

*Открытые системы:  
история и эволюция*

Автор: Абуталипов РН  
Кафедра: Информационные  
системы

## Общие ссылки по открытым системам

---

1. <http://www.intuit.ru/department/se/pposix/>
2. <http://www.intuit.ru/department/se/posix2/>
3. <http://www.ecolan.ru/terms.htm>
4. <http://www.opennet.ru/standard.shtml>
5. <http://www.opengroup.org/certification/idx/posix.html>

---

# История вопроса

- 
- Когда сетевая обработка стала реальностью, пользователи начали обращать внимание на совместимость и возможность интеграции как на необходимые атрибуты открытых систем
  - ISO в 1977-78 годах развернула интенсивные работы по созданию стандартов взаимосвязи в сетях открытых систем
  - На основе архитектуры SNA в ходе этих работ была создана семиуровневая модель взаимосвязи открытых систем OSI - Open Systems Interconnection Basic Reference Model
-

- Модель взаимосвязи открытых систем описывает общие принципы взаимосвязи открытых систем и используется в качестве основы для разработки стандартов ISO
- Тогда же впервые было введено определение открытой информационной системы.



В 1982 году был сделан первый  
RISC-процессор

- 
- Это событие не вызвало в то время больших откликов, однако оно в значительной степени определило развитие открытых систем до конца десятилетия и играет решающую роль и сегодня
-

- 
- RISC архитектура обеспечила существенное повышение производительности микропроцессоров
  - RISC архитектура предоставила аппаратную базу для реализации эффективной переносимости программ для процессоров разных производителей
-

---

RISC процессоры вне зависимости от конкретных реализаций, принадлежащих различным производителям, имеют ряд общих, особенностей

---

- 
- Большинство машинных команд исполняется за один машинный цикл
  - Регистровая архитектура: все команды обработки выбирают операнды и помещают результаты только в регистрах
-

- 
- Обращение к ОЗУ производится только командами чтения/записи
  - Конвейерная обработка команд, которая позволяет исполнять несколько команд одновременно
  - Простой фиксированный формат команд с небольшим количеством методов адресации ОЗУ
-

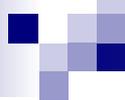
- 
- Наличие быстродействующей памяти: любая RISC машина имеет, по крайней мере, 32 регистра общего назначения и большую кэш-память
  - Упрощенный набор команд, из которого исключены многоцикловые команды, предназначенные для реализации сложных функций, что обеспечивает предельно короткий машинный цикл и высокое быстродействие аппаратуры.
-

- 
- Характерная для архитектуры RISC элементарность набора команд позволяет приблизить эффективность программ, написанных на языках высокого уровня, к эффективности программ в машинном коде и автоматизировать процесс настройки программ для их оптимизации
-

- 
- Использование стандартных компиляторов сделало возможным обеспечить на уровне языков высокого уровня эффективную мобильность программ
  - RISC процессоры обеспечили идеальные условия и для массового внедрения операционной системы (ОС) UNIX
-

- 
- С появлением мощных RISC-микропроцессоров с 32-х разрядной архитектурой UNIX проявила себя как наиболее перспективная открытая операционная среда.
  - Исторически ОС UNIX оказалась самым жизненным вариантом для создания общей базы переносимости.
  - Она удовлетворяет большинству требований, предъявляемых к открытым системам.
-

- 
- Прикладные программы, создаваемые для работы в UNIX, при определенных условиях могут иметь весьма высокую переносимость как в другие UNIX-подобные системы, так, во многих случаях, и в системы, удовлетворяющие стандартам на интерфейсы
-



Одна из причин рассматривать систему UNIX в качестве базовой ОС для использования в открытых системах состоит в том, что эта ОС почти целиком написана на языке высокого уровня, модульна и относительно гибка

- 
- ОС UNIX составлена из основных компонентов, включающих ядро, инструментальные утилиты и оболочку
  - Ядро, составляющее сердцевину UNIX`а, состоит из относительно маленького набора программ, предоставляющих системные ресурсы и непосредственно взаимодействующих с аппаратурой
-

# УТИЛИТЫ

- Утилиты - программы внешнего по отношению к ядру уровня - выполняют основные действия по обработке данных, обращаясь в определенной последовательности к процедурам ядра
- Отдельные утилиты, решающие простые задачи, могут объединяться с другими утилитами для выполнения более сложных действий

# ОБОЛОЧКА

- Оболочка предоставляет пользовательский интерфейс и действует в точности так же, как и любая другая программа
- Поскольку она не интегрирована в ядро, ее можно разработать заново при изменении требований.

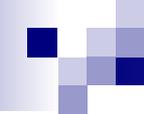
---

Хотя ОС UNIX машинно-  
независима, программы, которые  
реализуют некоторые службы, и  
часть кода зависят от аппаратуры

---



Прикладные системы,  
использующие особенности  
конкретной версии UNIX,  
также как в MS-DOS,  
реализационно зависимы.

- 
- 
- Привлекательный аспект, связанный с ОС UNIX, также состоит в том, что компания AT&T готова предоставлять лицензии на нее.

- 
- Однако это приводит также и к появлению множества различных и несовместимых реализаций
  - Не все поставщики выбирали лицензионные продукты, останавливаясь вместо этого на разработке систем подобных UNIX с различной степенью совместимости
-

- 
- Деятельность ряда организаций, таких как UniForum, POSIX и X/Open, направлена на поиск общего функционального ядра, которое позволило бы достичь переносимости между различными системами
-

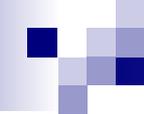
---

# Понятие подхода Открытых Систем

# IEEE POSIX 1003.0

- Открытая система - это система, реализующая открытые спецификации на интерфейсы, службы и форматы данных, достаточные для того, чтобы обеспечить:
  - - возможность переноса (мобильность) прикладных систем, разработанных должным образом, с минимальными изменениями на широкий диапазон систем;
  - - совместную работу (интероперабельность) с другими прикладными системами на локальных и удаленных платформах;
  - - взаимодействие с пользователями в стиле, облегчающем последним переход от системы к системе (мобильность пользователей)

- 
- Ключевой момент в этом определении - использование термина "**открытая спецификация**", что в свою очередь определяется как *"общедоступная спецификация, которая поддерживается открытым, гласным согласительным процессом, направленным на постоянную адаптацию новой технологии, и соответствует стандартам"*
-

- 
- 
- Согласно этому определению, открытая спецификация не зависит от конкретной технологии, т.е. не зависит от конкретных технических или программных средств или продуктов отдельных производителей.

- 
- Открытая спецификация одинаково доступна любой заинтересованной стороне
  - Более того, открытые спецификации находятся под контролем общественного мнения, так что все заинтересованные стороны могут принимать участие в ее развитии
-

# Общие свойства открытых систем

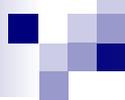
- расширяемость/масштабируемость - extensibility/scalability,
- мобильность (переносимость) - portability,
- интероперабельность (способность к взаимодействию с другими системами) - interoperability,
- дружелюбность к пользователю, в т.ч. - легкая управляемость - driveability.

- В идеале, в мире открытых систем:

- любые прикладные программы должны исполняться на любых системах,
- персонал должен либо очень мало, либо практически совсем не переучиваться при переходе от одной среды к другой
- все системы должны иметь возможность связываться друг с другом для свободного распространения и обмена информацией

Архитектура открытой системы оказывается иерархическим описанием ее внешнего облика и каждого компонента с точки зрения:

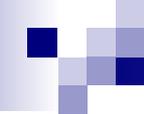
- пользователя (пользовательский интерфейс),
- проектировщика системы (среда проектирования),
- прикладного программиста (системы и инструментальные средства /среды программирования),
- системного программиста (архитектура ЭВМ),
- разработчика аппаратуры (интерфейсы оборудования).



Подход открытых систем  
пользуется успехом только  
потому, что обеспечивает  
преимущества для разного  
рода специалистов,  
связанных с областью  
компьютеров

**Для пользователя открытые системы обеспечивают следующее:**

- **НОВЫЕ ВОЗМОЖНОСТИ СОХРАНЕНИЯ СДЕЛАННЫХ ВЛОЖЕНИЙ** благодаря свойствам эволюции, постепенного развития функций систем, замены отдельных компонентов без перестройки всей системы

- 
- 
- освобождение от зависимости от одного поставщика аппаратных или программных средств, возможность выбора продуктов из предложенных на рынке при условии соблюдения поставщиком соответствующих стандартов открытых систем

- 
- дружелюбность среды, в которой работает пользователь, мобильность персонала в процессе эволюции системы;
  - возможность использования информационных ресурсов, имеющихся в других системах (организациях).
-

## Проектировщик информационных систем получает:

- возможность использования разных аппаратных платформ;
- возможность совместного использования прикладных программ, реализованных в разных операционных системах;
- развитые средства инструментальных сред, поддерживающих проектирование;
- возможности использования готовых программных продуктов и информационных ресурсов.

## Разработчики общесистемных программных средств имеют:

- новые возможности разделения труда, благодаря повторному использованию программ(reusability);
- развитые инструментальные среды и системы программирования;
- возможности модульной организации программных комплексов благодаря стандартизации программных интерфейсов.

---

# Открытые Системы и объектно-ориентированный подход



Основные свойства открытых систем хорошо поддерживаются объектно-ориентированным подходом к реализации системы

# Мобильность.

- Инкапсуляция позволяет хорошо скрыть машинно-зависимые части системы, которые должны быть реализованы заново при переходе на другую платформу.
- При этом гарантируется, что остальная часть системы не потребует изменений.
- При реализации новых машинно-зависимых частей многое может быть взято из уже существующей системы благодаря механизму наследования

# Расширяемость.

- Наследование позволяет сэкономить значительные средства при расширении системы, поскольку многое не нужно создавать заново, а некоторые новые компоненты можно получить, лишь слегка изменив старые.
- Кроме повторного использования, увеличивается также надежность, поскольку используются уже отлаженные компоненты.
- Возможность конструирования абстрактных типов данных для создания новых средств - обеспечивается самим понятием класса, объединяющего похожие объекты с одинаковым набором операций

# Интероперабельность.

- Способность системы взаимодействовать с другими системами хорошо поддерживается принципом послыки сообщения и соответствующими понятиями полиморфизма и динамического связывания
- В сообщении объекту (возможно удаленному) передается имя действия, которое должно быть им выполнено, и некоторые дополнительные аргументы сообщения

- 
- Как это действие выполнять - знает и решает только сам объект - получатель сообщения.
  - От него только требуется выдать в ответ результат.
  - Совершенно очевидно, что разные объекты будут по-разному реагировать на одинаковые сообщения (полиморфизм)
-

- Кроме того, очень удобно выбирать способ реализации в последний момент - при ответе на сообщение, в зависимости от текущего состояния системы (динамическое связывание).
- Для того, чтобы разные системы могли обмениваться сообщениями, необходима либо единая трактовка всех типов данных, в том числе абстрактных, либо индивидуальная процедура преобразования сообщения для каждой пары неодинаковых взаимодействующих систем

- 
- Простота понятия абстрактных типов данных в объектно-ориентированных системах существенно облегчает разработку такой процедуры

# Дружественность.

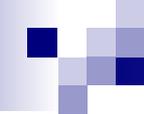
- Удобство взаимодействия человека с системой требует от последней наличия всех трех вышеуказанных качеств.
- Мобильность необходима ввиду быстрой смены старых и появления новых устройств, в частности, средств мультимедиа.
- Расширяемость требуется для разработки программной поддержки новых парадигм общения человека с машиной.
- Интероперабельность просто рассматривает человека как другую систему, с которой открытая система должна уметь взаимодействовать

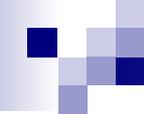
---

# Стандарты Открытых Систем

- 
- В настоящее время в мире существует несколько авторитетных сообществ, занимающихся выработкой стандартов открытых систем.
  - Однако исторически и, по-видимому, до сих пор наиболее важной деятельностью в этой области является деятельность комитетов **POSIX**
-

- Первая рабочая группа POSIX (Portable Operating System Interface) была образована в IEEE в 1985 г. на основе UNIX-ориентированного комитета по стандартизации /usr/group (ныне UniForum)

- 
- 
- Однако постепенно тематика работы рабочих групп POSIX (а со временем их стало несколько) расширилась настолько, что стало возможным говорить не о стандартной ОС UNIX, а о POSIX-совместимых операционных средах, имея в виду любую операционную среду, интерфейсы которых соответствуют спецификациям POSIX



Сейчас функционируют и  
регулярно выпускают  
документы следующие  
рабочие группы POSIX

---

- **POSIX 1003.0.**

- Рабочая группа, выпускающая "Руководство по POSIX-совместимым средам Открытых Систем".

- Это руководство содержит сводную информацию о работе и текущем состоянии документов всех других рабочих групп POSIX, а также других тематически связанных организаций, связанных со стандартизацией интерфейсов Открытых Систем.

- **POSIX 1003.1.**

- Интерфейсы системного уровня и их привязка к языку Си.

- В документах этой рабочей группы определяются обязательные интерфейсы между прикладной программой и операционной системой. С выпуска первой версии этого документа началась работа POSIX, и он в наибольшей степени связан с ОС UNIX, хотя в настоящее время интерфейсы 1003.1 поддерживаются в любой операционной среде, претендующей на соответствие принципам Открытых Систем

---

- **POSIX 1003.2. Shell и утилиты.**
- Рабочая группа специфицирует стандартный командный язык shell, основанный главным образом на Bourne shell, но включающий некоторые черты Korn shell.
- Кроме того, в документах этой рабочей группы специфицировано около 80 утилит, которые можно вызывать из процедур shell или прямо из прикладных программ.
- В документах серии 1003.2a описываются дополнительные средства, позволяющие пользователям работать с системой с помощью только ASCII-терминалов.
- **POSIX 1003.3.**
- Общие методы проверки совместимости с POSIX.
- Целью рабочей группы является разработка методологии проверки соответствия реализаций стандартам POSIX.
- Документы рабочей группы используются в различных организациях при разработке тестовых наборов.

- 
- **POSIX 1003.4.**
  - Средства, предоставляемые системой для прикладных программ реального времени.
  - В соответствии с определением 1003.4, системой реального времени считается система, обеспечивающая предсказуемое и ограниченное время реакции.
  - Работа ведется в трех секциях: файловые системы реального времени, согласованные многопоточковые (multithread) архитектуры, а также в секции, занимающейся такими вопросами, как семафоры и сигналы.
  - **POSIX 1003.5.**
  - Привязка языка Ада к стандартам POSIX.
  - В документах этой рабочей группы определяются правила привязки программ, написанных на языке Ада, к системным средствам, определенным в POSIX 1003.1.
-

- 
- **POSIX 1003.6.**
  - Расширения POSIX, связанные с безопасностью.
  - Разрабатываемый набор стандартов базируется на критериях министерства обороны США и будет определять безопасную среду POSIX.
  - **POSIX 1003.7.**
  - Расширения, связанные с администрированием системы.
  - Стандарт, разрабатываемый рабочей группой, будет определять общий интерфейс системного администрирования, в частности, разнородных сетей. Отправной точкой является модель OSI.
-

- 
- **POSIX 1003.8.**
  - Прозрачный доступ к файлам.
  - Будут обеспечены интерфейсы и семантика прозрачного доступа к файлам, распределенным в сети.
  - Работа основывается на анализе существующих механизмов: NFS, RFS, AFS и FTAM.
  - **POSIX 1003.9.**
  - Привязка языка Фортран.
  - Определяются правила привязки прикладных программ, написанных на языке Фортран, к основным системным средствам.
-

- 
- **POSIX 1003.10.**
  - Общие черты прикладной среды суперкомпьютеров (Application Environment Profile - AEP).
  - **POSIX 1003.11.**
  - Общие черты прикладной среды обработки транзакций (On-line Transaction Processing Application Environment - OLTP).
-

- 
- **POSIX 1003.12.**
  - Независимые от протоколов коммуникационные интерфейсы.
  - Разрабатываются два стандартных набора интерфейсов для независимых от сетевых протоколов коммуникаций "процесс-процесс".
  - Результаты должны обеспечивать единообразную работу с TCP/IP, OSI и другими системами коммуникаций.
  - **POSIX 1003.13.**
  - Общие черты прикладных сред реального времени. POSIX 1003.14.
  - Общие черты прикладных сред мультипроцессоров.
  - Помимо прочего, должны быть предложены соответствующие расширения стандартов других рабочих групп.
-

---

- **POSIX 1003.15.**

- Расширения, связанные с пакетной обработкой.
- Определяются интерфейсы пользователя и администратора и сетевые протоколы для пакетной обработки.

- **POSIX 1003.16.**

- Привязка языка Си.
  - Задача проекта, выполняемого реально рабочей группой 1003.1, состоит в выработке правил привязки международного стандарта языка Си (ISO 9989) к независимым от языка интерфейсам, определяемым POSIX 1003.1-1990 (ISO 9945-1).
-

- 
- **POSIX 1003.17.**
  - Справочные услуги и пространство имен.
  - Задачей рабочей группы является анализ и выработка рекомендаций по работе со справочниками и пространством имен в контексте X.500.
  - **POSIX 1003.18.**
  - Общие черты среды POSIX-платформы.
  - В одном документе должны быть специфицированы основные характеристики интерактивной многопользовательской прикладной платформы, соответствующей стандартам POSIX.
  - Работа выполняется группой 1003.1.
-