

---

# Системное ПО

Файлы, отображаемые на память

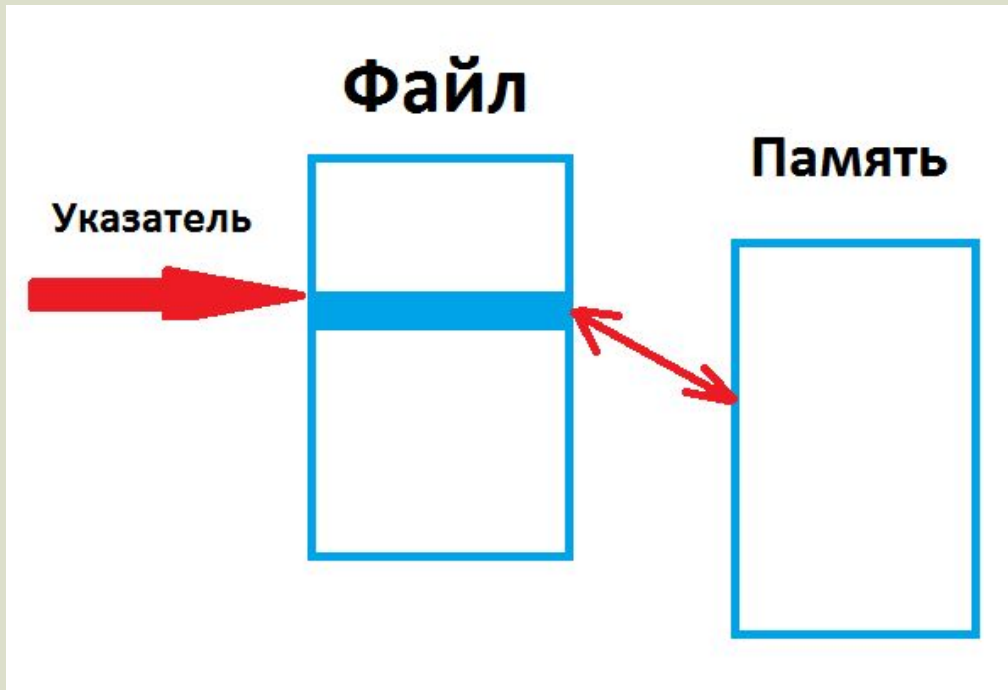
---

# Системное ПО

Файлы, отображаемые на память

## Способы работы с файлом

Классический



Можем скопировать в память/из памяти небольшой участок памяти, на который указывает указатель

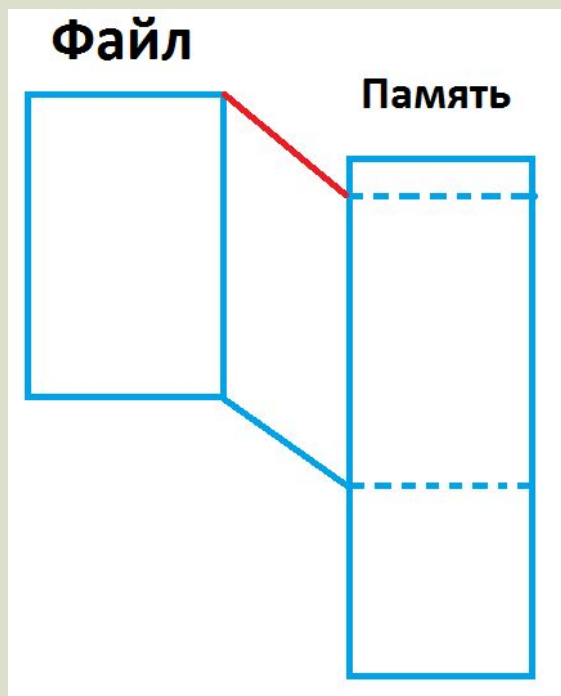
В случае сложной организации файла придется многократно «двигать» указатель и копировать нужный участок

# Системное ПО

Файлы, отображаемые на память

## Способы работы с файлом

Отображение на память



Можно работать с файлом, как с участком памяти

# Системное ПО

Файлы, отображаемые на память

```
HANDLE WINAPI CreateFileMapping(  
    _In_      HANDLE hFile,  
    _In_opt_ LPSECURITY_ATTRIBUTES lpAttributes,  
    _In_     DWORD flProtect,  
    _In_     DWORD dwMaximumSizeHigh,  
    _In_     DWORD dwMaximumSizeLow,  
    _In_opt_ LPCTSTR lpName  
);
```

Функция создает объект-отображение. Если размер отображения не задан – отображается весь файл

# Системное ПО

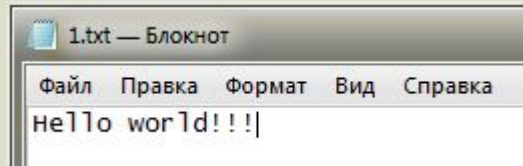
Файлы, отображаемые на память

```
LPVOID WINAPI MapViewOfFile(  
    _In_   HANDLE hFileMappingObject,  
    _In_   DWORD dwDesiredAccess,  
    _In_   DWORD dwFileOffsetHigh,  
    _In_   DWORD dwFileOffsetLow,  
    _In_   SIZE_T dwNumberOfBytesToMap  
);
```

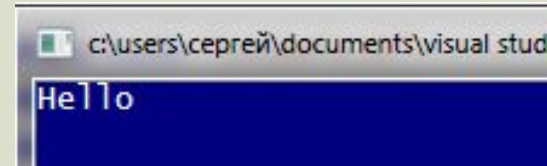
Функция отображает файл на память. Если задан нулевой размер, отображается весь файл.

# Системное ПО

Пример



Файлы, отображаемые на память



```
int _tmain(int argc, _TCHAR* argv[])
{
    HANDLE hFile,hFileMapping;
    char * lpMyFile;
    int i;

    hFile=CreateFile(L"C:\\1.txt",GENERIC_READ,0,NULL,OPEN_EXISTING,0,NULL);

    hFileMapping=CreateFileMapping(hFile,NULL,PAGE_READONLY,0,0,NULL);
    lpMyFile=(char*)MapViewOfFile(hFileMapping,FILE_MAP_READ,0,0,0);
    for(i=0;i<5;i++)
    {
        printf("%c",lpMyFile[i]);
    }
    _getch();
    return 0;
}
```