



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»
КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 4
Тема: файловая система /proc
Дисциплина: Операционные системы

Студент Лучина Е.Д

Группа ИУ7-61Б

Преподаватель Рязанова Н.Ю.

Москва.

2020 г.

В пользовательском режиме вывести на экран информацию об окружении процесса с комментариями	3
В пользовательском режиме вывести на экран информацию о состоянии процесса с комментариями	6
Вывести информацию из файла cmdline и директории fd	10
Часть 2	12

В пользовательском режиме вывести на экран информацию об окружении процесса с комментариями

Окружение (environment) или среда — это набор пар ПЕРЕМЕННАЯ=ЗНАЧЕНИЕ, доступный

каждому пользовательскому процессу. Иными словами, окружение — это набор переменных

окружения. Данная информация хранится в файле /proc/[pid]/environ. Вывести его содержимое можно следующим образом:

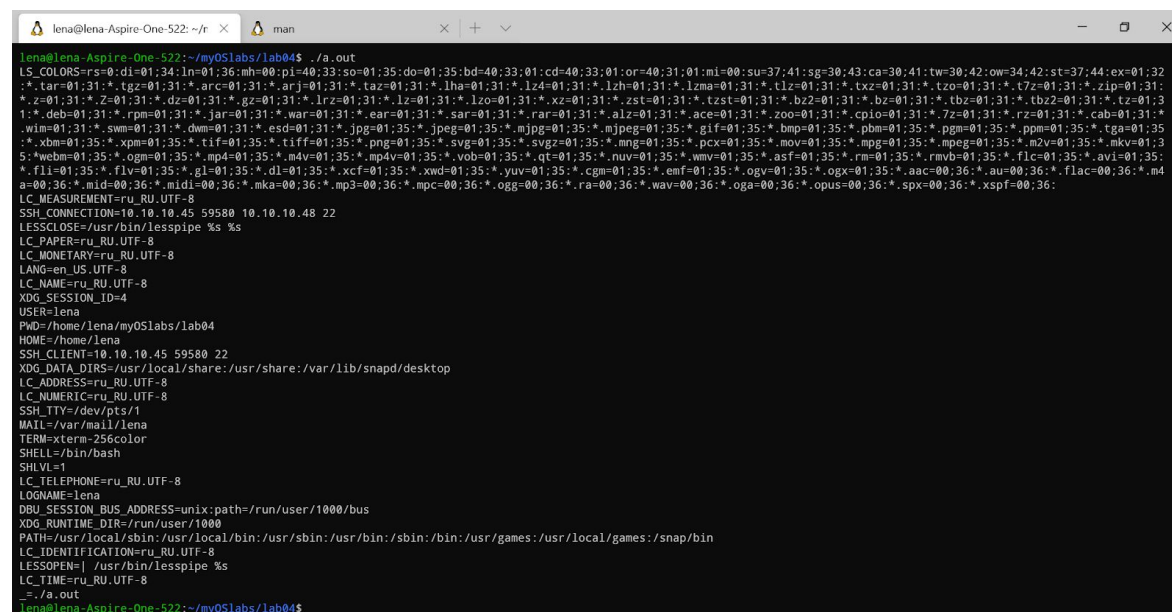
```
#include <stdio.h>
#define BUF_SIZE 1024

int main(void)
{
    int len;
    char buf[BUF_SIZE];

    FILE *f = fopen("/proc/self/environ", "r");
    while ((len = fread(buf, 1, BUF_SIZE, f)) > 0)
    {
        for (int i = 0; i < len; i++)
            if (buf[i] == 0)
                buf[i] = 10;

        buf[len-1] = 0;
        printf("%s", buf);
    }
    fclose(f);
}
```

результат



```
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out
LS_COLORS=rs=0:di=01:34:ln=01:36:mh=00:pi=40:33:so=01:35:do=01:35:bd=40:33:01:cd=40:33:01:or=40:31:01:mi=00:su=37:41:sg=30:43:ca=30:41:tw=30:42:ow=34:42:st=37:44:ex=01:32:.*=01:31:*.tgz=01:31:*.arc=01:31:*.arj=01:31:*.taz=01:31:*.lha=01:31:*.lzh=01:31:*.lma=01:31:*.tlz=01:31:*.txz=01:31:*.tzo=01:31:*.t7z=01:31:*.zip=01:31:*.z=01:31:*.Z=01:31:*.gz=01:31:*.lrz=01:31:*.lz=01:31:*.lzo=01:31:*.xz=01:31:*.zst=01:31:*.tzt=01:31:*.bz2=01:31:*.bz=01:31:*.tbz=01:31:*.tbz2=01:31:*.tz=01:31:*.deb=01:31:*.rpm=01:31:*.jar=01:31:*.war=01:31:*.ear=01:31:*.sar=01:31:*.rar=01:31:*.alz=01:31:*.ace=01:31:*.zoo=01:31:*.cpio=01:31:*.7z=01:31:*.rz=01:31:*.cab=01:31:*.wim=01:31:*.swm=01:31:*.dwm=01:31:*.esd=01:31:*.jpg=01:35:*.jpeg=01:35:*.mjpg=01:35:*.mjpeg=01:35:*.gif=01:35:*.bmp=01:35:*.pbm=01:35:*.pgm=01:35:*.ppm=01:35:*.tga=01:35:*.xbm=01:35:*.xpm=01:35:*.tif=01:35:*.tiff=01:35:*.png=01:35:*.svg=01:35:*.svgz=01:35:*.mng=01:35:*.pex=01:35:*.mov=01:35:*.mpg=01:35:*.mpeg=01:35:*.m2v=01:35:*.mkv=01:35:*.webm=01:35:*.ogm=01:35:*.mp4=01:35:*.m4v=01:35:*.m4p=01:35:*.vob=01:35:*.qt=01:35:*.nuv=01:35:*.wmv=01:35:*.asf=01:35:*.rm=01:35:*.rmvb=01:35:*.flc=01:35:*.avi=01:35:*.fli=01:35:*.flv=01:35:*.gl=01:35:*.dl=01:35:*.xcf=01:35:*.xwd=01:35:*.yuv=01:35:*.cgm=01:35:*.emf=01:35:*.ogv=01:35:*.ogx=01:35:*.aac=00:36:*.au=00:36:*.flac=00:36:*.m4a=00:36:*.mid=00:36:*.midi=00:36:*.mka=00:36:*.mp3=00:36:*.mpc=00:36:*.ogg=00:36:*.ra=00:36:*.wav=00:36:*.oga=00:36:*.opus=00:36:*.spx=00:36:*.xspf=00:36:
LC_MEASUREMENT=ru_RU.UTF-8
SSH_CONNECTION=10.10.10.45 59580 10.10.10.48 22
LESSCLOSE=/usr/bin/lesspipe %s %s
LC_PAPER=ru_RU.UTF-8
LC_MONETARY=ru_RU.UTF-8
LANG=en_US.UTF-8
LC_NAME=ru_RU.UTF-8
XDG_SESSION_ID=4
USER=lena
PWD=/home/lena/myOSlabs/lab04
HOME=/home/lena
SSH_CLIENT=10.10.10.45 59580 22
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/napd/desktop
LC_ADDRESS=ru_RU.UTF-8
LC_NUMERIC=ru_RU.UTF-8
SSH_TTY=/dev/pts/1
MAIL=/var/mail/lena
TERM=xterm-256color
SHELL=/bin/bash
SHLVL=1
LC_TELEPHONE=ru_RU.UTF-8
LOGNAME=lena
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
XDG_RUNTIME_DIR=/run/user/1000
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
LC_IDENTIFICATION=ru_RU.UTF-8
LESSOPEN=| /usr/bin/lesspipe %s
LC_TIME=ru_RU.UTF-8
lena@lena-Aspire-One-522:~/myOSlabs/lab04$
```

LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:	используется для определения цветов, с которыми будут выведены имена файлов при вызове ls.
LC_MEASUREMENT=ru_RU.UTF-8	Система измерения, используемая в регионе по умолчанию.
SSH_CONNECTION=10.10.10.45 59580 10.10.10.48 22	Определяет клиент и сервер конца соединения. Переменная содержит четыре значения через пробел: IP-адрес клиента, номер порта клиента, IP-адрес сервера и номер порта сервера.
LESSCLOSE=/usr/bin/lesspipe %s %s	определяет постобработчик файла, который открывается при вызове less.
LC_PAPER=ru_RU.UTF-8	размер бумаги по умолчанию для региона
LC_MONETARY=ru_RU.UTF-8	денежное форматирование
LANG=en_US.UTF-8	язык и кодировка пользователя
LC_NAME=ru_RU.UTF-8	применяется для представления имени и должности лица

XDG_SESSION_ID=4	переменная, необходимые для вызова xdg-open, использующейся для открытия файла или URL в пользовательском приложении.
USER=lena	имя пользователя, от чьего имени запущен процесс
PWD=/home/lena/myOSlabs/lab04	путь к рабочей директории
HOME=/home/lena	путь к домашнему каталогу текущего пользователя.
SSH_CLIENT=10.10.10.45 59580 22	IP-адрес клиента SSH, затем пробел и номер порта. Сервер SSH устанавливает это для команд, выполняемых в контексте службы.
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop	переменная, необходимые для вызова xdg-open, использующейся для открытия файла или URL в пользовательском приложении.
LC_ADDRESS=ru_RU.UTF-8	Конвенция, используемая для форматирования уличных или почтовых адресов
LC_NUMERIC=ru_RU.UTF-8	Числовое форматирование
SSH_TTY=/dev/pts/1	установлена на имя tty (путь к устройству), связанный с текущей оболочкой или командой. Если текущий сеанс не имеет управляющего терминала, эта переменная не установлена.
MAIL=/var/mail/lena	Расположение входящей локальной почты.
TERM=xterm-256color	тип запущенного терминала.
SHELL=/bin/bash	путь к предпочтительной оболочке командной строки.
SHLVL=1	уровень текущей командной оболочки
LC_TELEPHONE=ru_RU.UTF-8	Соглашения, используемые для представления телефонных номеров
LOGNAME=lena	имя текущего пользователя
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus	Переменная, необходимая dbus сервисам
XDG_RUNTIME_DIR=/run/user/1000	переменная, необходимые для вызова xdg-open, использующейся для открытия файла или URL в пользовательском приложении.

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin	список каталогов, в которых система ищет исполняемые файлы.
LC_IDENTIFICATION=ru_RU.UTF-8	невидимая пользователем категория, она содержит информацию о самой локали и редко полезна для пользователей или разработчиков
LESSOPEN= /usr/bin/lesspipe %s	определяет предобработчик файла, который открывается при вызове less.
LC_TIME=ru_RU.UTF-8	Локальные форматы времени и даты
_=./a.out	полная командная строка процесса

В пользовательском режиме вывести на экран информацию о состоянии процесса с комментариями

Информация о процессе содержится в файле /proc/self/stat. Содержимое файла выглядит следующим образом. (программа выводит просто все содержимое аналогично предыдущей программе)

```
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ gcc stat.c
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out
2885 (a.out) R 2293 2885 2293 34817 2885 4194304 70 0 0 0 0 0 20 0 1 0 681750 4612096 204 18446744073709551615 94706377551872 94706377554560 140736146462832 0 0 0 0 0 0 0 0 17 0 0 0 0 0 94706379652504 94706379653136 94706390257664 140736146466488 140736146466496 140736146466496 140736146468848 0lena@lena-Aspire-One-522:~/myOSlabs/lab04$
```

Представим это в более удобной форме.

```
#include <stdio.h>
#include <string.h>
#define BUF_SIZE 1024

int main(void)
{
    char *statCont[] = {
        "pid",
        ...
        "env_end",
        "exit_code"
    }; // 52 values

    char buf[BUF_SIZE];
    FILE *f = fopen("/proc/self/stat", "r");
    int len;
    while ((len = fread(buf, 1, BUF_SIZE, f)) > 0)
    {
        for (int i = 0; i < len; i++)
            if (buf[i] == 0)
                buf[i] = 10;

        buf[len-1] = 0;
        int len = strlen(buf);
        int cur = 0;
        char *pch = strtok(buf, " ");
        while (pch != NULL && cur < 52)
```

```

        {
            printf("%-20s\t%s\n", statCont[cur], pch);
            pch = strtok(NULL, " ");
            cur++;
        }
    }
    fclose(f);
}

```

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out
pid                2972
comm               (a.out)
state              R
ppid               2293
pgrp               2972
session            2293
tty_nr             34817
tpgid              2972
flags               4194304
minflt             69
cminflt            0
majflt             0
cmajflt            0
utime              0
stime              0
cutime             0
cstime             0
priority           20
nice               0
num_threads        1
itrealvalue        0
startttime         854371
vsize              4616192
rss                177
rsslim             18446744073709551615
startcode          93984213667840
endcode            93984213672520
startstack         140722403532640
kstkesp            0
kstkeip            0
signal             0
blocked            0
sigignore          0
sigcatch           0
wchan              0
nswap              0
cnsnap             0
exit_signal        17
processor           0
rt_priority         0
policy             0
delayacct_blkio_tics 0
guest_time         0
cguest_time        0
start_data         93984215772552
end_data           93984215773632
start_brk          93984234987520
arg_start          140722403534520
arg_end            140722403534528
env_start          140722403534528
env_end            140722403536880
exit_code          0
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ |

```

pid	2972	Идентификатор процесса
comm	(a.out)	имя исполняемого файла
state	R	состояние процесса - running
ppid	2293	идентификатор процесса-предка

pgrp	2972	идентификатор группы
session	2293	идентификатор сессии
tty_nr	34817	управляющий терминал
tpgid	2972	уникальный идентификатор группы управляющего терминала
flags	4194304	флаги
minflt	69	количество незначительных сбоев, которые возникли при выполнении процесса, и которые не требуют загрузки страницы памяти с диска
cminflt	0	количество незначительных сбоев, которые возникли при ожидании окончания работы процессов-потомков.
majflt	0	количество значительных сбоев, которые возникли при работе процесса, и которые потребовали загрузки страницы памяти с диска.
cmajflt	0	количество значительных сбоев, которые возникли при ожидании окончания работы процессов-потомков
utime	0	количество тиков, которые данный процесс провел в режиме пользователя
stime	0	количество тиков, которые данный процесс провел в режиме ядра
cutime	0	количество тиков, которые процесс, ожидающий завершения процессов-потомков, провёл в режиме пользователя
cstime	0	количество тиков, которые процесс, ожидающий завершения процессов-потомков, провёл в режиме ядра
priority	20	для процессов реального времени это отрицательный приоритет планирования минус один, то есть число в диапазоне от -2 до -100, соответствующее приоритетам в реальном времени от 1 до 99. Для остальных процессов это необработанное значение nice, представленное в ядре. Ядро хранит значения nice в виде чисел в диапазоне от 0 (высокий) до 39 (низкий), соответствующих видимому пользователю диапазону от -20 до 19.
nice	0	значение для nice в диапазоне от 19 (наиболее низкий приоритет) до -20 (наивысший приоритет)
num_threads	1	число потоков в данном процессе
itrealvalue	0	количество мигнов до того, как следующий SIGALARM будет послан процессу интервальным таймером. С ядра версии 2.6.17

		больше не поддерживается и установлено в 0.
starttime	840948	время в тиках запуска процесса после начальной загрузки системы.
vsize	4616192	размер виртуальной памяти в байтах. Имеем 4508 КБ или около 4.6 МБ
rss	189	резидентный размер: количество страниц, которые занимает процесс в памяти. Это те страницы, которые заняты кодом, данными и пространством стека. Сюда не включаются страницы, которые не были загружены по требованию или которые находятся в своппинге.
rsslim	18446744073709551615	текущий лимит в байтах на резидентный размер процесса.
startcode	94520998121472	адрес, выше которого может выполняться код программы.
endcode	94520998126152	адрес, ниже которого может выполняться код программ.
startstack	140735609978832	адрес начала стека.
kstkesp	0	текущее значение ESP (указателя стека)
kstkeip	0	текущее значение EIP (указатель команд).
signal	0	битовая карта ожидающих сигналов. Устарела, потому что не предоставляет информацию о сигналах реального времени, необходимо использовать /proc/[pid]/status.
blocked	0	битовая карта блокируемых сигналов. Устарела, потому что не предоставляет информацию о сигналах реального времени, необходимо использовать /proc/[pid]/status.
sigignore	0	битовая карта игнорируемых сигналов. Устарела, потому что не предоставляет информацию о сигналах реального времени, необходимо использовать /proc/[pid]/status.
sigcatch	0	битовая карта перехватываемых сигналов. Устарела, потому что не предоставляет информацию о сигналах реального времени, необходимо использовать /proc/[pid]/status.
wchan	0	“канал”, в котором ожидает процесс.
nswap	0	количество страниц на своппинге (не обслуживается)
cnsnap	0	суммарное nswap для процессов-потомков (не обслуживается)
exit_signal	17	сигнал, который будет послан предку, когда

		процесс завершится SIGCHLD - Child process has stopped or exited, changed (POSIX)
processor	0	номер процессора, на котором последний раз выполнялся процесс.
rt_priority	0	приоритет планирования реального времени, число в диапазоне от 1 до 99 для процессов реального времени, 0 для остальных
policy	0	политика планирования.
delayacct_blkio_ticks	0	суммарные задержки ввода/вывода в тиках.
guest_time	0	гостевое время процесса (время, потраченное на выполнение виртуального процессора на гостевой операционной системе) в тиках.
cguest_time	0	гостевое время для потомков процесса в тиках.
start_data	94521000226184	адрес, выше которого размещаются инициализированные и неинициализированные (BSS) данные программы.
end_data	94521000227264	адрес, ниже которого размещаются инициализированные и неинициализированные (BSS) данные программы.
start_brk	94521032708096	адрес, выше которого куча программы может быть расширена с использованием brk()
arg_start	140735609984696	адрес, выше которого размещаются аргументы командной строки (argv).
arg_end	140735609984704	адрес, ниже которого размещаются аргументы командной строки (argv).
env_start	140735609984704	адрес, выше которого размещается окружение программы.
env_end	140735609987056	адрес, ниже которого размещается окружение программы.
exit_code	0	статус завершения потока в форме, возвращаемой waitpid().

Вывести информацию из файла cmdline и директории fd

/proc/[pid]/cmdline - данный файл содержит полную командную строку процесса, если процесс не находится в состоянии зомби, иначе файл пуст. Аргументы командной строки представлены в виде набора строк, разделенных символом конца строки '\0'.

```
#include <stdio.h>
```

```

#define BUF_SIZE 1024

int main(int argc, char **argv)
{
    int len;
    char buf[BUF_SIZE];

    FILE *f = fopen("/proc/self/cmdline", "r");
    while ((len = fread(buf, 1, BUF_SIZE, f)) > 0)
    {
        for (int i = 0; i < len; i++)
            if (buf[i] == 0)
                buf[i] = 10;

        buf[len-1] = 0;
        printf("%s", buf);

    }
    fclose(f);
    printf("\n");
}

```

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ gcc cmdline.c
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out
./a.out
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out hello
./a.out
hello
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out hello world
./a.out
hello
world
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out hello world 1 2 34
./a.out
hello
world
1
2
34
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ |

```

/proc/[pid]/fd - данная поддиректория содержит одну запись для каждого файла, который открыт процессом. Имя каждой такой записи соответствует номеру файлового дескриптора и является символьной ссылкой на реальный файл. Так, 0 - это стандартный ввод, 1 - стандартный вывод, 2 - стандартный вывод сообщений об ошибках и т. д.

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <dirent.h>
#include <string.h>
#define BUFF_SIZE 1024

int main(void)
{
    struct dirent *dirp;
    DIR *dp;

```

```

char str[BUFF_SIZE];
char path[BUFF_SIZE];
dp = opendir("/proc/self/fd/");
while ((dirp = readdir(dp)) != NULL)
{
    if ((strcmp(dirp->d_name, ".") != 0) &&
        (strcmp(dirp->d_name, "..") != 0))
    {
        sprintf(path, "%s%s", "/proc/self/fd", dirp->d_name);
        readlink(path, str, BUFF_SIZE);
        printf("%s -> %s\n", dirp->d_name, str);
    }
}
closedir(dp);
}

```

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ gcc fd.c
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ./a.out
0 ->
1 ->
2 ->
3 ->
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ |

```

Часть 2

part2.c

```

#include <linux/module.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/proc_fs.h>
#include <linux/string.h>
#include <linux/vmalloc.h>
#include <asm/uaccess.h>
#include <linux/uaccess.h>

#define COOKIE_BUF_SIZE PAGE_SIZE

MODULE_LICENSE("GPL");
MODULE_AUTHOR("Luchina");

ssize_t fortune_read(struct file *file, char *buf, size_t count, loff_t
*f_pos);
ssize_t fortune_write(struct file *file, const char *buf, size_t count,
loff_t *f_pos);
int fortune_init(void);
void fortune_exit(void);

struct file_operations fops = {
    .owner = THIS_MODULE,
    .read = fortune_read,
    .write = fortune_write,
};

```

```

char *cookie_buf;
struct proc_dir_entry *proc_file;
unsigned int read_index;
unsigned int write_index;

ssize_t fortune_read(struct file *file, char *buf, size_t count, loff_t
*f_pos)
{
    int len;

    if (write_index == 0 || *f_pos > 0)
        return 0;

    if (read_index >= write_index)
        read_index = 0;

    len = sprintf(buf, "%s\n", &cookie_buf[read_index]);
    read_index += len;
    *f_pos += len;

    return len;
}

ssize_t fortune_write(struct file *file, const char *buf, size_t count,
loff_t *f_pos)
{
    int free_space = (COOKIE_BUF_SIZE - write_index) + 1;

    if (count > free_space)
    {
        printk(KERN_INFO "Cookie pot full.\n");
        return -ENOSPC;
    }

    if (copy_from_user(&cookie_buf[write_index], buf, count))
        return -EFAULT;

    write_index += count;
    cookie_buf[write_index-1] = 0;

    return count;
}

int fortune_init(void)
{
    cookie_buf = vmalloc(COOKIE_BUF_SIZE);

    if (!cookie_buf)
    {
        printk(KERN_INFO "Not enough memory for the cookie
pot.\n");
        return -ENOMEM;
    }

    memset(cookie_buf, 0, COOKIE_BUF_SIZE);
    proc_file = proc_create("fortune", 0666, NULL, &fops);

    if (!proc_file)
    {

```

```

        vfree(cookie_buf);
        printk(KERN_INFO "Cannot create fortune.\n");
        return -ENOMEM;
    }

    read_index = 0;
    write_index = 0;

    proc_mkdir("mydir", NULL);
    proc_symlink("mysymlink", NULL, "/proc/fortune");

    printk(KERN_INFO "Fortune module loaded.\n");
    return 0;
}

void fortune_exit(void)
{
    remove_proc_entry("fortune", NULL);

    if (cookie_buf)
        vfree(cookie_buf);

    printk(KERN_INFO "Fortune module unloaded.\n");
}

module_init(fortune_init);
module_exit(fortune_exit);

```

Makefile

```

ifneq ($(KERNELRELEASE),)
    obj-m := part2.o
else
    CURRENT = $(shell uname -r)
    KDIR = /lib/modules/$(CURRENT)/build
    PWD = $(shell pwd)

default:
    $(MAKE) -C $(KDIR) M=$(PWD) modules

clean:
    @rm -f *.o *.cmd *.flags *.mod.c *.order
    @rm -f *.*.cmd *~ *.*~ TODO.*
    @rm -fR .tmp*
    @rm -rf .tmp_versions

disclean: clean
    @rm *.ko *.symvers

endif

```

1) Загрузить модуль

```

lena@lena-Aspire-One-522: ~/r
lena@lena-Aspire-One-522: ~/r
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ make
make -C /lib/modules/5.3.0-28-generic/build M=/home/lena/myOSlabs/lab04 modules
make[1]: Entering directory '/usr/src/linux-headers-5.3.0-28-generic'
CC [M] /home/lena/myOSlabs/lab04/part2.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/lena/myOSlabs/lab04/part2.mod.o
LD [M] /home/lena/myOSlabs/lab04/part2.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.3.0-28-generic'
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ sudo insmod part2.ko
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ lsmod | head
Module              Size  Used by
part2                16384  0
ccm                  20480  9
uvcvideo             94208  0
videobuf2_vmalloc    20480  1 uvcvideo
videobuf2_memops      20480  1 videobuf2_vmalloc
videobuf2_v4l2        24576  1 uvcvideo
videobuf2_common      49152  2 videobuf2_v4l2,uvcvideo
videodev              208896  3 videobuf2_v4l2,uvcvideo,videobuf2_common
ath9k                 151552  0
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ modinfo part2.ko
filename:            /home/lena/myOSlabs/lab04/part2.ko
author:              Luchina
license:             GPL
srcversion:          D6BCA28FDA318645B09BC3B
depends:
retpoline:           Y
name:                part2
vermagic:            5.3.0-28-generic SMP mod_unload
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ dmesg | tail -2
[11691.321060] ---[ end trace 0c1ad2e638238d66 ]---
[11691.321064] Fortune module loaded.

```

- 2) Продемонстрировать передачу данных из режима пользователя в режим ядра и из режима ядра в режим пользователя с помощью скриншота.

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ echo "test1" > /proc/fortune
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ cat /proc/fortune
test1
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ echo "test2" > /proc/fortune
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ echo "test3" > /proc/fortune
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ cat /proc/fortune
test2
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ cat /proc/fortune
test3
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ |

```

- 3) Показать в каталоге созданный файл, директорию и поддиректорию.

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ls -l /proc | grep "my"
dr-xr-xr-x  2 root      root           0 июн  9 15:41 mydir
lrwxrwxrwx  1 root      root           13 июн  9 15:41 mysymlink -> /proc/fortune
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ ls -l /proc/fortune
-rw-rw-rw-  1 root root 0 июн  9 17:06 /proc/fortune
lena@lena-Aspire-One-522:~/myOSlabs/lab04$

```

- 4) Выгрузить модуль

```

lena@lena-Aspire-One-522:~/myOSlabs/lab04$ sudo rmmod part2.ko
lena@lena-Aspire-One-522:~/myOSlabs/lab04$ dmesg | tail -3
[11691.321060] ---[ end trace 0c1ad2e638238d66 ]---
[11691.321064] Fortune module loaded.
[11974.274045] Fortune module unloaded.
lena@lena-Aspire-One-522:~/myOSlabs/lab04$

```