Тема 2

1.Что такое операционная система?
2. Архитектурные особенности ОС
3. Классификация ОС

Структура вычислительной системы и место курса в общем цикле курсов по информатике

Пользователь

Алгоритмы и алгоритмические языки

Прикладные программы

Системные программы

Прочие системные программы

Операционная система

Современные операционные системы

Техническое обеспечение

Архитектура ЭВМ и язык ассемблера

Вычислительная система как совокупность ресурсов

- 5. Уровень прикладных систем
- 4. Уровень систем программирования
- 3. Уровень управления логическими/виртуальными ресурсами вычислительной системы
- 2. Уровень управления физическими ресурсами вычислительной системы
- 1. Аппаратный уровень вычислительной системы

Что такое операционная система ?

Основные точки зрения

- Виртуальная машина
- Менеджер ресурсов
- Защитник пользователей и программ
- Постоянно функционирующее ядро

История ОС

До компьютерной эры	Конец 40-х - середина 50-х годов XX века	Середина 50-х – начало 70-х	Начало 70-х – конец 80-х	90-е годы XX века – 00-е XXI века	Наше время
Аппаратное обеспечение (Железо)	Аппаратное обеспечение (Железо)	Аппаратное обеспечение (Железо)	Аппаратное обеспечение (Железо)	Аппаратное обеспечение (Железо)	Аппаратное обеспечение (Железо)
	Программное обеспечение (ПО)	Программное обеспечение (ПО)	Программное обеспечение (ПО)	Программное обеспечение (ПО)	Программное обеспечение (ПО)
		Операционная система (ОС)	Операционная система (ОС)	Операционная система (ОС)	Операционная система (ОС)
			Сетевое ПО/ОС	Сетевое ПО/ОС	Сетевое ПО/ОС
				Интернет-ПО/ОС	Интернет-ПО/ОС
					Облачное-ПО/ОС

Краткая история эволюции вычислительных систем

1-й период (1945 г. – 1955 г.)

- Ламповые машины
- Нет разделения персонала
- Нет операционных систем
- Ввод программы с пульта или с колоды перфокарт
- Отладка программы с пульта
- Одновременное выполнение только одной операции
- Появление прообразов первых компиляторов

Научно-исследовательская работа в области вычислительной техники

Краткая история эволюции вычислительных систем

2-й период (1955 г. – начало 1960х гг.)

- Транзисторные машины
- Происходит разделение персонала
- Бурное развитие алгоритмических языков
- Ввод задания с колоды перфокарт
- Отладка программы по изучению распечаток
- Пакеты заданий и системы пакетной обработки

Начало использования ЭВМ в научных и коммерческих целях

Краткая история эволюции вычислительных систем

3-й период (начало 1960х гг. – 1980 г.)

- Машины на интегральных схемах
- Использование спулинга (spooling)
- Планирование заданий
- Мультипрограммные пакетные системы

Влияние идеи мультипрограммирования на эволюцию вычислительных систем

Software

- Планирование заданий
- Управление памятью
- Сохранение контекста
- Планирование использования процессора
- Системные вызовы
- Средства коммуникации
- Средства синхронизации

Hardware

- Защита памяти
- Сохранение контекста
- Механизм прерываний
- Привилегированные команды

Краткая история эволюции вычислительных систем

3-й период (начало 60х гг. – 1980 г.)

- Машины на интегральных схемах
- Использование спулинга (spooling)
- Планирование заданий
- Мультипрограммные пакетные системы
- Системы разделения времени (time-sharing)
- Виртуальная память
- Интерактивная отладка программ
- Развитые файловые системы
- Семейства ЭВМ

Широкое использования ЭВМ в научных и коммерческих целях

Краткая история эволюции вычислительных систем

4-й период (1980 г. – наше время)

- Машины на больших интегральных схемах (БИС)
- Персональные ЭВМ
- Дружественное программное обеспечение
- Сетевые и распределенные операционные системы

Широкое использования ЭВМ в быту, в образовании, на производстве

Основные функции, которые выполняли классические ОС в процессе своей эволюции

- Планирование заданий и использования процессора
- Обеспечение программ средствами коммуникации и синхронизации
- Управление памятью
- Управление файловой системой
- Управление вводом-выводом
- Обеспечение безопасности

Основные понятия, концепции ОС

- Системные вызовы
- Прерывания
- Исключительные ситуации
- Файлы
- Процессы, нити

Системные вызовы (system calls)

- это интерфейс между операционной системой и пользовательской программой. Они создают, удаляют и используют различные объекты, главные из которых – процессы и файлы. Пользовательская программа запрашивает сервис у операционной системы, осуществляя системный вызов. Имеются библиотеки процедур, которые загружают машинные регистры определенными параметрами и осуществляют прерывание процессора, после чего управление передается обработчику данного вызова, входящему в ядро операционной системы. Цель таких библиотек – сделать системный вызов похожим на обычный вызов подпрограммы. Основное отличие состоит в том, что при системном вызове задача переходит в привилегированный режим или режим ядра (kernel mode). Поэтому системные вызовы иногда еще называют программными прерываниями, в отличие от аппаратных прерываний, которые чаще называют просто прерываниями. В большинстве операционных систем системный вызов осуществляется командой программного преры вания (INT). Программное прерывание – это синхронное событие, которое может быть повторено при выполнении одного и того же программного кода.

Прерывание (hardware interrupt)

– это событие, генерируемое внешним (по отношению к процессору) устройством. Посредством аппаратных прерываний аппаратура либо информирует центральный процес сор о том, что произошло какое-либо событие, требующее немедленной реакции (например, пользова тель нажал клавишу), либо сообщает о завершении асинхронной операции ввода-вывода (например, за кончено чтение данных с диска в основную память). Важный тип аппаратных прерываний – прерывания таймера, которые генерируются периодически через фиксированный промежуток времени. Прерывания таймера используются операционной системой при планировании процессов. Каждый тип аппаратных прерываний имеет собственный номер, однозначно определяющий источник прерывания. Аппаратное прерывание – это асинхронное событие, то есть оно возникает вне зависимости от того, какой код испол няется процессором в данный момент. Обработка аппаратного прерывания не должна учитывать, какой процесс является текущим.

Исключительная ситуация (exception)

– событие, возникающее в результате попытки выполнения программой команды, которая по каким-то причинам не может быть выполнена до конца. Примерами таких команд могут быть попытки доступа к ресурсу при отсутствии достаточных привилегий или обращения к отсутствующей странице памяти. Исключительные ситуации, как и системные вызовы, являются синхронными событиями, возникающими в контексте текущей задачи. Исключительные ситуации можно разделить на исправимые и неисправимые. К исправимым относятся такие исключительные ситуации, как отсутствие нужной информации в оперативной памяти. После устранения причины исправимой исключительной ситуации программа может выполняться дальше. Возникновение в процессе работы операционной системы исправимых исключительных ситуаций считается нормальным явлением. Неисправимые исключительные ситуации чаще всего возникают в результате ошибок в программах (например, деление на ноль). Обычно в таких случаях операционная система реагирует завершением программы, вызвавшей исключительную ситуацию.

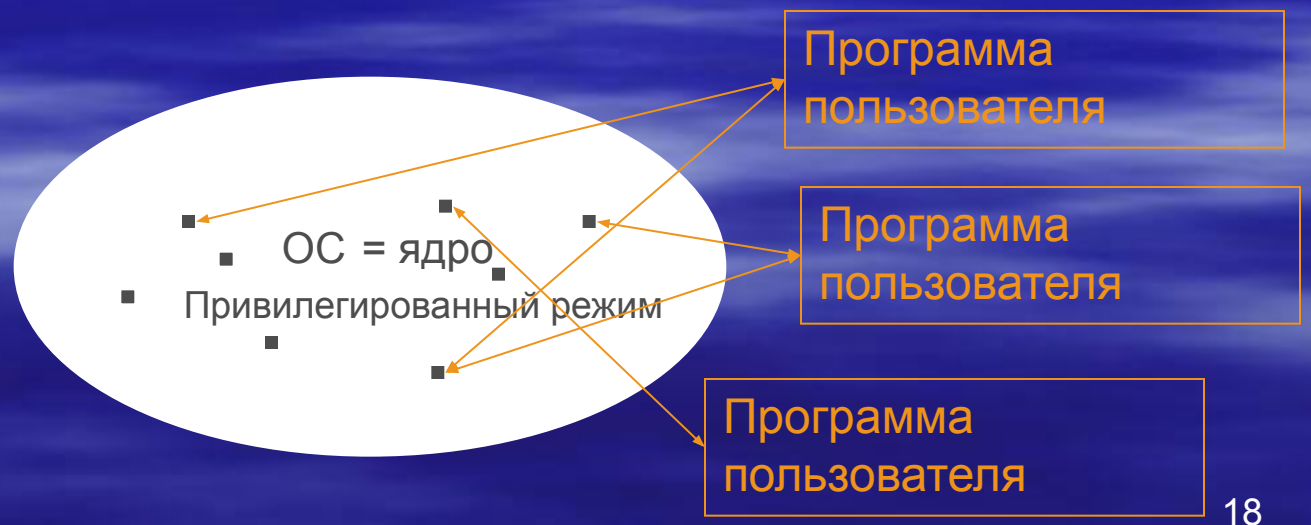
Файлы

Файлы предназначены для хранения информации на внешних носителях, то есть принято, что информация, записанная, например, на диске, должна находиться внутри файла. Обычно под файлом понимают именованную часть пространства на носителе информации. Главная задача файловой системы (file system) – скрыть особенности ввода-вывода и дать программисту простую абстрактную модель файлов, независимых от устройств. Для чтения, создания, удаления, записи, открытия и закрытия файлов также имеется обширная категория системных вызовов (создание, удаление, открытие, закрытие, чтение и т.д.). Пользователям хорошо знакомы такие связанные с организацией файловой системы понятия, как каталог, текущий каталог, корневой каталог, путь. Для манипулирования этими объектами в операционной системе имеются системные вызовы.

Внутреннее строение операционных систем

Монолитное ядро

- Каждая процедура может вызывать каждую
- Все процедуры работают в привилегированном режиме
- Ядро совпадает со всей операционной системой
- Пользовательские программы взаимодействуют с ядром через системные вызовы

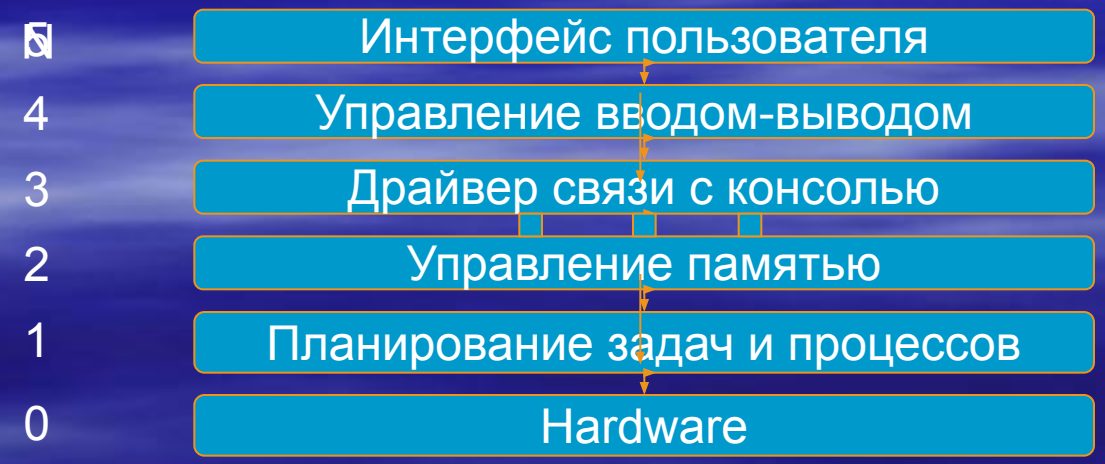


Внутреннее строение операционных систем

Многоуровневые системы (Layered systems)

- Процедура уровня N может вызывать только процедуры уровня N -1
- Все или почти все уровни работают в привилегированном режиме
- Ядро совпадает или почти совпадает со всей операционной системой
- Пользовательские программы взаимодействуют с ОС через интерфейс пользователя

Система THE



Внутреннее строение операционных систем

Микроядерная (microkernel) архитектура

Функции микроядра:

- взаимодействие между программами
- планирование использования процессора
- первичная обработка прерываний и операций ввода-вывода
- базовое управление памятью

Внутреннее строение операционных систем

Микроядерная (microkernel) архитектура

- Взаимодействие частей ОС между собой и с программами пользователей путем передачи сообщений через микроядро
- В привилегированном режиме работает только микроядро
- Микроядро составляет лишь малую часть ОС

Внутреннее строение операционных систем

Смешанные системы – почему?

- Монолитное ядро – необходимость перекомпиляции при каждом изменении, сложность отладки, высокая скорость работы.
- Многоуровневые системы – необходимость перекомпиляции при изменениях, отлаживается только измененный уровень, меньшая скорость работы
- Микроядро – простота отладки, возможность замены компонент без перекомпиляции и остановки системы, очень медленные

Внутреннее строение операционных систем

Смешанные системы – примеры

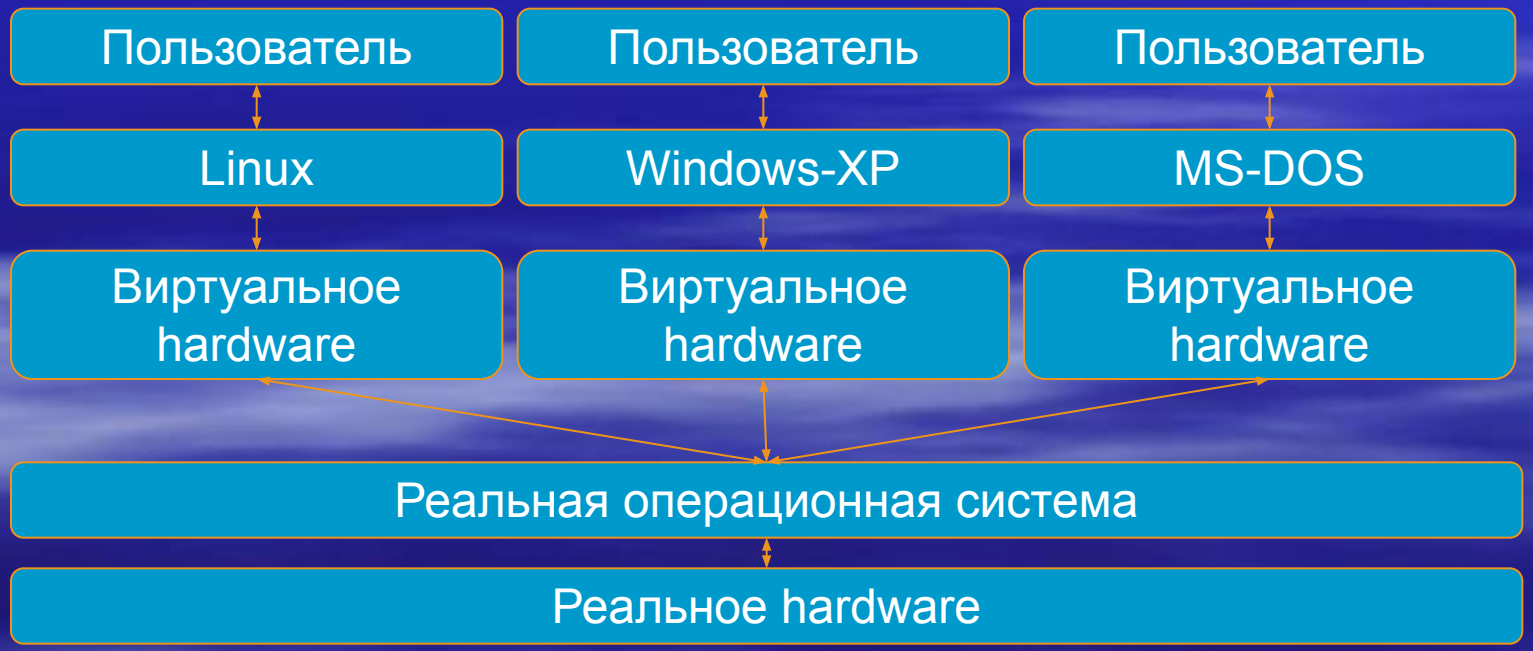
- Linux – монолитная система с элементами микроядерной архитектуры (подгружаемые модули).
- 4.4 BSD – запуск монолитной системы под управлением микроядра
- Windows NT – почти микроядерная система с элементами монолитности

Многоуровневый подход применяется почти во всех ОС в их отдельных компонентах

Внутреннее строение операционных систем

Виртуальные машины

Каждому пользователю предоставляется своя копия виртуального hardware



Классификация ОС

- По назначению ОС делятся на универсальные и специализированные
- По способу загрузки можно выделить загружаемые ОС (большинство) и системы, постоянно находящиеся в памяти вычислительной системы
- Поддержка многозадачности: однозадачные и многозадачные
- Поддержка многопользовательского режима: однопользовательские и многопользовательские
- Виды многопрограммной работы: не вытесняющая многопрограммность и вытесняющая многопрограммность
- Отсутствие или наличие средств поддержки многопроцессорной обработки
- Многопроцессорные ОС классифицируются по способу организации вычислительного процесса: асимметричные ОС и симметричные ОС

Классификация ОС

- По области использования и форме эксплуатации:
  - системы пакетной обработки;
  - системы разделения времени;
  - системы реального времени.
- По аппаратной платформе - типу вычислительной техники для которой они предназначаются:
  - Операционные системы для смарт-карт
  - Встроенные операционные системы
  - Операционные системы для персональных компьютеров
  - Операционные системы мини-ЭВМ
  - Операционные системы мэйнфреймов (больших машин)
  - Серверные операционные системы
  - Кластерные операционные системы

Операцио́нная систе́ма, сокр. **ОС** (англ. *operating system, OS*) - комплекс взаимосвязанных программ, предназначенных для управления ресурсами компьютера и организации взаимодействия с пользователем.

Операционная система, ОС (англ. operating system) базовый комплекс компьютерных программ, обеспечивающий интерфейс с пользователем, управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит. Таким образом, операционная система обеспечивает несколько видов интерфейса: интерфейса между пользователем и программно-аппаратными средствами компьютера (интерфейс пользователя); интерфейса между программным и аппаратным обеспечением (аппаратно-программный интерфейс); интерфейса между разными видами программного обеспечения (программный интерфейс).

Операционная система (ОС) - комплекс системных и управляющих программ, предназначенных для наиболее эффективного использования всех ресурсов вычислительной системы (ВС) и удобства работы с ней. Под ресурсом понимают любой логический или физический компонент ЭВМ и предоставляемые ими возможности (основными ресурсами являются процессорное время, память и устройства ввода-вывода).

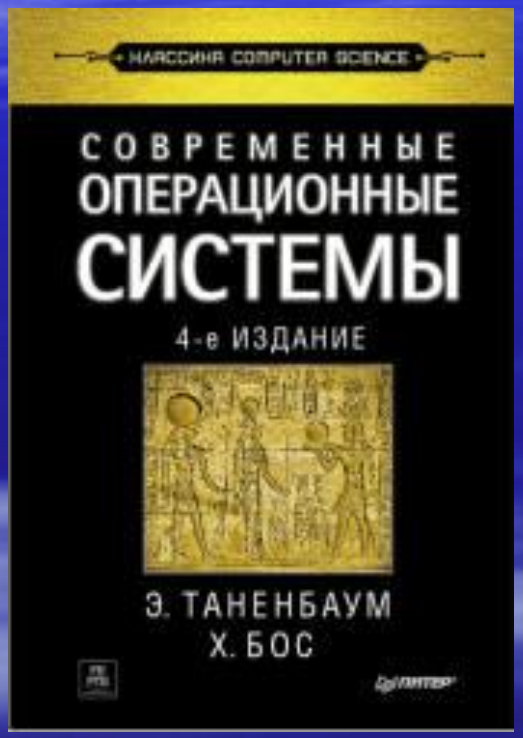
Литература

1. *Гордеев А. В.* Операционные системы: Учебник для вузов. - 2-е изд. - СПб.: Питер, 2007. - 416 с.
2. *Деннинг П. Дж., Браун Р. Л.* Операционные системы // Современный компьютер. - М., 1986.
3. *Иртегов Д. В.* Введение в операционные системы. - 2-е изд. - СПб.: BHV-СПб, 2007.
4. Карпов В. Е., Коньков К. А. Основы операционных систем. Курс лекций: учебное пособие. - Изд. 2-е, доп. и испр. - Москва : ИНТУИТ, 2005. – 531 с.
5. *Олифер В. Г., Олифер Н. А.* Сетевые операционные системы. - СПб.: Питер, 2002. - 544 с.

§ *Столлингс У.* Операционные системы = Operating Systems: Internals and Design Principles. - М.: Вильямс, 2004. - 848 с.

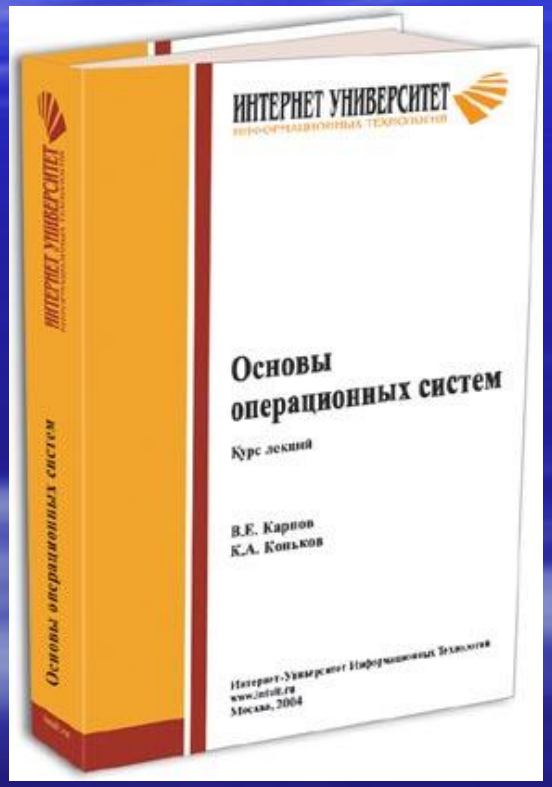
1. *Таненбаум Э. С.* Многоуровневая организация ЭВМ = Structured Computer Organization. - М.: Мир, 1979. - 547 с.
2. *Таненбаум Э. С.* Современные операционные системы = Modern Operating Systems. - 4-е изд. - СПб.: Питер, 2015. — 1120 с.
3. *Таненбаум Э. С., Вудхалл А. С.* Операционные системы. Разработка и реализация = Operating Systems: Design and Implementation. - 3-е изд. - СПб.: Питер, 2007. — 704 с.
4. *Шоу А.* Логическое проектирование операционных систем = The Logical Design of Operating Systems. - М.: Мир, 1981. - 360 с.
5. *Рэймонд Э. С.* Искусство программирования для UNIX = The Art of UNIX Programming. - М.: Вильямс, 2005. - 544 с.
6. Электронные ресурсы - https://www.intuit.ru/studies/courses/2192/31/lecture/998

Литература к курсу



Эндрю Таннебаум
Современные операционные системы, 4-е издание

Литература к курсу



В.Е.Карпов, К.А. Коньков
Основы операционных систем

Литература к курсу

Вильям Столлингс
Операционные системы
4-е издание