

## **ПРОБЛЕМА ВЫБОРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ПРЕПОДАВАНИЯ БАЗОВЫХ ДИСЦИПЛИН И ПОДДЕРЖКИ УЧЕБНОГО ПРОЦЕССА НА ФКН ОМГУ**

**Д.Н. Лавров**

В статье рассматриваются проблемы выбора легального программного обеспечения для преподавания основных дисциплин и обеспечения работы подразделений факультета компьютерных наук Омского государственного университета.

### **Введение**

Проблема выбора программного обеспечения – одна из сложнейших проблем – особенно актуальной становится в рамках преподавания компьютерных дисциплин. Огромное разнообразие предлагаемых программных продуктов различными производителями ставит разработчика учебного курса в условия множественного выбора.

С другой стороны, необходимо обеспечить работу структурных подразделений, лабораторий, деканата и кафедр.

Какие инструменты изучать в первую очередь? Какими критериями при этом руководствоваться?

Рассмотрим проблему выбора программного обеспечения (ПО) на примере факультета компьютерных наук (ФКН) Омского государственного университета. Выбор устанавливаемого ПО производится ежегодно при подготовке компьютерных классов к новому учебному году. Причин, радикально повлиявших на актуализацию этой проблемы, несколько.

Во-первых, особое внимание к свободному и открытому программному обеспечению, проявляемое со стороны государственных органов. Стремление использовать его в образовательном процессе средней школы, попытки создания национальной операционной системы – все это делает необходимым изучение этого программного обеспечения на факультете. Студенты должны владеть не только проприетарными продуктами, но и лучшим свободным и открытым ПО.

Нужно отметить, что открытое ПО не является противопоставлением проприетарному ПО, как это часто явно или неявно преподносится в средствах массовой информации. Проприетарное ПО является лишь противопоставлением свободному ПО в определении Р. Столлмена [1]. Бесплатное ПО может быть проприетарным, открытое ПО также может быть проприетарным. Противопоставлением бесплатному ПО является коммерческое. Операционные системы Microsoft являются закрытыми коммерческими продуктами, а Internet Explorer закрытым бесплатным продуктом. И то, и другое являются проприетарным ПО. Операционная система Solaris является проприетарным продуктом, исходный код которой практически полностью открыт в проекте Open Solaris

Вторая причина — это усиление борьбы правоохранительных органов с использованием нелицензионного программного обеспечения. Это стимулирует к приведению использования ПО в вузе в соответствие с законодательством. В частности, это привело к оформлению MSDNAA-подписке на программное обеспечение фирмы Microsoft на ФКН. Эта частично решило проблему, но только частично, так как использование продуктов Microsoft по данной подписке ограничено исключительно учебным процессом. Да и свобода ограничена некоммерческим использованием. После завершения обучения студенты должны деинсталлировать программные продукты Microsoft, полученные по подписке, и в случае необходимости их дальнейшего использования приобретать коммерческие лицензии. С точки зрения факультета нет смысла обучать программным продуктам, за использование которых потом придется платить фирмам, где будут работать выпускники.

Третья причина — это свобода исследования, модификации и распространения кода, принятая в научной среде. Эту свободу может обеспечить только свободное ПО и некоторые проекты с открытым исходным кодом. Лицензионное соглашение Microsoft, предлагаемое в рамках MSDNAA, предоставляет данные свободы достаточно ограниченно.

Итак, перед нами встала задача легализации программного обеспечения в рамках факультета. Внимание привлекло свободное и открытое программное обеспечение. Но какими критериями пользоваться при выборе этого ПО, а в некоторых случаях, может быть, лучше использовать коммерческий или проприетарный продукт?

## 1. Проблема выбора программного обеспечения

Все мы заложники своих привычек, знаний и требований к себе и другим. Часто это вызывает проблемы во взаимоотношении между людьми, когда один специалист пытается вольно или невольно навязать другому свое мировосприятие. Многие Linux-оиды и Unix-оиды тоже заложники своих привычек. Среди сторонников Windows тоже немало людей, которых можно назвать яркими поклонниками этой системы. Это хорошо, именно такие энтузиасты способны поддерживать, развивать и популяризировать любимую операционную систему. Но часто их любовь перерастает в культ. А культ определенного дистрибутива, будь то Linux или MS Windows, приводит к предвзятому отношению и необъ-

активным оценкам работы в альтернативной системе.

Истинная свобода заключается НЕ в том, чтобы выбрать одну операционную систему (ОС) навек: свободную или проприетарную, с открытым исходным кодом или закрытую, бесплатную или платную. Истинная свобода заключается в том, что мы должны уметь выбрать подходящую ОС, исходя из решаемых нами задач в соответствии с определенным критерием.

Каков же критерий? У каждого он свой. Перечислим некоторые из них:

- Я знаю эту ОС хорошо, следовательно, буду решать поставленную задачу с помощью имеющихся знаний о ней.
- Эта ОС работает с любым типом оборудования, поэтому решение будет строиться на ней.
- Эта система обеспечивает наилучший уровень безопасности, ведь это критично в нашей системе.
- Эта ОС универсальна, и если что, на ее основе можно построить другое решение, когда текущая задача будет решена.
- Я принимаю только решения на основе бесплатных или открытых решений.
- Я выбираю наиболее эффективную с точки зрения функционирования систему для данной задачи.
- Я выбираю наиболее экономически эффективную систему по внедрению и эксплуатации.

Можно продолжать, уточняя критерии. Выделим главное: композитный критерий, включающий три компоненты:

- Время внедрения.
- Стоимость внедрения и эксплуатации.
- Эффективность (обычно быстродействие и надежность) работы системы в рамках решаемых задач.

Исходя из задач могут быть добавлены и дополнительные критерии, такие, как безопасность, удобство управления системой и т.п. Но указанные три критерия будут присутствовать в любом проекте.

Например, для создания сетевой инфраструктуры, скорее всего, воспользуемся оборудованием и операционной системой Cisco, потому что именно она предназначена и оптимизированна для решения сетевых задач. Её эффективность доказана многими проектными решениями. Правда, стоимость ее внедрения может остановить; если масштаб задачи не слишком велик, тогда её экономическая эффективность не будет оправдана.

Очевидно, что нельзя выбрать одну ОС для решения всех задач. Каждая ОС имеет свои преимущества и недостатки. Чтобы получить свободу выбора, мы должны в целом представлять достоинства, недостатки каждой системы или хотя бы представителей семейства. Те же рассуждения применимы и к системам управления базами данных (СУБД), языкам программирования и другим программным продуктам.

У Microsoft и Sun Microsystems разные бизнес-модели. Не нужно думать, что какие-то фирмы, исходя из своей доброй воли и альтруистических побуждений,

делают использование своего программного обеспечения свободным и бесплатным (Sun, IBM). Если они от этого не будут в перспективе получать прибыль (как правило, от продажи поддержки), то они просто разорятся и некому будет поддерживать это открытое и бесплатное ПО. Время энтузиастов из университетских городков, создающих такое ПО для своих научных нужд, закончилось. Нет, конечно, такие люди остались, но их мало, и они не в состоянии в одиночку вести борьбу за свободу и открытость ПО.

С точки зрения эффективности обучения, нужно выбирать наиболее перспективные универсальные ОС, лидеров в своем направлении, с учетом заложенного в них потенциала развития. С другой стороны, нужно избежать повторов и слишком сильных пересечений в курсах по администрированию и программированию, особенно по программированию. Каковы цели вуза? Создать конкурентный продукт на рынке труда. Выпускник должен обладать такими знаниями, которые позволили бы ему быстро принять наиболее эффективное решение поставленных перед ним задач. Он должен обладать способностью к изучению новых перспективных технологий.

Подведем итоги, распротранив выводы с ОС на языки, среды программирования и СУБД:

1. Учить нужно разным ОС.
2. Этих ОС не должно быть много (не более трех). Предлагаемый список на настоящий момент, с которым можно спорить: MS Windows Server 2003/2008, OpenSolaris 2008.11/ 2009.05, CiscoIOS v.12. С точки зрения администрирования важно изучать именно серверных представителей и иметь пример специализированной ОС, например CiscoIOS.
3. Это должны быть наиболее интересные и перспективные в дальнейшем своем развитии ОС. Все три указанные системы обладают этими свойствами.
4. Учить нужно как администрированию этих ОС, так и программированию под них, если это возможно (Cisco IOS закрытая проприетарная система, под неё программировать затруднительно). Причем как скриптовым (пакетным) языкам, так и базовому языку, на котором написано ядро системы и драйверы. Это, как правило, язык C.
5. Программирование на языках высокого уровня необходимо изучать на межплатформенном универсальном языке. Выбор языка под задачу важен, но наша задача – обучение, а потому важна именно универсальность и многоплатформенность. Хорошо подходит для этого Java. C# и платформу .NET – придется отбросить в виду их явном замыкании на продукции Microsoft. Существующие реализации .NET типа Mono на Linux будут постоянно отставать в развитии от генеральной линии Microsoft.

Есть, конечно, масса интересных языков, это и JRuby, и Io, и много других замечательных языков. Но пока по статистике Java является самым

популярным языком программирования [2], большая часть корпоративных приложений пишется на Java, переубедить отказаться от Java будет трудно.

SQL как язык взаимодействия с базами данных необходимо изучать там, где идет речь о СУБД.

Обязательно включить в программу курса (или отдельным курсом) изучение технологий разработки и шаблонов проектирования от GRASP и GoF [3] и далее с переключением на шаблоны проектирования для корпоративных приложений, [4]. Конечно, обязательно изучение рефакторинга [5] и методики тестирования с JUnit или CUnit в зависимости от изучаемого языка.

6. Из универсальных скриптовых языков стоит изучать: 1) PHP, как наиболее распространенный язык для web-программирования; 2) Python, по мнению многих экспертов, один из мощнейших интерпретируемых языков, получивший широкое распространение. Кроме того, оба языка имеют хорошую связку с базами данных.
7. Учить нужно и разным СУБД, и их также не должно быть много: 1) Oracle, как лидер отрасли; 2) MySQL, как наиболее используемая для построения web-контента, универсальная и бесплатная. Возможно, из коммерческих MS SQL, но не больше. На наш взгляд, должен быть еще курс по базам данным, не привязанный к конкретным СУБД.
8. Инструменты разработчика зависят от ОС, но, к счастью, не всегда. NetBeans – многоплатформенная среда разработки (IDE) с поддержкой: Java, C, C++, PHP, Ruby и JRuby. Альтернативой является Eclipse, но с точки зрения обучения Netbeans предпочтительнее, так как использует идеологию «все в одном флаконе» и располагается в нескольких комплектах в зависимости от ваших потребностей и возможностей.
9. Все предыдущие предметы должны быть включены в основную программу обучения. А инновации следует преподавать в спецкурсах. Не все инновации приживаются, поэтому нет смысла их интегрировать в основную программу.
10. Самое трудное – подобрать замену математическим пакетам MatLab и Maple. К сожалению, из бесплатных и свободных продуктов полных аналогов со всей функциональностью найти невозможно. Наиболее сильная замена для MatLab, пожалуй, – Octave, а для Maple – Maxima.
11. Издательские системы. Выбор, очевидно, падает на свободный TEX. Open Office должен быть установлен на всех компьютерах факультета, но освоить его можно и самостоятельно.

Эти выводы представляют собой программу, которая реализуется на факультете. В настоящий момент изучение ряда дисциплин специальностей «Компьютерная безопасность» и «Вычислительные машины, комплексы, системы и сети» ведется в соответствии с этой программой: преподавание Java-технологий внесено в основную программу обучения, администрирование Cisco-оборудования изучается уже в течение пяти лет, на факультете преподаются учебные курсы Oracle. Примером успешного использования Java-технологии в предметах специализации является курс «Криптографические протоколы» [6].

## 2. Академические программы

После выбора ПО следует выбор академической программы производителя ПО. Самые известные академические программы:

- Cisco Network Academy
- Microsoft IT Academy и Microsoft Development Network Academic Alliance (MSDNAA)
- Sun Java Academy и Sun Campus Ambassador
- Oracle Academy
- IBM Academic Initiative
- Red Hat Academy
- Центр компетенции Mandriva Linux
- Kaspersky Lab Academy.

Участие в академической программе, имеет ряд преимуществ: предоставляется бесплатное обучение для преподавателей, льготное (для коммерческого ПО) и / или бесплатное предоставление продуктов для использования в учебном процессе и исследовательской работе, поддержка семинарами, льготная продажа оборудования, предоставление вакансий для выпускников и т.д.

Отметим, что на факультете уже работают академические программы Cisco, Microsoft, Sun, Oracle, Mandriva. В трех программах используется проприетарное коммерческое программное ПО: Cisco, Microsoft и Oracle. В двух – открытое с различными полусвободными и свободными лицензиями: Sun и Mandriva.

Выбор академической программы в общем случае так же сложен, как и выбор программного обеспечения, и может осуществляться на основе метода анализа иерархий [7].

## 3. Трудности

Перечислим основные трудности, с которыми приходится сталкиваться при переходе на новое ПО

- Издательские системы.
  - Требования научных издательств присылать статьи в формате Word.
  - Невозможность перехода на TeX в системе документооборота вуза.
  - Наличие заготовок и шаблонов документов в формате Word-a.
  - Несмотря на то что Open Office читает все форматы Microsoft Office, полного соответствия в форматировании не наблюдается. Особенно

это проявляется в математических формулах. В связи с этим приходится переделывать разработанные ранее методические указания и лабораторные работы.

- Операционные системы, прикладное и инструментальное ПО. Серьезных технических проблем не возникает, все проблемы решаются с помощью MSDNAA-подписки и систем виртуализации, например использования VirtualBox. Существуют проблемы психологического характера у преподавателей.
  - Необходимость перерабатывать материалы курсов с учетом специфики нового ПО.
  - Страх перед незнакомой техникой и программным обеспечением.
  - Сложившиеся психологические установки: ошибки в работе хорошо знакомого программного обеспечения расцениваются менее критично.

Обучение и переобучение может устранить переисленные проблемы. Важно, чтобы стоимость обучения и переобучения не превышали затрат на эксплуатацию коммерческого ПО.

## Заключение

Выбор ПО для учебного процесса — непростая задача. С одной стороны, необходимо обеспечить разнообразие ПО, с другой стороны, его не должно быть слишком много, так как множественный выбор снижает глубину получаемых знаний. Нужна некоторая разумная достаточность, и, как кажется автору данной статьи, этот баланс достигнут в настоящее время на ФКН.

По возможности нужно выбирать свободное ПО или хотя бы ПО с открытым исходным кодом. При отсутствии необходимого свободного или открытого ПО в рамках решаемых учебных задач, выбор падает на коммерческое ПО. Производители, лидеры своей отрасли предлагают академические программы, участие в которых позволяет минимизировать расходы по приобретению ПО и обучению и переобучению преподавателей.

Постоянно быть в курсе новых разработок, технологий, инструментов и оборудования можно только при тесном взаимодействии с производителями, что обеспечивается совместным участием в академических программах.

## ЛИТЕРАТУРА

1. Столлмен, Р. Почему «открытый код» проигрывает свободному программному обеспечению.  
URL: <http://www.gnu.org/philosophy/open-source-misses-the-point.ru.html>
2. TIОBE Programming Community Index for May 2009.  
URL: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
3. Ларман, К. Применение UML и шаблонов проектирования / К. Ларман. – Издательство: Вильямс, 2002. – 624 с.

- 
4. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер. – М.: «Вильямс», 2007. – С.544.
  5. Фаулер, М. Рефакторинг. Улучшение существующего кода / М. Фаулер. – Издательство: Символ-Плюс, 2008. – 432 с.
  6. Богаченко, Н.Ф. Использование языка Java и среды разработки NetBeans на занятиях по дисциплине «Криптографические протоколы» / Н.Ф. Богаченко, Д.Н. Лавров // Открытые информационные технологии: пути развития и внедрения: Материалы российской научно-практической конференции. – Уфа: Восточный университет, 2008. – С.65-69.
  7. Богаченко, Н.Ф. Применение метода анализа иерархий в задаче выбора академической программы / Н.Ф. Богаченко, Д.Н. Лавров // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. – Санкт-Петербург. Издательство Политехнического университета, 2008. – N.3(60). – С.187-191.