

Операционные системы и среды

ТЕМА Файловая система и ввод вывод информации.

Симферопольский политехнический
колледж

Задачи ОС по управлению файлами и устройствами

- ▶ Подсистема ввода-вывода ОС при обмене данными с внешними устройствами должна решать ряд общих задач:
 - ▶ Организация параллельной работы устройства ввода-вывода и процессора;
 - ▶ Согласование скоростей обмена и кэширования данных;
 - ▶ Разделение устройств и данных между процессами;
 - ▶ Обеспечение удобного логического интерфейса между устройствами и остальной частью системы;
 - ▶ Поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера;
 - ▶ Динамическая загрузка и выгрузка драйверов;
 - ▶ Поддержка файловых систем;
 - ▶ Поддержка синхронных и асинхронных операций ввода-вывода.

Организация параллельной работы устройства ввода-вывода и процессора

- ▶ Каждое устройство ввода-вывода - диск, принтер, терминал - имеет блок управления (*контроллер* устройства).
- ▶ Контроллер взаимодействует с *драйвером* - системным программным модулем, предназначенным для управления устройством. Под управлением контроллера устройство может работать некоторое время автономно от команд ОС.
- ▶ Подсистема ввода-вывода должна обеспечить работу - запуск и приостановку разнообразных драйверов, обеспечив приемлемое время реакции каждого драйвера на независимые события контроллера.
- ▶ С другой стороны, необходимо минимизировать загрузку процессора задачами ввода-вывода.

Согласование скоростей обмена и кэширования

данных

- ▶ При обмене информации в системе возникает задача согласования скорости выполняемых процессов. Согласование скорости осуществляется за счет буферизации данных в оперативной памяти и синхронизации доступа процессов к буферу.
- ▶ В некоторых случаях свободной оперативной памяти недостаточно для буферизации данных. Для размещения данных в буфере используются специальные файлы - спул-файлы.
- ▶ Другой способ - использование буферной памяти в контроллерах внешних устройств. Например, использование памяти, устанавливаемой на видеоадаптерах.

Разделение устройств и данных между процессами

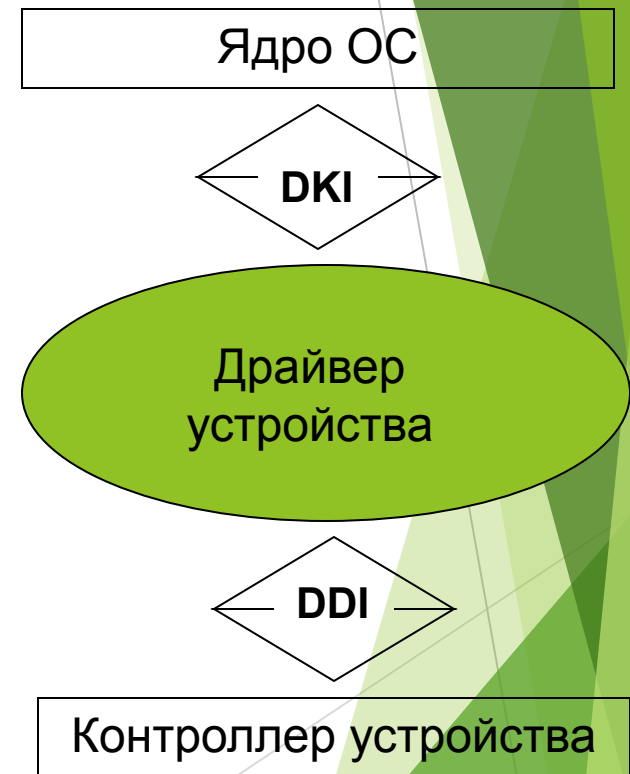
- ▶ Устройства ввода-вывода могут предоставляться процессам в монопольном и разделяемом режимах.
- ▶ Задача ОС обеспечить контроль доступа к данным ресурсам системы путем проверки прав пользователя, от имени которых выполняется процесс. Операционная система имеет возможность контролировать доступ не только к устройству в целом, но и к отдельным порциям данных.
- ▶ При разделении устройства между процессами возникает необходимость в разграничении порции данных от двух процессов. Для хранения очереди заданий применяется спул-файл, который синхронизирует скорости работы устройства и оперативной памяти.

Обеспечение удобного логического интерфейса между устройствами и остальной частью системы

- ▶ Разнообразие устройств ввода-вывода делает актуальной задачу создания экранирующего интерфейса между периферийными устройствами и приложениями.
- ▶ Современные ОС поддерживают файловую модель работы устройств, при которой устройства представляются набором байт, с которыми работают посредством унифицированных системных вызовов (read, write).
- ▶ Для детализации конкретных свойств используются специфические модели устройств конкретного типа - графическая подсистема, принтер, сетевые адаптеры и т.д.

Поддержка широкого спектра драйверов с возможностью простого включения в систему нового драйвера

- ▶ Достоинством подсистемы ввода-вывода операционной системы является разнообразие устройств, поддерживаемых данной ОС.
- ▶ Для создания драйверов необходимо наличие удобного и открытого интерфейса между драйверами и другими компонентами ОС.
- ▶ Драйвер взаимодействует, с одной стороны, с модулями ядра ОС, а с другой стороны - с контроллерами внешних устройств. Драйвер имеет два интерфейса DKI (driver kernel interface) и DDI (driver device interface).



Динамическая загрузка и выгрузка драйверов

- ▶ Другой проблемой работы с устройствами ввода-вывода является проблема включения драйвера в состав работающей ОС - динамическая загрузка/выгрузка драйверов.
- ▶ Способность системы автоматически загружать и выгружать из оперативной памяти требуемый драйвер повышает универсальность ОС.
- ▶ Альтернативой динамической загрузке драйверов при изменении текущей конфигурации внешних устройств является повторная компиляция кода ядра с требуемым набором драйверов. Пример - некоторые версии UNIX.

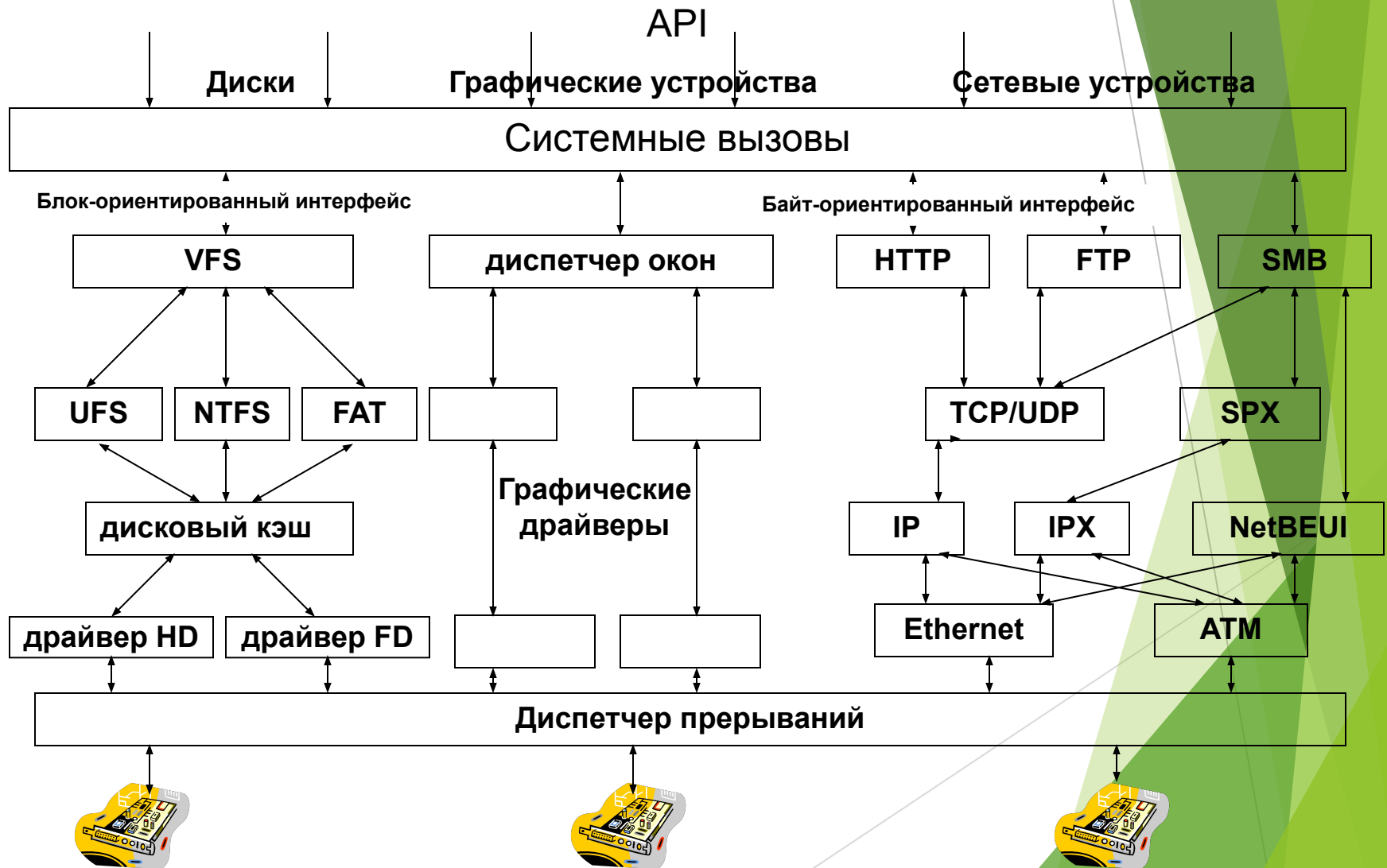
Поддержка файловых систем

- ▶ Внешняя память вычислительной системы представляет собой периферийные устройства, на которых хранится большая часть пользовательской информации и системных данных.
- ▶ Для организации хранения информации на внешних носителях используется файловая модель.
- ▶ Для обеспечения доступа к данным используется специальный программный слой, обеспечивающий поддержку работы с конкретной файловой системой - драйверы файловой системы.
- ▶ Для обеспечения возможности работы с несколькими файловыми системами применяется подход, основанный на применении специального слоя, с которым взаимодействуют приложения ОС - например, слой VFS (virtual file system) в некоторых версиях UNIX.

Поддержка синхронных и асинхронных операций ввода-вывода

- ▶ Операции ввода-вывода по отношению к программному приложению выполняются в синхронном или асинхронном режимах.
- ▶ **Синхронный режим** - приложение приостанавливает свою работу и ждет отклика от устройства.
- ▶ **Асинхронный режим** - приложение продолжает работу, параллельно с ожиданием отклика от устройства.
- ▶ Операционные системы для разных приложений должны обеспечить синхронную и асинхронную работу с устройствами.

Многослойная модель подсистемы ввода-вывода



Менеджеры ввода-вывода

- ▶ Для координации работы драйверов в подсистеме ввода-вывода выделяется специальный модуль, называемый менеджером ввода-вывода.
- ▶ Верхний слой менеджера составляют системные вызовы ввода-вывода, которые получают запросы от приложений и переадресуют их определенным драйверам.
- ▶ Нижний слой реализует взаимодействие с контроллерами внешних устройств, экранируя драйверы от особенностей аппаратной платформы компьютера.
- ▶ Еще одна функция менеджера ввода-вывода - организация взаимодействия модулей ввода-вывода с модулями других подсистем (управление процессами, виртуальной памятью и т.д.).

Специальные файлы

- ▶ Для унификации операций и структуризации программного обеспечения ввода-вывода устройства рассматриваются как некоторые специальные (виртуальные) файлы.
- ▶ Такой подход позволяет использовать общий набор базовых операций ввода-вывода для любых устройств, экранировать специфику устройства.
- ▶ Например, в операционных системах семейства UNIX, специальные файлы помещаются в каталог `/dev`. При появлении нового устройства администратор имеет возможность создать новую запись с помощью команды `mknod`.

Логическая организация файловой системы

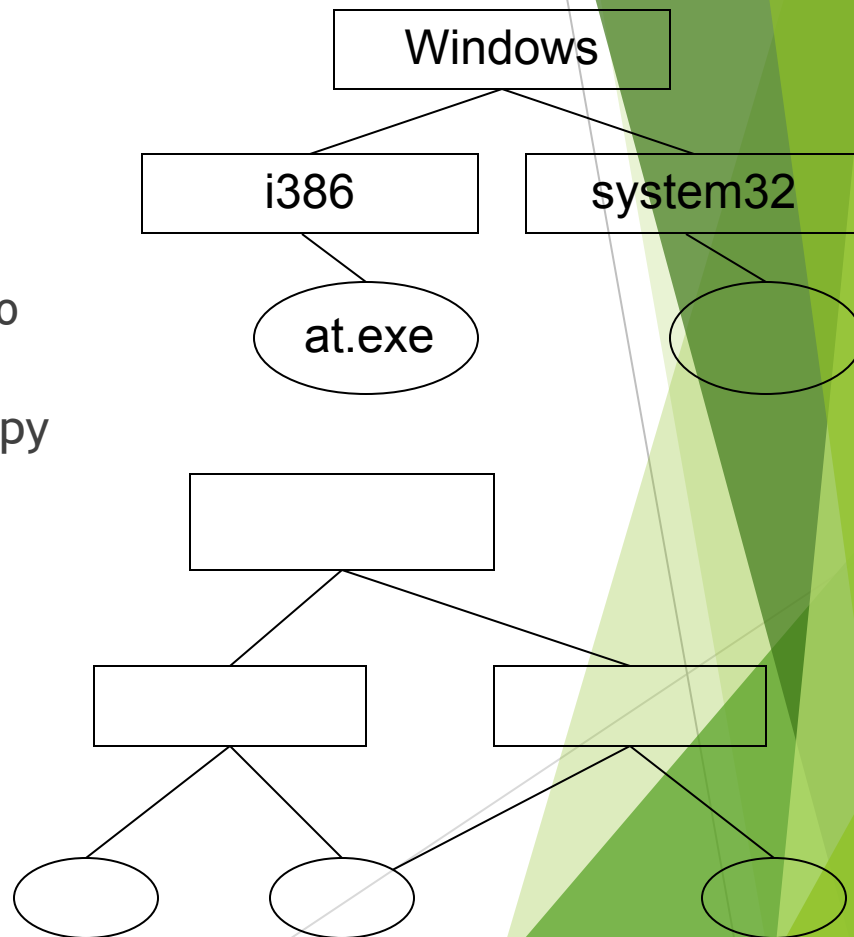
- ▶ Одной из основных задач ОС - предоставление удобного пользовательского интерфейса при работе с данными, хранящимися на носителях. Логическая модель в рамках ОС подменяет физическую модель размещения данных на носителях.
- ▶ **Файл** - именованная область внешней памяти, в которую могут записываться и откуда могут считываться данные. Применение файлов позволяет решить следующие задачи:
 - ▶ Долговременное хранение информации;
 - ▶ Совместное использование информации.

Файловая система

- ▶ **Файловая система** - часть ОС, включающая:
 - ▶ Совокупность всех файлов на дисках;
 - ▶ Наборы структур данных, используемых для управления (каталоги, дескрипторы файлов, таблицы распределения свободного и занятого пространства);
 - ▶ Комплекс системных программных средств, реализующих операции над файлами (создание, удаление, чтение, запись, именование и поиск файлов).
 - ▶ В многопользовательских системах добавляются функции по обеспечению защиты данных от несанкционированного доступа.
- ▶ Файловые системы поддерживают несколько функционально различных типов файлов:
 - ▶ Обычные файлы;
 - ▶ Каталоги;
 - ▶ Ссылки;
 - ▶ Именованные каналы;
 - ▶ Конвейеры и т.д.

Иерархическая структура файловой системы

- ▶ Пользователи обращаются к файлам по их символьным именам. Для удобства пользователя логическая структура хранения данных представляет иерархическую структуру.
- ▶ Граф, описывающий структуру файловой системы может представлять собой **дерево** или **сеть**.
- ▶ В Windows используется древовидная организация, в UNIX - сетевая.



Имена файлов

- ▶ Каждый файл имеет некоторое символьное имя. В иерархических системах выделяют три типа имен файлов:
 - ▶ Простое (имя файла в пределах одного каталога)
 - ▶ Полное (цепочка простых символьных имен всех каталогов, через которые проходит путь от корня до файла)
 - ▶ Относительное (имя включает имена каталогов, через которые проходит маршрут от текущего каталога к искомому файлу).
- ▶ В различных операционных системах есть свои ограничения на использование символов при присвоении имени, а также на длину относительного и полного имени файла.

Монтирование файловой системы

- ▶ В общем случае вычислительная система может иметь несколько устройств внешней памяти. Для обеспечения доступа к данным, хранящимся на разных носителях используются два подхода:
 - ▶ На каждом устройстве размещается автономная файловая система, со своим деревом каталогов (например, в MS-DOS накопители нумеруются а:, с: и т.д.).
 - ▶ Монтирование файловой системы - операция объединения файловых систем в единую файловую систему (например, в операционных системах семейства UNIX).

Атрибуты файла

Атрибут – информация, описывающая некоторые свойства файла, например:

- ▶ Тип файла
- ▶ Владелец файла
- ▶ Создатель файла
- ▶ Пароль для доступа к файлу
- ▶ Информация о разрешенных операциях к файлу
- ▶ Время создания, последнего доступа и модификации файла
- ▶ Признак «только для чтения»
- ▶ Признак «скрытый файл»
- ▶ Признак «системный файл»
- ▶ Признак «двоичный/символьный файл»
- ▶ Признак «временный файл»
- ▶ Признак блокировки
- ▶ Длина записи в файле
- ▶ Др.

Логическая организация файла

- ▶ В общем случае данные, хранящиеся в файле, имеют некоторую логическую структуру (формат хранения данных). Поддержание структуры данных в файле возлагается либо целиком на приложение, либо часть функций на файловую систему.
- ▶ Неструктурированная модель файла широко используется в большинстве современных ОС.
- ▶ Структурированный файл рассматривается ОС, как упорядоченная совокупность логических записей. Развитием данного подхода являются системы управления базами данных (СУБД).