



**Лекция 6. Операционные системы**

**NetCracker®**  
© 2013 NetCracker Technology Corporation Confidential

Операционные системы и  
основы проектирования  
информационных систем

## План лекции

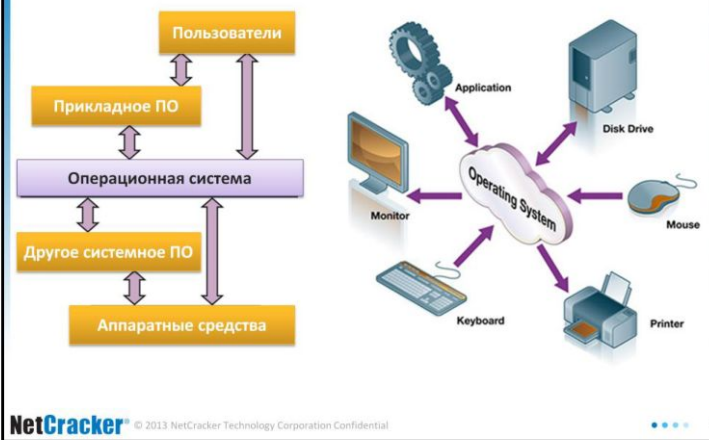
Основные функции операционных систем .

Классификация ОС.

Архитектурные особенности ОС.

Командная строка Windows.

## Основные функции операционных систем



## Основные функции операционных систем

### Операционная система

комплекс управляющих и обрабатывающих программ, которые:

- выступают как интерфейс между устройствами вычислительной системы и прикладными программами;
- предназначены для управления устройствами, вычислительными процессами, эффективного распределения вычислительных ресурсов между вычислительными процессами и организации надёжных вычислений.

ОС управляет  
четырьмя  
ключевыми  
асpekтами работы  
компьютера:

*процессами*

*распределением памяти*

*операциями ввода и вывода файлов*

*устройствами ввода/вывода*

NetCracker® © 2013 NetCracker Technology Corporation Confidential

••••• 4

Операционная система – это то, что обеспечивает среду для исполнения прикладных программ. Она обычно включает множество встроенных подпрограмм называемых функциями, которые могут вызывать приложения, чтобы выполнить основные задачи типа отображения строки текста на экране или вводе символа с клавиатуры. Можно ли написать программное обеспечение для компьютера без операционной системы? Да. Но это очень трудоемкая задача. Дело в том, что такое приложение должно решать тысячи задач, которые реализует операционная система. Кроме того, такое приложение было бы приспособлено для очень специфической аппаратной конфигурации.

## Классификация ОС

По числу одновременно выполняемых задач:

Однозадачные

Многозадачные

По числу одновременно работающих пользователей:

Однопользовательские

Многопользовательские

По назначению:

Локальные

Сетевые



В однозадачных системах используются средства управления периферийными устройствами, средства управления файлами, средства общения с пользователями. Многозадачные ОС используют все средства, которые характерны для однозадачных, и, кроме того, управляют разделением совместно используемых ресурсов: процессор, ОЗУ, файлы и внешние устройства.

#### **Невытесняющая многозадачность**

операционная система одновременно загружает в память два или более приложений, но процессорное время предоставляется только основному приложению. Для выполнения фонового приложения оно должно быть активизировано.

При *невытесняющей* многозадачности активный вычислительный процесс выполняется до тех пор, пока сама прикладная программа по собственной инициативе не отдаст указание операционной системе выбрать из очереди другой готовый к выполнению процесс. При *вытесняющей* многозадачности решение о переключении процессора с одного активного вычислительного процесса на другой принимается самой ОС, а не прикладной программой.

#### **Вытесняющая многозадачность**

операционная система сама передает управление от одной выполняемой программы другой в случае завершения операций ввода-вывода, возникновения событий в аппаратуре компьютера, истечения таймеров и квантов времени, или же поступлений тех или иных сигналов от одной программы к другой. В этом виде многозадачности процессор может быть переключен с исполнения одной программы на исполнение другой без всякого пожелания первой программы и буквально между любыми двумя инструкциями в её коде. Распределение процессорного времени осуществляется планировщиком процессов. К тому же каждой задаче может быть назначен пользователем или самой операционной системой определенный приоритет, что обеспечивает гибкое управление распределением процессорного времени между задачами (например, можно снизить приоритет ресурсоёмкой программе, снизив тем самым скорость её работы, но повысив производительность фоновых процессов). Этот вид многозадачности обеспечивает более быстрый отклик на действия пользователя.

## Классификация ОС



Наиболее существенно отличие заключается в наличии у многопользовательских систем механизмов защиты персональных данных каждого пользователя.

## Классификация ОС



Сетевая операционная система — операционная система со встроенными возможностями для работы в компьютерных сетях. Сетевая операционная система необходима для управления потоками сообщений между рабочими станциями и серверами. Она может позволить любой рабочей станции работать с разделяемым сетевым диском или принтером, которые физически не подключены к этой станции.



## Архитектурные особенности ОС

### ЯДРО

- центральная часть ОС, обеспечивающая приложениям координированный доступ к ресурсам компьютера, таким как :

- процессорное время,
- память,
- внешнее аппаратное обеспечение,
- внешнее устройство ввода и вывода информации.

Также обычно ядро предоставляет сервисы файловой системы и сетевых протоколов.

## Архитектурные особенности ОС

### Типы архитектур ядер операционных систем

Монолитное ядро

Модульное ядро

Микроядро

Экзодро

Наноядро

Гибридное ядро

## Архитектурные особенности ОС

### Монолитное ядро

**ОС с монолитным ядром** - система, в которой все ее компоненты являются составными частями системы, используют общие структуры данных и взаимодействуют друг с другом путем непосредственного вызова.

Для монолитной операционной системы ядро совпадает со всей системой.

**Примеры:** Традиционные ядра UNIX (такие как BSD), Linux; ядро MS-DOS, ядро KolibriOS.

Во многих операционных системах с монолитным ядром сборка ядра, то есть его компиляция, осуществляется для каждого компьютера, на который устанавливается операционная система. При этом можно выбрать список оборудования и программных протоколов, поддержка которых будет включена в ядро. Так как ядро является единой программой, перекомпиляция – это единственный способ добавить в него новые компоненты или исключить неиспользуемые. Следует отметить, что присутствие в ядре лишних компонентов крайне нежелательно, так как ядро всегда полностью располагается в оперативной памяти. Кроме того, исключение ненужных компонентов повышает надежность операционной системы в целом. Монолитное ядро – старейший способ организации операционных систем. Примером систем с монолитным ядром является большинство Unix-систем.

**Достоинства:** Скорость работы, упрощенная разработка модулей.

**Недостатки:** Поскольку всё ядро работает в одном адресном пространстве, сбой в одном из компонентов может нарушить работоспособность всей системы.

Примеры: Традиционные ядра UNIX (такие как BSD), Linux; ядро MS-DOS, ядро KolibriOS.

## Архитектурные особенности ОС

### Модульное ядро

современная, усовершенствованная модификация архитектуры монолитных ядер операционных систем компьютеров.

не требуют полной перекомпиляции ядра при изменении состава аппаратного обеспечения компьютера. Вместо этого модульные ядра предоставляют тот или иной механизм подгрузки модулей ядра, поддерживающих то или иное аппаратное обеспечение (например, драйверов). При этом подгрузка модулей может быть как динамической, так и статической.

Пример: VFS - «виртуальная файловая система».

Модульные ядра удобнее для разработки, чем традиционные монолитные ядра, не поддерживающие динамическую загрузку модулей, так как от разработчика не требуется многократная полная перекомпиляция ядра при работе над какой-либо его подсистемой или драйвером. Выявление, локализация, отладка и устранение ошибок при тестировании также облегчаются.

## Архитектурные особенности ОС

### Микроядро

Такая система, в которой большинство ее составляющих являются самостоятельными программами.

Микроядро работает в привилегированном режиме и обеспечивает взаимодействие между программами, планирование использования процессора, первичную обработку обращений к сервисам ОС, операции ввода-вывода и базовое управление памятью.

**Примеры:** Symbian OS; Windows CE; OpenVMS; Mach, используемый в GNU/Hurd и Mac OS X; QNX; AIX; Minix; ChorusOS; AmigaOS; MorphOS.

Остальные компоненты системы взаимодействуют друг с другом путем передачи сообщений через микроядро.

Основное достоинство микроядерной архитектуры — высокая степень модульности ядра операционной системы. Это существенно упрощает добавление в него новых компонентов. В микроядерной операционной системе можно, не прерывая ее работы, загружать и выгружать новые драйверы, файловые системы и т. д. Существенно упрощается процесс отладки компонентов ядра, так как новая версия драйвера может загружаться без перезапуска всей операционной системы. Микроядерная архитектура повышает надежность системы, поскольку ошибка на уровне непривилегированной программы менее опасна, чем отказ на уровне режима ядра.

В то же время микроядерная архитектура операционной системы вносит дополнительные накладные расходы, связанные с передачей сообщений, что существенно влияет на производительность. Для того чтобы микроядерная операционная система по скорости не уступала операционным системам на базе монолитного ядра, требуется очень тщательно проектировать разбиение системы на компоненты, стараясь минимизировать взаимодействие между ними.

**Достоинства:** Устойчивость к сбоям оборудования, ошибкам в компонентах системы.

Основное достоинство микроядерной архитектуры — высокая степень модульности ядра операционной системы. Это существенно упрощает добавление в него новых компонентов. В микроядерной операционной системе можно, не прерывая её работы, загружать и выгружать новые драйверы, файловые системы и т. д. Существенно упрощается процесс отладки компонентов ядра, так как новая версия драйвера может загружаться без перезапуска всей операционной системы. Компоненты ядра операционной системы ничем принципиально не отличаются от пользовательских

программ, поэтому для их отладки можно применять обычные средства.

Микроядерная архитектура повышает надежность системы, поскольку ошибка на уровне непривилегированной программы менее опасна, чем отказ на уровне режима ядра.

**Недостатки:** Передача данных между процессами требует накладных расходов.

## Архитектурные особенности ОС

### Экзоядро

ядро операционной системы компьютеров, предоставляющее лишь функции для взаимодействия между процессами и безопасного выделения и освобождения ресурсов.

Предоставляет лишь набор сервисов для взаимодействия между приложениями, а также необходимый минимум функций, связанных с защитой: выделение и высвобождение ресурсов, контроль прав доступа, и т. д.

Экзоядро не занимается предоставлением абстракций для физических ресурсов — эти функции выносятся в библиотеку пользовательского уровня (так называемую libOS).

NetCracker © 2013 NetCracker Technology Corporation Confidential

•••• 14

Основная идея операционной системы на основе экзоядра состоит в том, что ядро должно выполнять лишь функции координатора для небольших процессов, связанных только одним ограничением — экзоядро должно иметь возможность гарантировать безопасное выделение и освобождение ресурсов оборудования. В отличие от ОС на основе микроядра, ОС, базирующиеся на экзоядре, обеспечивают гораздо большую эффективность за счет отсутствия необходимости в переключении между процессами при каждом обращении к оборудованию.

Архитектуры на основе экзоядер являются дальнейшим развитием и усовершенствованием микроядерных архитектур и одновременно ужесточают требования к минималистичности и простоте кода ядра.

## Архитектурные особенности ОС

### Наноядро

архитектура ядра операционной системы, в рамках которой крайне упрощённое и минималистичное ядро выполняет лишь одну задачу — обработку аппаратных прерываний, генерируемых устройствами компьютера.

После обработки прерываний от аппаратуры наноядро, в свою очередь, посылает информацию о результатах обработки (например, полученные с клавиатуры символы) вышележащему программному обеспечению при помощи того же механизма прерываний.

Пример: KeyKOS.



## Архитектурные особенности ОС

### Гибридное ядро

модифицированные микроядра, позволяющие для ускорения работы запускать «несущественные» части в пространстве ядра.

Сочетает в себе элементы микроядерной и монолитной архитектуры. Часть функций, традиционно выполняемых ядром операционной системы, реализована в её монолитном ядре, однако некоторые реализованы в виде отдельных программ, исполняемых в своих собственных адресных пространствах, что соответствует идеям микроядерной архитектуры.

**Примеры:** Windows NT, Syllable, BeOS, Haiku, NetWare, Plan 9, ReactOS.

Все рассмотренные подходы к построению операционных систем имеют свои достоинства и недостатки. В большинстве случаев современные операционные системы используют различные комбинации этих подходов. Так, например сейчас, ядро «Linux» представляет собой монолитную систему с отдельными элементами модульного ядра. При компиляции ядра можно разрешить динамическую загрузку и выгрузку очень многих компонентов ядра — так называемых модулей. В момент загрузки модуля его код загружается на уровне системы и связывается с остальной частью ядра. Внутри модуля могут использоваться любые экспортируемые ядром функции.

Наиболее тесно элементы микроядерной архитектуры и элементы монолитного ядра переплетены в ядре Windows NT. Хотя Windows NT часто называют микроядерной операционной системой, это не совсем так. Микроядро NT слишком велико (более 1 Мбайт, кроме того, в ядре системы находится, например, ещё и модуль графического интерфейса), чтобы носить приставку «микро». Компоненты ядра Windows NT располагаются в вытесняемой памяти и взаимодействуют друг с другом путем передачи сообщений, как и положено в микроядерных операционных системах. В то же время все компоненты ядра работают в одном адресном пространстве и активно используют общие структуры данных, что свойственно операционным системам с монолитным ядром. Причина проста: чисто микроядерный дизайн коммерчески менее выгоден, поскольку менее эффективен (за счет накладных расходов на передачу сообщений там, где можно было обойтись вызовами функций).

## Командная строка Windows



### Командная строка Windows

- обеспечивает ввод команд MS-DOS и других компьютерных команд.
- позволяет выполнять задачи без помощи графического интерфейса Windows.

**Cmd.exe** — интерпретатор командной строки для операционных систем OS/2, Windows CE и для семейства Windows NT.

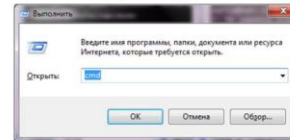
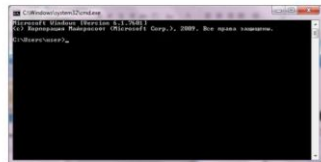
- Cmd.exe является аналогом COMMAND.COM, которая используется в семействах MS-DOS и Windows 9x.
- В Windows NT и OS/2 имеется и COMMAND.COM для совместимости со старыми программами.
- В Unix-подобных системах есть также свои аналоги.

## Командная строка Windows

Пуск -> Все программы -> Стандартные -> Командная строка

или

Пуск -> Выполнить и введите cmd



## Пакетные файлы



Пакетный файл  
(batch file)

Расширение:  
.bat, .cmd (в ОС  
семейства NT, OS/2)

Текстовый файл в MS-DOS, OS/2 или Windows, содержащий последовательность команд, предназначенных для исполнения командным интерпретатором.

После запуска пакетного файла, программа-интерпретатор читает его строка за строкой и последовательно исполняет команды.

Пакетный файл — аналог скриптовых файлов командной строки (shell script) в Unix-подобных операционных системах.

**Пакетные файлы полезны для автоматического запуска приложений и автоматизации рутинных операций.**

NetCracker © 2013 NetCracker Technology Corporation Confidential

•••• 19

Основная область применения — автоматизация наиболее рутинных операций, которые регулярно приходится совершать пользователю компьютера: например, копирование, перемещение, переименование, удаление файлов; работа с папками; архивация; создание резервных копий баз данных и т. п.

Вирусы могут быть написаны в виде пакетного файла, известны также генераторы вирусов, являющиеся пакетными файлами.

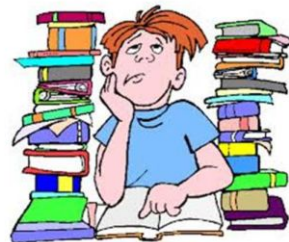
## Пакетные файлы

- Пример пакетного файла, вычисляющего выражения:

<code>:start</code>	Метка, создающая цикличность программы.
<code>@echo off</code>	Выключение вывода выполняющихся строк на экран.
<code>cls</code>	Очистка экрана.
<code>title Calculator</code>	Изменение заголовка окна Windows на строку «Калькулятор».
<code>color 71</code>	Изменение цвета шрифта и фона (тёмно-синий на светло-сером).
<code>echo Enter the equation</code>	Вывод строки «Введите уравнение».
<code>set /p exp=</code>	Создание переменной <code>exp</code> для хранения ввода пользователя.
<code>set /a result=%exp%</code>	Вычисление результата выражения и помещение его в переменную <code>result</code> .
<code>cls</code>	Очистка экрана.
<code>echo Your equation: %exp%</code>	Вывод строки «Ваше уравнение» и значения переменной <code>exp</code> .
<code>echo =%result%</code>	Вывод строки «=» и значения переменной <code>result</code> .
<code>pause&gt;nul</code>	Остановка выполнения кода до нажатия любой клавиши.
<code>goto start</code>	Переход на метку <code>start</code> , выполнение кода начнётся со следующей после метки команды.


## Дополнительные материалы

- <http://www.google.com>
- [http://kurs.ido.tpu.ru/courses/oper\\_system/tema3.htm](http://kurs.ido.tpu.ru/courses/oper_system/tema3.htm)
- [http://ru.wikipedia.org/wiki/Пакетный\\_файл](http://ru.wikipedia.org/wiki/Пакетный_файл)
- [http://en.wikipedia.org/wiki/Command\\_Prompt](http://en.wikipedia.org/wiki/Command_Prompt)



Q&A





**Thank you!**

• • • •

**NetCracker**® © 2013 NetCracker Technology Corporation Confidential 23