1. по каким соображениям в процессах Pentium запрещено вызывать менее привилегированные процедуры, но разрешено вызывать менее привилегированные задачи?

Для реализации косвенный вызова процедуры через шлюз. Первый способ в принципе не позволяет вызвать из пользовательской программы процедуру операционной системы, находящуюся в неподчиненном сегменте и имеющую более высокий уровень привилегий. С помощью второго способа могут быть вызваны процедуры ОС, находящиеся в подчиненном сегменте, однако они будут выполняться с пользовательским уровнем привилегий и не смогут обрабатывать системные данные, что нужно для большинства системных вызовов. Поэтому процессор Pentium предоставляет еще один способ вызова подпрограмм — через шлюз (вентиль), позволяющий пользовательскому коду вызывать привилегированные процедуры, которые будут работать со своим высоким уровнем привилегий

1. Поддерживает ли процессор Pentium приотеризацию запросов прерывания между несколькими внешними устройствами?

да

1. Основные отличия файловых систем FAT и NTFS.
	* NTFS рациональнее использует дисковое пространство.
	* FAT32 работает быстрее.
	* NTFS обеспечивает высокую производительность при работе с файлами больших размеров.
	* FAT32 поддерживает работу с файлами размером до 4 Гб.
	* FAT32 не требует большого объема ОЗУ.
	* NTFS обеспечивает высокую надежность работы системы.
	*
2. Основные функции работы с файлами (ОС)
	* HANDLE CreateFile(LPCTSTR lpName, DWORD dwAccess, DWORD dwShareMode, LPSECURITY\_ATTRIBUTES lpSecurityAttributes, DWORD dwCreate, DWORD dwAttrsAndFlags, HANDLE hTemplateFile)- создание файла
	* BOOL ReadFile(HANDLE hFile, LPVOID lpBuffer, DWORD nNumberOfBytesToRead, LPDWORD lpNumberOfBytesRead, LPOVERLAPPED lpOverlapped) – чтение из файла
	* BOOL WriteFile(HANDLE hFile, LPCVOID lpBuffer, DWORD nNumberOfBytesToWrite, LPDWORD lpNumberOfBytesWritten, LPOVERLAPPED lpOverlapped) – запись в файл
	* BOOL CloseHandle(HANDLE hObject) – закрытие файла
3. Способы копирования файлов

Пять различных вариантов реализации программ копирования файлов:

1. Программа cpC - использует библиотеку С
2. Программа cpW - является реализацией, в которой используются непосредственно средства Windows с буфером небольшого размера (256 байт)
3. Программа cpwFA — "быстрый" вариант реализации с использованием буфера большого размера (8192 байта)
4. Программа cpCF - использует функцию Windows CopyFile
5. Программа cpUC — реализация c использованием библиотеки совместимости с UNIX, использующая буфер небольшого размера
6. Функции управления каталогами
* BOOL CreateDirectory(LPCTSTR lpPathName, LPSECURITY\_ATTRIBUTES lpSecurityAttributes)- создание каталога
* BOOL RemoveDirectory(LPCTSTR lpPathName)- удаление каталога
* BOOL SetCurrentDirectory(LPCTSTR lpPathName) – установка рабочего каталога
* DWORD GetCurrentDirectory(DWORD cchCurDir, LPTSTR lpCurDir)- возвращение полного пути каталога
1. Поясните понятие реестр

реестр Windows — централизованная база данных, хранящая информацию о конфигурации системы, которой могут пользоваться как приложения, так и сама операционная система

1. Как получить структуру памяти системы?

Исследование структуры памяти в системе производится вызовом функции:

VOID GetSystemInfo(LPSYSTEM\_INFO lpSystemInfo), Параметром этой функции служит адрес структуры PSYSTEM\_INFO, в которой содержится информация относительно размера системной страницы, а также адресах физической памяти, доступных для приложений.

1. Основные функции для работы с пулами
* HANDLE GetProcessHeap(VOID) – получение дескриптора кучи процесса
* HANDLE HeapCreate(DWORD flOptions, SIZE\_T dwInitialSize, SIZE\_T dwMaximumSize) – создание дополнительной страницы кучи
* BOOL HeapDestroy(HANDLE hHeap)- уничтожение объекта кучи
1. При установлении размера кучи параметром dwInitialSize …

До размера страницы

1. Как управляется память кучи?
	* LPVOID НеарАllос(HANDLE hHeap, DWORD dwFlags, SIZE\_T dwBytes) – получение блока памяти из кучи
	* BOOL HeapFree(HANDLE hHeap, DWORD dwFlags, LPVOID lpMem) – освобождение блока памяти
	* LPVOID HeapReAlloc(HANDLE hHeap, DWORD dwFlags, LPVOID lpMem, SIZE\_T dwBytes) – повторное распределение блоков памяти
	* DWORD HeapSize(HANDLE hHeap, DWORD dwFlags, LPCVOID lpMem) – размер распределенного блока памяти
2. Понятие отображения файлов и получаемые преимущества

Динамическая память, распределенная в кучах, должна физически размещаться в файле подкачки. Управление перемещением страниц между физической памятью и файлом подкачки, а также отображением файла подкачки на виртуальное адресное пространство процесса осуществляется средствами ОС, ответственными за управление памятью. По завершении выполнения процесса физическое пространство в этом файле освобождается. Те же функциональные возможности Windows, которые обеспечивают отображение файла подкачки, позволяют отображать и обычные файлы. Отображение файлов дает следующие преимущества:

• Отпадает необходимость в выполнении операций файлового ввода/вывода

• Структуры данных, созданные в памяти, будут сохраняться в файле для последующего использования этой же или другими программами.

• Становится возможным применение удобных и эффективных алгоритмов, ориентированных на работу с файлами "в памяти", которые позволяют обрабатывать хранящиеся в файлах данные даже в тех случаях, когда размеры файлов значительно превышают доступный объем физической памяти.

• В некоторых случаях значительно повышается эффективность обработки файлов.

• Исчезает необходимость в управлении буферами и манипулировании содержащимися в них данными файлов.

• Обеспечивается возможность разделения памяти несколькими параллельно выполняющимися процессами.

 • Отпадает необходимость в расходовании излишнего пространства файла подкачки.

1. Объект отображения файлов и функция получения его дескриптора

Необходимо создать для открытого файла объект отображения файла (file mapping object), у которого имеется дескриптор, а затем отобразить этот файл или только некоторую его часть на виртуальное адресное пространство процесса. Объектам отображения можно присваивать имена, по которым к ним смогут обращаться другие процессы, разделяющие память совместно с данным процессом. Кроме того, объекты отображения файлов имеют параметры размера и атрибуты защиты.

HANDLE CreateFileMapping(HANDLE hFile, LPSECURITY\_ATTRIBUTES lpsa, DWORD dwProtect, DWORD dwMaximumSizeHigh, DWORD dwMaximumSizeLow, LPCTSTR lpMapName)

**Возвращаемое значение:** в случае успешного выполнения — дескриптор объекта отображения файла, иначе — NULL.

1. Понятие «базовый указатель»

базовые указатели (based pointers) являются фактически смещениями относительно других указателей, ссылающихся на область отображения файла, к которому впоследствии будет производиться обращение.