

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
(НИЯУ МИФИ)
Институт интеллектуальных кибернетических систем
Кафедра Кибернетики

**Лабораторная работа №4 по курсу
«Разработка ПО ОС UNIX»**

Выполнил студент группы Б15-501:

Огнянович Павел

Проверил:

Ктитров С.В.

Москва, 2018

Задание

Разработать программу для Solaris, печатающую и изменяющую по просьбе пользователя флаги, связанные с описателем файла. Установить флаг close-on-exec и проверить доступность описателя в запущенном процессе с указанным флагом и без него.

Код программы

Lab-4.c

```
#include <fcntl.h>

#include <unistd.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#define FILENAME "testfile123.txt"

int main(int argc, char *argv[]) {

    int fd = creat(FILENAME, S_IRWXG | S_IRWXU | S_IRWXO);

    int flags;

    char fdc = '0'+fd;

    char fdarr[10];

    sprintf(fdarr, "%d", fd);

    pid_t pid;

    int clex = fcntl(fd, F_GETFD);

    int i=0;

    char c;

    printf(">>>Set close-on-exec flag?(y/n)\n");

    c = getchar();

    char t = getchar();

    if(c=='Y' || c=='y') {

        if(clex==0) {
```

```

        fcntl(fd, F_SETFD, 1);
    } else{
        fcntl(fd, F_SETFD, 0);
    }
}

clex = fcntl(fd, F_GETFD);

printf(">>>1 - print all flags\n>>>2 - change close-on-exec(current: %d)\n>>>3 - test write to
fd\n>>>4 - exit\n", clex);

c = getchar();
t = getchar();
switch(c) {
    case '1':
//print all current flags
        flags = fcntl(fd, F_GETFL);
        printf("flags: %d\n", flags);
        break;
    case '2':
//change cloexec
        //(clex==0)?
        //    (fcntl(fd, F_SETFD, 1)):
        //    (fcntl(fd, F_SETFD, 0));
        break;
    case '3':
//test writing
        switch(pid = fork()) {
            case -1:
                printf(">>>FORK_ERROR!\n");
                break;
            case 0:
                printf(">>>CHILD_CASE!\n");
                //execl("example.exe", NULL);
                execl("lab-4-2.exe", "lab-4-2.exe", fdarr, NULL);
                printf("ALL IS BAD!!!");

```

```

        break;

    default:
        printf(">>>DEFAULT_AFTER_FORK!\n");
        break;

    }

    //write(fd, "HELLO", 5);

    break;

case '4':

    return 0;

default:

    printf(">>>UNKNOWN_COMMAND_NUMBER!\n");

}

//close(fd);

return 0;

}

```

Lab-4-2.c

```

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <sys/types.h>

#include <string.h>

```

```

int main(int argc, char *argv[]) {

    printf("CHILD CODE EXECUTION\n");

    char *ch = argv[1];

    if(ch==NULL) {

        printf("COMMAND_LINE_ARGUMENT_ERROR!");

        return 1;

    }
}

```

```
int fd = atoi(ch);  
printf("CHILD: fd=%d\n",fd);  
char buff[] = "1234567890";  
int size = strlen(buff);  
if(write(fd, buff, size)!=size)  
    printf("WRITE_ERROR!\n");  
close(fd);  
return 0;  
}
```