

Лабораторная работа №1 по курсу "Операционные системы"

Студент группы: М80-207Б-21, Крючков Артемий Владимирович

Контакты: artemkr2003@mail.ru

Работа выполнена: 17.09.2022

Преподаватель: Миронов Евгений Сергеевич

WRITE

НАЗВАНИЕ

write - производит запись в описатель файла

СИНТАКСИС

```
#include <unistd.h>
ssize_t write(int fd, const void *buf, size_t count);
```

ОПИСАНИЕ

write записывает до count байтов из буфера buf в файл, на который ссылается файловый описатель fd. POSIX указывает на то, что вызов write(), произошедший после вызова read() возвращает уже новое значение. Заметьте, что не все файловые системы соответствуют стандарту POSIX.

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

В случае успешного завершения возвращается количество байтов, которые были записаны (ноль означает, что не было записано ни одного байта). В случае ошибки возвращается -1, а переменной errno присваивается соответствующее значение. Если count равен нулю, а файловый описатель ссылается на обычный файл, то будет возвращен ноль и больше не будет произведено никаких действий. Для специальных файлов результаты не могут быть перенесены на другую платформу.

ММАР

НАЗВАНИЕ

mmap, munmap - отражает файлы или устройства в памяти или снимает их отражение

СИНТАКСИС

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

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

int munmap(void *start, size_t length);

#endif
```

ОПИСАНИЕ

Функция `mmap` отражает `length` байтов, начиная со смещения `offset` файла (или другого объекта), определенного файловым дескриптором `fd`, в память, начиная с адреса `start`. Последний параметр (адрес) необязателен, и обычно бывает равен 0. Настоящее местоположение отраженных данных возвращается самой функцией `mmap`, и никогда не бывает равным 0.

Аргумент `prot` описывает желаемый режим защиты памяти (он не должен конфликтовать с режимом открытия файла). Оно является либо `PROT_NONE` либо побитовым ИЛИ одного или нескольких флагов `PROT_*`.

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

При удачном выполнении `mmap` возвращает указатель на область с отраженными данными. При ошибке возвращается значение `MAP_FAILED` (-1), а переменная `errno` приобретает соответствующее значение. При удачном выполнении `munmap` возвращаемое значение равно нулю. При ошибке возвращается -1, а переменная `errno` приобретает соответствующее значение. (Вероятнее всего, это будет `EINVAL`).

ЕХЕС

ИМЯ

`execve` - выполнить программу

ОБЗОР

```
#include <unistd.h>
int execve(const char *filename, char *const argv [], char *const envp[]);
```

ОПИСАНИЕ

`execve()` выполняет программу, заданную параметром `filename`. Программа должна быть или двоичным исполняемым файлом, или скриптом, начинающимся со строки вида `"#! интерпретатор [аргументы]"`. В последнем случае интерпретатор -- это правильный путь к исполняемому файлу, который не является скриптом; этот файл будет выполнен как интерпретатор `[arg] filename`.

`argv` -- это массив строк, аргументов новой программы. `envp` -- это массив строк в формате `key=value`, которые передаются новой программе в качестве окружения

(environment). Как `argv`, так и `envp` завершаются нулевым указателем. К массиву аргументов и к окружению можно обратиться из функции `main()`, которая объявлена как `int main(int argc, char argv[], charenv[])`.

`execve()` не возвращает управление при успешном выполнении, а код, данные, `bss` и стек вызвавшего процесса перезаписываются кодом, данными и стеком загруженной программы. Новая программа также наследует от вызвавшего процесса его идентификатор и открытые файловые дескрипторы, на которых не было флага закрыть-при-`exec` (`close-on-exec`, `COE`). Сигналы, ожидающие обработки, удаляются. Переопределённые обработчики сигналов возвращаются в значение по умолчанию. Обработчик сигнала `SIGCHLD` (когда установлен в `SIG_IGN`) может быть сброшен или не сброшен в `SIG_DFL`.

Если текущая программа выполнялась под управлением `ptrace`, то после успешного `execve()` ей посылается сигнал `SIGTRAP`.

Если на файле программы `filename` установлен `setuid`-бит, то фактический идентификатор пользователя вызвавшего процесса меняется на идентификатор владельца файла программы. Точно так же, если на файле программы установлен `setgid`-бит, то фактический идентификатор группы устанавливается в группу файла программы.

Если исполняемый файл является динамически-скомпонованным файлом в формате `a.out`, содержащим заглушки для вызова разделяемых библиотек, то в начале выполнения этого файла вызывается динамический компоновщик `ld.so(8)`, который загружает библиотеки и компоует их с исполняемым файлом.

Если исполняемый файл является динамически-скомпонованным файлом в формате `ELF`, то для загрузки разделяемых библиотек используется интерпретатор, указанные в сегменте `PT_INTERP`. Обычно это `/lib/ld-linux.so.1` для программ, скомпилированных под Linux libc версии 5, или же `/lib/ld-linux.so.2` для программ, скомпилированных под GNU libc версии 2.

ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ

При успешном завершении `execve()` не возвращает управление, при ошибке возвращается `-1`, а значение `errno` устанавливается должным образом.