

Архитектура операционных систем

Лекция 2

Требования к современным ОС

- **Главные требования:**
- **выполнение основных функций эффективного управления ресурсами**
- **обеспечение удобного интерфейса для пользователя и прикладных программ**

Требования к современным ОС

1. **Расширяемость**
2. **Переносимость или многоплатформенность**
3. **Совместимость**
4. **Надежность и отказоустойчивость**
5. **Безопасность**
6. **Производительность**

1. Расширяемость ОС

- ОС всегда изменяются со временем эволюционно, и эти изменения более значимы, чем изменения аппаратных средств
- Если код ОС написан таким образом, что дополнения и изменения могут вноситься без нарушения целостности системы, то такую ОС называют расширяемой
- Расширяемость достигается за счет модульной структуры ОС (программы строятся из набора отдельных модулей, взаимодействующих только через функциональный интерфейс)

2. Переносимость или многоплатформенность

- В идеале код ОС должен легко переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на аппаратную платформу другого типа
- Переносимые ОС имеют несколько вариантов реализации для разных платформ

3. Совместимость

- Для пользователя, переходящего с одной ОС на другую, очень привлекательна возможность запуска в новой ОС привычного приложения.
- Если ОС имеет средства для выполнения прикладных программ, написанных для других ОС, то про нее говорят, что она обладает совместимостью с этими ОС
- Понятие совместимости включает также поддержку пользовательских интерфейсов других ОС

4. Надежность и отказоустойчивость

- ОС должна быть защищена как от внутренних, так и от внешних ошибок, сбоев и отказов.
- Действия ОС должны быть всегда предсказуемыми, а приложения не должны иметь возможности наносить вред ОС.
- Надежность и отказоустойчивость ОС прежде всего определяются архитектурными решениями, положенными в ее основу, а также качеством ее реализации
- Важно, включает ли ОС программную поддержку аппаратных средств обеспечения отказоустойчивости, таких, например, как дисковые массивы или источники бесперебойного питания.

5. Безопасность

- Современная ОС должна защищать данные и другие ресурсы вычислительной системы от несанкционированного доступа.
 - средства аутентификации — определения легальности пользователей
 - авторизации — предоставления легальным пользователям дифференцированных прав доступа к ресурсам
 - аудита — фиксации всех «подозрительных» для безопасности системы событий
- В сетевых ОС к задаче контроля доступа добавляется задача защиты данных, передаваемых по сети.

6. Производительность

- ОС должна обладать настолько хорошим быстродействием и временем реакции, насколько это позволяет аппаратная платформа
- На производительность ОС влияют:
 - архитектура ОС
 - многообразии функций
 - качество программирования кода
 - возможность исполнения ОС на высокопроизводительной (многопроцессорной) платформе

Основные принципы построения операционных систем

1. Модульности
2. Функциональной избирательности
3. Генерируемости
4. Функциональной избыточности
5. Виртуализации
6. Независимости программ от внешних устройств
7. Совместимости
8. Открытой и наращиваемой ОС
9. Мобильности (переносимости)
10. Обеспечения безопасности вычислений

Принцип модульности ОС

- Под модулем в общем случае понимают функционально законченный элемент системы, выполненный в соответствии с принятыми межмодульными интерфейсами.
- Модуль предполагает возможность без труда заменить его на другой при наличии заданных интерфейсов
- Наибольший эффект от его использования достигим, когда принцип распространен одновременно на операционную систему, прикладные программы и аппаратуру

Принцип функциональной избирательности

- В ОС выделяется некоторая часть важных модулей, которые должны постоянно находиться в оперативной памяти для более эффективной организации вычислительного процесса (ядро)
- При формировании состава ядра требуется учитывать два противоречивых требования.
 - 1) В состав ядра должны войти наиболее часто используемые системные модули.
 - 2) Количество модулей должно быть таковым, чтобы объем памяти, занимаемый ядром, был бы не слишком большим.
- **Транзитные программные модули загружаются в оперативную память только при необходимости и в случае отсутствия свободного пространства могут быть замещены другими транзитными модулями.**

Принцип генерируемости ОС

- Этот принцип позволяет настраивать центральную системную управляющую программу ОС, исходя из конкретной конфигурации конкретного вычислительного комплекса и круга решаемых задач.
- Эта процедура проводится редко, перед протяженным периодом эксплуатации ОС.
- Процесс генерации осуществляется с помощью специальной программы-генератора и соответствующего входного языка для этой программы, позволяющего описывать программные возможности системы и конфигурацию машины.
- **Принцип генерируемости существенно упрощает настройку ОС на требуемую конфигурацию вычислительной системы**

Принцип функциональной избыточности

- Этот принцип учитывает возможность проведения одной и той же работы различными средствами
- Позволяет:
 - быстро и наиболее адекватно адаптировать ОС к определенной конфигурации вычислительной системы
 - обеспечить максимально эффективную загрузку технических средств при решении конкретного класса задач
 - получить максимальную производительность при решении заданного класса задач

Принцип виртуализации

- Этот принцип позволяет представить структуру системы в виде определенного набора планировщиков процессов и распределителей ресурсов (мониторов) и использовать единую централизованную схему распределения ресурсов.
- Наиболее естественным и законченным проявлением концепции виртуальности является понятие *виртуальной машины*.
- Любая ОС скрывает от пользователя и его приложений реальные аппаратные и иные ресурсы, заменяя их некоторой абстракцией.

Принцип виртуализации

- Виртуальная машина, предоставляемая пользователю, воспроизводит архитектуру реальной машины, но архитектурные элементы в таком представлении выступают с новыми или улучшенными характеристиками:
 - единообразная по логике работы память (виртуальная) практически неограниченного объема.
 - произвольное количество процессоров (виртуальных), способных работать параллельно и взаимодействовать во время работы.
 - произвольное количество внешних устройств (виртуальных), способных работать с памятью виртуальной машины параллельно или последовательно, асинхронно или синхронно по отношению к работе того или иного виртуального процессора, которые инициируют работу этих устройств.

Принцип независимости программ от внешних устройств

- Связь программ с конкретными устройствами производится не на уровне трансляции программы, а в период планирования её исполнения
- В результате перекомпиляция при работе программы с новым устройством, на котором располагаются данные, не требуется.
- Принцип позволяет одинаково осуществлять операции управления внешними устройствами независимо от их конкретных физических характеристик

Принцип совместимости

- Это способность ОС выполнять программы, написанные для других ОС или для более ранних версий данной операционной системы, а также для другой аппаратной платформы.
- Двоичная совместимость достигается, когда можно запустить исполняемую программу на выполнение на другой ОС. Для этого необходимы:
 - совместимость на уровне команд процессора,
 - совместимость на уровне системных вызовов и даже на уровне библиотечных вызовов, если они являются динамически связываемыми.
- Совместимость на уровне исходных текстов требует наличия соответствующего транслятора в составе системного программного обеспечения, а также совместимости на уровне библиотек и системных вызовов. При этом необходима перекомпиляция имеющихся исходных текстов в новый выполняемый модуль.

Принцип открытой и наращиваемой ОС

- Открытая ОС доступна для анализа как пользователям, так и системным специалистам, обслуживающим вычислительную систему.
- Наращиваемая (модифицируемая, развиваемая) ОС позволяет не только использовать возможности генерации, но и вводить в ее состав новые модули, совершенствовать существующие и т. д.
- Необходимо, чтобы можно было внести дополнения и изменения, и не нарушить целостность системы.

Принцип мобильности (переносимости)

- Операционная система относительно легко должна переноситься с процессора одного типа на процессор другого типа и с аппаратной платформы одного типа на аппаратную платформу другого типа.
 - Большая часть ОС должна быть написана на языке, который имеется на всех системах, на которые планируется в дальнейшем ее переносить. То есть ОС должна быть написана на языке высокого уровня, предпочтительно стандартизованном.
 - Важно минимизировать или исключить части кода, которые непосредственно взаимодействуют с аппаратными средствами.
 - Если аппаратно-зависимый код не может быть полностью исключен, то он должен быть изолирован в нескольких хорошо локализуемых модулях. Аппаратно-зависимый код не должен быть распределен по всей системе.

Принцип обеспечения безопасности вычислений

- Правила безопасности определяют свойства:
 - защита ресурсов одного пользователя от других
 - установление квот по ресурсам для предотвращения захвата одним пользователем всех системных ресурсов
- Более безопасные системы не только снижают эффективность, но и существенно ограничивают число доступных прикладных пакетов, которые соответствующим образом могут выполняться в подобной системе

Классификация ОС

по числу одновременно выполняемых задач

- однозадачные (MS DOS) и
- многозадачные
- Системы пакетной обработки (ОС ЕС)
- Системы с разделением времени (Unix, Linux, Windows)
- Системы реального времени (RT11, QNX)

Классификация ОС

по числу одновременно работающих пользователей на ЭВМ ОС разделяются на

- **однопользовательские (MS DOS);**
- **многопользовательские (Unix, Linux, Windows 95 - XP)**

Классификация ОС

по типу лицензии:

- проприетарная (семейство Windows)
- открытая (большинство Linux и UNIX систем).

Проприетарное ПО - ПО, распространяемое с условиями, запрещающими его свободное дальнейшее распространение, использование получателем в собственном ПО, изучение, декомпиляцию, внесение изменений, либо требующими для таких действий специального отдельного соглашения с поставщиком или производителем ПО.

Классификация ОС

по архитектуре:

- микроядерные (VxWorks, QNX);
- монолитные (Windows XP);
- гибридные (Windows NT, большинство Linux);

Классификация ОС

по использованию процессора:

- однопроцессорные;
- многопроцессорные системы (начиная с OS/2, Net Ware, Widows NT, большинство современных ОС).

Классификация ОС

по применению:

- рабочих станций (DOS, MAC OS, Windows 98, XP, Vista),
- серверов (AIX, Windows 2000, Windows Server 2003, Windows Vista Server 2008),
- ОС реального времени;
- встроенные ОС (VxWorks, QNX, Nucleus),
- для мобильных устройств (Windows CE, Pocket PC, Windows Mobile, Palm OS, Symbian OS),
- для сетевых маршрутизаторов (IOS от Cisco),

Классификация ОС

ПО ВОЗМОЖНОСТИ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ:

- локальные (DOS);
- сетевые (Netware 3.x – 6.x, UNIX, Linux, FreeBSD).

ОС как система управления ресурсами

основные ресурсы современных вычислительных систем:

- Процессоры
- Основная память
- Таймеры
- Наборы данных
- Диски
- Принтеры
- Сетевые устройства
- и др.

Ресурсы распределяются между процессами.

Процесс

- Процесс (задача) – базовое понятие большинства современных ОС.
- *Процесс – программа в стадии выполнения.*
- Программа – это статический объект, представляющий собой файл с кодами и данными.
- Процесс — это динамический объект, который возникает в ОС после того, как пользователь или ОС решает «запустить программу на выполнение», то есть создать новую единицу вычислительной работы.

Задачи ОС по управлению ресурсами

Управление ресурсами вычислительной системы с целью наиболее эффективного их использования является назначением ОС.

Управление ресурсами включает решение следующих общих, не зависящих от типа ресурса задач:

- планирование ресурса — то есть определение, какому процессу, когда и в каком количестве (если ресурс может выделяться частями) следует выделить данный ресурс;
- удовлетворение запросов на ресурсы;
- отслеживание состояния и учет использования ресурса — то есть поддержание оперативной информации о том, занят или свободен ресурс и какая доля ресурса уже распределена;
- разрешение конфликтов между процессами.