1. **Что такое файл?**

**Файл** – набор логических записей; именованная область данных на носителе информации, используемая как базовый объект взаимодействия с данными в операционных системах.

1. **Перечислите основные характеристики (атрибуты) файла.**

**ARCHIVE -** файл изменён после резервного копирования или не был скопирован программами резервного копирования;

**HIDDEN –** файл скрывается от показа, пока не указано обратное;

**NORMAL** – файл не содержит другие атрибуты;

**READONLY -** файл запрещено редактировать;

**SYSTEM** – файл является критическим для работы операционной системы;

**TEMPORARY** -

1. **Что такое файловая система?**

**Файловая система -** система управления файлами, часть операционной системы обеспечивающая доступ к файлам. Устанавливает связь между логическим представлением и физическим расположением данных (абстракция над данными).

1. **Перечислите основные функции файловой системы.**
2. Создание/удаление каталогов;
3. Включение/исключение подкаталогов;
4. Включение/исключение файла в каталог;
5. Создание/удаление файла;
6. Открытие/закрытие доступа к файлу;
7. Чтение/запись логических записей файла;
8. Установка (поддержка) указателя файла.
9. **Перечислите 3 названия файловой системы.**

* **FAT**(16/32/64),
* **NTFS**,
* ext,
* ext(2/3/4),
* BtrFS(Oracle),
* ISO 9660(CD,DVD)

1. **Какая файловая система установлена на вашем компьютере под Windows? под Linux?**

NTFS для WINDOWS; XFS для Linux (file –s /dev/sda1)

1. **Что такое каталог файловой системы? Перечислите наименования специальных каталогов.**

**КаталОг** – файл содержащий информацию о месте расположения других файлов; специальные каталоги: .*(точка – текущий каталог), .. (две точки – родительский каталог)*.

1. **Поясните понятие «текущий каталог приложения».**

Текущим называется каталог, с которым работает ОС, если ей не указать другого каталога. Текущим каталогом для приложения является тот, с которым он работает в определённый момент времени. Он обозначается точкой (.) .

Для смены текущего каталога на другой используется команда cd

1. **Что такое специальные имена файлов? перечислите их, для чего они нужны.**

con, lpt1, prn, aux, com и т.д. - все эти сокращения от английских названий физических устройств и являются специальными именами файлов. Такие имена являются зарезервированными. Используются они преимущественно в пакетных командных файлах (BAT или CMD). Так же в консоли Win32.

1. **Для чего используются буферы ввода-вывода?**

**Буферы ввода/вывода** – области памяти для хранения физически считанных данных; необходимы для устранения несоответствия между физическим и логическим чтением/записью. Сначала заполняется буфер кластерами (физически считанными данными), а затем осуществляется чтение логических файлов.

1. **Поясните понятие «кэширование».**

**Кэширование ввода/вывода –** перемещение в быстродействующую память, наиболее часто используемых данных.

1. **Поясните понятие «указатель позиции файла».**

**Указатель файла** – объект файловой системы, позиционирующий логическую запись.

1. **Поясните понятие «маркер конца файла».**

**EOF** (end of file — конец файла) в компьютерной терминологии является индикатором операционной системы, означающим, что данные в источнике закончились. Источниками данных обычно являются файлы и потоки.

1. **Поясните понятие «блокировка файла».**

**Блокировка файла** — механизм, который ограничивает доступ к файлу, давая доступ в данный момент только одному пользователю или процессу.

1. **Windows. Функция OS API для создания файла.**

HANDLE CreateFile(

LPCTSTR lpFileName, // имя файла

DWORD dwDesiredAccess, // режим доступа

DWORD dwShareMode, // совместный доступ

LPSECURITY\_ATTRIBUTES lpSecurityAttributes, // SD (дескр. защиты)

DWORD dwCreationDisposition, // как действовать

DWORD dwFlagsAndAttributes, // атрибуты файла

HANDLE hTemplateFile // дескр.шаблона файла

);

1. **Windows. Функция OS API для открытия файла.**

HANDLE OpenFile(

LPCSTR lpFileName, // имя файла

LPOFSTRUCT lpReOpenBuff, // информация о файле

UINT uStyle // действия и атрибуты

);

1. **Windows. Функция OS API для удаления файла.**

BOOL DeleteFile(

LPCTSTR lpFileName // имя файла

);

1. **Windows. Функция OS API для записи в файл.**

BOOL WriteFile(

HANDLE hFile, // дескриптор файла

LPCVOID lpBuffer, // буфер данных

DWORD nNumberOfBytesToWrite, // число байтов для записи

LPDWORD lpNumberOfBytesWritten, // число записанных байтов

LPOVERLAPPED lpOverlapped // асинхронный буфер

);

1. **Windows. Функция OS API для чтения файла.**

BOOL ReadFile(

HANDLE hFile, // дескриптор файла

LPVOID lpBuffer, // буфер данных

DWORD nNumberOfBytesToRead, // число байтов для чтения

LPDWORD lpNumberOfBytesRead, // число прочитанных байтов

LPOVERLAPPED lpOverlapped // асинхронный буфер

);

1. **Windows. Назначение и отличие функций OS API: CopyFile, MoveFile, ReplaseFile.**

**CopyFile** создаёт копию файла с новым именем. В случае существования файла с таким именем в каталоге происходит его перезапись при передаче третьим параметром FALSE. При успешном выполнении возвращает 0, иначе – не нуль.

**MoveFile** перемещает уже существующий файл или каталог со всем содержимым в соответствии с новым именем файла или каталога, указанным в качестве второго параметра. При успешном выполнении возвращает 0, иначе – не нуль.

**ReplaceFile** заменяет один файл другим, с параметром создания резервной копии первоначального файла. Файл замены принимает имя замененного файла и его идентичность(атрибуты).

1. **Windows. Перечислите функции OS API, которые изменяют текущее значение указателя позиции файла.**

Функции SetFilePointer и SetFilePointerEx сдвигают указатель позиции файла.

1. **Windows. Перечислите функции OS API для блокировки и разблокировки файлов.**

BOOL LockFile(

HANDLE hFile, // дескриптор файла

DWORD dwFileOffsetLow, // младшее слово смещения

DWORD dwFileOffsetHigh, // старшее слово смещения

DWORD nNumberOfBytesToLockLow, // младшее слово длины

DWORD nNumberOfBytesToLockHigh // старшее слово длины

);

BOOL UnlockFile(

HANDLE *hFile*, // дескриптор файла

DWORD *dwFileOffsetLow*, // младшее слово начала

DWORD *dwFileOffsetHigh*, // старшее слово начала

DWORD *nNumberOfBytesToUnlockLow*, // младшее слово длины

DWORD *nNumberOfBytesToUnlockHigh* // старшее слово длины

);

1. **Windows. Поясните механизм «наблюдение за каталогом», перечислите набор функций OS API, позволяющий реализовать этот механизм.**

Механизм наблюдения за каталогом предполагает контроль над изменениями, производимыми с файлами и дочерними каталогами в каталоге: изменение имени, размера, атрибутов, удаление или создание. FindFirstChangeNotification, FindNextChangeNotification, FindCloseChangeNotification, WaitForSingleObject – функции, реализующие данный механизм.

1. **Windows. Перечислите функции OS API для работы с каталогами, поясните их назанчения.**

**CreateDirectory**, **CreateDirectoryEx** – создание каталога.

**FindFirstFile** – найти первое вхождение файла в каталоге.

**RemoveDirectory** – удаление каталога.

**MoveFile** – используется для перемещения файлов и каталогов.

**GetCurrentDirectory** и **SetCurrentDirectory** – определение и установка текущего каталога.

1. **Linux. Что такое FHS?**

**FileSystem Hierarchy Standard (FHS)** – стандарт, унифицирующий местонахождение файлов и каталогов с общим назначением в файловой системе UNIX.

1. **Linux. Перечислите типы файловых систем.**

**Ext2, Ext3, Ext4** или **Extended Filesystem** - это стандартная файловая система для Linux.

**XFS**- это высокопроизводительная файловая система для работы с большими файлами.

**devtmps** - файловая система для ядра OS, для описания устройств.

**tmpfs** позволяет размещать любые пользовательские файлы в оперативной памяти компьютера.

**procfs** - по умолчанию смонтирована в папку proc и содержит всю информацию о запущенных в системе процессах, а также самом ядре.

**sysfs** - с помощью этой файловой системы можно задавать различные настройки ядра во время выполнения.

1. **Linux. Что такое inode?**

**Inode** или **индексный дескриптор** - это структура данных, в которой хранятся метаданные файла и перечислены блоки с данными файла. Файловые системы Ext используют блоки для хранения данных. По умолчанию размер одного блока равен 4092 байта. В начале раздела расположен суперблок, в котором находятся метаданные всей файловой системы, а ним идут несколько зарезервированных блоков, а затем размещена таблица Inode и только после неё блоки с данными. Таким образом, все Inode размещены в начале раздела диска.

1. **Linux. Поясните назначение функций open, read, write, close, ioctl, stat, flush, lseek, lstat, fstat.**

**int open(const char \*filename, int access, unsigned mode)** – открытие файла данных; возвращает дескриптор файла (целочисленное значение, идентифицирующее открытый файл);

**int read(int fd, void \*buf, unsigned count)** ­– чтение данных из файла в буфер;

**int write(int handle, void \*buf, int count)** – запись информации из буфера в файл;

**int close(int fd)** ­– закрытие файла данных;

**ioctl** — средство управления аппаратными устройствами. Первым аргументом функции является дескриптор файла того устройства, которым требуется управлять. Второй аргумент — это код запроса, обозначающего выполняемую операцию. Разным устройствам соответствуют разные запросы. В зависимости от запроса функции ioctl() могут потребоваться дополнительные аргументы.

Функция **stat(char \*filename, struct stat \*statbuf)** вносит в структуру, на которую указывает statbuf, информацию, содержащуюся в файле, связанном с указателем filename.

**flush** – очищает выходную последовательность OS.

**long lseek(int handle, long offset, int origin) -**  устанавливает указатель положения в файле, указанном дескриптором handle, в положение, указанное аргументами offset и origin.

**stat** возвращает информацию о файле**. int lstat(const char \*file\_name, struct stat \*buf) –** идентична **stat**, но в случае, если обрабатывается символьная ссылка, то возвращаетсяинформация о самой ссылке, а не о файле, на который она ссылается.

**int fstat(int handle, struct stat \*statbuf) -** заполняет структуру, на кото­рую указывает statbuf, информацией о файле, связанном с дескриптором файла handle.