

Москалев А.А.

Архитектура операционных систем



Введение.

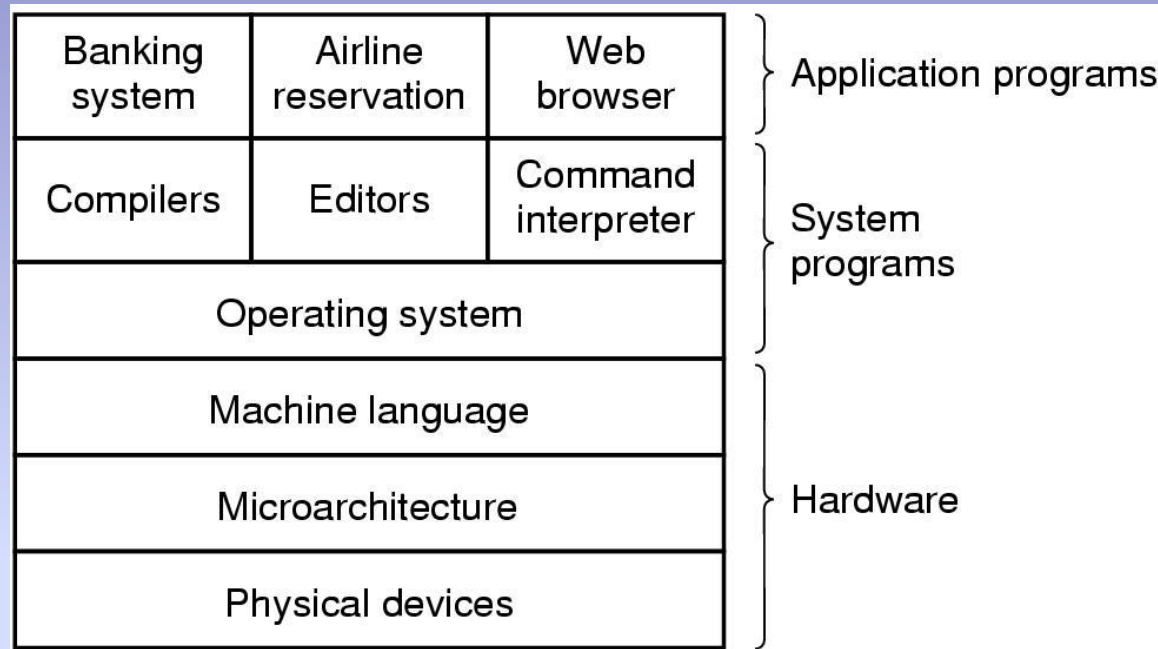
Понятие операционной системы»

Раздел 1

Введение

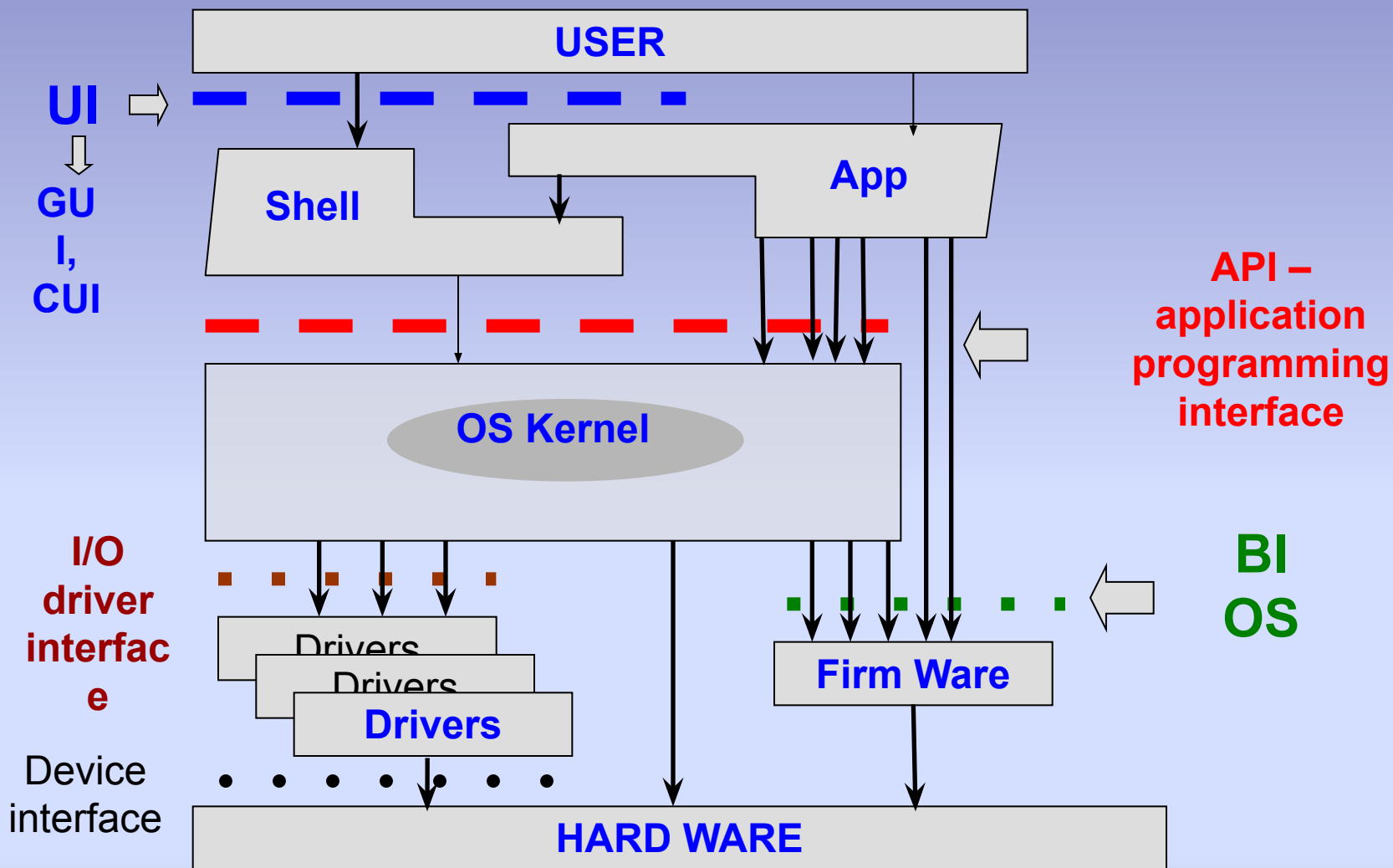
- 1.1 Понятие операционной системы
- 1.2 История развития операционных систем
- 1.3 Разновидности операционных систем
- 1.4 Аппаратные средства (повторение)
- 1.5 Концепции операционных систем
- 1.6 Системные вызовы
- 1.7 Структура операционной системы

Структура компьютерной системы



- Компьютерная система состоит из:
 - Аппаратуры (hardware)
 - Системных программ (system programs)
 - Прикладных программ (application programs)

ИНТЕРФЕЙСЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Понятие операционной системы

- ОС является расширенной машиной
 - Прячет массу мелких деталей реализации
 - Представляет пользователю виртуальную машину, более удобную в пользовании чем реальная аппаратура.
- ОС управляет ресурсами
 - Распределяет ресурсы во времени между программами
 - Распределяет ресурсы в пространстве

Понятие ресурса

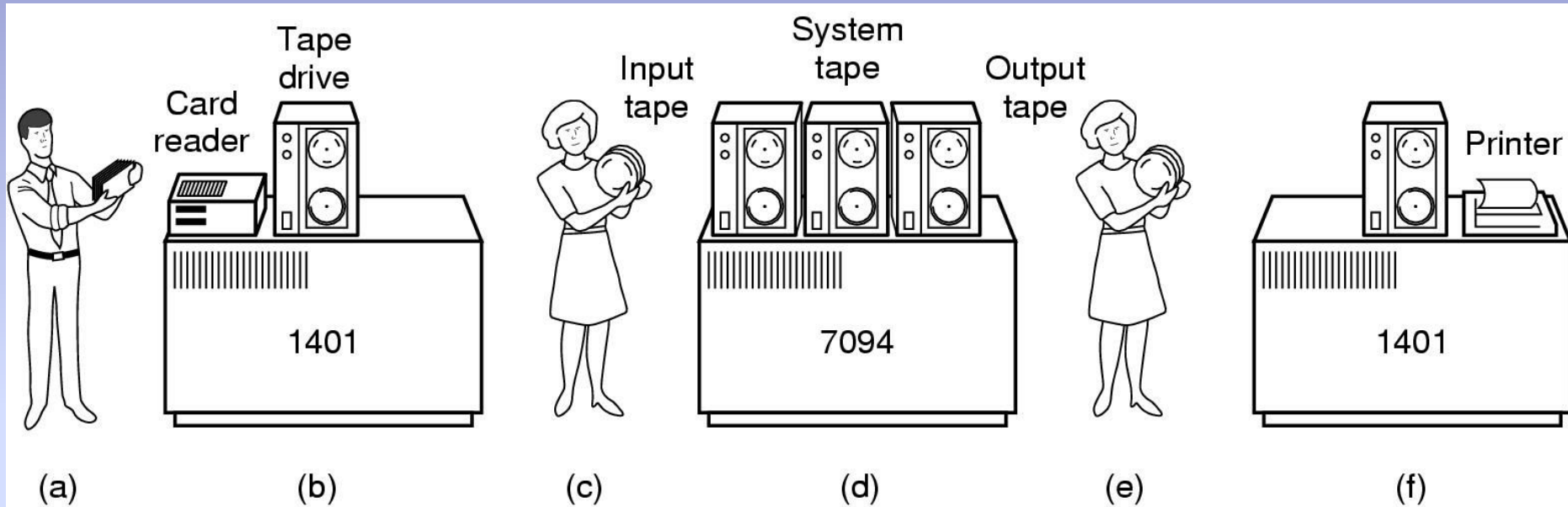
- Одна из задач ОС - обеспечить эффективный и бесконфликтный способ распределения ресурсов ЭВМ между пользователями

Всякий объект (независимо от формы его существования), обладающий некоторой практической ценностью для потребителя, является *ресурсом*

Эволюция ОС (1)

- Первое поколение 1945 - 1955
 - вакуумные лампы, коммутационные панели
- Второе поколение 1955 - 1965
 - транзисторы, пакетные системы
- Третье поколение 1965 – 1980
 - ИС и мультипрограммирование
- Четвертое поколение 1980 – наст. время
 - персональные компьютеры

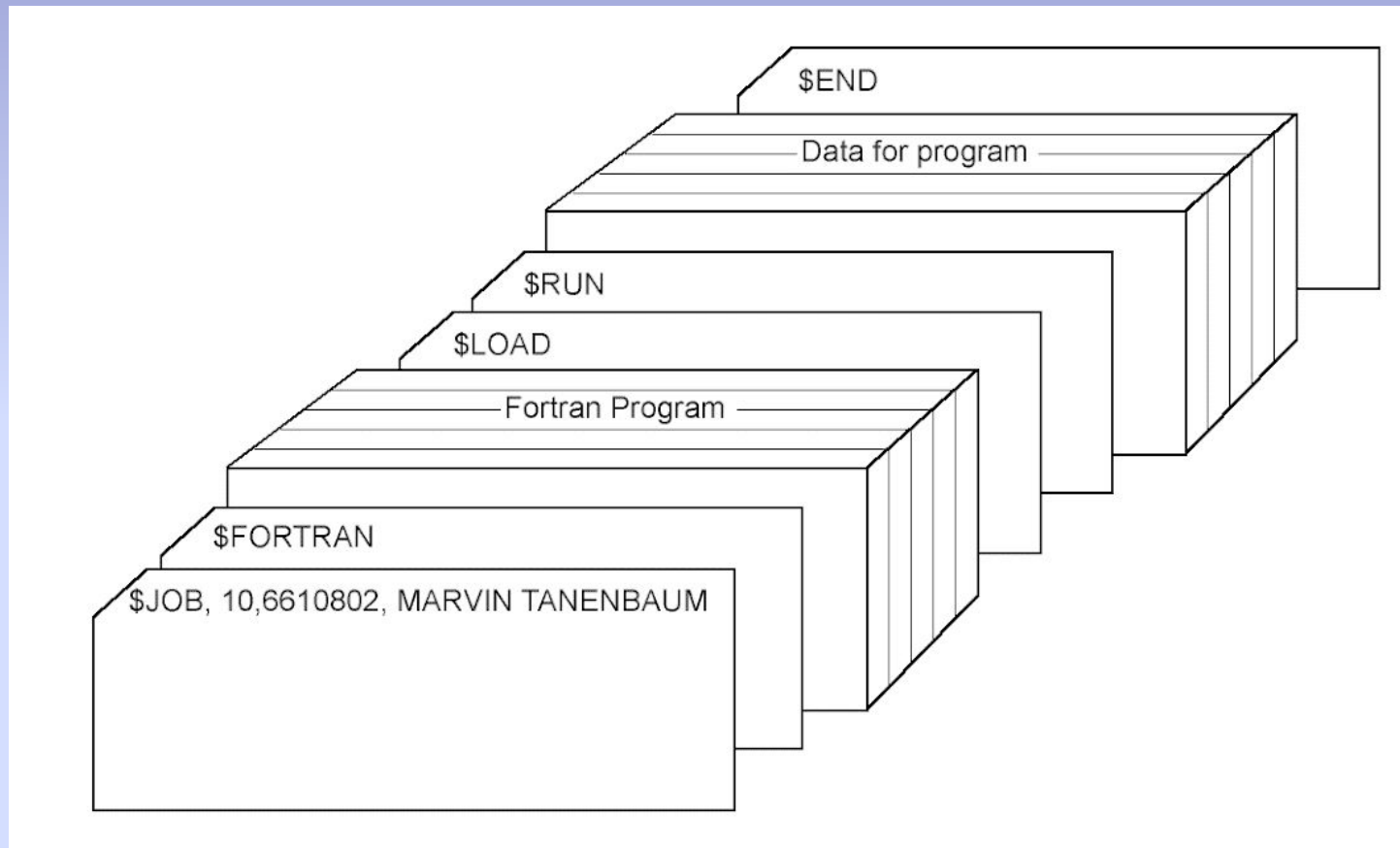
Эволюция ОС (2)



Системы пакетной обработки

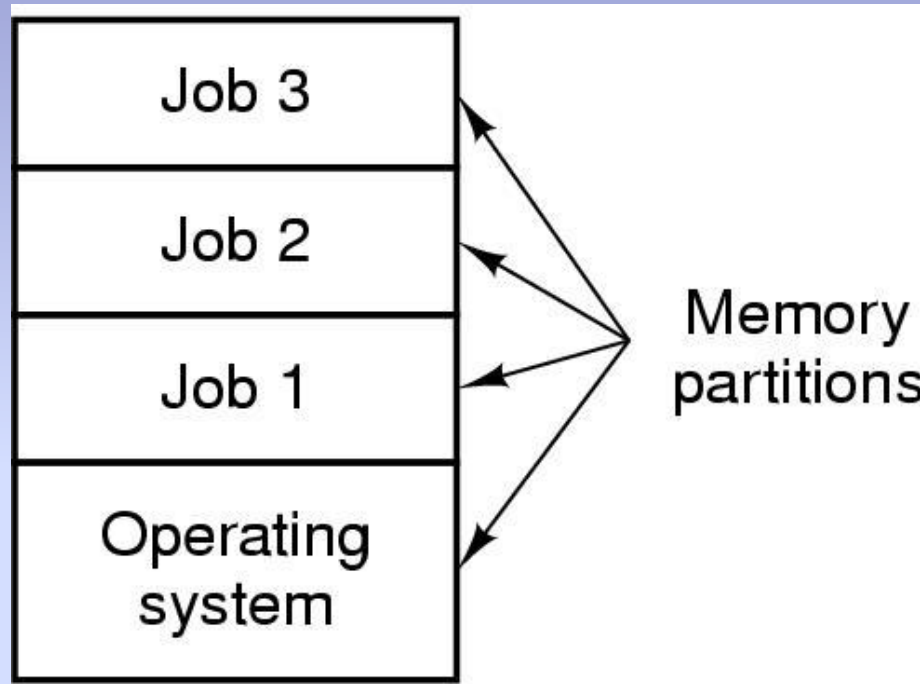
- Приносятся перфокарты на 1401
- Считываются карты на ленту
- Лента передается на 7094 где и производится обработка
- Лента с результатами передается на 1401 для вывода на печать

Эволюция ОС (3)



- Структура типового задания – 2-^е поколение

Эволюция ОС (4)

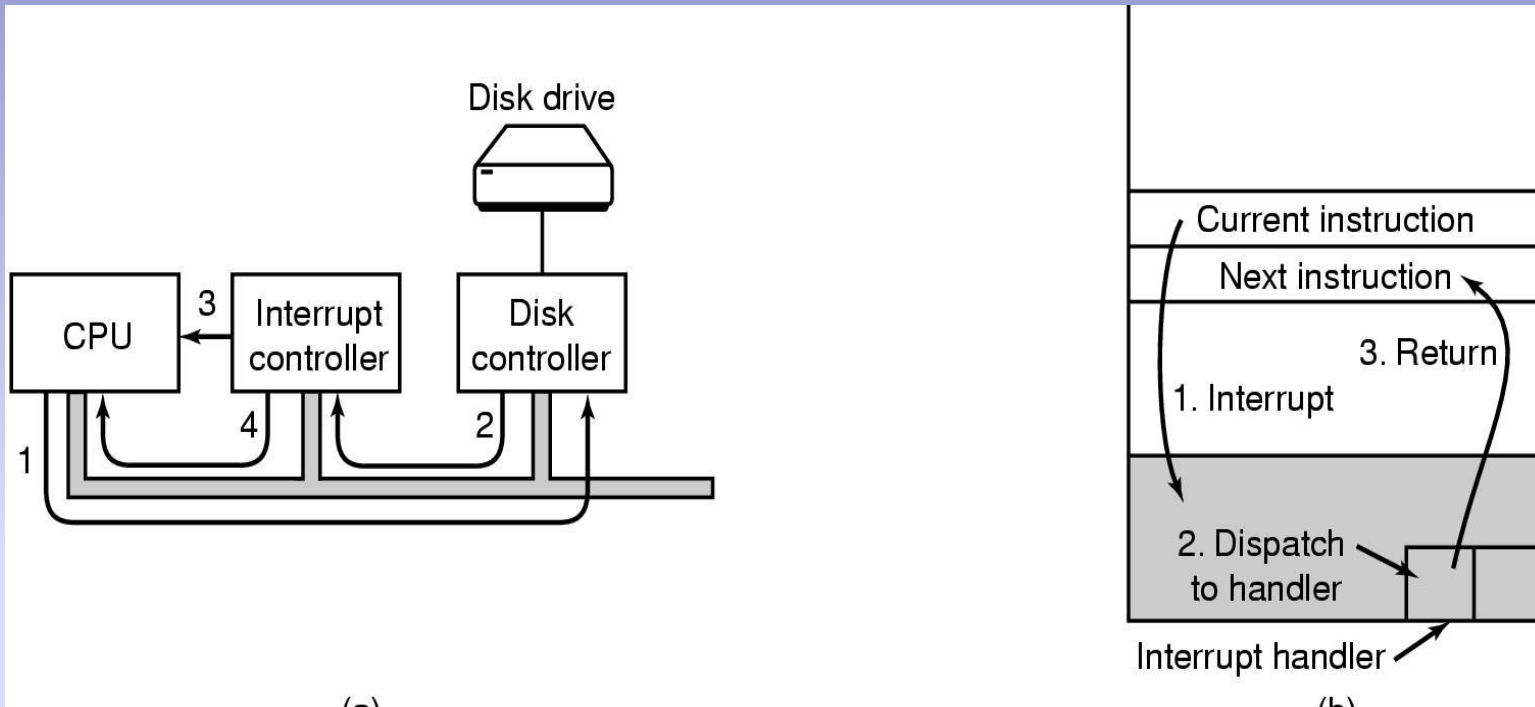


- Мультипрограммная система
 - четыре задачи в памяти – 3-^е поколение

Разновидности ОС

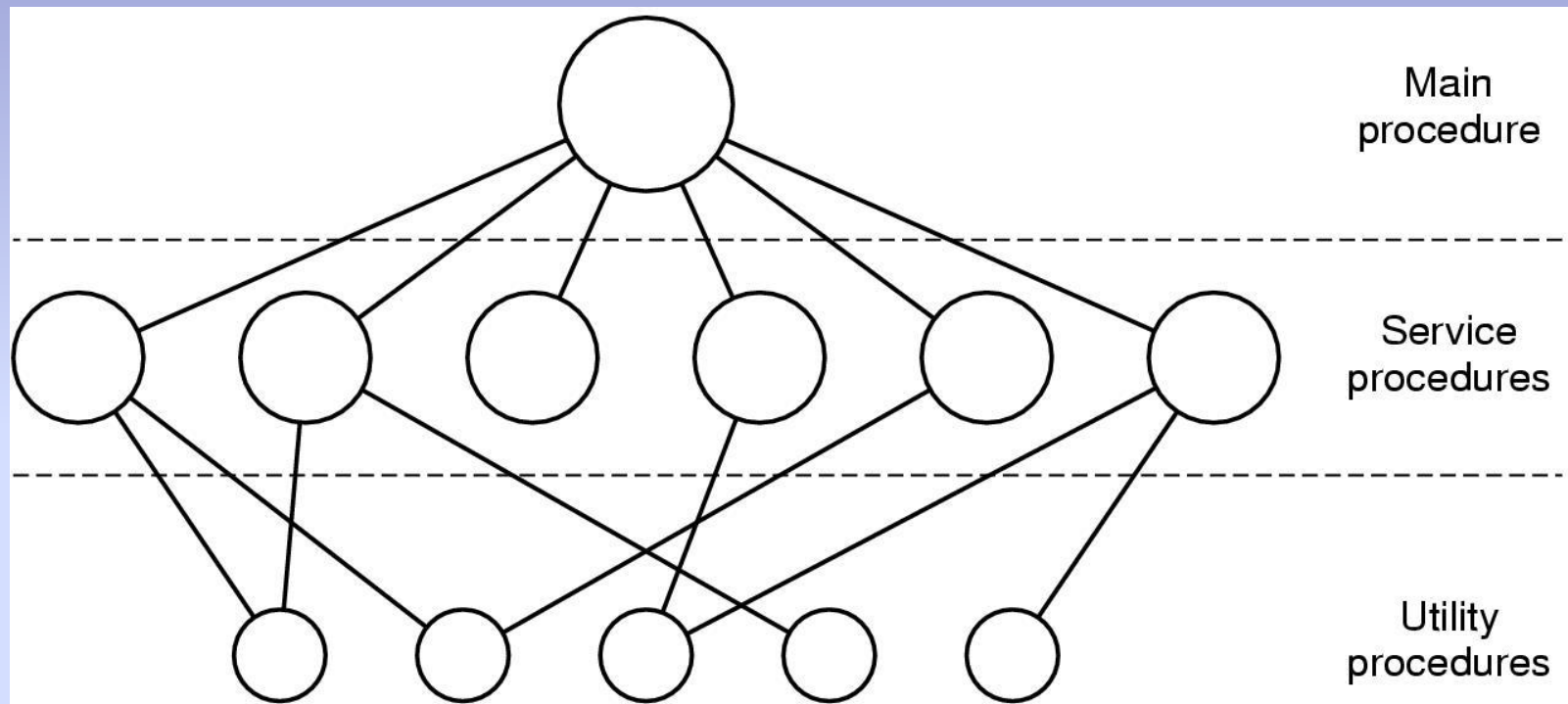
- Операционные системы майнфрэймов
- Серверные ОС
- Мультипроцессорные ОС
- ОС персональных компьютеров
- ОС реального времени
- Встроенные ОС
- Операционные системы смарт-карт
- и т.д.

Прерывания



- (a) Последовательность шагов обращения к устройству ввода/вывода и получение запроса прерывания
- (б) Выполнение прерывания

Структура ОС (1)



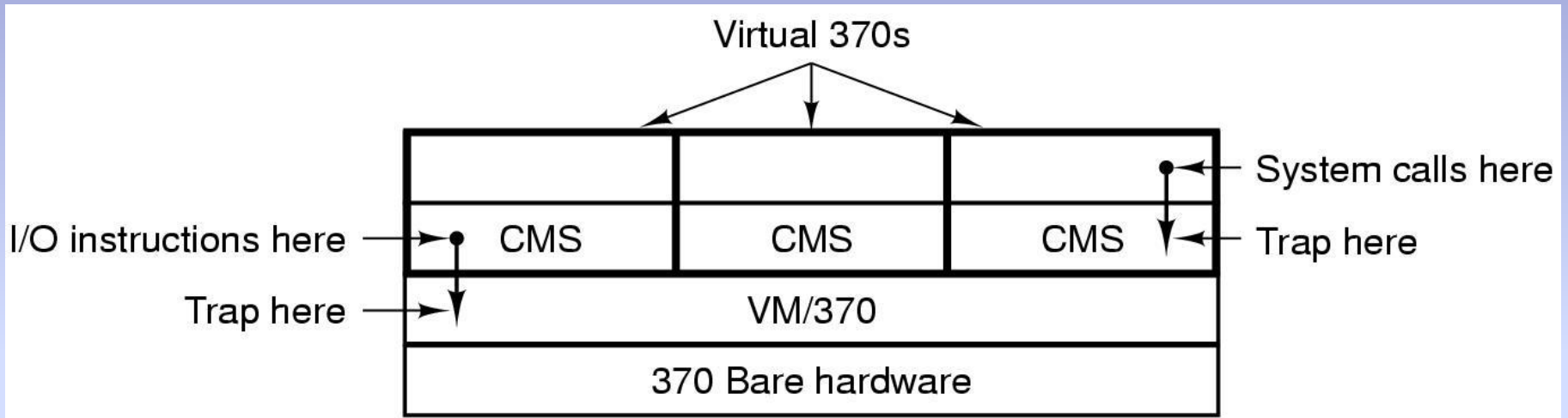
Пример структуры монолитной системы

Структура ОС(2)

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

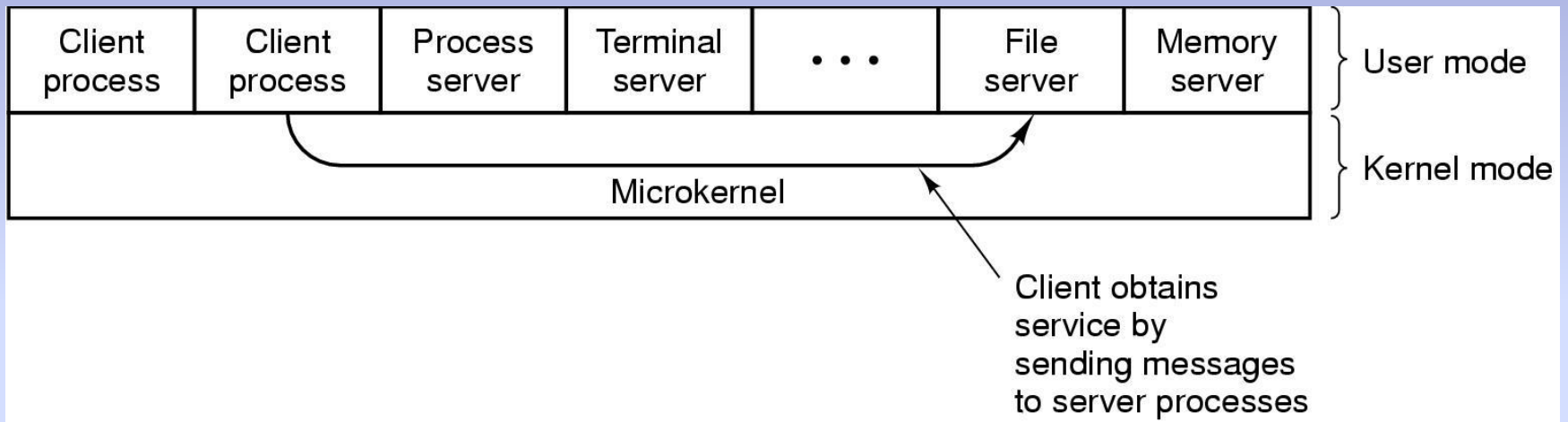
Структура операционной системы THE

Структура ОС(3)



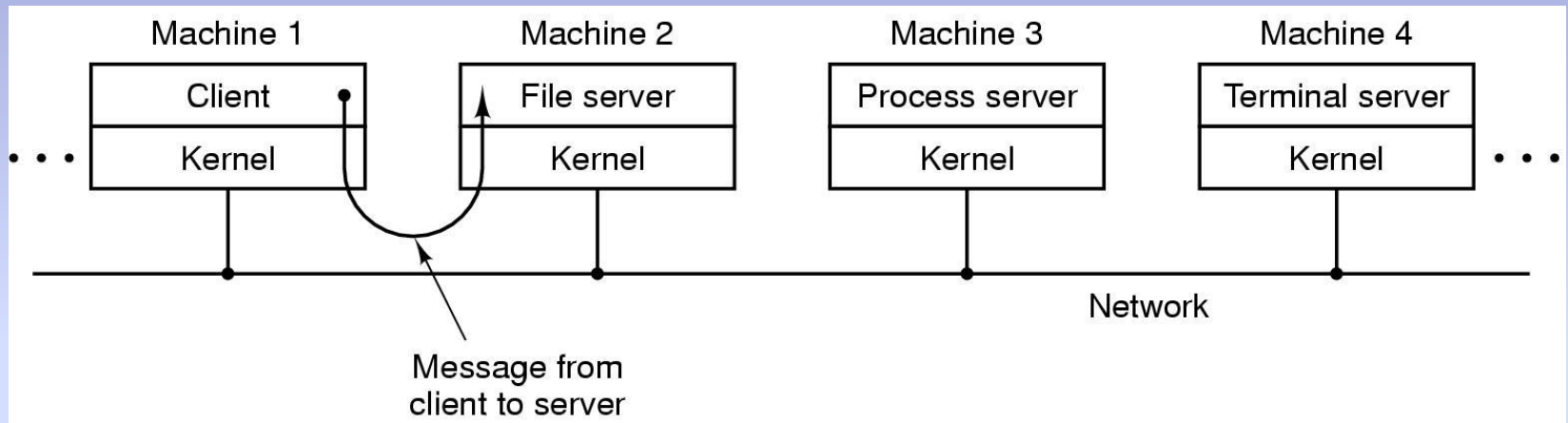
Структура VM/370 с CMS

Структура ОС(4)



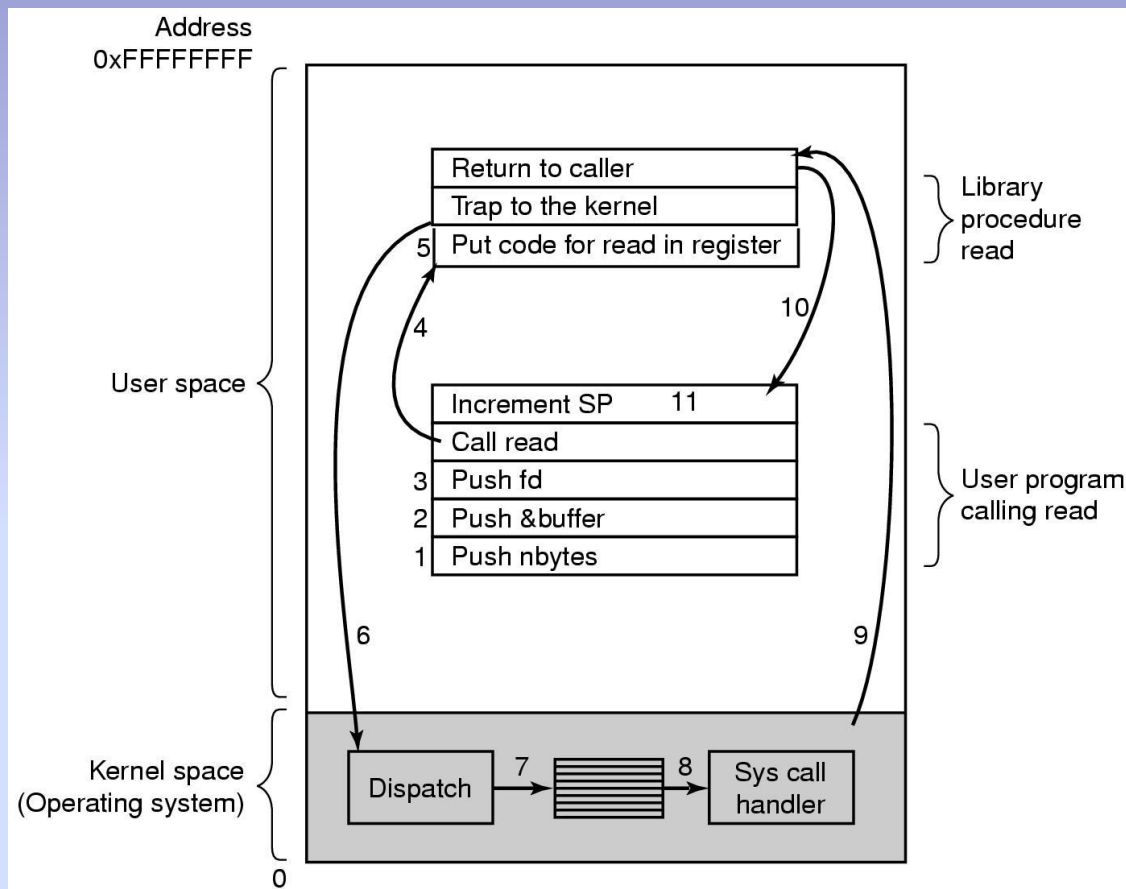
Клиент – серверная архитектура

Структура ОС(5)



Клиент – серверная архитектура в распределенной системе

Последовательность выполнения системного вызова



11 шагов выполнения системного вызова
read (fd, buffer, nbytes)

Некоторые системные вызовы для управления процессами

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status

Некоторые системные вызовы для управления файлами

File management

Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Некоторые системные вызовы для управления директориями

Directory and file system management

Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system

Некоторые системные вызовы для разных целей

Miscellaneous

Call	Description
<code>s = chdir(dirname)</code>	Change the working directory
<code>s = chmod(name, mode)</code>	Change a file's protection bits
<code>s = kill(pid, signal)</code>	Send a signal to a process
<code>seconds = time(&seconds)</code>	Get the elapsed time since Jan. 1, 1970

Сравнение системных вызовов

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time