# Семинар 3 ОС (09.03.2022)

## Виртуальная файловая система proc

Почему эта файловая система называется виртуальной? Потому что она не монтируется … ,а … на лету, при обращении.

… процессам, то есть приложениям, информации о ресурсах, о самих процессах.

Сама виртуальная файловая система proc – структура ядра. (Должна предоставлять информацию … пользовательским процессам)

Будет лабораторная по proc из двух частей (сдавать можно отдельно):

1. Используем интерфейс proc в режиме пользователя для получения соответствующей информации. \*(Позволяет получать информацию обычными функциями работы с файлами – обращаемся к информации как к файлам или символическим ссылкам)

Фактически proc представляет собой интерфейс ядра, позволяющий … читать и изменять данные в адресном пространстве ядра, получать информацию о процессах и ресурсах, которые используются процессами, при этом используется стандартный интерфейс файловой системы и …. ?Осуществление доступа через установление прав доступа к файлам? Предоставляет интерфейс к структурам данных ядра (Из мануала). Монтируется автоматически при инициализации системы, но можно написать и руками:  
mount -t proc proc/proc

Не существует, не находится ни на одном устройстве вторичной памяти, создаётся на лету.

(У нас в литературе принято называть виртуально, у них псевдо – почти. Имеется ввиду что её на самом деле нет, но она есть)

/proc/[pid]

pid = getpid();

/proc/setf/ - обратится к информации о самом себе

|  |  |  |
| --- | --- | --- |
| элемент | тип | содержание |
| cmdline | файл | Указание на директорию процесса (Фактически путь директории) |
| cmd | символическая ссылка |  |
| einviron | файл | Большой объём информации (называется список окружения процесса) |
| exe | символическая ссылка | Указывает на образ процесса |
| fd (file decriptor) | директория | Ссылки на открытые процессом файлы |
| root | символьная ссылка |  |
| stat | файл | Большая, объёмная информация о процессе (большое кол-во полей) |
| io | файл | Статистическая информация о работе процесса с устройствами ввод вывода (большой объём полей) |
| maps | файл | В мануале описывается какую информацию предоставляет этот файл (речь идёт об адресном пространстве…), фактически это регионы адресного пространства… |
| pagemap | файл |  |

cmdline будет содержать полную командную строку для …

символическая ссылка – особый вид файлов …

#include <stdio.h>  
int main (…)  
{  
 FILE \*f = foppen(“/proc/setf/environ”, “r”);  
 while((len=fread(buf, 1, BUF\_SIZE, f)) > 0)  
 {  
 for (i=0; i<len; i++)  
 if(buf[i] == 0)  
 buf[i]=10; //for  
 buf[len] = 0;  
 printf(“%s ”, buf);  
 }  
 fclose(f);  
 return(0);  
}

10 или 0x0A (конец файла)  
BUF\_SIZE <– #define 0x100

…

То есть надо записать всю эту информацию в файл.

…

#cat /prco/<pid>/io

rchar: (кол-во байтов которые этим заданием были записаны или обработаны из памяти)  
wcahr: (w - write)  
syscr: (read syscalls – кол-во выполненных вызовов read/write)  
syscw:   
read\_bytes:   
write\_bytes:   
…

Всё это вывести в файл…  
Смысл этих действий заключается в том, что мы волей-неволей знакомимся с той информацией, которую предоставляет вирт. файловая система proc (в user mode)

dev  
08:02

старший и младший номер устройства – major, minor

Записать последовательно эту информацию в один файл (чтобы это было читаемо).

Это информация, которая находится в директории каждого процесса: /proc/[pid]

Для этого нам надо будет запустить какой-то процесс (например, курсовую по КГ, демона)

proc предоставляет информацию просто, не связанную с процессам а именно:

**/proc/filesystems** (это файл, содержащий файловые системы, которые определены ядром. Это имена файловых систем, которые были скомпилированы в ядро и которые ядро загрузило в текущий момент). При этом если файловая система определена nodev (nodevice) это означает что данная файловая система не требует монтирования на блочное устройство, то есть это virtual file system (может быть или VFS, или network file system).

**/proc/fs** В данном случае эта поддиректория содержит информацию о файлах смонтированной файловой системы

**/proc/interrupts** здесь можно посмотреть кол-во прерываний на процессор и на устройство ввода/вывода

**/proc/modules** это текстовый файл, который содержит перечень модулей, которые были загружены системой

**/proc/ioports** (порты ввод/вывода) э то список зарегистрированных в текущий момент времени ввода/вывода (?ввод это диапазон адресов? Надо запастись флешками…)

**/proc/apm** информация о состоянии батареи ноутбука

## Вторая часть лабораторной работы

?Это уже работа в ядре.?

Очевидно, что в ядре мы также можем выполнять ?работу? с виртуальной файловой системой proc

Файлы и поддиректории файловой системы proc могут создаваться, их можно регистрировать и прекращать их регистрацию динамически, поэтому ф.с. proc часто используется с загружаемыми модулями ядра.

Файлы и поддиректории файловой системы proc используют структуру  
typedef int(read\_proc\_t)(char \*page, char \*\*start, off\_t off, int count, int \*eof, void \*data)  
typedef int (write\_proc\_t)(struct file \*file, const char \_\_user \*buffer, unsigned long count, void \*data);  
struct proc\_dir\_entry  
{  
 unsigned int low\_ino;  
 unsigned short namelen;  
 const char \*name; //имя виртуального файла  
 mode\_t mode; //права доступа, режим доступа к файлу  
 hlink\_t hlink; // кол-во линков  
 uid\_t uid;  
 gid\_t gid;  
 loff\_t size; //loff\_t – unsigned\_t?  
 const struct inode\_operations \*proc\_iops;  
 const struct file\_operations \*proc\_fops;  
 struct proc\_dir\_entry \*next, \*parent, \*subdir;  
 void \*data;  
 read\_proc\_t \*read\_proc;  
 write\_proc\_t \*write\_proc;  
…

В структуре определён указатель на struct file\_opeartions. В структуре file\_opeartions имеются все функции в системе для работы с файлами. Обычно это структура используется для регистрации собственных функций чтения и записи в системе (в нашем загружаемом модуле)

Смысл 2ой части этой ЛР – передача информации из user\_mode в kernel и из kernel в user\_mode. Существует несколько механизмов, мы рассмотрим 2:

1. Фартунки (copy\_to\_user, copy\_from\_user)
2. seg\_file