

Лекция 3. Стандарты и лицензии на программное обеспечение

3.1. Стандарты семейства UNIX

Причиной появления стандартов на операционную систему UNIX стало то, что она была перенесена на многие аппаратные платформы. Ее первые версии работали на аппаратуре PDP, но в 1976 и 1978 годах система была перенесена на Interdata и VAX. С 1977 по 1981 годы оформились две конкурирующие ветви: UNIX AT&T и BSD. Наверное, цели разработки стандартов были разными. Одна из них – узаконить главенство своей версии, а другая – обеспечить переносимость системы и прикладных программ между различными аппаратными платформами. В связи с этим говорят и о мобильности программ. Такие свойства имеют отношение как к исходным текстам программ, так и исполнимым программам.

Дальнейший материал приводится в хронологическом порядке появления стандартов.

Стандарты языка программирования C

Этот стандарт не относится непосредственно к UNIX. Но поскольку C является базовым как для этого семейства, так и других ОС, упомянем о стандарте этого языка программирования. Начало ему было положено выходом в 1978 году первой редакции книги Б.Кернигана и Д.Ритчи. Этот стандарт часто называют K&R. Программисты, авторы этого труда, работали над UNIX вместе с Кеном Томпсоном. При этом первый из них предложил название системы, а второй изобрел этот язык программирования. Соответствующий текст доступен в Интернете [45].

Однако промышленный стандарт языка программирования C был выпущен в 1989 году ANSI и имел имя X3. 159 – 1989. Вот что написано про этот стандарт [46]:

"Стандарт был принят для улучшения переносимости написанных на языке Си программ между различными типами ОС. Таким образом, в стандарт, кроме синтаксиса и семантики языка Си, вошли рекомендации по содержанию стандартной библиотеки. О наличии поддержки стандарта ANSI C говорит предопределенное символьное имя `_STDC`".

В 1988 году на основе этого стандарта языка программирования была выпущена вторая редакция книги Кернигана и Ритчи о C. Заметим, что фирмы, производящие программные продукты для разработки программ на языке C, могут формировать свой состав библиотек и даже несколько расширять состав других средств языка.

System V Interface Definition (SVID)

Другое направление развития стандартов UNIX связано с тем, что не только энтузиасты задумывались о создании "эталонов". Основные разработчики системы с появлением многих "вариантов" решили издавать собственные документы. Так появляются стандарты, выпускаемые USG, организацией, разрабатывающей документацию версий UNIX AT&T с того момента, когда для создания операционной системы была образована эта дочерняя компания. Первый документ появился в 1984 году на основе SVR2. Он имел название SVID (System V Interface Definition). Четырехтомное описание было выпущено после выхода в свет версии SVR4. Эти стандарты дополнялись набором тестовых программ SVVS (System V Verification Suite). Основное назначение этих средств состояло в том, чтобы разработчики имели возможность судить, может ли их система претендовать на имя System V [14].

Отметим, что положение дел со стандартом SVID в чем-то сходно со стандартом языка программирования C. Изданная авторами этого языка программирования книга является одним из эталонов, но не единственным. Выпущенный позже стандарт C является результатом коллективного труда, прошел этап обсуждения широкой общественности и, видимо, может

претендовать на ведущую роль в списке стандартов. Так и SVVS является набором тестов, позволяющих судить, достойна ли система соответствовать имени System V, только одной из версий UNIX. При этом не учитывается весь опыт разработки операционных систем от разных производителей.

Комитеты POSIX

Работа по оформлению стандартов UNIX началась группой энтузиастов в 1980 году. Была сформулирована цель – формально определить услуги, которые операционные системы обеспечивают приложениям. Такой стандарт программного интерфейса стал основой документа POSIX (Portable Operating System Interface for Computing Environment – переносимый интерфейс операционной системы для вычислительной среды) [14]. Первая рабочая группа POSIX была образована в 1985 году на основе UNIX-ориентированного комитета по стандартизации /usr/group, также называемой UniForum [47]. Название POSIX было предложено родоначальником GNU Ричардом Столмэном.

Ранние версии POSIX определяют множество системных сервисов, необходимых для функционирования прикладных программ, которые описаны в рамках интерфейса, специфицированного для языка C (интерфейс системных вызовов). Заложенные в нем идеи использовались комитетом ANSI (American National Standards Institute) при создании стандарта языка C, упомянутого ранее. Исходный состав функций, закладываемый в первые версии, опирался на UNIX AT&T (версия SVR4 [48]). Но в дальнейшем происходит отрыв спецификаций стандартов POSIX от этой конкретной ОС. Подход к организации системы на основе множества базовых системных функций был применен не только в UNIX (например, WinAPI фирмы Microsoft).

В 1988 году был опубликован стандарт 1003.1 – 1988, определяющий API (Application Programming Interface, программный интерфейс приложений). Через два года был принят новый вариант стандарта IEEE 1003.1 – 1990. В нем были определены общие правила программного интерфейса, как для системных вызовов, так и для библиотечных функций. Далее утверждаются дополнения к нему, определяющие сервисы для систем реального времени, "нитей" POSIX и др. Важным является стандарт POSIX 1003.2 – 1992 – определение командного интерпретатора и утилит.

Имеется перевод [1] этих двух групп документов, которые получили такие названия: POSIX.1 (интерфейс прикладных программ) и POSIX.2 (командный интерпретатор и утилиты – интерфейс пользователя). В упомянутом переводе содержатся три главы: основные понятия, системные услуги и утилиты. Глава "Системные услуги" разделена на несколько частей, каждая из которых группирует сходные по функциям услуги. Например, в одном из разделов "Базовый ввод/вывод" седьмая часть, посвященная операциям с каталогами, описывает три функции (opendir, readdir и closedir). Они определяются в четырех пунктах: "Синтаксис", "Описание", "Возвращаемое значение" и "Ошибки".

Для тех, кто знаком с алгоритмическим языком программирования C, приведем пример фрагментов описания. Фактически такое описание дает представление о том, как специфицируется "Интерфейс системных вызовов". В пункте "Синтаксис" про функцию readdir приведены такие строки:

```
#include <sys/types.h>
#include <dirent.h>
struct dirent *readdir(DIR *dirp);
```

Второй пункт ("Описание") содержит следующий текст:

"Типы и структуры данных, используемые в определениях с каталогами, определяются в файле dirent.h. Внутренний состав каталогов определяется реализацией. При чтении с помощью функции readdir формируется объект типа struct dirent, содержащий в качестве поля символьный массив d_name, в котором находится завершаемое символом NUL локальное имя файла.

Readdir читает текущий элемент каталога и устанавливает указатель позиции на следующий элемент. Открытый каталог задается указателем dirp. Элемент, содержащий пустые имена, пропускается".

А вот что приводится в пункте "Возвращаемое значение":

" Readdir при успешном завершении возвращает указатель на объект типа struct dirent, содержащий прочитанный элемент каталога. Прочитанный элемент может заноситься в статическую память и перекрывается очередным таким вызовом, примененным к тому же открытому каталогу. Вызов readdir для различных открытых каталогов не перекрывает читаемую информацию. В случае ошибки или достижении конца файла возвращается нулевой указатель".

В пункте "Ошибки стандарта" указано следующее:

" Readdir и closedir обнаруживают ошибку. [EBADF] Dirp не являются указателем на открытый каталог".

Этот пример показывает, как описываются представляемые приложением услуги. Требования к операционной системе (реализации) заключается в том, что она "...должна поддерживать все обязательные служебные программы, функции, заголовочные файлы с обеспечением специфицированного в стандарте поведения. Константа _POSIX_VERSION имеет значение 200112L [49]".

В мире компьютерных технологий существует такое словосочетание: "программирование POSIX". Этому можно научиться, используя различные руководства по системному программированию UNIX и операционным системам (например, [5]). Есть отдельная книга с таким названием [3]. Заметим, что в предисловии к этой книге сказано, что она описывает ". . . стандарт втройне . . .", так как она опирается на последнюю версию POSIX 2003 года, в основе которой три стандарта: IEEE Std 1003.1, технический стандарт Open Group и ISO/IEC 9945.

Как же проверить соответствие конкретной системы стандарту POSIX? Формализация такого вопроса не так проста, как кажется на первый взгляд. В современных версиях предлагается 4 вида соответствия (четыре семантических значения слова "соответствие": полное, международное, национальное, расширенное).

В рассматриваемых документах приводятся списки двух видов интерфейсных средств: обязательные (по возможности предполагается его компактность) и факультативные. Последние должны либо обрабатываться предписанным образом, либо возвращать фиксированное значение кода ENOSYS, означающего, что функция не реализована.

Отметим, что набор документов POSIX изменяется уже много лет. Но разработчики новых версий всегда стараются максимально сохранить преемственность с предыдущими версиями. В более свежих редакциях может появиться что-то новое. Например, в документе 2004 года были объединены четыре части [50]:

- Base Definitions volume (XBD) – определение терминов, концепций и интерфейсов, общих для всех томов данного стандарта;
- System Interfaces volume (XSH) – интерфейсы системного уровня и их привязка к языку Си, где описываются обязательные интерфейсы между прикладными

программами и операционной системой, в частности – спецификации системных вызовов;

- Shell and Utilities volume (XCU) – определение стандартных интерфейсов командного интерпретатора (т.н. POSIX-shell), а также базовой функциональности Unix-утилит;
- Rationale (Informative) volume (XRAT) – дополнительная, в том числе историческая, информация о стандарте.

Как и первые редакции, документ в своей основной части описывает группы представляемых услуг. Каждый элемент там описан в следующих пунктах: NAME (Имя), SINOPSIS (Синтаксис), DISCRIPTION (Описание), RETURN VALUE (Возвращаемое значение), ERRORS (Ошибки) и в заключении EXAMPLE (Примеры).

Современные версии стандарта определяют требования как к операционной системе, так и к прикладным программам. Приведем пример [51].

Функция `readdir()` должна возвращать указатель на структуру, относящуюся к очередному элементу каталога. Возвращаются ли элементы каталога с именами "точка" и "точка-точка", стандартом не специфицировано. В этом примере возможно четыре исхода, а требование к прикладной программе состоит в том, что она должна быть рассчитана на любой из них.

И в заключение приведем отрывок из курса лекций Сухомлинова ("ВВЕДЕНИЕ В АНАЛИЗ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ", Сухомлинов В.А. Часть V. Методология и система стандартов POSIX OSE), посвященный области применимости стандартов [52]:

"Область применимости стандартов POSIX OSE (Open System Environment) – обеспечение следующих возможностей (называемых еще свойствами открытости) для разрабатываемых информационных систем:

- Переносимость приложений на уровне исходных текстов (Application Portability at the Source Code Level), т.е. предоставление возможности переноса программ и данных, представленных на исходных текстах языков программирования, с одной платформы на другую.
- Системная интероперабельность (System Interoperability), т.е. поддержка взаимосвязанности между системами.
- Переносимость пользователей (User Portability), т.е. обеспечение возможности для пользователей работать на различных платформах без переобучения.
- Адаптируемость к новым стандартам (Accommodation of Standards), связанным с достижением целей открытости систем.
- Адаптируемость к новым информационным технологиям (Accommodation of new System Technology) на основе универсальности классификационной структуры сервисов и независимости модели от механизмов реализации.
- Масштабируемость прикладных платформ (Application Platform Scalability), отражающая возможность переноса и повторного использования прикладного программного обеспечения применительно к разным типам и конфигурациям прикладных платформ.
- Масштабируемость распределенных систем (Distributed System Scalability), отражающая возможность функционирования прикладного программного обеспечения независимо от развития топологии и ресурсов распределенных систем.
- Прозрачность реализаций (Implementation Transparency), т.е. сокрытие от пользователей за интерфейсами систем особенностей их реализации.

- Системность и точность спецификаций функциональных требований пользователей (User Functional Requirements), что обеспечивает полноту и ясность определения потребностей пользователей, в том числе в определении состава применяемых стандартов."

Это позволяет решать следующие задачи:

- интеграция информационных систем из компонент различных изготовителей;
- эффективность реализаций и разработок, благодаря точности спецификаций и соответствию стандартным решениям, отражающим передовой научно-технический уровень;
- эффективность переноса прикладного программного обеспечения, благодаря использованию стандартизованных интерфейсов и прозрачности механизмов реализации сервисов систем.

Также в стандартах формально определяются такие важные понятия операционных систем: пользователь; файл; процесс; терминал; хост; узел сети; время; языково-культурная среда. Там не приводятся формулировки такого определения, но вводятся применяемые к ним операции и присущие им атрибуты.

Всего в списке стандартов POSIX более трех десятков элементов. Их имена традиционно начинаются буквой "P", после которой расположено четырехзначное число с дополнительными символами. Существуют также групповые имена стандартов POSIX1, POSIX2 и т.д. Например, POSIX1 связан со стандартами на базовые интерфейсы ОС (P1003.1x, где вместо x либо пусто, либо символы от a до g; таким образом, в этой группе 7 документов), а POSIX3 – методы тестирования (два документа – P2003 и P2003n).

X/Open, OSF и Open Group

Основанная в 1984 году рядом компьютерных фирм организация X/Open своей основной задачей ставила согласование и утверждение для разных версий UNIX стандартов общего программного интерфейса и программной среды для приложений. В 1988 году появился документ XPG3 (X/Open Portability Guide3). Он включал POSIX 1003.1 – 1988 и стандарт на графическую систему X Window System (MTI). Этот документ был развит, включая последние идеи UNIX версии BSD и System V. Он получил название XPG4.2 [14].

X/Open объединила свои усилия с Open Software Foundation, создав The Open Group, которая до настоящего времени работает над идеологией открытых систем. С момента создания ей принадлежит торговая марка UNIX. Фирма активно работает (вместе с IEEE, ISO и IEC) над последними стандартами операционных систем.

Заметим, что OSF была образована рядом организаций в ответ на объединение Sun Microsystems и AT&T для разработки операционной системы, претендующей на универсальность. Сегодня организация The Open Group является главным держателем стандартов UNIX. Она размещает на своем сайте много самой разнообразной информации, начиная от истории описания "What is UNIX" ("Что такое UNIX") и заканчивая серией стандартов Single Unix, Unix95, Unix98, Unix03, ISO/IEC 9945:2003, а также UNIX System API Table.

В этот неправительственный консорциум входят около 200 членов. В их числе правительственные организации, учебные заведения и представители бизнеса разных стран. Приведем (в алфавитном порядке) нескольких представителей компьютерного бизнеса членов The Open Group:

AT&T	Hewlett-Packard	IBM
Intel	NEC Corporation	Oracle
Red Hat (R), Inc.	Sun Microsystems	SUSE LINUX Products GmbH

В заключение этого раздела заметим, что существует еще много менее значимых стандартов на программное обеспечение. Один из них, Linux Standard Base (LSB), создавался Free Standards Group. Но последняя объединилась с Open Source Development Labs (OSDL) и образовала новую организацию The Linux Foundation. Назовем и еще один документ – лицензию BSD (программная лицензия университета Беркли, применяемая для распространения UNIX-подобных операционных систем BSD).

3.2. Лицензии на программное обеспечение и документацию

С появлением Linux и подобных ей систем распространяемые свободно программы стали вытеснять коммерческие продукты. Для разных платформ существует много бесплатных программ семейства UNIX/Linux. Большая часть из них разрабатывалась в соответствии со специальной лицензией GPL (General Public License), которая была издана в рамках проекта GNU, начатого в 1984 году Ричардом Столменом (Richard Stallman) [16].

Информацию о Ричарде Столмене как одном из плеяды "выдающихся программистов мира" можно посмотреть в Интернете [53]. Ричард Столмен окончил Гарвардский университет по специальности "физика". Затем поступил на работу в Массачусетский технологический институт, где участвовал в нескольких проектах по разработке программного обеспечения. К примеру, он написал включенный во многие версии UNIX текстовый редактор emacs. С 1984 года он работает над проектом, первоначальной целью которого было создание на основе идей UNIX свободно распространяемой (бесплатной) операционной системы. Для ее разработки были нужны другие программные средства, например, транслятор с одного из языков программирования и редактор текстов. Но они также должны были быть бесплатными, иначе их авторы могут заявить свои права на часть созданной операционной системы.

Мысли Столмена были перенаправлены на создание новых методов разработки программного обеспечения. Для этого была создана лицензия GPL, в рамках которой разрабатываются свободно распространяемые программы. Для развития такого направления основывается FSF (Free Software Foundation), который возглавил Столмен. Его идеи заключаются в том, что программы обязательно должны иметь открытые исходные тексты. Любой программист может воспользоваться фрагментом чужой программы, но открыв исходный текст, созданный им самим. Кроме изменений, связанных с возможностью использовать чужие фрагменты, такой метод разработки программ улучшает и процедуру тестирования программ.

Удачные алгоритмы применяются многими программистами и подвергаются неоднократной и разнообразной проверке. Вообще Столмен сравнивал такой способ разработки программ с обменом кулинарными рецептами. Заметим, что разрабатываемые в соответствии с GPL программы не обязательно должны быть бесплатными. Можно включить программы других авторов как часть своего продукта и продавать последний. Конечно, при этом надо указать всех авторов всех частей проекта.

Что же такое свобода программного обеспечения по Столмену [8]?

1. Разрешается запускать программу и использовать ее по назначению в любых целях.
2. Разрешается изучать устройство программы, то, как она создана. При этом можно и даже необходимо использовать ее свободно предоставляемые исходники.

3. Разрешается копировать программу в любых количествах и распространять бесплатно всем, кому она нужна.
4. Разрешается изменять код программы, изменять ее в соответствии со своими представлениями и распространять как на коммерческой основе, так и на некоммерческой (платно или бесплатно).

Приведем и еще одну интерпретацию четырех пунктов "свободы" для разработчиков программ по Столмену [54]. Разработка свободно распространяемого ПО была очень важным шагом, но еще большей заслугой Р. Столмена следует признать создание "Стандартной Общественной Лицензии GNU" (GNU General Public License, или GPL). На русский язык это название разные авторы переводят по-разному: "Универсальная общественная лицензия", "Обобщенная Публичная Лицензия" и т.п. Но считается, что юридическую силу имеет только английский вариант этой лицензии. Основная идея GPL состоит в том, что пользователь должен обладать следующими четырьмя правами (или четырьмя свободами):

- правом запускать программу для любых целей (свобода 0);
- правом изучать устройство программы и приспособлять ее к своим потребностям (свобода 1), что предполагает доступ к исходному коду программы;
- правом распространять программу, имея возможность помочь другим (свобода 2);
- правом улучшать программу и публиковать улучшения в пользу всего сообщества (свобода 3), что тоже предполагает доступ к исходному коду программы.

Публичная лицензия первой версии была выпущена в 1989 году. Через пару лет вышла ее вторая версия, а третья была написана в 2005 году, но окончательный вариант был принят в 2007 году. Эти лицензии обозначаются так: GPL vX (где X может быть 1, 2 или 3). Из-за ограниченности размера книги приводим только название частей второй версии GPL:

0. Определения
1. Право на копирование и распространение.
2. Изменение программы.
3. Требование предоставления исходного кода.
4. Прекращение действия лицензии при нарушении ее условий.
5. Акты, означающие принятие лицензии.
6. Запрещение дополнительных ограничений при дальнейшем распространении.
7. Внешние ограничения не снимают обязательства выполнять условия лицензии.
8. Возможность географических ограничений.
9. Будущие версии GNU GPL.
10. Запросы на исключения из правил.
11. Отказ от предоставления гарантий.
12. Отказ от ответственности.

Иногда, в противовес правам на интеллектуальную собственность (в том числе и на программы), именуемым *copyright*, программы, распространяемые в соответствии с лицензией, разработанной Столменом, связывают с термином *copyleft* (копилефт-лицензии).

Также сегодня, в противовес чисто коммерческому направлению разработки и распространения программного обеспечения, существует и другое направление – "открытые исходники" (Open Source). Его определение сформулировал Брюс Перенс (Bruce Perens) в 1997 году. С изменениями оно было опубликовано на сайте [54]. В Интернете об этом движении

много самой разнообразной информации. Дадим только одну ссылку [55], содержащую адреса этой тематики в Рунете.

Отметим, что Open Source не эквивалентен GNU или FSF. Яркие последователи каждого из них часто высказывают свое несогласие между собой. Сам же разработчик Linux (Торвальдс) старается держаться "поодаль" от перечисленных и других подобных движений. Эти два термина отличаются расстановкой приоритетов. Сторонники open source делают упор на эффективность открытых исходников как метода разработки, модернизации и сопровождении программ. Сторонники free software считают, что именно права на свободное распространение, модификацию и изучение программ является достоинством свободного ПО.

Linux – один из самых ярких представителей программного продукта, реализованного по методу открытых исходников. Но в этой разработке есть и нечто большее. Об этом ярко написал Эрик С. Рэймонд в статье "Базар и Собор" (The Cathedral and the Bazaar) Русский перевод можно найти в [56]. Там в противовес централизованному методу разработки программ предлагается другой метод – параллельный. При его использовании, разрабатывая программу, надо публиковать ее исходный текст с ранних стадий. Тогда создаются условия участия в проекте, например, на уровне обсуждения идей или частичной отладки, многих программистов. Об этом можно прочитать и статью Безрукова [57, 58].

Open Source имеет как много сторонников, так и противников. Его сторонники собираются на различные мероприятия, обсуждают свои проблемы в открытой печати и Интернете. Среди противников, что естественно, находим, прежде всего, представителей компьютерного бизнеса. Глава Microsoft неоднократно высказывался об Open Source. Например, в интернете есть публикация "Гейтс о бесплатном ПО" [59].

Исходные тексты своих программ публикуют и самые мощные представители компьютерного бизнеса. Это сделали, к примеру, Sun и даже Microsoft. Правда, последнюю фирму вряд ли можно "заподозрить" в приверженности к Open Source. Просто они оказались вынужденными передать исходные тексты своих программ, например, операционной системы Windows, под давлением [60].

Заметим, что параллельно с выпуском GPL v2 был разработан и в 1991 году оформлен документ, названный GNU Lesser General Public License (англ. "Стандартная общественная лицензия ограниченного применения GNU", сокращенно – GNU LGPL). Она была основана на GNU Library General Public License (англ. "Стандартная общественная лицензия GNU для библиотек"). Эти лицензии действуют на свободное программное обеспечение и одобрены Фондом свободного программного обеспечения. Их цель – достигнуть компромисса между GPL и простыми разрешительными лицензиями (например, BSD License, MIT License, Mozilla Public License). LGPL была написана в 1991 году, а затем обновлена в 1999 и 2007 годах Ричардом Столлманом и Эбеном Могиленом. На странице "Лицензии открытого ПО" Википедии приведен список из более 50 элементов. Естественно, это создает определенные трудности.

В семейства GNU есть еще одна лицензия. Ее имя FDL, а с описанием можно познакомиться на http://ru.wikipedia.org/wiki/GNU_Free_Documentation_License – "Свободная лицензия GNU на документацию". Может рассматриваться как дополнение к основной лицензии GPL.

Эта копилефт-лицензия первоначально разрабатывалась для пользовательских руководств, учебников и документации, сопровождающей программы для компьютеров. Как и основная лицензия GNU (GPL), предполагает возможность воспроизведения, распространения и изменения исходных документов (в том числе и в коммерческих целях). При этом обязательно указывать авторов первоисточника. Заметим, что последний может содержать неизменяемые разделы.