1. Что такое файл отображенный в память?

**Файл, отображённый в памяти** (memory-mapped file), — это механизм, который позволяет процессу работать с файлами как с обычной памятью. Вместо того, чтобы читать или записывать данные в файл через системные вызовы, процесс может отображать файл в своё адресное пространство.

1. Windows: что такое объект «проекции файла»?

**Проекция файлов** — это механизм, который позволяет отображать содержимое файла в адресное пространство процесса, как если бы это была часть оперативной памяти.

**Объект «проекция файла»** - примитивы, на основе которых диспетчер памяти реализует общую память.

Используется для отображения виртуальных адресов в основную память, в страничный файл или другой файл.

1. Windows: что такое представление отображения?

**Представление секции (section view)** – это часть секции, которая фактически видна процессу. Процесс создания представления для секции называется **отображением представления секции.**

Позволяет процессам экономить адресное пространство, потому что в память отображаются только представления объекта секции, необходимого в данный момент.

1. Windows: какие бывают отображения?

Приватные и разделяемые

1. Windows: какие функции входят в API для управления файлами отображенными в память?

**Как же использовать проецируемые в память файлы?**

Для этого нужно выполнить операции:

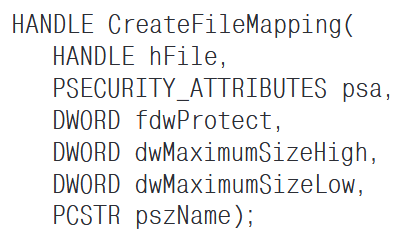
1. **Создать или открыть объект ядра «файл»**

[**CreateFile**](https://learn.microsoft.com/en-us/windows/win32/fileio/creating-and-opening-files)**:** имеет 3 важные параметра: первый параметр, **pszFileName**, идентифицирует имя создаваемого или открываемого файла; второй параметр, **dwDesiredAccess**, указывает способ доступа к содержимому файла (чтение, запись/чтение); третий параметр, **dwShareMode**, указывает тип совместного доступа к данному файлу.

Вызвав CreateFile, указали операционной системе, где находится физическая память для проекции файла

1. **Создать объект ядра «проекция файла» (CreateFileMaping -** возвращает дескриптор объекта**)**

Теперь сообщите системе, какой объем физической памяти нужен проекции файла. Для этого используем функцию [**CreateFileMapping**](https://learn.microsoft.com/en-us/windows/win32/memory/creating-a-file-mapping-object):



Первый параметр, ***hFile***, идентифицирует описатель файла, проецируемого на адресное пространство процесса. Этот описатель Вы получили после вызова CreateFile

При использовании **секций на базе страничных файлов** первый параметр должен быть установлен в INVALID\_HANDLE\_VALUE.

При создании объекта «проекция файла» система не резервирует регион адресного пространства и не увязывает его с физической памятью из файла

Но, как только дело дойдет до отображения физической памяти на адресное пространство процесса, системе понадобится точно знать атрибут защиты, присваиваемый страницам физической памяти. Поэтому в ***fdwProtect*** надо указать желательные атрибуты защиты.

Следующие два параметра этой функции (***dwMaximumSizeHigh*** и ***dwMaximumSizeLow***) самые важные. Основное назначение CreateFileMapping – гарантировать, что объекту «проекция файла» доступен нужный объем физической памяти. Через эти параметры мы сообщаем системе максимальный размер файла в байтах.

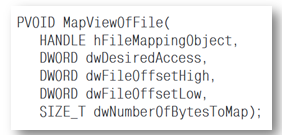
Вызов **CreateFileMapping** с флагом PAGE\_READWRITE заставляет систему проверять, чтобы размер соответствующего файла данных на диске был не меньше, чем указано в параметрах ***dwMaximumSizeHigh*** и ***dwMaximumSizeLow***. Если файл окажется меньше заданного, **CreateFileMapping** **увеличит его размер** до указанной величины.

1. **Проецирование файловых данных на адресное пространство процесса. Указать системе, как отобразить в адресное пространство процесса объект «проекция файла» – целиком или частично.**

Возвращает указатель на вид файла в адресном пространстве процесса.

Выделяется память, когда мы к ней обращаемся. Либо по умному зарезервированная память переходит в зафиксированную

Когда объект «проекция файла» создан, нужно, чтобы система, зарезервировав регион адресного пространства под данные файла, передала их как физическую память, отображенную на регион. Это делает функция [**MapViewOfFile**](https://learn.microsoft.com/en-us/windows/win32/memory/creating-a-file-view):



Параметр ***hFileMappingObject*** идентифицирует описатель объекта «проекция файла», возвращаемый предшествующим вызовом **CreateFileMapping.**

Параметр ***dwDesiredAccess*** идентифицируетвид доступа к данным.

Остальные три параметра относятся к резервированию региона адресного пространства и к отображению на него физической памяти.

Проецируя на адресное пространство процесса представление файла, нужно сделать две вещи. *Во-первых*, сообщить системе, какой байт файла данных считать в представлении первым. Для этого предназначены параметры ***dwFileOffsetHigh*** и ***dwFileOffsetLow***

*Во-вторых*, от Вас потребуется указать размер представления, т. е. сколько байтов файла данных должно быть спроецировано на адресное пространство в параметре ***dwNumberOfBytesToMap***

Заметьте, что смещение в файле должно быть кратно **гранулярности** выделения памяти в данной системе (64 КБ)

1. **Отключение файла данных от адресного пространства процесса**

Когда необходимость в данных файла отпадет, освободите регион вызовом [**UnmapViewOfFile**](https://learn.microsoft.com/en-us/windows/win32/memory/closing-a-file-mapping-object)**.**

Единственный параметр, ***pvBaseAddress***, указывает базовый адрес возвращаемого системе региона.

1. **Закрытие объектов «проекция файла» и «файл» -** функция CloseHandle.
2. Linux: какие бывают отображения и в чём их особенности?

Отображения могут производиться для двух типов объектов:

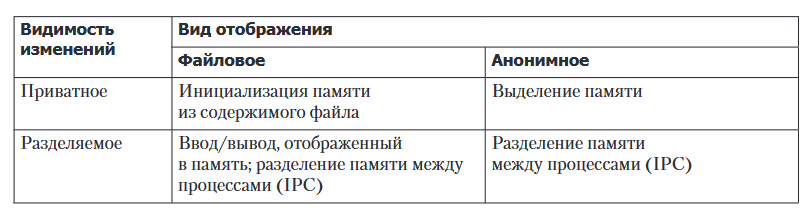
* **Обычный файл в файловой системе Linux**

Секция файла разделена на фрагменты размером со страницу, каждый из которых содержит исходное содержимое виртуальной страницы. Из-за подкачки по требованию ни одна из этих виртуальных страниц фактически не загружается в физическую память до тех пор, пока центральный процессор сначала не затронет страницу;

* **Анонимный файл**

Область памяти также может быть сопоставлена с анонимным файлом, созданным ядром, который заполнен нулями (файл бинарный).

1. **Приватное отображение** (MAP\_PRIVATE). Изменения, вносимые в содержимое отображения, остаются невидимыми для других процессов; в случае с файловым отображением они не передаются в исходный файл
2. **Разделяемое отображение** (MAP\_SHARED). Изменения, вносимые в содержимое отображения, доступны другим процессам, которые разделяют то же отображение; в случае с файловым отображением изменения передаются в исходный файл



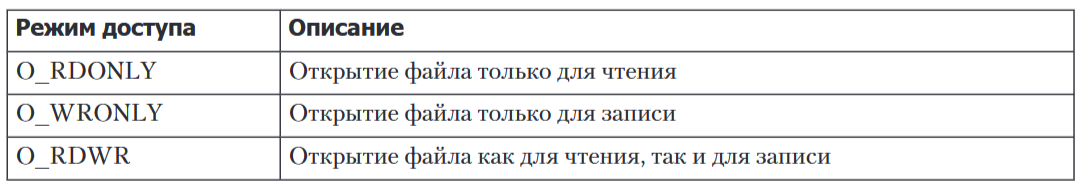
1. Linux: какие функции входят в API для управления файлами отображенными в память?

**Как же использовать проецируемые в память файлы?**

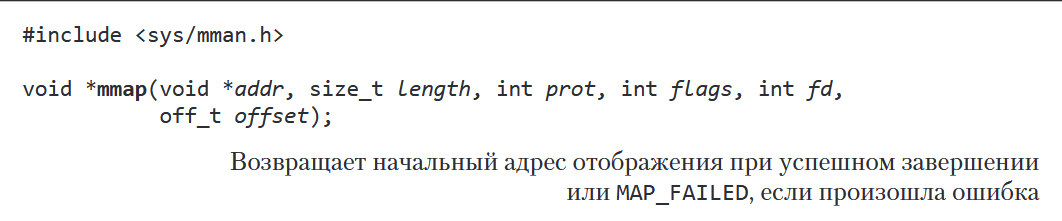
1. **Создание или открытие объекта ядра «файл». Получить дескриптор файла с помощью вызова open.**

**Pathname.** Если в этом аргументе находится символьная ссылка, она разыменовывается

Аргумент ***flags*** является битовой маской, указывающей режим доступа к файлу с использованием одной из констант



1. **Создание отображения**. Передать этот файловый дескриптор вызову mmap в виде аргумента fd



Аргумент ***addr*** обозначает виртуальный адрес, по которому будет находиться отображение.

Аргумент ***length*** обозначает размер отображения в байтах.

Аргумент ***prot*** представляет собой битовую маску, позволяющую задать защиту отображения.

Битовая маска ***flags*** состоит из параметров, управляющих различными аспектами работы отображения. В их число обязательно должно входить одно из следующих значений:

* MAP\_PRIVATE
* MAP\_SHARED

Оставшиеся аргументы, ***fd*** и ***offset***, используются в сочетании с файловыми отображениями (и игнорируются анонимными).

Аргумент ***fd*** – это файловый дескриптор файла, который нужно отобразить

Аргумент ***offset*** задает начальную позицию отображения в файле

1. **Удалить отображение файла**

Системный вызов [**munmap**](https://pubs.opengroup.org/onlinepubs/9799919799/functions/munmap.html)удаляет отображение из виртуального адресного пространства вызывающего процесса.

Системный вызов [msync](https://pubs.opengroup.org/onlinepubs/9799919799/functions/msync.html) предоставляет приложению возможность самостоятельно синхронизировать разделяемое отображение с отображенным файлом

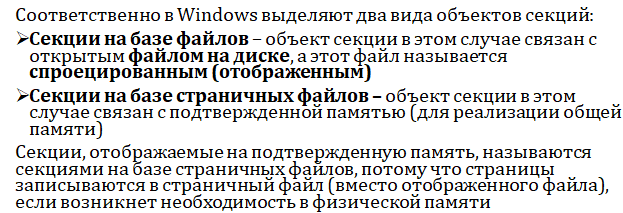
1. **Закрыть объект ядра «файл»**

Для закрытия объекта «файл» вызовите функцию [**close**](https://pubs.opengroup.org/onlinepubs/9799919799/functions/close.html)

1. Что такое Copy On Write?

Методика **копирования при записи**. То есть когда процесс пытается изменить содержимое страницы, ядро создает для него ее копию (и корректирует таблицу страниц процесса).

В связи с этим отображение MAP\_PRIVATE иногда называют **приватным, копируемым при записи.**

****

**Объект ядра** — это структура данных в операционной системе, которая используется ядром для управления ресурсами и синхронизации между процессами.

**Страничный файл** (или файл подкачки) — это файл на диске, который операционная система использует для управления памятью. Он позволяет расширить доступную оперативную память, временно храня данные, которые не помещаются в RAM. Это помогает избежать нехватки памяти, но доступ к данным на диске медленнее, чем в оперативной памяти.

**Рабочий набор** (working set) — это термин, используемый в области операционных систем и управления памятью, который описывает набор страниц памяти, активно используемых процессом в данный момент времени.

**Страницы памяти**:

* Рабочий набор состоит из страниц, которые являются минимальными единицами управления памятью. Размер страницы обычно составляет 4 КБ

**Область памяти** — это участок памяти, который выделяется для хранения данных, кода или других информационных единиц в компьютерной системе.

