлектронная система для измерений вибраций поверхности объектов (доклад) / 20-я конференция по исследовательской и испытательной технике в аэрокосмической отрасли ICIAF'03: Сб.трудов конф. Август 2003г. Изд-во DLR, Геттинген, Германия. -№ 4.3. С. 115-117.
- [4] Электрический захват [электронный ресурс]. Режим доступа: https://aliexpress.ru/item/32721105267.html?spm=a2g0o.productlist.0.0.51173397KQazns&algo_pvid=ab0ecd7f-6ae7-49e6-9fe8-cbbe2f59eb29&algo_expid=ab0ecd7f-6ae7-49e6-9fe8-cbbe2f59eb29-21&btsid=

7946904c-92ca-472c-ae20-81afff94b7f3&ws_ab_test=searchweb0_0, searchweb201602_8, searchweb201603_55 (дата обращения: 30.01.2020).

- [5] Программный код разработки [электронный ресурс]. Режим доступа: https://github.com/Vladimir28091959/Robot_zahvat.git (дата обращения: 30.01.2020).

Альховская Марина, Симонов Владимир
Москва, ФГБОУ ВО «Российский государственный социальный университет»
https://github.com/amari04/reaction_project.git

Программно-аппаратный комплекс на базе СПО для оценки реакции человека в условиях повышенных нагрузок

Аннотация

Комплексы оценки психологического и психофизиологического состояния человека, а также его реакций занимают значительную роль в современном мире. Многие работодатели прибегают к использованию аппаратных комплексов оценки реакций и состояния человека, чтобы иметь представление о психологическом или психофизиологическом состоянии кандидата, программные комплексы оценки реакций широко используются в спортивных отраслях, как для поддержания рефлексов спортсменов, так и для их улучшения, чем дольше спортсмен тренирует простую реакцию, тем быстрее ее подвижность.

Время реакции — это промежуток времени между стимулом (свет, звук, и т.д.) и реакцией организма.

Время реакции человека ограничено и зависит от вида сигнала-раздражителя, интенсивности, настроенности на принятие сигнала, возраста и сложности реакции индивида, т.е. чем слабее раздражитель, тем дольше человек на него реагирует и наоборот, также зависит от натренированности и настроенности на восприятие сигнала человеком, а также возраста и пола.

Появление в современном мире компьютеров существенно упростило организацию хронометрических исследований. Методики регистрации реакций активно разрабатываются для разных сфер деятельности, таких как спорт, образование, здравоохранение и других. Как

правило, каждая методика основывается на разных типах реакции организма человека.

Для оценки уровня тревожности были произведены мозговые исследования тревожности посредством латентного времени сенсомоторных реакций.

Для оценки сенсомоторной реакции разрабатываются комплексы оценки реакции, основанные на тестах РДО (реакция на движущийся объект), для детей широко используются тесты OZO (проведенные тесты показали ухудшение времени реакции к концу рабочего дня у 5 из 20 испытуемых)

Принцип функционирования разработки комплекса оценки реакции состоит в следующем. Заказчик выбирает режим, испытуемому предоставляется инструкция перед выполнением каждого вида теста, после ознакомления с инструкцией испытуемый приступает к выполнению тестирования.

Практическая значимость программно-аппаратного комплекса состоит, в первую очередь, в исследовании влияния нагрузок на реакцию человека, сборе и получении данных о изменении реакции человека в условиях повышенных нагрузок.

Ниже будет описан комплекс, являющийся учебным проектом (преследует в первую очередь учебные цели), поэтому не претендует на выполнения множества известных методов оценки реакции человека, как например Комплекс «НС-Психотест», Устройство психофизиологического тестирования УПФТ-1/30 «Психофизиолог» и др..

Алгоритм работы устройства состоит в следующем:

- выбор варианта тестирования. Представлено 3 варианта тестирования реакции: 1-ый на основе методики OZO, 2-ой на основе методики РДО и 3-ий оценка реакции на звук (суть методики проста, испытуемому необходимо как можно быстрее среагировать на звуковой сигнал, который подается через случайные промежутки времени);
- после выбора теста испытуемый знакомится с инструкцией и приступает к выполнению выбранного режима тестирования. Для наилучшей оценки реакции рекомендуется пройти все три тестирования;
- по завершении тестов на экран выводятся результаты тестирований.

В проекте использовалась следующая программная и аппаратная база:

- аппаратная база: Arduino Mega 2560, Troyka-Shield, плата расширения с возможностью подключения периферийных Arduino - модулей, RGB-матрица 64x32 тип P3.0, три кнопки Трема-модуль V2.0, один звуковой излучатель («пищалка»);
- программная база: среда разработки Arduino IDE, язык программирования C++ (Wiring).

Разработанный авторами программный код проекта представлен в [4].

Таким образом был спроектирован и реализован комплекс оценки реакции человека в условиях повышенных нагрузок. Указанное устройство является студенческим проектом и используется в учебном проекте при обучении студентов направления, «Информационные системы и технологии».

Литература

- [1] Грибанов А.В., Кожевникова И.С., Нехорошкова А.Н., Джос Ю.С. Латентное время сенсомоторных реакций у детей 10-11 лет с высоким уровнем тревожности. – URL: <https://cyberleninka.ru/article/n/latentnoe-vremya-sensomotornyh-reaktsiy-u-detey-10-11-let-s-vysokim-urovнем-trevozhnosti/viewer> (дата обращения: 25.01.2020).
- [2] Физиологические методы исследования. Методы оценки функционального состояния человека [Электронный ресурс] : метод. указания / И. Ю. Мышкин, Н. Н. Тятенкова, Яросл. гос. ун-т. – Ярославль : ЯрГУ, 2007. – 67 с. – Режим доступа: <https://rucont.ru/efd/206941> (дата обращения: 29.01.2020).
- [3] Репин Д.С., Дегтярев Н.В., Петухов И.В. Микропроцессорный комплекс оценки времени реакции человека на движущийся объект // фундаментальные исследования. – 2011. – № 8-1. – С. 167-171; – URL: <http://www.fundamental-research.ru/ru/article/view?id=26806> (дата обращения: 29.01.2020).
- [4] Программный код проекта (СПО) по оценке реакции человека посредством звукового и зрительного раздражителей; – URL: https://github.com/amari04/reaction_time.git (дата обращения: 30.01.2020).

Дмитрий Климов, Александр Гусин

Пермь, Федеральное государственное бюджетное образовательное учреждение высшего образования «Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Методические рекомендации по организации рабочих мест для студентов и сотрудников ВУЗа на базе Alt Linux 9.0

Внедрение импортозамещения в ВУЗе является сложным комплексным проектом. Данный процесс регламентируются документами вышестоящих организаций[1][2], которые определяют основные этапы перехода на преимущественное использование отечественного программного обеспечения.

Согласно плана организационно-технических мероприятий[3][4], принятого в ВУЗе, проведено обследование существующей информационно-технологической инфраструктуры, в том числе прикладного программного обеспечения автоматизированных информационных систем и иных компонентов программной инфраструктуры. Обследование проводилось при помощи специализированных средств мониторинга и инвентаризации информационно-технической среды вуза. В частности, были использованы возможности системы GLPI (ServiceDesk система GNU GPL 2), её компонента OCS Inventory и специализированных скриптов на языке автоматизации PowerShell для сбора необходимой информации из службы каталога Active Directory сети ВУЗа.

Анализ собранных данных позволил выделить рабочие места, которые позволяют проведение перехода на преимущественное использование отечественного программного обеспечения. Часть рабочих мест (для нашего ВУЗа примерно 68%) требует планирование и проведение дополнительных мероприятий и задач, направленных на устранение факторов и барьеров, препятствующих переходу. Проведена классификация остальных рабочих мест по критерию «переход на использование отечественного офисного программного обеспечения».

Выделены 3 типа рабочих мест, для которых подготовлены типовые конфигурации программного обеспечения базе платформы Alt Linux 9.0 : «Учебный тип» (рабочие места в учебных классах), «Бухгалтер» (рабочие места сотрудников, использующих конфигурации

1С) и «Стандарт» (для использования на рабочих местах сотрудников ПНИПУ).

Конфигурация «Учебный тип» разрабатывалась еще на базе «Альт Образование 8.2». В качестве тестирования, был развернут учебный класс на 13 рабочих мест в 2018/2019 учебном году на кафедре ИТАС ПНИПУ. Функциональные возможности дистрибутива обеспечивают проведение занятий в рамках направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия». Опыт этого внедрения будет учитываться для дальнейшего развертывания учебных рабочих мест уже на базе «Альт Образование 9».

Конфигурация «Бухгалтер» основана на использовании дистрибутива «Альт Рабочая станция 9.0». Основным отличием, этого типа является: настройка работы с конфигурациями 1С, автоматического монтирования сетевых ресурсов для работы системы «Консультант +» и внутренней системы документооборота ИАС «ВУЗ», установкой офисного пакета «Р7-Офис». Выбор пакета «Р7-Офис» обусловлен совместимостью применяемых технологий обработки данных, в том числе электронных документов, отчетных и иных форм, форматов и расширений электронных документов с MS Office.

Конфигурация «Стандарт» основана на использовании дистрибутива «Альт Рабочая станция 9.0». Для коммерческого и некоммерческого ПО под платформу ОС Windows давно существуют свободно-распространяемые аналоги:

Опыт показывает, что принципиальных сложностей у студентов при работе с Alt Linux не наблюдается. Пользователи уже отметили экономное расходование ресурсов компьютера и отсутствие особой необходимости в антивирусном ПО. Длительная эксплуатация ПК не приводит к существенному падению производительности системы, как это неоднократно наблюдалось при загрузке Windows 7.

Вместе с тем переход большинства сотрудников, подразумевает дополнительные мероприятия: по настройке среды работы пользователя (максимально напоминающей Windows), по обучению приемам работы в новой системе, дополнительной настройке уникального программного обеспечения.

Эта информация необходима при исполнении дальнейших этапов перехода на преимущественное использование отечественного программного обеспечения (планирование финансовых ресурсов, достижения целевых показателей).

Назначение компонента	ПО для платформы Windows	ПО в составе ОС «Альт Образование»
Офисный пакет	Microsoft Office	LibreOffice
Электронная почта	MS Outlook	Evolution
Математические расчеты и моделирование	Matlab	Octave, SciLab, wxMaxima
Создание трёхмерной графики	Autodesk 3DS Max	Blender
Выполнение приложений для ОС Windows	ОС Windows	Wine

Литература

- [1] Приказ №335 от 04.07.2018, Минкомсвязи России «Об утверждении методических рекомендаций по переходу органов исполнительной власти субъектов Российской Федерации и органов местного самоуправления муниципальных образований Российской Федерации на использование отечественного офисного программного обеспечения, в том числе ранее закупленного офисного программного обеспечения»
- [2] Приказ №156 от 18.04.2019, Минкомсвязи России «О внесении изменений в приказ Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 20.09.2018 № 486 «Об утверждении методических рекомендаций по переходу государственных компаний на преимущественное использование отечественного программного обеспечения, в том числе отечественного офисного программного обеспечения»
- [3] Приказ №486 от 20.09.2018, Приказ Минкомсвязи России «Об утверждении методических рекомендаций по переходу государственных компаний на преимущественное использование отечественного программного обеспечения, в том числе отечественного офисного программного обеспечения»
- [4] Приказ №1220 от 14.12.2017, Минобрнауки «Об организации в Министерстве образования и науки Российской Федерации работы по переходу на использование отечественного офисного программного обеспечения на период 2017 — 2018 годов и на плановый период до 2020 года»

Андрей Иванов

Туринск, МАОУ Усениновская СОШ

Опыт внедрения и использования ОС Альт в образовательном процессе

АльтЛинукс 4.0, Ubuntu, ^{Аннотация} SymplyLinux, Альт Образование 8.0, Альт Образование 9.0. Проблемы внедрения и их решение. Использование Альт Образования на уроках информатики и в учебном процессе образовательного учреждения. Взаимодействие с документами, созданными в пакете MS Office. Дальнейшие перспективы развития СПО в школе.

Мой опыт внедрения Linux в образовательный процесс начался в 2007 с известного дела Поносова, которое привело к большим планам по внедрению СПО, в частности 21 сентября 2007 и. о. первого вице-премьера Дмитрий Медведев заявил о том, что «в течение ближайших трех лет все отечественные школы должны быть избавлены от пиратских компьютерных программ и перейти преимущественно на отечественные. До конца 2009 г., согласно распоряжению правительства от 18 октября 2007 г. №1447-р, Минобрнауки и Минкомсвязи должны обеспечить внедрение СПО во всех российских школах.

Прошло 12 лет и ничего не изменилось. Школы в большинстве как использовали Windows и проприетарное ПО, так и используют, миллионы бюджетных средств уходят на их оплату. Мы снова слышим обещания, читаем указы, новые документы). «А воз и ныне там».

Когда ПО удобно и устраивает пользователя, то он его возьмёт и купит, а если при работе возникает много проблем, то программы не будут брать, даже под соусом «зато отечественное и бесплатное». Начинать переход на СПО необходимо конечно с системы образования, чтобы школьники и студенты, будущие пользователи не были привязаны к связке Windows+MSO. Долгое внедрение СПО в школе натывается на проблемы, которые и хочется описать.

Установка ОС и поддержка оборудования

АльтЛинукс 4.0 в отличие от Windows требовал больше знаний и создавал много проблем при установке, особенно это касалось уста-

новки драйверов. Поэтому все эксперименты с СПО проводились в кабинете информатики, компьютеры в учебных классах не были задействованы.

Большую помощь в работе оказал сайт <http://freeschool.altlinux.ru>. В 2008 году МАОУ Усениновская СОШ стала победителем национального проекта «Образование», на полученный грант были закуплены компьютеры с установленной ОС Windows Vista, одной из неудач MS, что предопределило дальнейшую работу с СПО.

Несколько старых компьютеров были отправлены в библиотеку с установленной Альт.Линукс 4.0, где и работают до сих пор, только была немного увеличена память и установлена SimplyLinux. В кабинете информатики из-за возникающих проблем с Windows Vista и неудовлетворенностью Альт.Линукс была установлена второй ОС Ubuntu.

Непрерывная борьба с Windows, ежегодное требование оплаты «Первой помощи» от Microsoft, окончание срока поддержки, привели к решению полностью отказаться от Windows Vista в кабинете информатики. В 2017 году была установлена SimplyLinux, а в 2018 году установлена Альт Образование 8.0 KDE, в 2019 — Альт Образование 9.0 KDE. Если в SimplyLinux нужно было доустанавливать программы для учебного процесса, то в Альт Образование всё необходимое уже имеется при установке дистрибутива. Был сомнения по поводу работы KDE на 2 Гб оперативной памяти, но ОС работает хорошо, без «тормозов».

Из коробки ставится достаточное для уроков информатики ПО: офисный пакет, графические пакеты, системы программирования, в том числе среда Кумир, обработка звука и видео, браузеры. Установка ОС прошла успешна с помощью USB-образа. В связи с тем, что многие образовательные ресурсы уходят в он-лайн, этого более чем достаточно для успешной работы. Проблем и непонимания у учащихся при работе с ОС не возникает.

Сложность установки ОС уже в прошлом, лучше поддержка дополнительного оборудования, хотя в каждом конкретном случае надо уже при покупке учитывать имеющиеся драйвера, но это не критично, как 10 лет назад. То есть производители идут навстречу потребителям Linux. Подводя итог по внедрению Альт Образование в компьютерном классе, отзыв только положительный.

Использование программного обеспечения

ПО в основном ориентированно на Windows (Электронные учебники в том числе). Издательства, связанные с образованием, обязательно это отмечают.

Можно конечно пробовать запускать это ПО в эмуляторах, виртуальных средах, но как это себя поведет, нужно смотреть отдельно в каждом конкретном случае. И так может случиться, что необходимая и нужная программа учителю не запускается в Linux, то его не убедить переходить на эту ОС. Но с учетом того, как было сказано ранее, обучение уходит в сеть, то данный пункт будет терять свою значимость в будущем.

Зависимость от офисного пакета MS Office

Для перехода на СПО в учебных кабинетах это один из значимых факторов. Пусть сейчас выдвинуты планы на переход отечественного ПО и СПО, но большинство пользователей, в том числе и педагоги, работают в MSO. Пользователю важно, чтобы его документы открывались без изменений и редактировались правильно в других офисных пакетах. Сейчас в школе используется пакет libreoffice 6.2, пока проблем с чтением MSO не обнаружено, может быть это только сложные эл.таблицы и презентации, но для учебного процесса это не значимо.

Заключение

Внедрение Linux в образовании процесс долговременный и требующий поддержки государства. Основы уже заложены единым реестром российских программ для ЭВМ и баз данных <https://reestr.minsvyaz.ru/reestr/>, принятые приказы и постановления федеральными органами РФ.

Наша школа в 2019-2020 учебном году будет увеличивать долю СПО . Сейчас это 16 компьютеров из 48 (30%), желательно довести до 90%.

Стас Фомин

Udaff — русский пиктографический Python. От элементарных алгоритмов до гомоморфного шифрования

Аннотация

Автор часто встречал изобретения «специальных языков для обучения программированию», кривых и убогих, с жалкой инфраструктурой и поддержкой, отрезанных от мирового мейнстрима программирования и обучения, при этом навязываемых в школах и ВУЗах. Иногда необходимость таковых обосновывается требованиями «русскоязычности» ключевых слов.

Предлагается шире использовать Python — язык, годный от обучения дошкольников до глубокого профессионального использования во многих областях. А если кому-то нужна именно русскоязычность ключевых слов — туда ее можно добавить, что и проделал автор, не нарушив совместимости. Кроме русскоязычности добавлены и пиктограммы, эмоджи — возможно это тоже поможет в обучении начинающим, а возможно это пригодится и взрослым — для компактизации описаний нетривиальных алгоритмов.

Проблема

В русскоязычном пространстве наблюдается странный эффект изобретения «русскоязычных» языков программирования, и попыток навязать их обучающимся в Школе и ВУЗе.

См. например Кумир [1], SLang [2] (не путать с еще одной самоделкой — СЛанг [3])... множество их.

С одной стороны, вроде как основания есть — из-за специфики высшего образования в РФ (бесплатное образование, оплачивается ВУЗам МинОбром подушевым образом, отчислять невыгодно, и даже запрещено [4]), в ВУЗы на околопрограммистские специальности попадает множество немотивированных и функционально необразованных кадров, не способных понимать даже текст с десятком ключевых слов на английском. С другой стороны, возможно успех таких платформ типа 1С именно этим и обусловлен, да и есть немало энтузиастов, которые считают, что русификация ЯП — полезна [5], и учить на русифицированных языках эффективно [6].

Однако проблема всех этих самоделок в том, что язык — это не только синтаксис грамматики в BNF на полстраницы, а это инфраструктура:

- Редакторы, IDE, поддерживающие 100500 удобных фиш, не говоря уже о обязательной «построчной» отладке.
 - Разработчики «Русских ЯП» пытаются делать некие подобию IDE, благо сейчас это можно слепить из каких-нибудь готовых компонентов (модуль текстового редактора с подсветкой, MDI интерфейс с менюшками) — но все что получается, это скажем прямо, уровень 90x, и сравнивая это с бесплатным и open-source «швейцарским ножом» Visual Studio Code, (не говоря уже о коммерческих IDE) хочется только плакать от жалости.
- Инфраструктура пакетов (сами библиотеки, пакетные менеджеры), 100500 пакетов для решения всего скучного и типового, не говоря уже о нетривиальных платформах (быстрый вход в разработку игр, математические методы и AI и т.п.)
- Правильные концепции и парадигмы языка, проверенные десятилетиями обкатки на миллионном комьюнити профессионалов.
- Возможность получения профессиональных НАВЫКОВ, вшитых на уровень костного мозга, которых можно применить в профессиональной разработке для решения реальных проектов. Чтобы стартовав с элементарных алгоритмов и простых подделий, можно было эволюционировать в профессионала.

Но если «показать что-то про программирование на русском», на уровне «операторы-ветвление-цикл-функция-рекурсия» для совершенно левых людей (которым максимум в 1С в жизни придется что-то подправить) — допустимо, то совращать «девственно» чистых школьников кривыми поделиями, отрубая им прямой выход к реальной разработке и отбивая желание программировать («пробовал ваше программирование на К... » — ничего не работает, тормозит, криво, неудобно, долбайтесь сами), как минимум неэтично, хотя к сожалению, уголовно ненаказуемо.

Кстати, иногда изобретают «еще одну платформу для обучения» даже не для русификации языка, а для, скажем так, «реанимации стюардессы», типа паскаля, но проблемы остаются те же — какое-то свое подмножество языка, унылые среды разработки, все сбоку от майнстрима и сообщества [7].

Возникает дилемма — как бы убить двух зайцев — пользуясь одной платформой дать возможность неодаренным и негодным студентам и школьникам [8] «попробовать программирование», а продвинутым, тут же, в том же классе, на том же софте, дать возможность уйти в отрыв и стать настоящими профессионалами.

Решение

Итак, уже есть Python, идеальный язык для обучения, рожденный как язык для обучения, десятилетиями использующийся от везде — от уровня младших школьников, до профессионалов, как в программировании, так и в куче научных областей — будь то астрономия, биоинформатика, статистика... везде.

Знание его полезно, если не сказать необходимо, даже тем, кто не программист, если деятельность хоть как-то интеллектуальна (да, даже если экономист-юрист уровня выше чем «за рубль покупаем, за три продаем, на эти два процента и живем»).

К нему есть куча IDE, включая прекрасную поддержку даже в бесплатном и свободном Visual Studio Code (не нужно изобретать страшные велосипеды [9]), есть 100500 пакетов, платформы для написания всего — игр, десктоп и вебприложений, даже мобильной и ИОТ разработки... Впрочем, все это банально и очевидно.

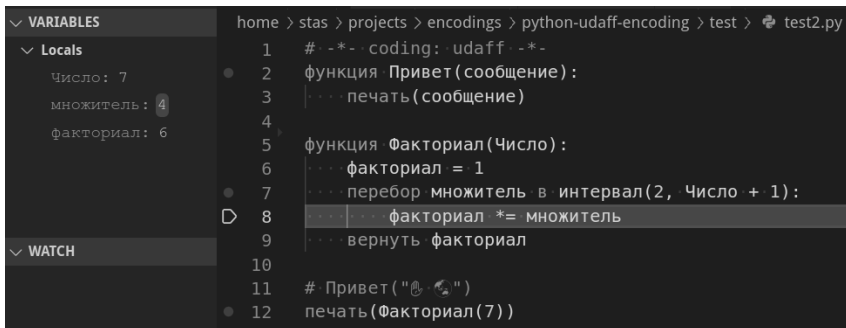
Наша идея — добавить еще «маленькую лесенку» снизу, разрешив писать ключевые слова и базовые функции на русском, для тех, кто вот только стартует, не потеряв никаких возможностей Python.

Для этого можно воспользоваться идеей «пользовательских кодировок», перекодирующих файл с программным кодом при открытии. В результате, несколько десятков ключевых слов Python приводятся к каноническому английскому виду, их понимает и язык, и отладчики и т.п. А обучающийся имеет полную возможность, по мере изучения, постепенно заменить «русские аналоги» их оригинальными ключевыми словами, и с улыбкой (но без ненависти) забыть свои самые первые программы.

Английский	Русский
and	и
as	как
assert	проверить

Английский	Русский
break	прервать
class	класс
continue	продолжить
def	функция
del	удалить
elif	ежели
else	иначе
except	случись
exec	выполни
finally	наконец
for	перебор
from	из
global	глобальное
if	если
import	подключить
in	в
is	суть
lambda	лямбда
not	не
or	или
pass	ничего
print	печать
raise	паника
return	вернуть
try	пробовать
while	повторять
with	пусть
yield	вернуть
range	интервал

Лично я (Стас Фомин), не считаю, что это необходимо. На мой взгляд лучше так выучить эти пару десятков ключевых слов Python на английском, и отсеять «электорат» с уровнем IQ меньше веса. Но идя путем «выбора меньшего зла» (*toss a coin to...*), я уверен, что этот подход лучше изобретения кривых велосипедов от скучающих преподавателей, которые будут калечить поколения школьников и студентов. То есть, я был бы рад, если все это не пригодились при



```
home > stas > projects > encodings > python-udaff-encoding > test > test2.py
1  #-*-coding:udaff-*-
2  функция Привет(сообщение):
3  ... печать(сообщение)
4
5  функция Факториал(Число):
6  ... факториал = 1
7  ... перебор множитель в интервал(2, Число + 1):
8  ... факториал *= множитель
9  ... вернуть факториал
10
11 # Привет("@_☺")
12 печать(Факториал(7))
```

Рис. 1: Пример отладки факториала.

обучении, но при встрече с изобретателем очередного «национального языка», у меня будет куда его конструктивно послать.

С другой стороны, может это пригодится где-то, где Python используется как DSL для какой-нибудь бизнес-логики — короткие функции, которых должны править и вычитывать специалисты в предметной области, может это пригодится и там.

Более того, можно привлечь к программированию, совсем, так сказать, правополушарных людей, и использовать вместо идентификаторов функций и переменных значки emoji.

Если серьезно, то можно использовать и имеющуюся в юникоде «математику», и символы-эмодзи, для того, чтобы компактифицировать реальные работающие алгоритмы, для преподавания или верификации, особенно нетривиальных алгоритмов (автор пробовал описывать блокчейны и гомоморфное шифрование).

Ну а название «udaff», отсылает к популярному времен начала Рунета ресурсу, который прекрасно иллюстрировал идею, что буквоедство и грамотность не так важны, как суть и контент, а иногда такое «грязное языковое хакерство», как кстати, в предложенном решении, даже весело. Кстати, тут несложно сделать и поддержку разных версий «падонкоффского языка».

Литература

[1] Доклады о языке Кумир, <http://0x1.tv/Kumir>

```

# -*- coding: udaff -*-
подключить hashlib

класс 𐄂():
    функция __init__(мое, 𐄂, 𐄃, 𐄄, 𐄅):
        мое.𐄂 = 𐄂
        мое.𐄃 = 𐄃
        мое.𐄄 = 𐄄
        мое.𐄅 = 𐄅
        ничего

    функция 𐄆(мое):
        𐄇 = hashlib.sha256()
        функция 𐄈(𐄉):
            wtf = str(𐄉)
            𐄇.update(wtf.encode('utf-8'))
            ничего

        𐄇 (мое.𐄂)
        𐄇 (мое.𐄃)
        𐄇 (мое.𐄄)
        𐄇 (мое.𐄅)

    вернуть 𐄇

```

Рис. 2: Часть работающего «русифицированного» блокчейна

- [2] «Обучающая среда по программированию на базе СПО (Валерий Лаптев, OSEDUCONF-2019)», <http://0x1.tv/20190127E>
- [3] «Проект СЛанг — текущее состояние и перспективы (Алексей Канатов, SECR-2017)», <http://0x1.tv/20171021CA>
- [4] «Отчислять отстающих запрещено», <https://youtu.be/LDdgdKI20cU?t=1351>
- [5] «Русификация языка – это отдельная большая тема, и её польза лично мне очевидна», <http://www.0x1.tv/20191205AD#comment-4763683278>
- [6] «учить на русифицированных языках эффективно», <https://youtu.be/LDdgdKI20cU?t=1503>

- [7] «Несколько причин забыть PascalABC.Net», <https://habr.com/ru/post/417229/>
- [8] «Ваши студенты олигофрены?», <https://youtu.be/LDdgdKI20cU?t=1144>
- [9] «IDE для изучения Python (Николай Попов, OSEDUCONF-2015)», <http://0x1.tv/20150124G>

Д.А. Слинкин

Шадринск, ФГБОУ ВО Шадринский государственный педагогический университет

Проект: Проект RubiRobot <http://rubirobot.ru/>

Проект RUBIROBOT: управления роботами Lego Mindstorms EV3 с Использованием языка программирования Free Pascal.

Аннотация

Проект RubiRobot раскрывает возможности языка программирования Free Pascal в создании программ для робототехнической платформы Lego Mindstorms EV3. Разработка программ возможна как на контроллере, так и на хостовой системе средствами кросс-компиляции. Проект включает в себя объектно-ориентированную программную библиотеку поддержки внешних и внутренних устройств платформы, более 15 модулей и 30 демонстрационных примеров, документацию на русском языке, консольные и графические утилиты, образ операционной системы Debian Linux проекта ev3dev для загрузки на контроллере EV3. В качестве хостовой операционной системы проект использует ALT Linux седьмой, восьмой или девятой платформ. Основная область применения проекта — обучение робототехнике школьников среднего и старшего звена, обучение образовательной робототехнике студентов педагогических колледжей и вузов, подготовка к участию в робототехнических олимпиадах.

Ключевые слова: Lego Mindstorms EV3, RubiRobot, образовательная робототехника, программирование, Free Pascal

Робототехническая платформа Lego Mindstorms EV3¹ (в дальнейшем — EV3) прочно вошла в образовательный процесс школ, кружков

¹Знакомство с EV3: <https://www.lego.com/ru-ru/mindstorms/about-ev3>

детского творчества, педагогических колледжей и вузов. Высокая стоимость конструктора компенсируется надежностью аппаратных решений и развитой образовательной экосистемой, включающей в себя большой объем методических материалов, курсов для учителей и спонсирование робототехнических олимпиад. Продвижению платформы также способствует полностью открытая программная прошивка, в основе которой лежит операционная система Linux и столь же открытая спецификация на платформу и периферийные устройства². Штатная среда программирования LEGO MINDSTORMS Education EV3 позволяет достаточно эффективно начать изучение программирования с младшего школьного возраста, но при этом обладает рядом недостатков, один из которых — отсутствие в качестве платформы исполнения операционной системы Linux. Данный факт стал одной из причин создания множества альтернативных средств программирования EV3, среди которых следует выделить leJOS³, представляющий собой минимальный дистрибутив Linux с запущенной виртуальной машиной Java, и ev3dev⁴ — полноценный дистрибутив Linux Debian, с возможностью использования любых языков программирования, доступных для ОС Linux. Именно ev3dev лег в основу свободного авторского проекта RubiRobot⁵. Это единственный на сегодняшний день проект, использующий язык Free Pascal⁶ для программирования EV3. С учетом широкого распространения различных диалектов языка Pascal при обучении программированию в российских школах, проект RubiRobot позволяет эффективно актуализировать знания, полученные на уроках информатики.

Проект состоит из нескольких компонентов:

1. RubiRobotLib. Это программная библиотека, включающая в себя более 15 модулей для поддержки встроенных возможностей и периферийных устройств EV3, более 30 демонстрационных примеров различного уровня сложности. Обеспечивает объектно-ориентированный доступ ко всем штатным датчикам и моторам робототехнической платформы, звукогенератору, дисплею, кнопкам, батарее, цветоинди-

²MINDSTORMS EV3 Developer Kits – Support — LEGO Education [Электронный ресурс]. URL: <https://education.lego.com/en-us/support/mindstorms-ev3/developer-kits>

³leJOS. Java for LEGO Mindstorms: <http://www.lejos.org>

⁴ev3dev is your EV3 re-imagined: <http://www.ev3dev.org>

⁵О роботах и не только: <http://rubirobot.ru>

⁶Open source compiler for Pascal and Object Pascal: <http://freepascal.org>

кации и т. д. Поддерживает возможность параллельного исполнения кода. Составной частью библиотеки является русскоязычная документация и консольный менеджер, обеспечивающий коммуникацию между хостовым компьютером и EV3 по протоколу SSH.

2. *ev3dev-RubiRobot*. Это модифицированный образ операционной системы Debian Linux проекта *ev3dev*, с установленным компилятором Free Pascal. Образ обеспечивает поддержку русской локали и графических шрифтов, очищен от посторонних сервисов, оптимизирован по скорости загрузки, содержит библиотеку *RubiRobotLib* для обеспечения разработки непосредственно на контроллере EV3.

3. *RubiRobotCross*. Это инсталлятор, предназначенный для выполнения на седьмой, восьмой и девятой платформах операционной системы ALT Linux⁷, в том числе — на дистрибутивах ALT Linux, входящих в реестр российского программного обеспечения⁸. Устанавливает и конфигурирует библиотеку *RubiRobotLib*, *RubiRobotManager* (графический менеджер доступа к EV3), кроссбиблиотеки *ev3dev*. Установленный набор ПО позволяет разрабатывать программы на хостовом компьютере с использованием кросскомпилятора *FreePascal*, загружать и запускать их на EV3, что на порядок ускоряет цикл разработки по сравнению с программированием непосредственно на контроллере. Кросскомпилятор *FreePascal* автоматически создается из исходного кода компилятора *Free Pascal* во время работы инсталлятора.

Проект *RubiRobot* развивается более 2-х лет и активно применяется для подготовки школьников и студентов к участию в робототехнических соревнованиях различного уровня^{9 10 11 12}. Различные аспекты использования проекта раскрываются в его электронной документации¹³ и в авторских печатных изданиях [1,2,3,4]. В настоящее время готовится релиз версии 0.2.6 проекта, одной из важных особенностей которого является поддержка фото и видео-съемки с использованием веб-камеры, подключаемой к USB-порту EV3, с сохранением

⁷ALT Linux (дистрибутив Linux): [https://ru.wikipedia.org/wiki/ALT_Linux_\(дистрибутив_Linux\)](https://ru.wikipedia.org/wiki/ALT_Linux_(дистрибутив_Linux))

⁸Единый реестр российских программ для электронных вычислительных машин и баз данных: <https://reestr.minsvyaz.ru/>

полученных материалов в файлах и «живой» видеотрансляцией на дисплее контроллера.

Литература

- [1] Подготовка турниров по робототехнике: практико-методический аспект: монография / Д.А. Слинкин, В.Е. Евдокимова, В.М. Гордиевских, Д.М. Бельков, И.Н. Слинкина, М.Е. Козловских; Шадр. Гос. пед. Ун-т. — Шадринск: ШГПУ, 2019. — 120 с.
- [2] Слинкин Д.А. Использование языка программирования Free Pascal и программной библиотеки RubiRobotLib для управления роботами на платформе LEGO MINDSTORMS EV3 // Информатика в школе. — 2018. — №7(140). с. 8–12.
- [3] Слинкин Д.А. От Среды разработки LEGO MINDSTORMS Eeducation EV3 к программной библиотеке RUBIROBOTLIB // Образовательная робототехника: перспективы роста: материалы Всерос. конф., 15 марта 2019 г. / Шадр. гос. пед. ун-т; отв. ред. В.Е. Евдокимова. — Шадринск: ШГПУ, 2019. — 156 с. с.126–132
- [4] Слинкина В., Слинкин Д.А. Образовательная робототехника: основы взаимодействия между наставником и командой // Информатика в школе. 2019. № 4 (147). С. 8-16.

⁹13 декабря 2019 года состоялся открытый областной турнир по робототехнике «ШГПУ: история в деталях» | VK-группа RubiRobot: https://vk.com/rubirobot?w=wall-174353824_16

¹⁰Команда «Нолидей» заняла 1 место в категории «Любитель» среди студентов на соревновании «Гонки роботов», прошедших 7 декабря 2018 года в Курганском Государственном Университете | VK-группа RubiRobot: https://vk.com/rubirobot?w=wall-174353824_8

¹¹Юные робототехники взяли золото на чемпионате России | Молодёжный портал Зауралья: https://prospekt45.ru/1_reg_news/34239/

¹²RubiRobotLib в сказочном коворккачестве | VK-группа RubiRobot: <https://vk.com/@rubirobot-cover-2019-01-21>

¹³Релиз документации библиотеки RubiRobotLib, формат PDF/A | Проект RubiRobot: <http://rubirobot.ru/files/src/rubiro.manual.archive.pdf>

Пустыгин А.Н.
Челябинск

О разработке стандартов анализа исходных текстов программного обеспечения на материалах свободного ПО

Аннотация

Феномен программного обеспечения с открытым исходным текстом открывает исследователю широкие перспективы изучения алгоритмов и использования данных в большом объеме доступных проектов

Текст программного модуля на исходном языке построен неформально и является синтаксически правильной нотацией алгоритма на языке программирования. Согласно действующим стандартам ЕСПД, исходный текст должен снабжаться пакетом программной документации, это увеличивает трудоемкость изготовления программного продукта, однако обеспечивает сопровождение в течении жизненного цикла.

Количество языков программирования растет, возникают модификации. Сведения о числе естественных языков и языков программирования приблизительны: число естественных языков составляет от 6000 до 7000 [3], число языков программирования с вариантами по разным источникам — от 2500 до 3000 [4,5,6], 8512! [7,8].

Существует мнение, что количества естественных языков и языков программирования уже сравнимы, и если количество живых естественных языков имеет тенденцию к сокращению [3], то тенденция развития языков программирования пока состоит в устойчивом росте. По материалам [9] эта зависимость похожа на линейную, хотя критерии отбора языков программирования при анализе явно не названы.

Тенденция «раздувания программного обеспечения» [10,12] может считаться одной из проблем прикладной информатики. Увеличение размера ПО и как следствие памяти выгодно и разработчикам массового ПО и изготовителям вычислительной техники массового применения.

На основании ряда источников в одной системе координат приближённо отражены в логарифмическом масштабе кривые роста примерно до 2012 года:

- числа слов в википедии [13],
- число оригинальных сигнатур вредоносных программ по данным Лаборатории Касперского[14],
- число оригинальных сигнатур вирусов по данным компании Semantec[15],
- объём потребляемой оперативной памяти Microsoft Windows [11],
- объём информации в internet[16],
- число строк ядра linux & freebsd[17,18].

Помимо общих выводов о похожих скоростях роста некоторых из зависимостей при экстраполяции в диапазонах, дополняющих до интервала 1995-2012 г.г. (тонкие линии на рисунке), можно заключить, что темп рост объема ПО на модели сигнатур вирусов (по двум независимым источникам) совпадает и хорошо согласуется с ростом содержания википедии. Главным выводом из представленных зависимостей является общая тенденция экспоненциального роста объемов информации как на естественных языках, так и на языках программирования. Это обстоятельство не оставляет никаких сомнений в необходимости автоматизации процесса извлечения знаний из исходных текстов для противодействия порабощению человечества информацией, которую не представляется возможным воспринимать.

Если принять за базу объём трудозатрат для изучения исходного текста Windows95, то объём трудозатрат по решению аналогичной задачи для NT2000 увеличивается в 8 раз, а для Windows 7 — в 125 раз, что делает такую задачу практически не решаемой без инструментальных средств исследования исходных текстов. Если предположить, что характер рассматриваемых зависимостей не претерпит качественных изменений, можно спрогнозировать контрольные цифры для каждой из величин к 2020 году, и эти величины весьма велики.

Построение эквивалентных представлений (ЭП) исходных текстов позволяет подвести единую методическую базу для процесса анализа, в этом случае ЭП могут принимать форму технических комментариев [1]. Построение ЭП выгодно выполнять в два этапа, на первом их которых формируется промежуточное представление в виде сериализованного дерева разбора, являющееся текстом на языке разметки [2]. Если это представление стандартизовано, то дальнейший анализ может выполняться универсальными инструментами построения ЭП.

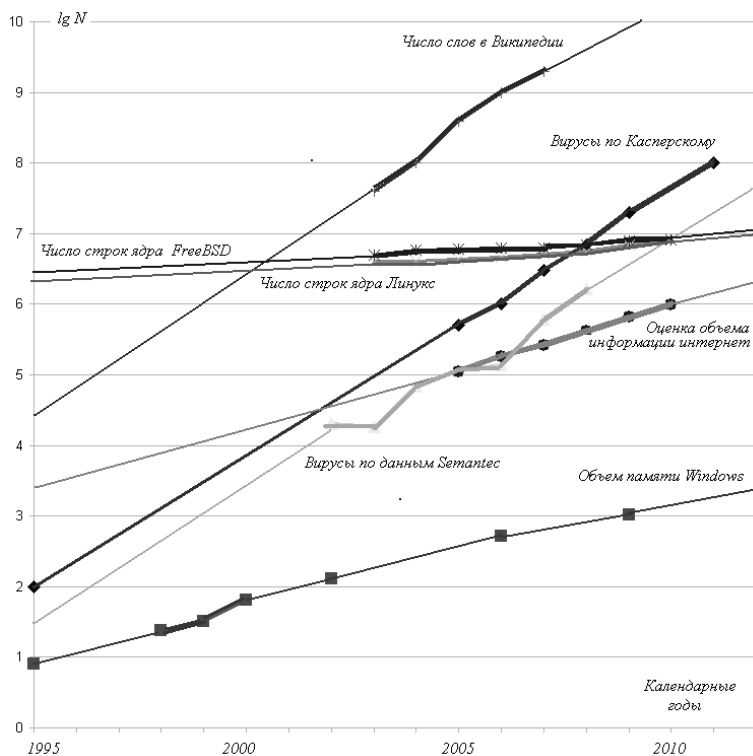


Рис. 1: Приближённые графики роста объёмов информации в сфере программного обеспечения.

Таблица. Предварительный состав подлежащих разработке стандартов анализа исходных текстов программного обеспечения (первой очереди)

На основании прототипов инструментов анализа исходных текстов открытого ПО в опытных работах построения эквивалентных представлений созданы спецификации стандартов, расширяющих существующую методическую базу программной документации.

Наименование проекта стандарта	Назначение стандарта
1. ГОСТ Р «Информационные технологии. Общие требования к программным средствам формирования технических комментариев исходных кодов программ»	Определяет состав требований к программным средствам формирования технических комментариев, касающихся назначения средства, состава программных конструкций, подлежащих комментированию, порядка формирования промежуточных эквивалентных представлений, объема комментария и др.
2. ГОСТ Р «Информационные технологии. Основные этапы формирования машинных комментариев исходных кодов программ, формы машинных комментариев, способы получения»	Определяет состав требований к процессу последовательного формирования технических комментариев, касающихся этапов преобразования исходного текста, наборов эквивалентных представлений исходного текста, подлежащих комментированию, обеспечения навигации как между эквивалентными представлениями, так и между эквивалентными представлениями и исходным текстом и др.
3. ГОСТ Р «Информационные технологии. Форматы представления машинных комментариев исходных кодов программ»	Регламентирует форматы промежуточных и конечных эквивалентных представлений технических комментариев, обеспечивая единство эквивалентного представления независимо от языка исходного текста
4. ГОСТ Р «Информационные технологии. Общие требования к мультиязыковой поддержке средств формирования технических комментариев исходных кодов программ»	Определяет состав требований к языковой среде формирования технических комментариев, состав словарей национальных языков, которые могут подключаться к автоматическому комментатору, требования к интерфейсам

Литература

- [1] Энциклопедия «SEO: Поисковая Оптимизация от А до Я» Бесплатный учебник по поисковой оптимизации и раскрутке сайтов. Комментарии (программирование) [http://www.seobuilding.ru/wiki/Комментарии_\(программирование\)](http://www.seobuilding.ru/wiki/Комментарии_(программирование)) — 2010
- [2] Расширяемый язык разметки XML. <http://ru.wikipedia.org/wiki/XML> — 2011
- [3] А.Е. Кибрик База естественного человеческого языка и ее основные параметры <http://www.dialog-21.ru/dialog2011/materials/html/3.htm> — 2011
- [4] Antula: студия вэб-дизайна <http://www.antula.ru/power-language.htm> — 2011

- [5] B. Kinnersley Collected Information On About 2500 Computer Languages, Past and Present. <http://people.ku.edu/nkinners/LangList/Extras/langlist.htm> — 2011
- [6] Языки программирования <http://www.ikiznanie.ru/ru-wz/index.php/> — 2011
- [7] http://ru.wikipedia.org/wiki/Язык_программирования
- [8] И.Н.Бекман Компьютерные науки. Курс лекций. Лекция 8. Языки программирования – profbeckman.narod.ru/Комп.files/Lec8.pdf — 2011
- [9] Хронология языков программирования http://ru.wikipedia.org/wiki/Хронология_языков_программирования — 2011.
- [10] E. Perratore et al., «Fighting Fatware», Byte, Vol. 18 No. 4 Apr. 1993, pp. 98-108.
- [11] https://ru.wikipedia.org/wiki/Раздутое_программное_обеспечение
- [12] Никлаус Вирт Долой «жирные» программы Открытые системы №06, 1996. <http://www.osp.ru/os/1996/06/179017/>
- [13] Антивирусный прогноз погоды: облачно <http://www.infosecurity.ru/cgi-bin/mart/arts.pl?a=101001> — 2011
- [14] Полноценные хранилища свободных знаний — Википедия и Проект Гутенберг <http://rus-linux.net/lib.php?name=/MyLDP/freesoft/impossible-2.html>
- [15] «Лаборатория Касперского»: экскурсия и взгляд изнутри http://www.thg.ru/business/kaspersky_tour/print.html
- [16] Антивирусные вендоры ищут выход из технологического тупика http://www.anti-malware.ru/antivirus_trends
- [17] Семенов Ю.А. (ГНИЦ ИТЭФ) Направления и тенденции развития ИТ-технологий <http://book.itep.ru/4/7/resources.htm>
- [18] Количество строк кода http://ru.wikipedia.org/wiki/Количество_строк_кода
- [19] Почти объективно на тему «чем FreeBSD лучше Linux» <http://eax.me/freebsd-vs-linux/>

Лукин В. Н., Чернышов Л. Н.
Москва, Московский авиационный институт

Система онлайн-тестирования с открытым кодом

Аннотация

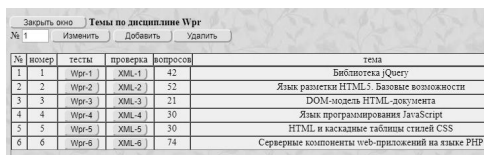
В работе рассмотрена система тестирования знаний студентов для проведения контрольных мероприятий. Также система может использоваться для подготовки фонда оценочных средств

Тестирование студентов с целью проверки знаний — изъезженная тема. Существует большое число программных средств для автоматизации тестирования, в том числе и средств, доступных через интернет (например, система на сайте www.quizful.net [1]).

Новые образовательные стандарты ФГОСЗ++ требуют наличия фонда оценочных средств (ФОС), для которых тестовые задания — одна из форм. Для преподавателей это оказалось удобной формой — относительно небольшая трудоемкость и реальная полезность. Существующие в вузах бюрократические структуры типа центров тестирования не очень удобны. Преподаватель зачастую должен подготовить разного рода документы, общаться с «всезнающими» методистами, соблюдать установленные правила и регламенты. При этом нет возможности оперативно вносить изменения в базы тестовых заданий. Выход — система с открытыми кодами, которую легко скачать, установить и использовать по своему усмотрению. В работе рассматривается именно такая система тестирования, с помощью которой можно проводить обычные контрольные мероприятия.

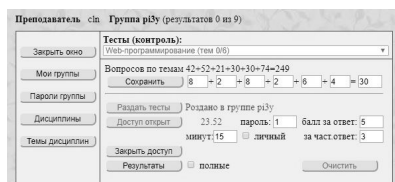
Администратор системы вводит дисциплины согласно учебному плану, ведет списки преподавателей и групп студентов. Также он может загружать базы тестовых заданий и проверять их на корректность.

Тестовые задания задаются в простом текстовом формате, а затем переводятся в стандартный XML-формат, а также JSON, который удобнее для мобильных приложений [2]. Имеется возможность проверить правильность ввода путем прохождения тестов в том виде, как они будут представлены студентам. Задания группируются по темам (рис.1) для того, чтобы при проведении тестирования можно было сформировать выборку по конкретным темам.



№ номер	тесты	проверка	вопросов	тема	
1	1	Wpr-1	XML-1	42	Библиотека jQuery
2	2	Wpr-2	XML-2	52	Язык разметки HTML5. Базовые возможности
3	3	Wpr-3	XML-3	21	DOM-модель HTML-документа
4	4	Wpr-4	XML-4	30	Язык программирования JavaScript
5	5	Wpr-5	XML-5	30	HTML и каскадные таблицы стилей CSS
6	6	Wpr-6	XML-6	74	Серверные компоненты web-приложений на языке PHP

Рис. 1: Ввод тестовых заданий по дисциплине.



Преподаватель: cin Группа р1у5 (результатов 0 из 9)

Тесты (контроль): [Выбор-программирование (тем 0/6)]

Вопросов по теме: 42+52+21+30+30+74+249

Скорректировать: 8 +2 +8 +2 +6 +4 =30

Раздать тесты: [Раздано в группе р1у5]

Доступ открыт: 23.52 пароль: 1 балл за ответ: 5

минут: 15 [] личный за част. ответ: 3

Закрыть доступ: [] полные [] Очистить

Рис. 2: Интерфейс преподавателя

Режим проведения контрольных мероприятий мало отличается от систем подобного типа: преподаватель «раздает» заданное количество заранее подготовленных тестовых заданий, устанавливает время на их выполнение и после завершения вносит результаты в итоговые документы. Для очного обучения тестирование проводится в компьютерном классе. Перед началом тестирования преподаватель также может задать групповой пароль для входа в систему и установить критерии оценивания. Предусмотрен балл за полностью правильный ответ и за частично правильный (при нескольких правильных вариантах студент не выбрал все). Вид экрана преподавателя представлен на рисунке 2.

На этом же экране преподаватель может закрыть доступ и посмотреть результаты. Тестовые задания выбираются случайно из множества тестовых заданий по изучаемой теме. Варианты ответов перемешиваются.

Экран прохождения теста студентом на обычном компьютере показан на рисунке 3. Вопросы выдаются последовательно. Когда все задания пройдены, указывается количество правильных и частично правильных ответов, а также число набранных баллов. Преподаватель имеет возможность продемонстрировать результаты и показать совершенные студентами ошибки.

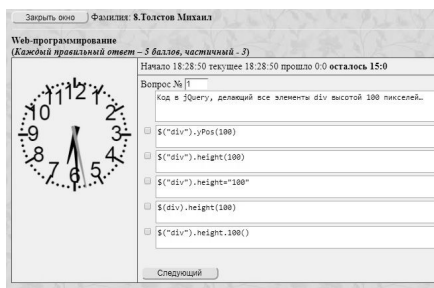


Рис. 3: Интерфейс студента

Завершение тестирования происходит либо по истечении времени, либо по указанию преподавателя. Результаты автоматически формируются в виде таблицы. Преподаватель может проанализировать ответы каждого студента, а также процент правильных ответов на каждое задание. Если успеваемость ведется в Google-таблицах, как это сделано в [3], результаты могут легко перенесены из системы в Google-приложение.

В перспективе предполагается расширение функций и возможностей системы в различных направлениях. Во-первых, это накопление БТЗ по разным темам и дисциплинам, увеличение числа типов заданий, совершенствование механизма генерации заданий. Во-вторых, более гибкая система оценивания и анализа результатов тестирования. Серверная компонента системы может быть интегрирована с корпоративным порталом вуза для того, чтобы результаты тестирования непосредственно передавались в электронные зачетные ведомости студентов, прошедших тестирование через указанную систему.

Система выложена на Github вместе с базами тестовых заданий по дисциплине «Web-программирование».

Литература

- [1] Сервис онлайн-тестирования Quizful [электронный ресурс: www.quizful.net (дата обращения: 17.05.2017)]
- [2] Чернышов Л.Н., Харитоновна Е.Н. Разработка системы тестирования знаний студентов на мобильных устройствах. В сборнике: Взгляд молодых ученых на проблемы устойчивого развития Материалы III Между-

народного конгресса молодых ученых по проблемам устойчивого развития. 2017. С.33-39.

- [3] Лукин В.Н., Чернышов Л.Н. Дистанционное обучение программированию на Google-платформе. — Четырнадцатая конференция «Свободное программное обеспечение в высшей школе»: Сборник материалов конференции / Переяславль, 25-27 января 2019 года. М.: МАКС Пресс, 2019., С.97-101.

Сергей Бессонов, Артём Попцов

Нижний Новгород, ООО ГК «Азь»

Проект: Азбук <http://sisyphus.ru>

Азбук: как решение для образования оказалось востребовано и в коммерческих компаниях

Феномен Google ChromeBook

ChromeBook — это изначально проект Google по созданию ноутбука со специализированной ОС, включающей в себя только браузер Google Chrome и минимум приложений, синхронизирующийся с сервисами Google. Сегодня на ChromeOS можно ставить приложения Android, но это другая история.

Не смотря на изначальную, опять же, ориентацию на технически непродвинутых пользователей, этот проект оказался очень популярен в образовании. В образовательных учреждениях очень востребовано, что операционную систему на ноутбуке нельзя сломать или перенастроить, приложения на ноутбуке нельзя установить или удалить, а данные на диске синхронизированы с центральным сервером и легко восстанавливаются в случае потери, кражи или поломки ноутбука.

Рождение азбука

У ChromeOS есть одна особенность, которая мешает использованию в образовании или госорганах в России — плотнейшая завязка этой ОС на сервисах Google. ChromeOS невозможно использовать, не имея учётной записи Google, браузер синхронизируется только через учётную запись Google и так далее. Даже открытый проект

ChromiumOS всё равно по-прежнему синхронизируется только через сервисы Google.

Казалось бы, на поверхности лежит идея — взять любой дистрибутив, оставить там только браузер и минимум приложений, сделать синхронизацию через какие-то сервисы и радоваться. Но реально, сообщество разработчиков Linux подобный дистрибутив не предлагает.

Из интересных проектов отметим:

- CloudReady <https://www.neverware.com/pricing> Коммерческий клон ChromiumOS. Всё также завязан на Google.
- Peppermint OS <https://peppermintos.com/> Дистрибутив на базе Ubuntu для работы в веб. Синхронизация, впрочем, отсутствует.
- Joli OS <https://www.jolicloud.com/jolios/> Проект заброшен. Синхронизация только с проприетарным облаком производителя.
- Liri OS <https://liri.io/download/> Проект по воссозданию облика ChromeOS целиком с интерфейсом на базе Qt и Wayland (!). Синхронизация пока отсутствует, но в планах. К сожалению, разрабатывается в одиночку и степень готовности оставляет желать лучшего.

Таким образом, мы подошли к тому, что при всей кажущейся простоте задачи, её до нас никто не решил. Поэтому мы решили попробовать сделать операционную систему для образовательных учреждений, которая бы отвечала следующему набору требований, актуальному для образования:

- пользователь не имеет возможности установить или удалить ПО, не имеет прав доступа к системе.
 - предустановленный набор ПО включает браузер и несколько программ для работы — видеоплеер, просмотрщик документов, текстовый редактор.
 - все данные пользователя и настройки браузера синхронизируются с центральным сервером при помощи учётной записи пользователя.
 - после разлогинивания пользователя все данные удаляются, можно передать компьютер другому студенту.
- ... и никакого администрирования.

И если первые два пункта элементарно решаются в любом дистрибутиве, автоочистка тоже не особая проблема, то синхронизация оказалась куда более сложной задачей. Большинство облачных систем синхронизации используют для этого протокол WebDAV. Например, Nextcloud и Owncloud используют свои клиенты для решения этой задачи, но такие клиенты не работают с любым WebDAV хранилищем. Мы попробовали монтировать домашний каталог по WebDAV на лету — получается очень ненадёжно и медленно, с гигантскими задержками при обращении к данным. Кроме того, хотелось бы, чтобы операционная система продолжала работу и при недоступном сервере синхронизации. Использование rsync по sftp в данном контексте совершенно неприемлемо. В итоге пришлось писать свою собственную реализацию.

Коммерческие перспективы

Общаясь с потенциальными коммерческими заказчиками, мы выяснили, что их потребности и потребности образовательных учреждений схожи. Им также требуется, чтобы сотрудник, работающий с ОС, не имел возможности её сломать, установить вредоносное ПО, а его данные должны храниться централизованно под надзором системного администратора, и, конечно, с применением резервного копирования. Разумеется, сервер для хранения данных должен находиться во владении предприятия, а не быть публичным, отсюда требование использовать стандартный протокол синхронизации.

Также, многие заказчики хотели бы видеть удаление всех данных после разлогинивания пользователя с целью сохранения секретных данных компании в случае кражи оборудования.

Кроме вышеописанных задач, коммерческие заказчики также хотят видеть возможность формировать список ПО, предустановленного в системе. По всей видимости, потребуется создание ПО, которое будет формировать образ для разворачивания Азбука в учебном заведении или коммерческой компании. Такое ПО позволит задать доменные имена серверов синхронизации, установить их сертификаты и сформировать список предустановленного ПО.

Николаев Константин Сергеевич, Суворов Артём Викторович
Москва, НИУ МИЭТ

Разработка программного средства отслеживания субъекта при динамических расстройствах организма

Средство отслеживания субъекта при динамических расстройствах организма

Большой интерес в этой области представляет лечение пациентов, подверженных эпилептическим приступам. Во время обследования и проведения видео-ЭЭГ (электроэнцефалография) мониторинга оператор вручную манипулирует камерой наблюдения. Выполнение такой работы на протяжении нескольких часов сказывается на качестве записи. Применение современных информационных технологий поможет избежать ошибок и неточностей в записи.

В настоящее время особое внимание представляют динамические расстройства людей с точки зрения лечения этих расстройств при помощи информационных технологий. В связи с декларированием президентом РФ национальных проектов на основе цифровизации, разработка программных средств для лечения указанных расстройств с помощью информационных технологий являются весьма актуальными.

Большой интерес в этой области представляет лечение пациентов, подверженных эпилептическим приступам. Во время обследования и проведения видео-ЭЭГ (электроэнцефалография) мониторинга оператор вручную манипулирует камерой наблюдения. Выполнение такой работы на протяжении нескольких часов сказывается на качестве записи, так как это связано с утомляемостью оператора (человеческим фактором). В результате приступ обследуемого пациента либо записан плохо, либо не записан вовсе, что увеличивает время обследования.

С целью улучшения качества видео-ЭЭГ мониторинга и увеличения быстродействия, поставлена задача разработки автоматизированного программного средства для вращающейся видеокамеры (PTZ-IP), которое обладает следующим функционалом:

- поиск метки (маркера), находящейся на одежде пациента;

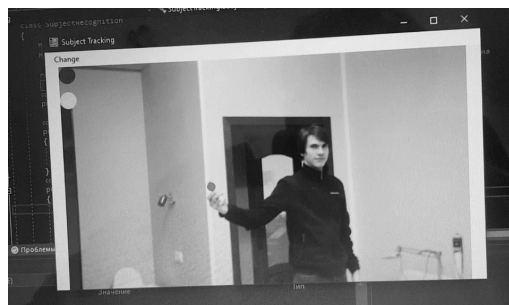


Рис. 1: Фотография с примером работы программного средства.

- фокусировка на маркере;
- при движении пациента сдвиг камеры в ту же сторону с минимальной задержкой;
- сбор данных об активности пациентов;
- хранение в базах данных.

При разработке заявленного функционала разработан тривиальный алгоритм вычисления позиции метки относительно центра объекта (см. рис. 1). Существенным плюсом такого подхода является высокая скорость обработки входящего видеопотока. Недостатком является то, что поворот камеры невозможен при съёмке в ночном режиме (при отсутствии цветного изображения), однако это несущественно, поскольку субъект в это время малоподвижен.

Для реализации алгоритма используется язык программирования C# и OpenCV — библиотеки компьютерного зрения с открытым исходным кодом. Тестирование программного средства проводится с помощью специализированной видеокамеры Sony SNC-ER580 (см. рис. 2).

В перспективе планируется разработка и тестирование быстрых алгоритмов, основанных на машинном обучении, для повышения точности детектирования субъекта и решения проблемы ночной съёмки.

Преимуществом разработанного программного средства является минимизация человеческого фактора и повышения качества обследования, то есть уменьшения временных и денежных затрат на лечение динамического расстройства, в данном случае эпилепсии.



Рис. 2: Видеокамера Sony SNC-ER580.

Данное программное средство разработано по заказу «Epilepsy Center» («Центр эпилепсии», <https://центр-эпилепсии.рф>) и будет использоваться в штатном режиме.

Список литературы

1. MSDN — библиотека официальной технической документации для разработчиков под ОС Microsoft Windows., <https://docs.microsoft.com/>
2. OpenCV — библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом., <https://opencv.org/>

Софья Зверева

Москва, НКО «OpenSpace», Общество Развития Взаимовыручки Ценовичок
github.com/keremet/pricer

Органайзинг в сфере СПО. Привлечение студентов к разработке и использованию СПО.

Аннотация

Одно из важнейших преимуществ свободного кода, которое, к сожалению, не всегда используется на практике — возможность вовлечь в процесс разработки, поиска ошибок (bug) и формулирования идей по доработкам (feature request) большое количество пользователей. В докладе сформулирована идея практического использования данного преимущества СПО на примере наращивания численности активных участников сообщества за счет студентов, изучающих программирование.

Как использовать преимущества СПО

Увеличение числа участников проекта напрямую способствует повышению *качества* программ и скорости разработки.

Кроме того, в сообществе при этом появятся больше заинтересованных в развитии продукта пользователей, которые будут находить возможности улучшить продукт. Например, сделать доработки, тестирование, поиск багов, идеи. . . Потому, что эти улучшения нужны им самим. А всё сообщество сможет применять результаты их труда. Это снизит нагрузку платных разработчиков.

Свобода ПО даёт возможность мобилизовать на разработку такие ресурсы, которые не сможет себе позволить даже транснациональная корпорация. Вовлечение в сообщество пользователей и студентов, изучающих программирование позволяет реализовать данную возможность.

За счет принадлежности к коллективу/сообществу с общей целью, разработчики смогут вложить в производство продукта столько творчества и усилий, сколько они не выжмут из себя просто ради денег.

С целью повышения конверсии пользователей в разработчиков может быть использован метод комьюнити-органайзинга. Этот метод организации сообществ предполагает модель участия заинтересованных

людей в развитии продукта. Другими словами, профессиональные оргайзеры объединяют в сообщество людей с общими интересами для взаимовыгодного достижения цели.

Вовлечение в разработку СПО студентов, изучающих программирование.

Применительно к высшей школе, оргайзинг позволит вовлечь множество студентов и преподавателей в сообщество. Программирование изучается в каждом регионе, численность студентов достаточно высока, это тысячи потенциальных участников сообщества. Через участие в свободных проектах студенты могут лучше осознать и на практике увидеть преимущества свободного ПО, такие как возможность пользователя участвовать в развитии продукта, возможность узнать, как работает программа и убедиться в отсутствии закладок, возможность сделать форк, etc. Понимание плюсов свободного ПО приведёт к сознательному использованию продуктов сообщества. Если люди будут использовать свободные программы там, где это можно, и только при отсутствии аналога пользоваться проприетарными сервисами, они будут находить больше багов и недостатков, а функционал СПО будет быстро увеличиваться. Еще одним ожидаемым результатом от вовлечения студентов можно считать появление специалистов, способных сопровождать СПО, облегчит его внедрение.

Наличие доступа к интернету даёт пользователю возможность участия в разработке, в какой точке земного шара он бы ни находился. Помимо этого, участвовать в развитии продукта могут даже относительно далёкие от программирования студенты. Чтобы позволить потенциальному участнику сообщества принять участие в разработке или регистрации ошибок с наименьшими затратами сил и времени, можно создавать подробную документацию по компиляции программ, по регистрации заявок на доработку, а также виртуальные машины со всеми необходимыми для компиляции библиотеками и программами.

Студенты пишут программы для: лабораторных, курсовых, дипломов, прохождения практики, получения «автомата» по предмету, а в ряде случаев, из личного интереса.

На сайте проекта должна быть возможность найти идеи разных уровней сложности и временных затрат.

Польза для студентов

- Возможности для дальнейшего трудоустройства (коммиты можно указать в резюме)
- Помощь со стороны более опытных разработчиков, контроль качества кода
- Доказанная полезность разработки

Что можно сделать?

Организаторы могут вести пропаганду среди студентов. Проводить лектории с приглашением специалистов или студентов, которые могут поделиться опытом участия в сообществах разработчиков. Взаимодействовать с преподавателями, доносить через них преимущества СПО.

Преподаватели, в свою очередь, могут оказывать помощь с выбором тем для курсовых и дипломов (описание тем на сайте проекта), или просто содействовать студентам в участии в развитии СПО.

Фирмы-разработчики могут платить деньги за доработки СПО.

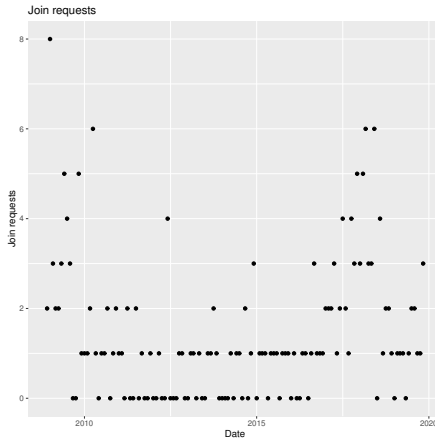
Игорь Чудов, Евгений Синельников
Саратов, Базальт СПО

Проект: Стажёрская программа компании "Базальт СПО"
<http://altlinux.org/>

Приобщение к участию в разработке свободных программ на примере стажировки в компании Базальт СПО

Аннотация

В рамках доклада рассмотрена проблема обучения новых сотрудников в компаниях, работающих с СПО. Предложены методики для ускорения процедуры включения людей в рабочие процессы. Представлены примеры вводных документов.



Проблематика

В сравнении с компаниями, ориентированными на outsourcing-разработку для множества различных заказчиков, небольшие продуктовые компании, особенно если они связаны с разработкой СПО, испытывают дефицит кадров.

В качестве примера предлагается рассмотреть выборку заявок на Join с сайта <http://bugzilla.altlinux.org/> за весь доступный период (2008-2020 годы):

Данные представляют из себя формальную характеристику потока потенциальных сотрудников. Не смотря на длительный период выборки можно отметить, что количество поступающих заявок невелико, что может служить сигналом о нескольких проблемах (при рассмотрении вопроса извне):

- проблема с процедурой подачи заявок (неочевидность, сложность, бюрократия, длительность);
- проблема с методом коммуникации (устаревание инструмента и методики, недостаточность усилий по преодолению барьеров коммуникации);
- также сказывается специфика предметной области с которой связана работа.

Со стороны же тех, кто вовлечён в работу с СПО, инструменты коммуникации типа IRC, e-mail и других подобных представляют ценность за счёт:

- доступность на разных платформах;
- широкая аудитория со всего мира;
- надёжность, стабильность во времени;
- простота.

В связи с неоднозначным восприятием ситуации разными группами людей возникает конфликт интересов. Для решения этой проблемы компаниям, работающим с СПО, приходится находить способы транслировать свою культуру в массы с целью формирования сообщества, разделяющего эти ценности. В противном случае работодатель сталкивается с дефицитом кадров, который не может восполнить за счёт основной массы рынка рабочей силы так как требуемый для работы с СПО уровень квалификации на данный момент таков, что сотрудник способен достичь его только в том случае, если он взаимодействует с СПО не только в рамках должностных обязанностей.

Для снятия части нагрузки с основных сотрудников, а также для наработки опыта передачи знаний и культурных ценностей, в Саратовском офисе компании "Базальт СПО" была организована стажёрская программа.

Для участия в стажёрской программе была размещена заявка на HeadHunter со следующими результатами:

- 28 входящих заявок получено;
- из них 6 человек отправили свои резюме и пришли на собеседование;
- из 6 человек было отобрано 4 стажёра;
- один стажёр отказался от работы через 4 дня.

Наблюдаемая по заявке ситуация оказалась такова: Поступающие кадры незнакомы с требуемыми инструментами, а также не воспринимают принятые в сообществе разработчиков СПО средства коммуникации.

Методика преодоления проблемы

В процессе жизнедеятельности человек связанный с СПО проходит несколько стадий:

- пользователь - использует СПО в повседневной деятельности;
- разработчик - пишет СПО (возможно, на коммерческой основе);
- мейнтейнер - не только пишет СПО, но и внедряет его в открытые продукты/дистрибутивы ОС.

В данном случае мейнтейнер рассматривается как высшая ступень эволюции инженера. Организованная в компании "Базальт СПО" стажёрская программа преследовала цель прохождения данных этапов за счёт оплачиваемого вовлечения инженеров в процесс решения задач с использованием СПО за счёт привлечения людей с рынка труда.

Для стажёров были проведены вводные лекции, определён список задач с нарастанием их сложности, а также выделено время для помощи с возникающими вопросами.

Анализ результатов

В рамках месячной стажировки получены следующие результаты:

- один из стажёров разработал утилиту ALT Media Writer дополнив и улучшив один из открытых проектов;
- второй стажёр занимался тестированием Samba и в результате сильно продвинул проект разбирая большое количество use-cases;
- третий стажёр занимался написанием документации к Samba на публичной Wiki <http://www.altlinux.org/>. В результате он получил как общее понимание проекта, так и сделал его доступнее для русскоязычных пользователей;
- Сотрудники компании получили необычный опыт наставничества и провели разбор допущенных в процессе ошибок.

На основании полученных результатов сделаны выводы:

- специфика отрасли требует найма сотрудников и стажёров высокой квалификации;
- приходящие с рынка кадры в большинстве своём недообучены и обладают недостаточной квалификацией;
- работа со стажёрами недостаточной квалификации забирает непозволительно много времени у основного состава сотрудников, что понижает рентабельность найма стажёров;

- найм сотрудников из сообщества может быть несравнимо более результативен по сравнению с общепринятыми методами, но только при наличии самого сообщества.

Библиография

Литература

[1] ALT Linux Bugzilla <http://bugzilla.altlinux.org/>

[2] ALT Linux Wiki <http://www.altlinux.org/>

Алексей Ерпелев, Владимир Симонов
Москва, ФГБОУ ВО «Российский государственный социальный университет»

Использование СПО в учебном процессе на примере разработки портативной метеостанции на RaspberryPi

Аннотация

В статье рассматривается использование свободного программного обеспечения в учебном процессе при обучении студентов направлений подготовки ИВТ, ИСТ и ПИН основам программирования и электроники с использованием Raspberry Pi. Данная тематика позволяет изучить принципы построения электронных устройств, сетевого взаимодействия, а также основы программирования, создание и эксплуатация баз данных. Метеостанция построена на базе Raspberry Pi Zero W с использованием цифрового датчика температуры и влажности DHT11.

Современное технологическое развитие дало возможность создавать приборы и устройства для отслеживания климатических показателей в режиме реального времени, сбора и анализа. Такие устройства могут быть построены практически на любых доступных средствах: микроконтроллер Arduino, IskraJS, Raspberry Pi и так далее [3]. Они могут работать как одиночно, так и совместно (например, связка Raspberry Pi + Arduino, одна платформа собирает данные, а вторая анализирует и аккумулирует). Данные устройства работают не только на проприетарном (На Raspberry Pi ядра GPU работают

под управлением операционной системы реального времени ThreadX. Данная ОС с закрытыми исходниками управляет системой без ведома ядра Linux [1]), но и на свободном программном обеспечении. Так, например, Raspberry Pi работает на свободной операционной системе Raspbian, можно использовать свободные компоненты такие, как язык Python, СУБД MySQL, база данных SQLite и так далее. На платформе Raspberry Pi имеется большое количество свободного программного обеспечения, многие из них распространяются по лицензии GPL. Данное устройство используется в обучении студентов по направлениям подготовки ИВТ, ИСТ и ПИН для освоения профильных компетенций - способностью разрабатывать средства автоматизированного проектирования информационных технологий (ПК-13), 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) [2].

С целью демонстрации использования свободного программного обеспечения в учебном процессе на примере разработки портативной метеостанции, авторами была поставлена и решена задача разработки соответствующего макета-устройства (Raspberry Pi Zero W и цифровой датчик температуры и влажности DHT11).

Разработанный макет-устройство позволяет записывать текущие климатические показатели в базу данных SQLite с датчика DHT11, а также выводить их на экран мобильного или десктопного устройства путём создания веб-страницы. В работе данного устройства участвуют такие компоненты, как база данных SQLite, веб-фреймворк Flask. Вся логика устройства написана на языке программирования Python. Устройство может самостоятельно записывать климатические показатели в автоматическом режиме или в ручном режиме. Для ручного режима Raspberry Pi может становиться Wi-Fi-роутером, к которому необходимо подключиться, далее открываем веб-страницу устройства через браузер и на экране у нас будет доступны следующие функции: просмотр текущих показателей и все данные из БД, или использовать внешнюю Wi-Fi точку для подключения к устройству [4].

При выполнении данного проекта студенты изучают язык программирования Python, получают навыки по созданию собственных веб-приложений, сборки устройств из электронных компонентов и опыт проектирования схем устройств в виртуальных средах, напри-

мер TinkerCad. Студенты используют полученные знания по дисциплине Проектирование и администрирование информационных систем на практике путём моделирования структуры базы данных и написания необходимых запросов для выполнения поставленных задач. Также студенты учатся составлять алгоритмы работы модулей и устройств обработки информации. 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 макет-устройство демонстрирует возможности свободного программного обеспечения для освоения студентами компетенций по направлениям подготовки ИВТ, ИСТ и ПИН. При этом данные устройства являются базовыми для изучения студентами технических специальностей программирования, электроники, автоматики, основ робототехники и прототипирования [3, 5]. Развитие данного направления в обучении студентов предполагает дальнейшее усложнение проекта в сторону увеличения количества используемых функций (экспортирование данных на сторонние сервисы, установка дополнительных датчиков и так далее).

Литература

- [1] Что не так с Raspberry Pi [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/440584/>. Дата обращения: 12.01.2020.
- [2] Использование свободного ПО в учебном процессе на примере разработки устройств «умного дома» с использованием микроконтроллеров Arduino и Iskra JS [Текст]* / Ерпелев А.В., Симонов В.Л., Рубанкова А.П. // Четырнадцатая конференция «Свободное программное обеспечение в высшей школе»: материалы конференции. — Москва: Издательство МАКС Пресс, 2019. — С. 34–36.
- [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] Creating Wireless Router using Raspberry Pi Zero W [Электронный ресурс] — Режим доступа: <https://scribles.net/creating-wireless-router-using-raspberry-pi-zero-w/>. Дата обращения: 12.01.2020.
- [5] Справочник по отечественным и иностранным электронным компонентам [Электронный ресурс] — Режим доступа: <http://www.chipinfo.ru/>. Дата обращения: 12.01.2020.

Алексеев Е.Р., Бондаренко Р.А., Виноградова А. Г.
Краснодар, Кубанский Государственный Университет

Использование программы Maxima при обучении студентов педагогического направления в Кубанском Государственном Университете

Аннотация

Представлена практика использования пакета Maxima при подготовке учителей по направлению «Педагогическое образование» (профили «Математика, Информатика») Кубанского Государственного Университета

Свободное программное обеспечение довольно широко используется при обучении студентов направления подготовки 44.03.05 «Педагогическое образование (с двумя профилями подготовки): математика, информатика» Кубанского Государственного Университета (КубГУ). При изучении программирования используются компиляторы `gcc`, `g++`, интерпретатор `python3`, интегрированные среды разработки Lazarus, PascalABC, Geany, Spyder. Знакомство с математическими пакетами проходит в курсах:

- программное обеспечение ЭВМ;
- математические пакеты и их применение в естественно-научном образовании;
- информационные системы в практике учителя.

Специфика профессиональной деятельности учителей математики и информатики определяет использование ими и студентами педагогических направлений математических пакетов. В связи с этим на

кафедре информационных образовательных технологий КубГУ большое внимание уделяется системам компьютерной алгебры на примере свободного пакета Maxima.

Первое знакомство с пакетом Maxima происходит в курсе «Программное обеспечение ЭВМ» (1-2 семестры). В этом курсе студенты получают начальные знания по различным математическим пакетам: MathCad Prime, Scilab, Maxima. Будущие учителя знакомятся с диалоговым интерфейсом Maxima, с графической оболочкой xMaxima, с графической интегрированной средой wxMaxima. В этом курсе студенты учатся производить простейшие вычисления и строить графики.

В третьем семестре студенты полноценно изучают пакет Maxima в курсе «Математические пакеты и их применение в естественнонаучном образовании», они продолжают изучение пакета и рассматривают использование Maxima при решении следующих задач:

- уравнения и неравенства;
- задачи линейной алгебры;
- вычисление пределов;
- вычисление производной;
- разложение функции в ряд Тейлора;
- вычисление определённых и неопределённых интегралов.

По окончании дисциплины студенты выполняют индивидуальное задание, в котором они представляют описание и решение задачи средствами Maxima.

Методической основой этой дисциплины является курс «Использование систем компьютерной математики (СКМ) Maxima и Scilab при решении задач в средней школе», который был разработан одним из авторов по заказу компании Базальт (<https://kurs.basealt.ru/course/view.php?id=38>).

В 9-10 семестре бакалавры изучают курс «Информационные системы в практике учителя», в котором будущие учителя информатики знакомятся с информационными ресурсами, которые могут быть использованы в практике современного учителя математики и информатики: Google Classroom (<https://classroom.google.com/h>), Exelearning (<https://exelearning.net/en/>) и рядом других. Кроме этого, в этом курсе есть блок, посвящённый разработке электронных образовательных ресурсов (электронных конспектов) с помощью Maxima.

Для разработки методических материалов для курса «Информационные системы в практике учителя» были привлечены студенты старших курсов кафедры информационных образовательных технологий КубГУ, которые, используя Maxima, в рамках курсовых работ и производственной практики разрабатывают электронные конспекты по различным разделам школьных курсов алгебры и геометрии. Разработанные электронные конспекты можно увидеть по адресу https://vk.com/maxima_kubsu. Эти материалы будут использоваться студентами старших курсов и магистрантами кафедры информационных образовательных технологий КубГУ при изучении специальных дисциплин («Информационные системы в практике учителя» и др.) и в курсовых работах, а также в практике выпускников кафедры – учителей математики и информатики.

Алексеев Е.Р., канд.тех. наук, доцент, Лутошкин Д.А., магистр, Стародумов В. В., аспирант
Краснодар, Киров, Кубанский Государственный Университет, Вятский Государственный Университет

Свободная система математического моделирования Simfor на базе компилятора gfortran (об одном свободном студенческом проекте)

Система математического моделирования Simfor [1] была разработана в рамках выпускных магистерских работ на кафедре «Прикладная математика и информатика» в Вятском Государственном Университете (ВятГУ).

Как известно, современные задачи моделирования, встречающиеся при решении инженерных задач, проблем биологии и во многих других отраслях человеческой деятельности являются довольно сложными и имеют ряд особенностей:

- сложные дифференциальные уравнения, описывающие физические, технические, биологические и др. процессы;
- при дискретизации дифференциальных уравнений получаются матричные разностные уравнения, содержащие сотни тысяч и даже получающиеся в результате требуют огромного количества вычислений;

- кроме того, разрабатываемое специализированное программное обеспечение использует ряд довольно стандартных алгоритмов (решение огромных систем линейных алгебраических уравнений, операции над матрицами большой размерности ($n \sim 10\,000 - 1\,000\,000$) и т.д.).

Для разработки современных программ математического моделирования программист должен знать предметную область, численные методы (в том числе и параллельные). Также от разработчика требуется умение разрабатывать высокоэффективные параллельные приложения.

Не всегда просто использовать опыт и коды разработчиков других систем, изложенных в современных научных работах. Ряд современных библиотек, реализующих эффективные параллельные алгоритмы вычислительной математики, являются проприетарными. Разработка ПО для решения задачи математического моделирования является сложной, дорогостоящей, громоздкой и длительной по времени.

Авторам неизвестны свободные универсальные интегрированные среды разработки параллельных приложений, которые бы работали в локальном компьютере и на вычислительном кластере.

В рамках выпускной магистерской работы была *поставлена задача разработки системы математического моделирования*, которая бы упростила разработку специализированных программ моделирования физических, технологических, биологических, инженерных и др. процессов.

Были сформулированы требования к системе моделирования.

1. В основе системы должен лежать язык программирования, который имеет свободный транслятор компилирующего типа;
2. Язык программирования должен поддерживать все современные технологии параллельного программирования.
3. Свободная библиотека, реализующая основные алгоритмы вычислительной математики.
4. Свободная библиотека графического вывода результатов.
5. Отладчик, который позволяет отлаживать как последовательные, так и параллельные приложения.
6. Универсальная свободная интегрированная среда разработки, позволяющая вести разработку приложений, работающих как на локальном компьютере, так и в вычислительном кластере.

В качестве языка программирования был взят современный язык Фортран. Вся система моделирования построена на базе компилятора `gfortran`. Были разработаны библиотека графического вывода и параллельная библиотека вычислительной математики. Свободная графическая библиотека на языке Фортран была представлена на тринадцатой конференции «Свободное программное обеспечение в высшей школе» в Переславле-Залесском 26-28 января 2018 года. Одной из особенностей библиотеки является возможность построения графиков при использовании консольных операционных систем. В этом случае библиотека формирует графические изображения в файле.

Ядром системы моделирования является разработанная авторами библиотека численного анализа.

Одним из главных требований к разрабатываемой системе является функционирование среды разработки как на локальном компьютере, так и на кластере. В связи с этим, было принято решение строить систему путём расширения функционала текстового редактора Emacs. В состав редактора был добавлен ряд существующих модулей. Был модифицирован модуль подсветки синтаксиса программы. Кроме того, на языке `elisp` были написаны следующие функции, превратившие текстовый редактор в полноценную интегрированную среду разработки:

1. Функции (*`compile-cc()`*, *`compile-f()`*) позволяют компилировать код на языках C/C++ и Фортран, кроме того, на базе этих функций написан шаблон, с помощью которого пользователь может изменять функционал функций *`compile-cc()`*, *`compile-f()`* при необходимости.
2. Функции *`compile-f-lib()`*, *`compile-f-plotter()`* предназначены для добавления в строку компиляции необходимых ключей и подключения библиотек.
3. Функция *`send-from-local()`* для рассылки исполняемого файла с локального компьютера.
4. Функция *`send-from-cluster()`* для отправки с узла кластера на оставшиеся узлы.
5. Функции *`run-program()`*, *`run-plotter()`*, *`run-program-cluster()`* для запуска разрабатываемых приложений на кластере.

Для установки системы на локальный компьютер или вычислительный кластер был написан скрипт, который собирает и добавляет биб-

блиотеки параллельных алгоритмов численного анализа и вывода графики в текстовый редактор Emacs, преобразует редактор Emacs в специализированную интегрированную среду разработки.

Разработанная система ориентирована на работу с любыми linux-подобными операционными системами. В данный момент прошло тестирование системы на ОС Ubuntu, Mint, Debian. Сейчас проходит адаптация системы под отечественные ОС – BaseAlt, Rosa, Astra. В систему могут быть добавлены необходимы библиотеки на Фортране и C/C++ с открытым исходным кодом. Система доступна на сайте <https://www.github.com/MatSysLib>.

Литература

- [1] Алексеев, Евгений Ростиславович; Лутошкин, Денис Александрович; Стародумов, Вячеслав Владимирович. Свободная система математического моделирования simfor на базе компилятора gfortran. Международный научный журнал «Современные информационные технологии и ИТ-образование», [S.l.], v. 15, n. 4, p. 820-827, dec. 2019. ISSN 2411-1473. Доступно на: <http://sitito.cs.msu.ru/index.php/SITITO/article/view/585>. Дата доступа: 23 января 2020 doi: <https://doi.org/10.25559/SITITO.15.201904.820-827>.

А.А. Дивинец, А.А. Маркина, Р. В. Разумейчик
Брест, Брестский государственный технический университет

О модернизации подхода к преподаванию основ алгоритмизации и программирования

Аннотация

Рассмотрен набор подходов к интенсификации изучения основ алгоритмизации и программирования студентами первой ступени высшего образования. Приводится перечень методических приёмов и их результативность в привязке к изучению теоретического материала и практическим занятиям для языка С. Рассматривается использование свободного программного обеспечения в свободной части курса и его связь с последующими курсами соответствующих специальностей.

Введение

Кафедра Электронных вычислительных машин и систем Брестского технического университета ведет подготовку студентов первой и второй ступеней высшего образования, и одной из основных специальных дисциплин, читаемых на первой ступени, является предмет «Основы алгоритмизации и программирования» (ОАиП). Цель преподавания данной дисциплины - освоение студентами методики постановки, подготовки и решения инженерно-технических задач на современных компьютерах. В качестве задач ставятся изучение процедурно-ориентированного языка программирования (при этом в базовых учебных программах для всех специальностей жёстко закреплен язык Си) и автоматизированных средств разработки (IDE), закладывание основ стиля программирования, приобретение навыков разработки и отладки, развитие алгоритмического мышления. Дополнительно особенностями преподавания дисциплины для специальностей первого курса являются: обучение студентов самоконтролю, формирование интереса к программированию, создание мотивации для саморазвития.

Традиционными проблемами являются низкая активность студентов в плане их участия в учебном процессе, адаптация к требованиям ВУЗа после школы, а также низкие показатели алгоритмического мышления и навыков разработки программ. Недостаточно качественное решение этих проблем на первом курсе оказывает сильный негативный эффект на последующие дисциплины. Более того, актуальность эффективного решения данных проблем в рамках дисциплины предсказуемо обострились после разделения высшего образования на первую и вторую ступени привело к сокращению числа учебных часов (причем значительно пострадали лабораторные занятия).

В ходе попыток решения обозначенных проблем в курс были внесены методические изменения, которые будут рассмотрены ниже.

Технологии преподавания

На протяжении последних двух лет преподавания данной дисциплины мы используем ряд методов активного обучения на лекциях в виде символической и текстовой наглядности для улучшения восприятия и усваиваемости материала:

- Лекция-визуализация позволяет схематично и наглядно представить опорную информацию, дополняется видеоматериалом. Этот вид лекции наиболее важен для студентов-визуалов, т. к. с помощью него происходит максимальное усвоение материала. Кроме этого, он повышает знания в области алгоритмизации, развивает системное мышление, позволяет получить наглядность в оформлении и оптимизации кода. Применён на всех темах лекционных занятий.
- Лекция-провокация. Данный вид лекции позволяет проконтролировать осознанность воспринимаемого материала слушателями, развивает у них критическое мышление, повышает дух соперничества и удовлетворенности от образовательного процесса. В начале лекции преподаватель озвучивает сколько ошибок будет допущено в лекции, в конце – проверяются найденные студентами упущения, проверяются и обсуждаются. Применён для следующих тем лекционных занятий: «Циклы», «Массивы», «Рекурсия», «Списки», «Деревья двоичного поиска».
- Лекция-конференция. Построена в форме научно-практического занятия с заслушиванием докладов и выступлений студентов (в малых группах) по ранее подготовленной проблеме в рамках учебной программы, а преподаватель подводит итоги, уточняет информацию, формулирует основные выводы. Данный вид лекции позволяет развить коммуникационный навык, здоровую конкуренцию, критичность и удовлетворение от процесса. Применён для итоговых лекций, завершающих отдельные разделы дисциплины.
- Проблемная лекция проводится в виде диалога со студентами, в процессе которого происходит совместный поиск решения задачи, разбираются типичные ошибки в работах. Такой подход позволяет развить коммуникационный навык молодых людей, а также увеличить осознание алгоритмов и основ процедурного программирования. Применён для лекций, проходящих после промежуточного контроля.

Кроме этого, в ходе лекций происходит комбинирование прослушивания и проверки знаний студентов (2 раза за лекцию), что позволяет более качественно удерживать внимание слушателей.

В ходе проведения лабораторных занятий нами используются:

- Метод «дискуссия», которая применяется во время обсуждения выданных заданий. Она позволяет максимально активизировать мыслительную деятельность студентов и увеличить усвояемость теоретических выводов. Кроме этого, данный метод позволяет увеличить мотивацию учения.
- Метод мозгового штурма направлен на снижение самокритичности студентов, повышение уверенности в себе и обучение навыку творческого подхода к решению проблемы. Применяется в ходе лабораторных задач по темам: «Циклы», «Массивы», «Рекурсия», «Стек», «Деревья двоичного поиска».
- Эвристическая беседа применяется нами при защите лабораторных работ. Этот активный метод обучения строится на вопросах проблемного характера, на который студент дает ответы. В итоге происходит активизация мышления и приобретаются новые знания.

Заметим, что наименее эффективной среди перечисленных показала себя эвристическая беседа: в ряде случаев студенты терялись в ходе попыток ответа (возможно, внешняя причина - язык программирования, изначально создававшийся отнюдь не для обучения студентов). В меньшей степени проблемы возникали с методом мозгового штурма для части студентов, испытывающих затруднения с импровизацией и свободным изъяснением своих мыслей.

В ходе изучения дисциплины имеются формы текущего (два раза в семестр) и итогового контроля. Для повышения заинтересованности и мотивации учебной деятельности мы используем кредитно-рейтинговую систему оценки в течение учебного семестра: студенты знакомятся со списком обязательных работ, требованиями преподавателей, ведётся учет активности и посещаемости. В конце семестра выстраивается рейтинг студентов по результатам их работы, и далее он учитывается в итоговом контроле.

Используемые программные средства

Кросс-платформенность языка С позволяет студентам использовать любые ОС и среду разработки на собственных устройствах. Однако в качестве основных систем в учебных классах установлены Ubuntu Linux и IDE Qt Creator. Преимуществом использования данных инструментальных сред является изначальное формирование у

студентов привычки к многоликости графических оболочек и интерфейсов. Кроме того, элементы работы с командной строкой и средой облегчает преподавание ряда дисциплин на следующих курсах, включая такие как «Конструирование программ и языки программирования» (курс включает создание графических приложений в Qt Creator), «Архитектура ЭВМ» (в рамках которой приобретаются навыки написания элементарных модулей ядра Linux), и др.

Заключение

Хотя используемые меры не являются панацеей, прошедшие два года обучения демонстрируют положительную динамику в овладении знаниями студентов, повышается интерес к будущей деятельности, возрастает мотивация к учебе, легче переносится адаптационный период обучения.

Татьяна Сундукова, Галина Ванькина

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

Использование свободного программного обеспечения для учебных курсов в контексте федерального проекта «Кадры для цифровой экономики»

Аннотация

Экономические и технологические особенности современного информационного общества требуют создания и реализации подходов по содействию гражданам в освоении ключевых компетенций цифровой экономики, обеспечении массовой цифровой грамотности и персонализации образования. С целью реализации направления «Кадры для цифровой экономики» национальной программы «Цифровая экономика Российской Федерации» образовательными организациями, являющимися площадками пилотного проекта, разрабатываются актуальные и востребованные учебные программы, преподавание по которым целесообразно осуществлять на основе свободного программного обеспечения.

Цифровое общество предъявляет к членам социума требования, характеризующиеся постоянным изменением содержания. Такая тенденция связана с расстановкой приоритетов в области структуризации знаний, расширением областей применения технологий, увеличением количества и сложности задач, требующих анализа и построения стратегии принятия решения. Компетентностный подход к уровню подготовки специалиста ставит перед системой высшего образования проблему в области формирования содержания образовательных программ, способствующих обучению конкурентоспособных специалистов.

Глобальные общественные процессы связаны с быстрым изменением технологий и условий жизни, поэтому к членам социума предъявляются новые требования, основанные на многоуровневой системе образования: школьное образование (начальный этап обучения), формирование умения учиться и способности к адаптации в новых условиях. В основу таксономии компетенций современного члена социума положена система датского ученого Йенса Расмуссена, предполагающая распределение всех работников по трем категориям (Правило, Умение и Знание). Согласно принятой концепции, наиболее востребованными и гибкими являются компетенции из категории Знание, поэтому для достижения требуемого уровня компетентности граждан государство предлагает различные формы организации обучения населения, федеральные стратегии и проекты [2]. С целью реализации направления «Кадры для цифровой экономики» национальной программы «Цифровая экономика Российской Федерации» образовательными организациями, являющимися площадками пилотного проекта, разрабатываются актуальные и востребованные учебные программы. В результате к 2024 году будет выстроена преемственная на всех уровнях система образования, включающая выявление и поддержку талантов в областях математики и информатики, подготовку высококвалифицированных кадров, отвечающих требованиям к компетенциям цифровой экономики, реализацию программ переподготовки по востребованным профессиям и перспективных образовательных проектов. Система предоставления персональных цифровых сертификатов включает инструменты поддержки образования и профессионального развития в течение всей жизни для формирования у трудоспособного населения компетенций, востребованных в условиях цифровой экономики. Перечень компетенций цифровой экономики, методика отбора конкретных программ из списка потенциально воз-

можных, принципы отбора обучающихся, которые смогут воспользоваться персональными цифровыми сертификатами будут доведены до всех участников постепенно от апробации к внедрению.

Одним из курсов, допущенных к апробации на пилотной площадке ТГПУ им. Л.Н. Толстого, является «Базы данных и работа с информацией», цель которого – формирование и совершенствование профессиональных компетенций в области баз данных. Курс предназначен для начинающих разработчиков приложений баз данных и для специалистов, повышающих квалификацию или проходящих профессиональную переподготовку.

Программа курса ориентирована на изучение следующих тем:

- Основные понятия в области баз данных.
- Методы проектирования и модели данных.
- Метод нормальных форм.
- Реляционная алгебра и исчисление.
- Введение в SQL.
- Выборка данных (оператор SELECT).
- Манипулирование данными.
- Создание объектов базы данных.
- Представления (VIEW).
- Определение прав доступа пользователей к данным.

Работа в сфере информационных технологий невозможна без взаимодействия с наборами данных. Для систематизации и оптимизации процессов хранения и обработки данных информацию структурируют в базу данных, которая доступна для реализации запросов. Стандартным ядром языка запросов к структурированным данным является SQL, команды которого применяются для создания, модификации и управления данными [1]. В качестве программного обеспечения для реализации практической части курса целесообразно выбрать свободное программное обеспечение, в частности, сервер баз данных MySQL. Выделим особенности MySQL:

- стандартизированное (SQL:2003, SQL:2008) и бесплатное программное обеспечение;
- возможность использования в качестве удаленного 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 счет 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.

Успешное освоение курса позволит развитию у слушателей социально-поведенческих, когнитивных, цифровых навыков, определенных как Создание систем (навыки программирования, разработки приложений, проектирования производственных систем) и Управление информацией (навыки обработки и анализа данных), что соответствуют уровню Знание в таксономии компетенций.

Список литературы

1. Сундукова, Т.О, Ванькина, Г.В. Роль свободного программного обеспечения в преподавании дисциплины Технологии разработки баз данных в вузе //Свободное программное обеспечение в высшей школе. Материалы конференции / Переславль. / отв. ред. Черный В.Л. – М.: МАКС Пресс – 2019. – С. 135-139.
2. Цифровая экономика: социально-психологические и управленческие аспекты: Коллективная монография / Е.В. Камнева [и др.]; под ред. Е.В. Камневой, М.М. Симоновой, М.В. Полевой. – М.: Прометей, 2019. – 172 с.

Екатерина Лавренова, Лариса Павликова, Михаил Бочаров
г. Москва, Государственное автономное образовательное учреждение
высшего образования города Москвы «Московский городской
педагогический университет» (ГАОУ ВО МГПУ)

Проект: Повышение квалификации педагогических работников,
осуществляющих функции администраторов информационных систем в
общеобразовательных организациях, в области применения пакета
свободного программного обеспечения в учебном процессе
<https://www.mgpu.ru/obrazovanie/institutes/ide/>

Организация проектно-практической деятельности студентов вуза по участию в процессе внедрения свободного программного обеспечения в общеобразовательные учреждения

Аннотация

Данное исследование посвящено анализу опыта организации проектно-практической деятельности студентов ГАОУ ВО МГПУ, участвовавших в реальном проекте организованном Министерством образования Московской области совместно с АНО ДПО «Софтлайн Эдьюкейшн» по обучению педагогических работников общеобразовательных организаций Московской области применению свободного программного обеспечения в учебном процессе.

Ключевые слова: проектно-практическая деятельность, работа со взрослой аудиторией, обучение студентов вуза, свободное программное обеспечение.

Значительной проблемой в ходе организации обучения студентов вуза является создание условий для применения изученного материала на практике, которые позволили бы закрепить полученные знания, довести уровень владения ими до навыков, сформировать необходимые требуемые образовательной программой компетенции и создать мотивацию к получению новых знаний и к качественному их усваиванию [1]. Для решения этой проблемы ГАОУ ВО МГПУ не только осуществляет мониторинг проектов, организуемых в социальной сфере, образовании и бизнесе, но и выступает инициатором социально-образовательных проектов с привлечением студентов к проектно-практической деятельности [2].

Институт цифрового образования (ИЦО) ГАОУ ВО МГПУ в рамках проекта Министерства образования Московской области по обучению педагогических работников общеобразовательных организаций Московской области применению свободного программного обеспечения в учебном процессе совместно с АНО ДПО «Софтлайн Эдюкейшн». В ИЦО под руководством директора института Екатерины Владимировны Лавреновой была создана рабочая группа по разработке механизмов реализации проекта с участием студентов старших курсов в роли тьюторов в данном проекте.

Перед административным и профессорско-преподавательским составом ИЦО стояла задача подготовки студентов к деятельности в качестве тьюторов, что подразумевало формирование на базе полученных в вузе знаний практических умений работы с программным продуктом Альт Линукс СПТ 7.0 и педагогических компетенций по работе со взрослой аудиторией [3]. Для этого администрация ИЦО разработала механизм встраивания процесса подготовки тьюторов в учебно-образовательный процесс подготовки студентов через организацию прохождения студентами плановых практик (учебных, производственных, педагогических), определила механизмы мотивации и поощрения участников данного проекта.

Администрация и профессорско-преподавательский состав ИЦО, ориентируясь на Государственную программу Московской области «Цифровое Подмосковье» на 2018-2021 годы [4] и проект Министерства образования Московской области по оснащению школ компьютерной техникой [5] определили: уровень необходимой подготовки учителей по владению программным продуктом Альт Линукс СПТ 7.0; условия проведения занятий; программно-техническую базу для организации занятий. Администрация ИЦО также тесно взаимодействовала и с АНО ДПО «Софтлайн Эдюкейшн» по формированию модели организации подготовки студентов тьюторов в ИЦО ГАОУ ВО МГПУ, встраиванию методических материалов АНО ДПО «Софтлайн Эдюкейшн» в образовательный процесс и итоговой аттестации студентов-тьюторов для подтверждения необходимой квалификации и соответственного получения допуска к участию в проекте по требуемой подготовке учителей.

Профессорско-преподавательский состав ИЦО участвовавший в данном проекте: проводил анализ соответствия компетенций которыми обладают студенты требуемому уровню подготовки тьюторов [6]; разрабатывал курс по адаптации полученных практических навыков

к требованиям подготовки тьюторов [7]; проводил курс по формированию базовых практических навыков работы в операционной системе Линукс и обучению методическим особенностям практической работы со взрослой аудиторией.

В результате участия ИЦО ГАОУ ВО МГПУ в проекте организованном Министерством образования Московской области совместно с АНО ДПО «Софтлайн Эдюкейшн» по обучению педагогических работников общеобразовательных организаций Московской области применению свободного программного обеспечения в учебном процессе было подготовлено 23 студента-тьютора, на итоговой аттестации в АНО ДПО «Софтлайн Эдюкейшн» они продемонстрировали высокий уровень подготовки и все получили удостоверение, дающее право в рамках проводимого проекта участвовать в подготовке учителей в качестве тьюторов. Студенты ИЦО ГАОУ ВО МГПУ в качестве тьюторов участвовали в процессе подготовки 1500 педагогических работников, осуществляющих функции администраторов информационных систем в общеобразовательных организациях Московской области.

По результатам участия студентов в проектно-практической деятельности Проектным офисом ИЦО проведен анализ проектной деятельности по подготовке студентов-тьюторов и их участию в проектной деятельности. В частности, было выявлено какие новые навыки и умения приобрели студенты, с какими трудностями по содержанию обучения они столкнулись, легко или сложно им было работать со взрослой аудиторией, что помогало им находить общий язык, какие отзывы у слушателей были об организации обучения.

Таким образом практика реализации представленного проекта показала перспективным организацию практики в рамках проектов, реализуемых совместно с коммерческими компаниями и государственными учреждениями. Так у студентов, участвовавших в проектах, увеличилась мотивация к учебной деятельности, они систематизировали и закрепили полученные в вузе знания и навыки, получили опыт профессионального общения.

Литература

- [1] Лавренова Е.В. Разработка учебного технологического пакета «новые технологии» для организации проектной деятельности бакалавров (на примере учебного модуля «промышленное производство»)/Педагогическое образование в России. 2014. № 1. С. 146–149.

- [2] Лавренова Е.В., Готская И.Б. Обучение современным технологиям бакалавров и магистров технологического образования на основе учебных технологических пакетов / В сборнике: Электронное обучение в ВУЗе и в школе. материалы сетевой международной научно-практической конференции. 2014. С. 170–173.
- [3] Лавренова Е.В., Теплякова А.Ю. К вопросу о развитии информационно-технологической компетентности взрослого населения России / В сборнике: Новые информационные технологии в образовании и науке. Материалы X международной научно-практической конференции. 2017. С. 299–304.
- [4] Государственная программа Московской области «Цифровое Подмосковье» на 2018-2021 годы / Утверждена постановлением Правительства Московской области от 17.10.2017 № 854/38.
- [5] Более 750 миллионов рублей выделили на покупку компьютеров для школ Подмосковья/Правительство московской области. URL: <https://mosreg.ru/sobytiya/novosti/news-submoscow/podmoskovnye-shkoly-poluchat-15-tys-edinic-kompyuternoj-tekhniki> (дата обращения: 17.05.2019).
- [6] Бочаров М.И. Структура и способы межвузовского взаимодействия сообщества разработчиков свободной автоматизированной информационной системы вуза на базе вузовских информационно-технологических консорциумов / Информатизация образования и науки. 2013. № 2 (18). С. 170–186.
- [7] Бочаров М.И., Ростовцев А.Н. Составляющие программного продукта и технологические этапы его разработки в проектной деятельности: учеб. пособие / Новокузнецк: РИО НФИ КемГУ, 2005. — 95 с.

Сергей Мартишин, Марина Храпченко

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

Организация облачных вычислений над конфиденциальными данными на СПО

Аннотация

Рассмотрены¹ вопросы организации вычислений над конфиденциальными данными большого объема. Сформулировано понятие дедуктивно безопасного запроса. Описаны свойства и применение лямбда-архитектуры. Определен комплекс СПО для практической реализации проекта.

В связи со своей деятельностью, в течение нескольких последних лет различные фирмы, компании, предприятия, государственные службы и т.д. накопили значительный объем данных, подчас разнородных и неструктурированных. Появление термина «Big Data» («Большие данные»), как раз и связано с появлением огромного объема разнородной информации. При этом накопленные данные содержат в себе полезную информацию, которая после определенной обработки может быть использована для повышения эффективности работы организаций, анализа статистической информации при принятии стратегических решений и т.п. Например, в процессе строительства и эксплуатации зданий и сложных инженерных сооружений (особенно стратегических объектов) возникает необходимость контроля различных показателей, изменяющихся динамически. Это осуществляется при помощи различного рода датчиков, показания с которых постоянно передаются в базу данных. При этом, для контроля за состоянием объекта данные анализируются и на основе анализа принимаются решения. Аналогичная ситуация наблюдается в плане исследования медицинских данных (анализ эффективности того или иного препарата, загруженность койко-мест, оценка эффективности работы персонала) и пр.

Обработка больших данных всегда представляла собой сложную проблему. В настоящее время существует достаточно много алгоритмов и технологий, позволяющих хранить и обрабатывать большие данные. Однако вопросы быстрейшего и безопасного вычисления

¹Работа выполнена при финансовой поддержке РФФИ, проект № 19-01-00702 А

по-прежнему остаются в фокусе внимания. Это тем более актуально, что во многих случаях никакие конкретные сведения об объекте исследования не должны быть раскрыты. Таким образом при работе с базами конфиденциальных данных одним из наиболее важных является вопрос безопасности.

В статьях [1,2] была разработана и исследована модель облачных вычислений, состоящая из участников (к их числу относятся облачный сервер, пользователи, хранящие свои данные на облаке, клиенты, проводящие вычисления с этими данными, и криптосерверы), сеть связи, данные, вычисляемые функции, контроль доступа (центр аутентификации проверяет полномочия данного клиента на вычисление запрашиваемой функции и, при наличии таковых, санкционирует выполнение запроса).

В статье [2] также было сформулировано требование криптографической стойкости, в предположении, что при заданных условиях пассивный противник способен получить лишь такую информацию о конфиденциальных данных пользователей, которую можно извлечь из значений функций, вычисляемых по запросам клиентов. Однако очевидно, что определенные запросы клиентов позволяют получить конфиденциальные данные пользователей. Поэтому важным моментом является аутентификация запросов на вычисления, которая позволяет определить разрешенное множество функций.

Там же [3] было сформулировано определение дедуктивной безопасности, подразумевающее, что клиент базы данных, получив ответы на все доступные ему запросы, не может получить дедуктивно (в виде логического следствия) конфиденциальные данные пользователей. И было доказано, что симметрические запросы являются дедуктивно безопасными, поскольку призваны собирать лишь статистические данные (например: сумму, медиану, среднее квадратическое отклонение и др.) о количественных данных, содержащихся в базе, но не раскрывают никакой специальной информации (например, имен), относящейся к отдельным пользователям базы данных.

В плане быстрого действия некоторое время назад была предложена новая масштабируемая и отказоустойчивая архитектура обработки данных — лямбда-архитектура (Lambda Architecture), которая способна производить вычисления с огромными объемами данных на основе методов пакетной обработки [4]. Заметим, что в плане организации хранения в лямбда-архитектуре используются репликация и шардинг.

В основе лямбда-архитектура имеет структуру, состоящую из трех уровней [4].

Пакетный уровень предназначен для управления архивом необработанных данных и предварительного вычисления пакетных представлений. Важно, что внесённые данные остаются неизменными, происходит лишь добавление новых (исключение составляет удаление заведомо неверных данных).

Уровень обслуживания индексирует пакетные представления. Основная цель индексации — возможность выдавать запрашиваемые пакетные представления с низкой задержкой.

Уровень ускорения отвечает за обработку данных, поступающих в систему в реальном времени. На этом уровне к уже обработанным данным добавляются актуальные данные, то есть ответ на запрос выдается при помощи объединения результатов из пакетных представлений и полученных представлений в реальном времени.

Для практической реализации работы с большими данными описанный выше подход был осуществлен с использованием следующего свободного программного обеспечения (СПО, free software): файловой системой HDFS (Hadoop Distributed File System — для хранения необработанных данных) [5], PostgreSQL (хранение пакетных представлений) [6], Apache Hive [7] (для выполнения запросов, агрегирования и анализа данных, хранящихся в Hadoop). Данный выбор СПО не является единственно возможным, но в виду наличия требуемого функционала, хорошей совместимости и удобства использования, полностью отвечает требованиям разработки.

Таким образом, сочетание использования лямбда-архитектуры с ограничением на разрешенные дедуктивно безопасные запросы к базам данных позволяет реализовать облачные вычисления над конфиденциальными данными на СПО.

Литература

- [1] Варновский Н.П., Мартишин С.А., Храпченко М.В., Шокуров А.В. // Пороговые системы гомоморфного шифрования и защита информации в облачных вычислениях, Программирование. 2015. № 4. С. 47–51.
- [2] Варновский Н.П., Захаров В.А., Шокуров А.В. К вопросу о существовании доказуемо стойких систем облачных вычислений // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. 2016. № 2. С. 32–38.

- [3] Варновский Н.П., Захаров В.А., Шокуров А.В., О дедуктивной безопасности запросов к базам конфиденциальных данных в системе облачных вычислений // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. 2017. № 1. С. 38а–44.
- [4] Марц Н., Большие данные. принципы и практика построения масштабируемых систем обработки данных в реальном времени // Марц Н., Уоррен Д., — Санкт-Петербург: Вильямс, 2016. — 368с.
- [5] Apache Hadoop [Электронный ресурс] — Режим доступа: <https://hadoop.apache.org/>, свободный. Яз. англ. — Дата обращения: 11.01.2020.
- [6] PostgreSQL [Электронный ресурс] — Режим доступа: <https://www.postgresql.org/>, свободный. Яз. англ. — Дата обращения: 11.01.2020.
- [7] Apache Hive [Электронный ресурс] — Режим доступа: <https://hive.apache.org/>, свободный. Яз. англ. — Дата обращения: 11.01.2020.

Зацепин Н. А.

г. Москва, Московский авиационный институт

https://github.com/CiFFiR0/template_test_tools

Генератор тестовых заданий для системы Moodle

Аннотация

В докладе рассматривается способ генерации тестовых заданий. Приведено описание веб-сервиса для работы с данным видом шаблонов. Описываются способы интеграции данного генератора тестовых заданий с системой дистанционного обучения Moodle.

В современной системе получения высшего образования тестирование студентов играет важную роль. Написание тестов представляет собой трудоемкий, творческий процесс, который в данное время слабо автоматизирован. А ведь автоматизация этого процесса хотя бы частично позволила бы преподавателю уделять больше времени другим своим важным обязанностям. Над этой проблемой ранее работали авторы статей [1, 2, 3, 4, 5]. Автором предложена новая форма шаблона тестового задания, использующая заданные преподавателем продукционные правила. Они дают возможность генерировать из группы

таких шаблонов тесты в формате GIFT. Все это дает возможность преподавателю написать шаблон теста вместо самостоятельного написания всех вариантов данного теста, что должно облегчить труд и сэкономить время. Предлагаемый подход может быть использован в работе преподавателя как инструмент для подготовки тестирования в зависимости от его цели и характера изучаемой дисциплины.

Тестовые задания по дисциплинам программирования имеют особенности, не характерные для других дисциплин. Тестовое задание может включать фрагмент кода на некотором языке программирования, а в ответе необходимо указать результат выполнения этого кода. При составлении такого задания для полной надежности желательно действительно выполнить этот фрагмент и получить результат. Но если подобные фрагменты параметризовать так, что число возможных вариантов задания станет очень большим, их проверка станет очень трудоемкой. Возможное решение этой проблемы — исполнение сгенерированного фрагмента непосредственно после его создания в генераторе заданий. Именно это решение предлагается в описываемом методе.

Структура шаблона тестового задания состоит из следующих элементов:

- Заголовок шаблона — метainформация относительно семантического содержания шаблона;
- Тип шаблона — открытый ответ, закрытый ответ с одним правильным вариантом или закрытый ответ с несколькими правильными вариантами;
- Грамматика — КС-грамматика, задаваемая пользователем и являющиеся подмножеством РБНФ.
- Текст задания — фактически является правой частью правила для начального нетерминала и также может содержать вхождения нетерминалов, объявленных в грамматике;
- Скрипт обратной связи — скрипт на языке JavaScript. Исполнение этого скрипта позволяет сформировать правильный ответ на поставленный в задании вопрос в зависимости от сгенерированного текста задания. Для этого в скрипте есть возможность, используя встроенные переменные и функции, получить информацию о том, какие конкретно альтернативы были выбраны для соответствующих нетерминалов и каковы их значения. Данный

скрипт выполняется при генерировании тестового задания из его шаблона. Результатом выполняемого скрипта обратной связи является ответ(ы) и неверные варианты выбора. После того, как будет получен этот результат, он вместе со сгенерированным текстом тестового задания транслируется в тестовое задание формата GIFT.

Описанная техника генерации тестовых заданий, а также по аналогии и обычных заданий реализована в веб-сервисе. Веб-сервис имеет следующие функциональные возможности:

- регистрация и авторизация пользователя;
- просмотр документации;
- хранение, загрузка на сервер, загрузка с сервера, создание, редактирование и удаление шаблонов теста, тестовых заданий, заданий и групп заданий с помощью редакторов;
- загрузка архива со сгенерированными вариантами контента в формате — GIFT для шаблонов тестовых заданий и тестов, TXT для шаблонов заданий и групп заданий.

Создаются и редактируются шаблоны в редакторах. В них шаблоны представляются как форма, которую необходимо заполнить. Редакторы для шаблонов тестовых и обычных заданий имеют подсветку для грамматики, текста заданий и скрипта обратной связи. Она реализована с помощью библиотеки CodeMirror. Редактор с каждым изменением формы проверяет шаблон на корректность и при появлении в нем ошибки пользователь сразу об этом уведомляется. Для формата хранения шаблонов был выбран JSON. Сервер использует СУБД MySQL. Шаблоны хранятся в БД в сжатом бинарном представлении. При их загрузке пользователь получает JSON-файлы. Веб-сервис написан на Node.JS с использованием сторонних модулей.

Для интеграции описанных шаблонов с системой Moodle есть два способа: импорт тестов и написание модуля. Для импорта достаточно из шаблона теста сгенерировать тесты в формате GIFT и импортировать их в систему. Однако тесты устаревают и для последующих курсов надо будет повторять данное действие. Модуль для системы Moodle, в который встроен данный генератор, лишен данного недостатка. Можно будет единожды загрузить шаблон теста и для каждого тестирования модуль будет сам генерировать тесты.

Литература

- [1] *Sosnovsky S., Shcherbinina O., Brusilovsky P.* Web-based Parameterized Question as a Tool for Learning, in Allison Rossett (ed): Proceedings of E-Learn 2003, Phoenix, Arizona USA, November 7–11 — 2003. — P. 2151–2154.
- [2] *Сергушичева А.П., Швецов А.Н.* Гибридный подход к синтезу тестовых заданий в тестирующих системах // Математика, Компьютер, Образование: Сборник научных трудов. — 2006. — Выпуск 13. Том 1 2006. — С. 215–228.
- [3] *Крутасов А. М., Швецов А. Н.* Метод автоматизированной генерации заданий для тестов контроля знаний из текстов учебных пособий // Современные информационные технологии и ИТ-образование. — 2013. № 9. — С. 218–228.
- [4] *Лантев В. В.* Морфологический синтез вариантов заданий в обучающей системе по программированию // Вестн. Астрахан. гос. техн. ун-та. Сер.: Информационные технологии в образовательной деятельности. — 2015. — № 1. — С. 140–152.
- [5] *В.В. Кручинин, Л.И. Магазишников, Ю.В. Морозова* Модели и алгоритмы компьютерных самостоятельных работ на основе генерации тестовых заданий // Известия Томского политехнического университета. Сер.: Технологии инженерного образования. — 2006. — Т. 309. №8. — С. 258–262.

Василий Романов, Ольга Солнцева, Анастасия Трескова
Ульяновск, ФГБОУ ВО Ульяновский ГАУ

Опыт перехода на альтернативное программное обеспечение в учебном процессе УлГАУ

Аннотация

Рассмотрены вопросы перехода на свободно распространяемое или альтернативное программное обеспечение. Проведено сравнение свободно распространяемых операционных систем семейства Linux с широко распространенной операционной системой Microsoft Windows.

Несколько последних лет РФ уделяет особое внимание вопросам импортозамещения. Этот процесс коснулся и сферы приобретения и

использования ПО. Вышло несколько нормативных документов обязывающих все бюджетные организации переходить на отечественное ПО. Причем не только в процессе его будущего приобретения, но и ранее приобретенного [1][2].

В связи с этим все государственные предприятия, включая учебные заведения, переходят либо на ПО включенное в «Единый реестр российских программ для электронных вычислительных машин и баз данных», либо на свободно распространяемое программное обеспечение на основе операционных систем семейства Linux.

Практика перехода на свободное ПО распространена во всём мире. Масштабные мероприятия по внедрению и использованию открытого программного обеспечения можно наблюдать в Бразилии, Италии, Великобритании и других странах [3].

Несомненными плюсами Windows являются: популярность и распространенность, огромное количество прикладных программ, хорошая поддержка актуального оборудования. Основным минусом, на наш взгляд является то, что Windows — это самая высоко «атакуемая» вирусами система, в силу ее большой популярности. Так же стоит отметить, что система платная.

Минусы Windows оборачиваются для свободно распространяемого программного обеспечения преимуществами. В частности — это открытость и бесплатность. В последние годы можно добавить наличие довольно хорошей поддержки актуального оборудования. Так же преимуществом является защищенность от вирусов. Неоспоримые преимущества Linux систем над Windows признает даже президент и главный исполнительный директор Microsoft Стив Балмер [4].

В последнее время бизнес и государство всё чаще начинают «смотреть» в сторону свободно распространяемого или альтернативного программного обеспечения. Как следствие, возникают вопросы готовности технических служб и пользователей к такому переходу. Что требует от учебных заведений способности подготовить будущих специалистов для работы в этих программных продуктах.

В 2019 г. в Ульяновском ГАУ после тестирования свободно распространяемого ПО и ПО включенного в Единый реестр российских программ принята для внедрения отечественная операционная система «АльтОбразование». На начальном этапе осуществляется использование ОС «АльтОбразование» и ряда приложений входящих в состав дистрибутива в учебном процессе по дисциплинам: «Информатика», «Основы компьютерных технологий», «Информационные технологии

в профессиональной деятельности» по всем основным образовательным программам Ульяновского ГАУ для студентов 1 — 3 курсов.

Поскольку наш ВУЗ не связан с подготовкой специалистов в области ИТ, основной упор делается на подготовку пользователей применяющих офисные приложения и специализированное ПО в своей профессиональной деятельности. Из приложений общего назначения в учебном процессе на первом этапе используется пакет LibreOffice.

К началу учебного года проведена подготовка преподавателей для работы с ОС «Альт Образование», LibreOffice, Scribus, GIMP и Inkscape, разработаны методические материалы по LibreOffice.

Практика внедрения ОС «Альт Образование» и приложений работающих в её среде показала, что особых проблем у пользователей при работе в этой системе не возникает. Естественно, требуется время на освоение интерфейса программ. Для большинства пользователей имеющих опыт работы с MS Office, графические значки интерфейса LibreOffice хорошо «ассоциируются» с их функционалом. Некоторое время требуется в основном для определения их местоположения на панели.

Таким образом, для рядового массового пользователя, использующего ограниченный набор прикладных программ, в основном из офисного пакета, программная платформа, на которой работает вычислительное средство, не так важна. При использовании современного графического интерфейса отличий между Windows и систем из семейств Linux пользователь может просто не заметить.

Более проблемная составляющая перехода на ОС «Альт Образование» и приложений работающих в её среде — готовность технического персонала и состояние парка компьютерной техники. Из-за отсутствия практики работы с ПО семейства Linux, потребовалось много времени на обретение необходимых навыков по установке ОС и приложений.

Поскольку компьютерные классы комплектовались в разное время, «железо не однородно», возникают частные проблемы, связанные с конфигурацией конкретного компьютера. На часть компьютеров установить ОС не удалось. Так же возникают проблемы с тем, что некоторые приложения работают некорректно, например Base в LibreOffice 5.3. Приложение запускается, но при попытке создания объекта, например таблицы, Base закрывается.

Таким образом, несмотря на некоторые проблемы, переход на использование свободно распространяемого или альтернативного программного обеспечения в учебных заведениях и других бюджетных

организация, к тому же простимулированный «административным ресурсом», является оправданным шагом, позволяющим повысить информационную безопасность и значительно сэкономить бюджетные средства.

Литература

- [1] Распоряжение Правительства России от 26 июля 2016 г. №1588-р.
- [2] Приказ № 334 от 29 июня 2017 г. Министерства связи и массовых коммуникаций российской федерации «Об утверждении методических рекомендаций по переходу федеральных органов исполнительной власти и государственных внебюджетных фондов на использование отечественного офисного программного обеспечения, в том числе ранее закупленного офисного программного обеспечения»
- [3] Перечень школьного программного обеспечения, <https://academic.ru/dic.nsf/ruwiki/310737>
- [4] Панишев А., Пингвин против Империи, <http://rus-linux.net/lib.php?name=/MyLDP/histori/iv-pan/iv-pan.html>