**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Для каких целей используется механизм проецирования в память?
   1. загрузки и выполнения EXE- и DLL-файлов Это позволяет существенно экономить как на размере страничного файла, так и на времени, необходимом для подготовки приложения к выполнению,
   2. доступа к файлу данных, размещенному на диске Это позволяет обойтись без операций файлового ввода-вывода и буферизации его содержимого,
   3. разделения данных между несколькими процессами, выполняемыми па одной машине (В Windows есть и другие методы для совместного доступа разных процессов к одним данным — но все они так или иначе реализованы на осно ве проецируемых в память файлов.)
2. Чем механизм проецирования в память отличается от работы с виртуальной памятью?

2.1 Как и виртуальная память, проецируемые файлы позволяют резервировать регион адресного пространства и передавать ему физическую память. Различие между этими механизмами заключается в том, что в последнем случае физическая память не выделяется из страничного файла, а берется из файла, уже находящегося на диске. Как только файл спроецирован в память, к нему можно обращаться так, будто он целиком в нее загружен.

1. Опишите последовательность действий при проецировании в память.
   1. Создайте объект “файл”, вызовом функции CreateFile.

3.2Создайте объект “проекция файла”, вызовом функции CreateFileMapping.

* 1. Закройте описатель файла, если он вам больше не нужен, вызовом CloseHandle.
  2. Отобразите весь или часть файла на адресное пространство, вызовом функции MapViewOfFile. (Таблица 5).
  3. Для отключения отображения воспользуйтесь UnmapViewOfFile.
  4. После окончания работы с проекцией файла закройте его описатель, вызовом функции CloseHandle.

1. Что случится, если при завершении проецирования в память не закрывать дескрипторы объектов с помощью функции CloseHandle ?

Закончив работу с любым открытым Вами объектом ядра, Вы должны его закрыть, иначе в процессе начнется утечка ресурсов.

**5.** Можно ли при проецировании в память увеличить размер файла?

Нет.  
**6.** Каким образом выполняется проецирование в память файлов, размер которых больше 4 Гбайт?

Вам придется проецировать не весь файл, а его представление, содержащее лишь некую часть данных Вы начнете с того, что спроецируете представление самого начала файла Закончив обработку данных в этом представлении, Вы отключите его и спроецируете представление следую щей части файла — и так до тех пор, пока не будет обработан весь файл Конечно, это делает работу с большими файлами, проецируемыми в память, не слишком удобной, но утешимся тем, что длина большинства файлов достаточно мала. Этот алгоритм проецирует представления по 64 Кб (в соответствии с гранулярностью выделения памяти) или менее Кроме того, функция *MapViewOfFile*требует, чтобы передаваемое ей смещение в файле тоже было кратно гранулярности выделения памяти. Подпрограмма проецирует на адресное пространство сначала одно представление, подсчитывает в нем количество нулей, затем переходит к другому представлению, и все повторяется. Спроецировав и просмотрев все 64-килобайтовые блоки, подпрограмма закрывает объект «проекция файла».