

Сергей Юрьевич Шилов

Системное программирование для современных платформ

Системное программирование для современных платформ

5. Ввод-вывод через прямое отображение в память (Direct mapping I/O)

Системное программирование для современных платформ

PRO

- работа с файлом как с массивом данных в памяти
- высокая производительность
- графические и текстовые редакторы, издательские системы, приложения видеомонтажа
- загрузка программного кода

CONTRA

- повышенные требования к объёму физической памяти
- возможна загрузка вредоносного программного кода

Системное программирование для современных платформ

```
#include <unistd.h>
#include <sys/mman.h>
```

```
void * mmap(void *start, size_t length, int prot , int flags, int
fd, off_t offset);
```

ОПИСАНИЕ

Функция **mmap** отражает **length** байтов, начиная со смещения **offset** файла, определенного дескриптором **fd**, в память, начиная с адреса **start**. Последний параметр (адрес) необязателен, и обычно бывает равен **NULL**. Настоящее местоположение отраженных данных возвращается самой функцией **mmap**, и никогда не бывает равным **NULL**. Аргумент **prot** описывает желаемый режим защиты памяти (он не должен конфликтовать с режимом открытия файла и общими ограничениями, накладываемыми операционной системой). Он является либо **PROT_NONE** либо побитовым **OR** одного или нескольких флагов **PROT_***.

PROT_EXEC - данные в страницах могут исполняться;

PROT_READ - данные можно читать;

PROT_WRITE - в эту область можно записывать информацию;

PROT_NONE - доступ к этой области памяти запрещен.

Системное программирование для современных платформ

Параметр **flags** задает тип отражаемого объекта, опции отражения и указывает, принадлежат ли отраженные данные только этому процессу или их могут читать другие. Он состоит из комбинации следующих флагов:

MAP_FIXED - не использовать другой адрес, если адрес задан в параметрах функции. Если заданный адрес не может быть использован, то функция **mmap** вернет сообщение об ошибке. Если используется **MAP_FIXED**, то **start** должен быть выровнен по размеру страницы. Использование этой опции не рекомендуется. Многие ОС допускают её использование только для суперпользователя.

MAP_SHARED - Разделить использование этого отражения с другими процессами, отражающими тот же объект. Запись информации в эту область памяти будет эквивалентна записи в файл. Файл может не обновляться до вызова функций **msync(2)** или **munmap(2)**.

MAP_PRIVATE - Создать неразделяемое отражение с механизмом **copy-on-write**. Запись в эту область памяти не влияет на файл. Не определено, являются или нет изменения в файле после вызова **mmap** видимыми в отраженном диапазоне. Один из флагов **MAP_SHARED**, либо **MAP_PRIVATE** должен быть установлен.

Значение **offset** должно быть выровнено по размеру страницы, получаемому при помощи функции **getpagesize(2)**.

Отображение в памяти наследуется через **fork(2)** с сохранением всех атрибутов. **exec(2)** не наследует отображение файлов в памяти.

Если размер отображаемой области не выровнен по границе страниц, она дополняется до этой границы нулями. Запись в зону выравнивания не сохраняется в файле. Поведение при изменении размера отображаемого файла неопределено.

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

При удачном выполнении **mmap** возвращает указатель на область с отображенными данными. При ошибке возвращается значение **-1**, а переменная **errno** приобретает соответствующее значение.

ЗАМЕЧАНИЯ

Поддержка **PROT_EXEC** в **PROT_READ** зависит от платформы. Портируемые программы должны всегда устанавливать **PROT_EXEC** если они намерены исполнить код. На многих платформах использование **PROT_EXEC** разрешено только суперпользователю.

Системное программирование для современных платформ

```
#include <unistd.h>
#include <sys/mman.h>
```

```
int munmap(void *start, size_t length);
int msync(void *start, size_t length, int flags);
```

ОПИСАНИЕ

Системный вызов **munmap** удаляет все отражения из заданной области памяти, после чего все ссылки на данную область будут вызывать ошибку **invalid memory reference**. Отображение удаляется автоматически при завершении процесса. С другой стороны, закрытие файла не приводит к снятию отображения

msync записывает на диск изменения, внесённые в файл, отображённый в память при помощи функции **mmap(2)**. Если не использовать эту функцию, то нет никакой гарантии, что изменения будут записаны в файл до вызова **munmap(2)**. На диск записывается часть файла, начинающаяся в памяти с адреса **start** длиной **length**. Параметр **flags** состоит из комбинации битов **MS_ASYNC**, **MS_SYNC** и **MS_INVALIDATE**. Одновременное использование флагов **MS_ASYNC** и **MS_SYNC** не допускается. **MS_ASYNC** даёт системе задание на запись и немедленно возвращается в вызывающий процесс. **MS_SYNC** даёт системе задание на запись и ждет его исполнения. **MS_INVALIDATE** приказывает системе обновить другие отражения этого файла так, чтобы они содержали изменения, внесённые этим вызовом.

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

При удачном выполнении **munmap** и **msync** возвращаемое значение равно нулю. При ошибке возвращается **-1**, а переменная **errno** приобретает соответствующее значение.

Системное программирование для современных платформ

Спасибо за внимание!
:-)