Операционные системы

Введение. История. Основные понятия [1]

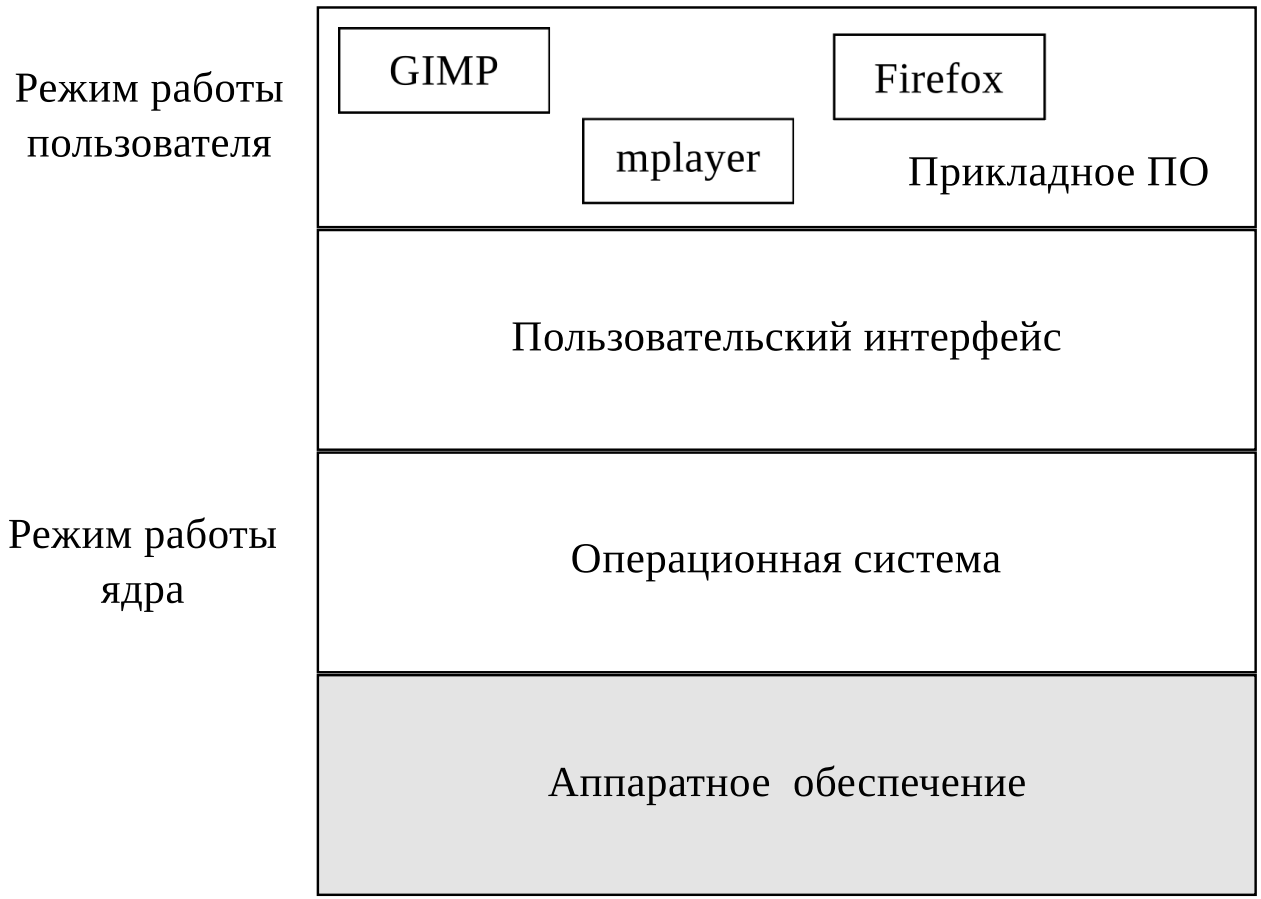
Соловьев А. В.

ПетрГУ – КИИСиФЭ

(Rev. 2018 06 07)

[1] По материалам «Таненбаум Э., Бос Х. Современные операционные системы. СПб.: Питер, 2015.»

Место ОС в структуре ПО



Что такое ОС?

ОС определяется через её функции:

- предоставление прикладному ПО набора ресурсов взамен неупорядоченного набора аппаратного обеспечения (*ОС как расширенная машина*);

- управление ресурсами аппаратного обеспечения в интересах прикладного ПО (*ОС как менеджер ресурсов*).

Мультиплексироване ресурсов во времени (принтер, процессор) и в пространстве (память, диск).

История (1)

- Первое поколение (1945–1955): машины на электронных лампах (1944, Колоссус (GB); 1944, Марк I (US); 1945, ЭНИАК (US))

Программирование путём сборки электрических схем на коммутационных панелях. Чуть позже – машинные коды на перфокартах.

- Второе поколение (1955–1965): транзисторы и системы пакетной обработки (IBM 7094)

Программы для них составлялись в основном на Фортране и ассемблере, а типичными операционными системами были FMS (Fortran Monitor System) и IBSYS – программы-мониторы.

- Третье поколение (1965–1980): интегральные схемы и многозадачность (IBM System/360, миникомпьютеры серии PDP)

OS/360 – операционная система для единого семейства (многозадачность, спулинг). CTSS (Compatible Time Sharing System) – режим разделения времени (у каждого пользователя свой диалоговый терминал). Проект MULTICS.

История (2)

- Четвертое поколение (с 1980 года по наши дни): персональные компьютеры

CP/M, MS-DOS, Mac OS X, Windows, UNIX и его клоны.
Графический интерфейс пользователя. Сетевые и распределённые ОС.
История UNIX: `https://www.levenez.com/unix/`

- Пятое поколение (с 1990 года по наши дни): мобильные компьютеры (1996, Nokia 9000; 1997, Ericsson GS88 Penelope, ...)

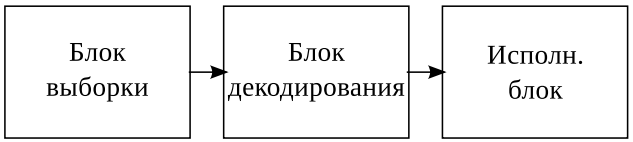
Объединение в одном устройстве телефона и компьютера.
Symbian OS. RIM Blackberry OS. Windows Phone. Apple iOS.
Google Android (с 2008 г.) – открытый исходный код, доступна по разрешительной лицензии. Приложения в основном на Java.

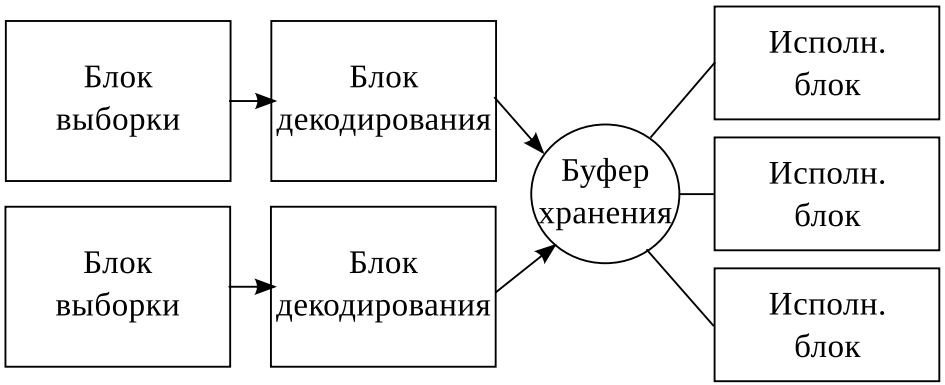
Обзор аппаратного обеспечения

- Центральный процессор + MMU
- Память
- Видеоконтроллер + дисплей
- Контроллер клавиатуры + клавиатура
- Контроллер USB + периферийные устройства (принтер, сканер)
- Контроллер жёсткого диска + дисковые накопители

Обзор аппаратного обеспечения: процессоры

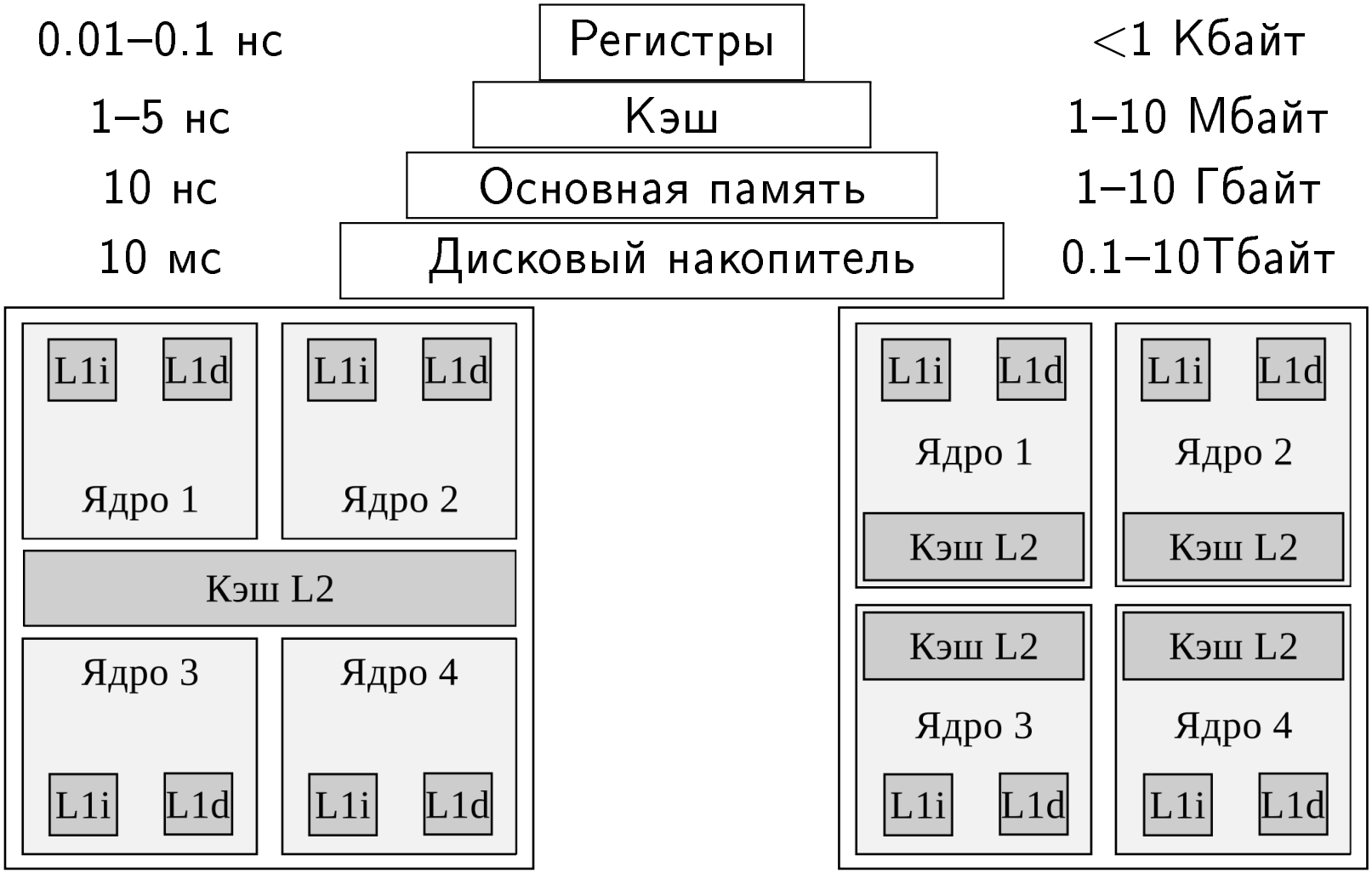
- Набор команд
- Регистры общего назначения
- Специальные регистры (счётчик команд, указатель стека, состояния и управления)
- Режимы работы: пользователя, супервизора; способ переключения (TRAP, INT, CALL).
- Конвейеризация





- Многопоточность (hyperthreading), многоядерность.

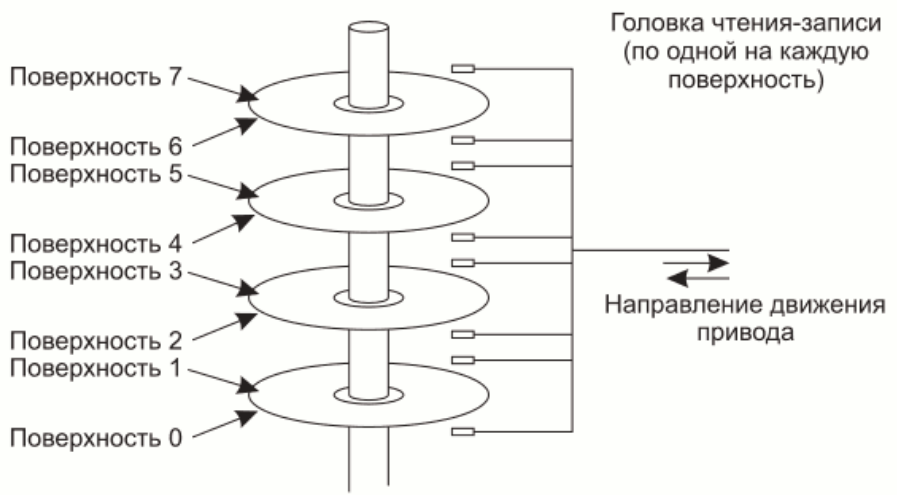
Обзор аппаратного обеспечения: память



Реализация кэширования в многоядерных процессорах.

Обзор аппаратного обеспечения: диски

Дисковый накопитель в пересчете на бит информации на два порядка дешевле, чем ОЗУ, его ёмкость зачастую на два порядка выше, скорость произвольного доступа к данным примерно на три порядка медленнее.



Поверхность — цилиндр (дорожка) — сектор (512 байт).
Подход к соседнему цилиндру – 1 мс. Подход к сектору – 5–10 мс. Скорость чтения сектора – 50–200 Мбайт/с.
SSD – нет движущихся частей (скорость произвольного доступа больше, но дольше запись и ресурс зависит от числа перезаписей).

Обзор аппаратного обеспечения: ввод/вывод (1)

Периферийное устройство обычно подключается к шине ЦП через контроллер. Управление устройством может быть очень сложно и требует высокого уровня детализации, поэтому задачей контроллера является предоставление ОС простого интерфейса.
Программа, предназначенная для общения с контроллером, выдачи ему команды и получения поступающих от него ответов, называется *драйвером устройства*. Каждый производитель контроллеров должен поставлять вместе с ними драйверы для каждой поддерживаемой ОС. Драйвер может работать в режиме ядра (чаще всего) или в режиме пользователя.
Способы включения драйвера в ядро: перекомпиляция ядра ОС (старинные UNIXы), включение при загрузке ОС (Windows), динамическая загрузка (совр. ОС).

Обзор аппаратного обеспечения: ввод/вывод (2)

Способы доступа к регистрам контроллеров:

- через обычное адресное пространство;
- через пространство портов ввода-вывода (IN/OUT).

Способы организации ввода-вывода:

- активное ожидание;
- по прерыванию;
- прямой доступ к памяти (DMA).

Типы операционных систем

- ОС мейнфреймов (OS/360, OS/390, z/OS)
- Серверные ОС (FreeBSD, Linux, Solaris, Windows Server)
- Многопроцессорные ОС (современные ОС, в т.ч. Linux и Windows)
- ОС персональных компьютеров (Windows, OS X, Linux)
- ОС КПК и смартфонов (Android, iOS)
- Встроенные ОС + ОС реального времени (жёсткие/мягкие) (Embedded Linux, QNX, VxWorks, FreeRTOS, RTEMS)
- ОС сенсорных узлов (датчиков) (TinyOS, Contiki, FreeRTOS)

Основные термины и понятия: процесс

Процесс – это программа во время её выполнения, или контейнер, в котором содержится вся информация, необходимая для работы программы:

- собственное адресное пространство (образ памяти),
- регистры процессора,
- список открытых файлов, рабочий каталог,
- необработанные предупреждения (сигналы),
- список связанных (дочерних) процессов,
- атрибуты процесса (UID, GID, SID, ...) и т. п.

Иерархия процессов (родительский – дочерний).
Межпроцессное взаимодействие (например, сигналы).

Основные термины и понятия: адресное пространство

ОС создаёт абстракцию *адресного пространства* в виде набора адресов, на которые может ссылаться процесс. Адресное пространство отделено от физической памяти машины и может быть как больше, так и меньше неё.
Технология *виртуальной памяти* – ОС хранит часть адресного пространства в ОЗУ, а часть – на диске, по необходимости меняя их фрагменты местами.
Обычно каждому процессу отводится для использования некоторый непрерывный набор адресов от нуля и до некоторого максимума.
Чтобы исключить взаимные помехи между процессами (и помехи работе ОС), нужен защитный механизм. Несмотря на то что этот механизм должен входить в состав оборудования, управляется он ОС.
Разрядность архитектуры процессора: 32 бита, 64 бита, ...

Основные термины и понятия: файл и ФС

Для сокрытия специфики дисков и других устройств ввода-вывода ОС предоставляет программисту удобную и понятную абстрактную модель – *файловую систему*. Компоненты этой модели – *файлы* и *каталоги* (каталог – контейнер для объединения файлов в группы).
Иерархическая (древовидная) структура файловой системы:

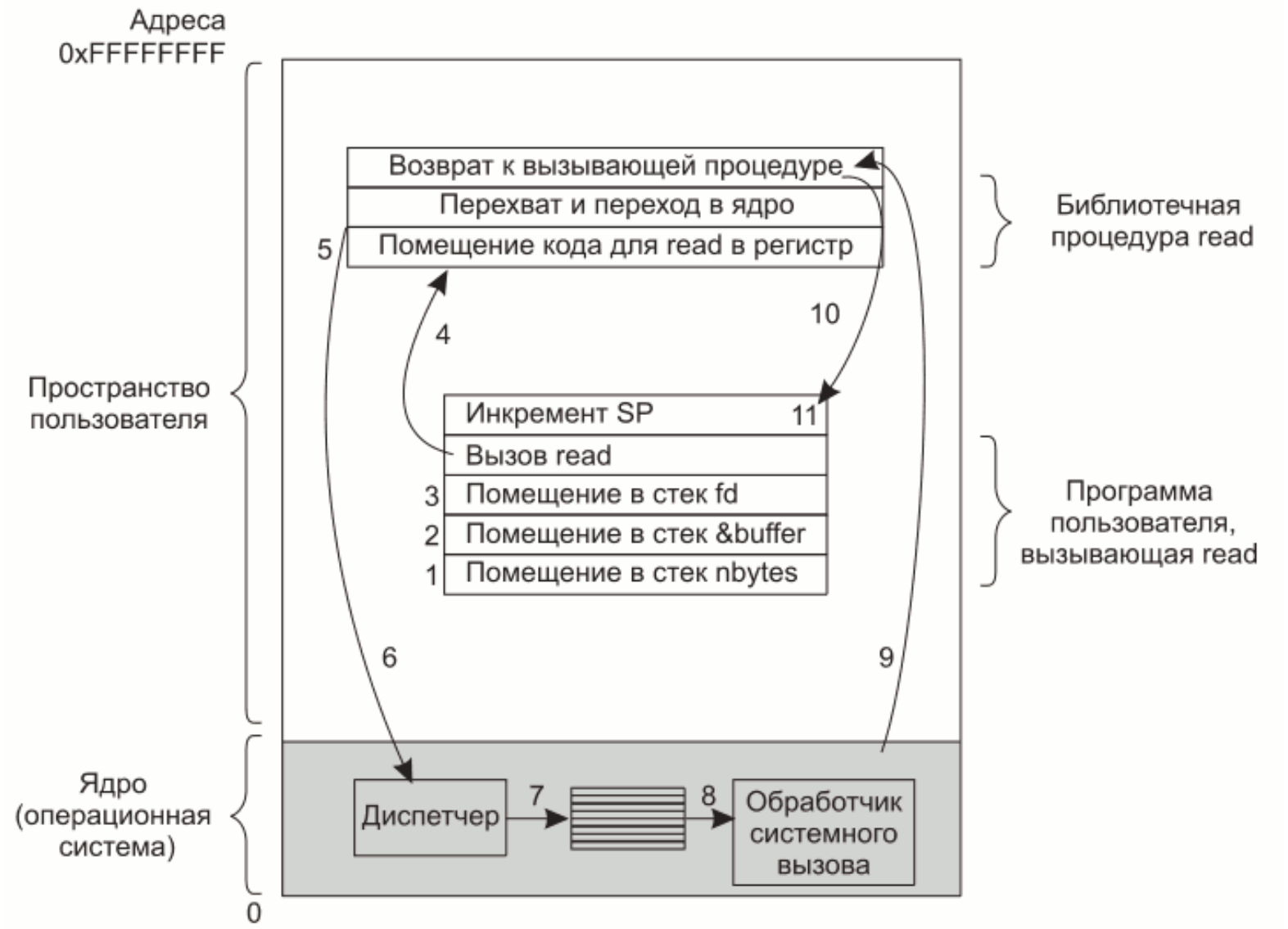
- большая глубина (у процессов – 3–4, у файлов – может быть более 10);
- длинный период существования (у процессов – несколько минут, у файлов – месяцы, годы)
- развёрнутые механизмы защиты (у процессов – родитель-потомок, у файлов – владелец-группа-остальные или СКД).

Полное имя (абсолютный путь).
Рабочий каталог. Относительное имя (путь относительно рабочего каталога).
Типы файлов: обычные, каталоги, специальные файлы (файлы устройств).
Монтирование файловых систем.

Основные термины и понятия: оболочка

ОС – программа, исполняющая системные вызовы. Редакторы, компиляторы, утилиты и интерпретаторы команд (оболочки) по определению не являются частью ОС. Тем не менее, оболочка – главное средство доступа ко многим функциям ОС и служит хорошим примером использования системных вызовов.
Оболочка – реализация интерфейса с пользователем. Может быть графической (Explorer в Windows; GNOME или KDE в Linux) или текстовой (консольной) (Command Prompt в Windows; bash, csh, ksh, ... в UNIX).
Оболочка запускается после входа в систему любого пользователя.

Выполнение системного вызова



Примеры системных вызовов (API ОС)

POSIX	Win32	Описание
fork	CreateProcess	Создает новый процесс
waitpid	WaitForSingleObject	Ожидает завершения процесса
exit	ExitProcess	Завершает выполнение процесса
open	CreateFile	Создает/открывает файл
close	CloseHandle	Закрывает файл
read	ReadFile	Читает данные из файла
write	WriteFile	Записывает данные в файл
lseek	SetFilePointer	Перемещает указатель файла
stat	GetFileAttributesEx	Получает различные атрибуты файла
mkdir	CreateDirectory	Создает новый каталог
rmdir	RemoveDirectory	Удаляет пустой каталог
link	—	Создание жёсткой связи
unlink	DeleteFile	Удаляет существующий файл (связь)
chdir	SetCurrentDirectory	Изменяет рабочий каталог
kill	—	Отправка сигнала процессу
time	GetLocalTime	Получает текущее время

POSIX (ISO/IEC/IEEE 9945:2009)

Структуры ОС: монолитные системы

- Набор процедур, связанных вместе в одну большую исполняемую программу. Каждая процедура может свободно вызвать любую другую процедуру. Все части монолитного ядра работают в одном адресном пространстве.
- Возможность вызвать любую нужную процедуру приводит к весьма высокой эффективности работы системы, но наличие нескольких тысяч процедур, которые могут вызывать друг друга сколь угодно часто, нередко делает ее громоздкой и непонятной.
- Отказ в любой из этих процедур приведет к аварии всей ОС.
- Полностью отсутствует сокрытие деталей реализации – каждая процедура видна любой другой процедуре.

Современные монолитные ядра позволяют во время работы динамически (по необходимости) подгружать и выгружать модули (драйверы), выполняющие часть функций ядра.
Примеры: Linux, FreeBSD, Solaris.

Структуры ОС: микроядра (1)

ОС разбивается на небольшие, вполне определённые модули. Только один из них – микроядро – запускается в режиме ядра, а все остальные запускаются в виде относительно слабо наделённых полномочиями обычных пользовательских процессов.

Типичные функции микроядра:

- Paging (управление страничным механизмом);
- Scheduling (управление потоками/процессами);
- IPC (межпроцессные коммуникации).

Остальные функции (драйверы устройств, реализации ФС, стеки TCP/IP, USB, ...) работают в пространстве пользователя в виде отдельных процессов (сервисов/серверов), взаимодействуют с ядром с помощью системных вызовов, взаимодействуют друг с другом с помощью IPC.

Структуры ОС: микроядра (2)

Достоинства:

- простота реализации и отладки;
- безопасность и надёжность;
- модульность.

Недостатки:

- низкая производительность (из-за накладных расходов на IPC).

Примеры: Minix, QNX, OS X (на ядре Mach), Symbian.

Структуры ОС: гибридные ядра

Гибридные ядра – модифицированные микроядра, позволяющие для ускорения работы запускать «несущественные» части в пространстве ядра (например, WinNT).

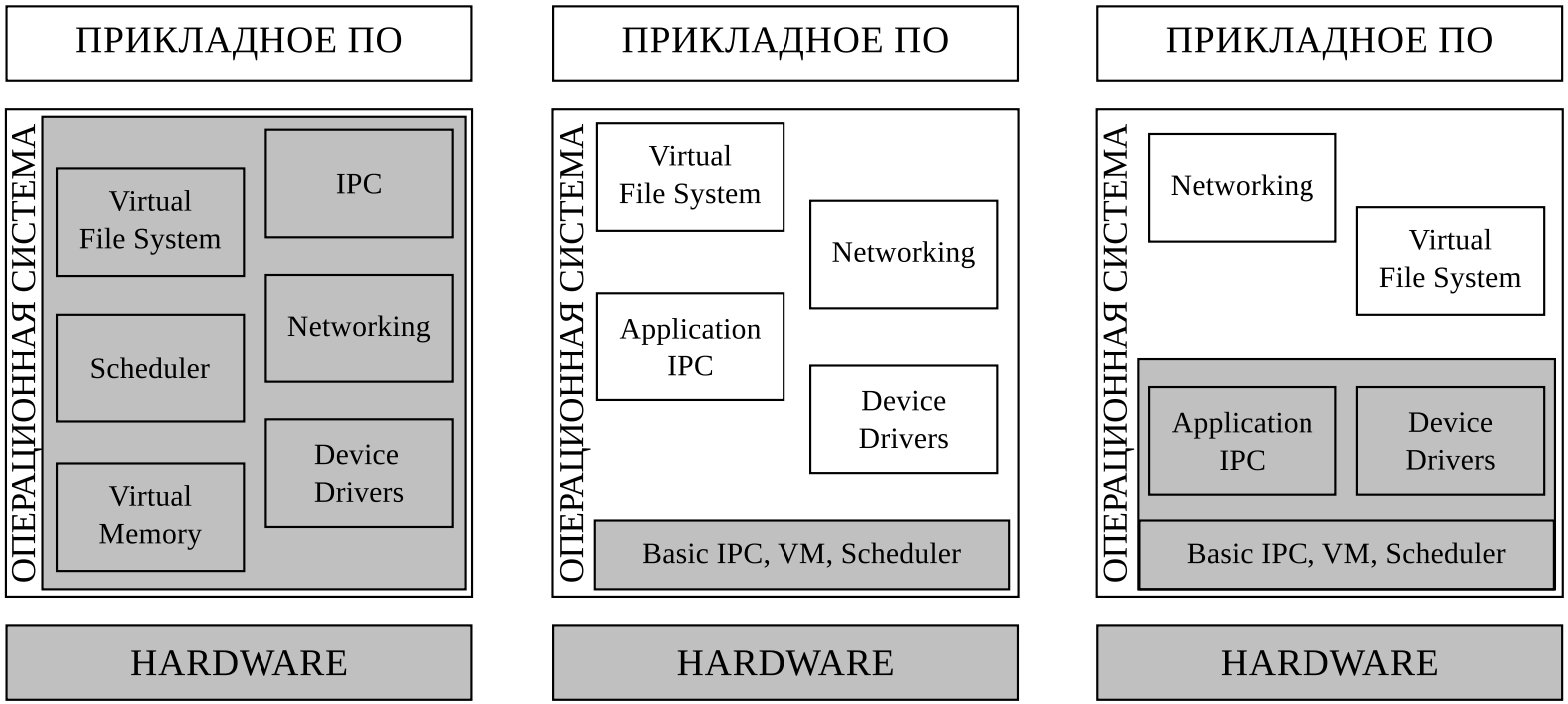


Рис.: Монолитное ядро – Микроядро – Гибридное ядро