



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
по операционным системам**

Студент Колганов О.С.

Группа ИУ7 — 62Б

Преподаватель Рязанова Н.Ю.

Москва.
2020 г.

Задание 1:

- В пользовательском режиме вывести на экран информацию об окружении процесса с комментариями;
- В пользовательском режиме вывести на экран информацию о состоянии процесса с комментариями;
- Вывести информацию из файла cmdline и директории fd.

Листинг функции вывода информации об окружении :

```
int output_info_from_file(const char* filename)
{
    FILE* f = fopen(filename, "r");
    if (!f) {
        printf("Can't open file!");
        return FILE_OPEN_ERROR;
    }

    size_t len = 0;
    char buf[BUF_SIZE];

    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);
    return 0;
}
```

Результат выполнения для /proc/self/environ:

```
environ:
SHELL=/bin/bash
SESSION_MANAGER=local/Moxxx1e:@/tmp/.ICE-unix/1706,unix/Moxxx1e:/tmp/.ICE-unix/1706
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK_IM_MODULE=ibus
QT4_IM_MODULE=ibus
GNOME_SHELL_SESSION_MODE=ubuntu
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
SSH_AGENT_PID=1625
GTK_MODULES=gall:atk-bridge
PWD=/home/oleg/Документы/OS/lab_04
LOGNAME=oleg
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
XAUTHORITY=/run/user/1000/gdm/Xauthority
GJS_DEBUG_TOPICS=JS ERROR;JS LOG
WINDOWPATH=2
HOME=/home/oleg
USERNAME=oleg
IM_CONFIG_PHASE=1
LANG=ru_RU.UTF-8
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;4
2: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:*.lzh=01;31:*.lzm=01;31:*.tlz
=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=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:*.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=0
1;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:*.huv=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=1;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:*.o
gg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
```

```
XDG_CURRENT_DESKTOP=ubuntu:GNOME
VTE_VERSION=5802
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/4a9ef7d1_aa1f_4e51_b284_b8c1cbfff0d7
INVOCATION_ID=2c40cbaaa4234c7792e77a6056bb7363
MANAGERPID=1508
CLUTTER_IM_MODULE=ibus
GJS_DEBUG_OUTPUT=stderr
LESSCLOSE=/usr/bin/lesspipe %s %s
XDG_SESSION_CLASS=user
TERM=xtem-256color
LESSOPEN=| /usr/bin/lesspipe %s
USER=oleg
GNOME_TERMINAL_SERVICE=:1.1555
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1000
JOURNAL_STREAM=9:37383
XDG_DATA_DIRS=/usr/share/ubuntu:/usr/local/share:/usr/share:/var/lib/napd/desktop
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
GDMSESSION=ubuntu
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
OLDPWD=/home/oleg/Документы/OS
_=./lab_04
```

Результат выполнения для /proc/self/cmdline:

```
cmdline:
./lab_04
```

Листинг функции для вывода информации о состоянии процесса:

```
#define BUF_SIZE 0x200
void output_statement()
{
    char buf[BUF_SIZE];
    FILE* f = fopen("/proc/self/stat", "r");
    fread(buf, 1, BUF_SIZE, f);
    char* pch = strtok(buf, " ");
    printf("stat: \n");
    int i = 0;
    while (i < NUMBER_OF_COMMENTS && pch != NULL) {
        printf("%s: ", comments_for_stat[i]);
        printf("%s\n", pch);
        pch = strtok(NULL, " ");
        i++;
    }
    fclose(f);
}
```

Результат работы функции:

```
stat:
1) pid: 14255
2) comm: (lab_04)
3) state: R
4) ppid: 14239
5) pgrp: 14255
6) session: 14239
7) tty_nr: 34817
8) tpgid: 14255
9) flags: 4194304
10) minflt: 76
11) cminflt: 0
12) majflt: 0
13) cmajflt: 0
14) utime: 0
15) stime: 0
16) cutime: 0
17) cstime: 0
18) priority: 20
19) nice: 0
20) num_threads: 1
21) itrealvalue: 0
22) starttime: 1558270
23) vsize: 2539520
24) rss: 193
25) rsslim: 18446744073709551615
26) startcode: 94515894194176
27) endcode: 94515894200309
28) startstack: 140723477378720
29) kstkesp: 0
30) kstkeip: 0
31) signal: 0
32) blocked: 0
33) sigignore: 0
34) sigcatch: 0
35) wchan: 0
36) nswap: 0
37) cnsnap: 0
```

```
38) exit_signal: 17
39) processor: 3
40) rt_priority: 0
41) policy: 0
42) delayacct_blkio_ticks: 0
43) guest_time: 0
44) cguest_time: 0
45) start_data: 94515894209872
46) end_data: 94515894211008
47) start_brk: 94515903062016
48) arg_start: 140723477386239
49) arg_end: 140723477386248
50) env_start: 140723477386248
51) env_end: 140723477389295
52) exit_code: 0
```

Листинг функции для вывода информации из директории fd:

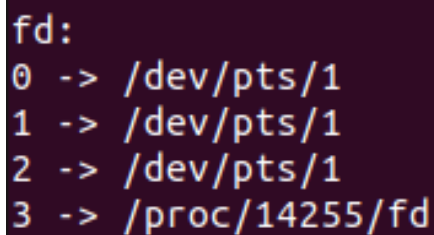
```
void output_fd()
{
    printf("fd:\n");

    struct dirent* dirp;
    DIR* dp;
    char str[BUF_SIZE];
    char path[BUF_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, BUF_SIZE);
            printf("%s -> %s\n", dirp->d_name, str);
        }
    }

    closedir(dp);
}
```

Результат работы функции:



```
fd:
0 -> /dev/pts/1
1 -> /dev/pts/1
2 -> /dev/pts/1
3 -> /proc/14255/fd
```

Задание 2:

Написать программу – загружаемый модуль ядра (LKM) – которая поддерживает чтение из пространства пользователя и запись в пространство пользователя. После загрузки модуля пользователь может загружать в него строки с помощью команды echo, а затем автоматически считывать их с помощью команды cat.

Листинг программы:

```
#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>

MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Fortune cookie Kernel Module");
MODULE_AUTHOR("Kolganov Oleg, IU7-62B");

ssize_t fortune_read(struct file *filp, char* __user buf, size_t count, loff_t *f_pos);
ssize_t fortune_write(struct file *filp, const char* __user buf, size_t len, 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,
};

#define MAX_COOKIE_LENGTH PAGE_SIZE

static struct proc_dir_entry* proc_entry;
static char* cookie_pot;
static int cookie_index_write;
static int cookie_index_read;
ssize_t fortune_read(struct file *filp, char* __user buff, size_t count, loff_t *f_pos)
{
    int len;

    if (!cookie_index_write || *f_pos > 0)
        return 0;

    if (cookie_index_read >= cookie_index_write)
        cookie_index_read = 0;

    len = sprintf(buff, "%s\n", &cookie_pot[cookie_index_read]);
    cookie_index_read += len;
    *f_pos += len;

    return len;
}

ssize_t fortune_write(struct file* filp, const char* __user buff, size_t len, loff_t* data)
{
    size_t space_available = (MAX_COOKIE_LENGTH - cookie_index_write) + 1;

    if (len > space_available) {
        printk(KERN_INFO "fortune: cookie pot is full!\n");
        return -ENOSPC;
    }

    if (copy_from_user(&cookie_pot[cookie_index_write], buff, len)) {
        return -EFAULT;
    }

    cookie_index_write += len;
    cookie_pot[cookie_index_write - 1] = '\0';

    return len;
}
```

```

#define FILE_NAME "fortune"
#define DIR_NAME "fortune_dir"
#define SYMLINK_NAME "fortune_l"
int init_fortune_module(void)
{
    int ret = 0;

    cookie_pot = (char*)vmalloc(MAX_COOKIE_LENGTH);
    if(!cookie_pot) {
        ret = -ENOMEM;
        return ret;
    }

    memset(cookie_pot, 0, MAX_COOKIE_LENGTH);
    proc_entry = proc_create(FILE_NAME, 0666, NULL, &fops);

    if (proc_entry == NULL) {
        ret = -ENOMEM;
        vfree(cookie_pot);
        printk(KERN_INFO "fortune: Couldn't create proc entry!\n");
        return ret;
    }

    cookie_index_write = 0;
    cookie_index_read = 0;

    proc_mkdir(DIR_NAME, NULL);
    proc_symlink(SYMLINK_NAME, NULL, "/proc/" FILE_NAME);

    printk(KERN_INFO "fortune: Module loaded.\n");
    return ret;
}

void cleanup_module(void)
{
    remove_proc_entry(SYMLINK_NAME, NULL);
    remove_proc_entry(FILE_NAME, NULL);
    remove_proc_entry(DIR_NAME, NULL);
    if (cookie_pot) {
        vfree(cookie_pot);
    }
    printk(KERN_INFO "fortune: Module unloaded.\n");
}

module_init(init_fortune_module);
module_exit(cleanup_module);

```

Загрузка модуля:

```

oleg@Moxxx1e:~/Документы/OS/OS/lab_04_2$ sudo insmod fortune.ko
oleg@Moxxx1e:~/Документы/OS/OS/lab_04_2$ ls
cmake-build-debug  fortune.c  fortune.ko  fortune.mod  Makefile
oleg@Moxxx1e:~/Документы/OS/OS/lab_04_2$ lsmod | grep fortune
fortune            16384  0

```


Созданы также директория и символическая ссылка:

-rw-rw-rw-	1	root	root	0	anp	1	17:43	fortune
dr-xr-xr-x	2	root	root	0	anp	1	17:43	fortune_dir
lrwxrwxrwx	1	root	root	13	anp	1	17:43	fortune_l -> /proc/fortune
10	15	1780	1880	2044	390	930		toauavg
1010	150	1786	1888	2627	397	931		locks
1053	1508	1787	1897	263	398	934		mdstat
11	15083	1795	1923	264	399	983		meminfo
12	1509	1799	1924	2666	4	988		misc
12046	15213	18	1937	2676	40	989		modules
1212	1529	1804	1968	2695	41	acpi		mounts
1225	153	1808	1993	27	42	asound		mtrr
1228	1532	1811	2	28	447	brcm_monitor0		net
1230	1537	18197	2009	29	472	buddyinfo		pagetypeinfo
12338	1539	1822	2063	296	510	bus		partitions
12494	154	18258	21	297	517	cgroups		pressure
12650	1541	18303	2189	3	5273	cmdline		sched_debug
13	1545	18306	22	30	540	consoles		schedstat
13047	1555	1831	23	33	5403	cpuinfo		scsi
1319	157	1836	23149	3325	561	crypto		self
136	15888	1838	23171	34	6409	devices		slabinfo
137	15914	18387	23201	342	6417	diskstats		softirqs
138	15915	18388	23221	35	6443	dma		stat
14	16	1841	23234	36	6557	driver		swaps
140	16014	18420	23695	361	827	execdomains		sys
141	1625	18422	23734	365	828	fb		sysrq-trigger
14152	1646	18426	23794	366	890	filesystems		sysvipc
14168	1648	1847	24	367	891	fortune		thread-self
142	1656	1857	241	369	9	fortune_dir		timer_list
14217	16584	1862	242	37	911	fortune_l		tty
14228	1660	1863	243	371	912	fs		uptime
14239	168	1868	2432	374	913	interrupts		version
143	1681	187	244	375	915	iomem		version_signature

Та же информация с ключом -all

dr-xr-xr-x	2	root	root	0	anp	1	17:43	driver
-r--r--r--	1	root	root	0	anp	1	17:43	execdomains
-r--r--r--	1	root	root	0	anp	1	17:43	fb
-r--r--r--	1	root	root	0	anp	1	17:43	filesystems
-rw-rw-rw-	1	root	root	0	anp	1	17:43	fortune
dr-xr-xr-x	2	root	root	0	anp	1	17:43	fortune_dir
lrwxrwxrwx	1	root	root	13	anp	1	17:43	fortune_l -> /proc/fortune
dr-xr-xr-x	5	root	root	0	anp	1	11:02	fs
-r--r--r--	1	root	root	0	anp	1	17:42	interrupts
-r--r--r--	1	root	root	0	anp	1	17:43	iomem
-r--r--r--	1	root	root	0	anp	1	17:43	ioports
dr-xr-xr-x	28	root	root	0	anp	1	11:02	irq

Демонстрация работы:

```
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ echo "Success is an individual proposition. Thomas Watson" > /proc/fortune
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ echo "If a man does his best, what else is there? Gen. Patton" > /proc/fortune
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ cat /proc/fortune
Success is an individual proposition. Thomas Watson
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ cat /proc/fortune
If a man does his best, what else is there? Gen. Patton
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ cat /proc/fortune
Success is an individual proposition. Thomas Watson
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ cat /proc/fortune
If a man does his best, what else is there? Gen. Patton
oleg@Moxxxie:~/Документы/OS/OS/Lab_04_2$ cat /proc/fortune
Success is an individual proposition. Thomas Watson
```


Выгрузка модуля:

```
oleg@Moxxx1e:~/Документы/OS/OS/Lab_04_2$ sudo rmmod fortune
oleg@Moxxx1e:~/Документы/OS/OS/Lab_04_2$ dmesg | grep fortune | tail -2
[25178.422460] fortune: Module loaded.
[25314.477488] fortune: Module unloaded.
oleg@Moxxx1e:~/Документы/OS/OS/Lab_04_2$ lsmod | grep fortune
oleg@Moxxx1e:~/Документы/OS/OS/Lab_04_2$ ls /proc
```