

Одесский национальный политехнический университет

Институт компьютерных систем

Кафедра информационных систем

Лабораторная работа № 8

По дисциплине: «Операционные системы»

**Тема: «Программирование управлением процессами в ОС Unix»**

Выполнила:

Студентка группы АИ-205

Колдунова А. А.

Проверил:

Блажко О.А.

Одесса 2020

**Цель работы:** получение навыков в управлении процессами в ОС Unix на уровне языка программирования С

**План работы.**

1 Теоретические сведения

1.1 Расширенный функционал команд оболочки ОС интерфейса командной строки и утилит командной строки

1.2 Редактирование каталогов и файлов файловой системы

1.3 Перенаправление потоков данных

1.4 Конвейеризация команд

1.5 Команды оболочки и утилиты командной строки по обработке текста

1.6 Конфигурация работы оболочки

2 Задание к исполнению

3 Требования к оформлению протокола выполнения лабораторной работы

**Решение:**

**Задание 1:**

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

int main(int argc, const char * argv[]) {
    pid_t pid = getpid();
    pid_t ppid = getppid();
    pid_t uid = getuid();
    pid_t gid = getgid();
    pid_t sid = getsid(pid);
    printf("my sid=%d\n", sid);
    printf("My pgrp=%d\n", getpgrp());
    printf("My pid=%d\n", pid);
    printf("My ppid=%d\n", ppid);
    printf("My uid=%d\n", uid);
    printf("My gid=%d\n", gid);
    return 0;
}
```

[ Read 17 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```
[[koldunova_anastasiya@vpsj3IeQ ~]]$ nano main.c
[[koldunova_anastasiya@vpsj3IeQ ~]]$ gcc -o main main.c
```

```
[[koldunova_anastasiya@vpsj3IeQ ~]]$ nano main.c
[[koldunova_anastasiya@vpsj3IeQ ~]]$ ./main
my sid=15216
My pgrp=19548
My pid=19548
My ppid=15216
My uid=54414
My gid=54420
[[koldunova_anastasiya@vpsj3IeQ ~]]$
```

**Задание 2:**

```

#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern char** environ;
int main(int argc, const char * argv[]) {
    char* echo_args[] = {"echo", "Process of Koldunova got signal\n", NULL};
    pid_t pid = fork();
    if(pid == 0)
        printf("Child prosecc: pid = %d\n",getgid());
    else{
        printf("Parent prosecc: pid = %d\n",getgid());
        execve("/bin/echo", echo_args, environ);
    }
    return 0;
}

```

[ Read 17 lines ]  
 ^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
 ^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text^T To Spell

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ nano information.c
[[koldunova_anastasiya@vpsj3IeQ ~]$ gcc -o information information.c
```

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ ./information
Parent prosecc: pid = 54420
Child prosecc: pid = 54420
Process of Koldunova got signal
```

```
[koldunova_anastasiya@vpsj3IeQ ~]$
```

### Задание 3:

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ nano task3.c
[[koldunova_anastasiya@vpsj3IeQ ~]$ gcc -o task3 task3.c
[[koldunova_anastasiya@vpsj3IeQ ~]$ ./task3
```

```

#include <stdio.h>
#include <signal.h>
#include <unistd.h>

static void sig_usr(int signo) {
    if (signo == SIGUSR2) {
        printf("i got SIGUSR2\n");
    }
}

int main(void) {
    if (signal(SIGUSR2, sig_usr) == SIG_ERR) {
        fprintf(stderr, "Error");
    }
    for ( ; ; ) {
        pause();
    }
    return 1;
}

```

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ nano task3.2.c
[[koldunova_anastasiya@vpsj3IeQ ~]$ gcc -o task3.2 task3.2.c
```

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

pid_t pid = 18144;

int main(void) {
    if (!kill(pid, SIGUSR2))
        printf("Send signal go pid=%d", pid);
    else
        fprintf(stderr, "Error");
        return 1;
}
```

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ ./task3.2
Error[[koldunova_anastasiya@vpsj3IeQ ~]$ ]
```

#### Задание 4:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/types.h>

int main(void) {
    int i;
    pid_t pid = fork();
    if(pid != 0){
        printf("I am parent with pid = %d. My child pid = %d\n", getpid(), getppid());
        sleep(2);
        _exit(0);
    } else {
        for(i = 0; i<5; i++) {
            printf("I am child with pid=%d. My parent id=%d", getpid(), getppid());
            sleep(1);
        }
    }
    return 0;
}
```

```
[[koldunova_anastasiya@vpsj3IeQ ~]$ gcc -o task4 task4.c
[[koldunova_anastasiya@vpsj3IeQ ~]$ ./task4
I am parent with pid = 23321. My child pid = 15216
[[koldunova_anastasiya@vpsj3IeQ ~]$ I am child with pid=23322. My parent id=23321
I am child with pid=23322. My parent id=23321I am child with pid=23322. My parent id=1I am child with pid=23322. My parent id=1I am child with pid=23322. My parent id=1[[koldunova_anastasiya@vpsj3IeQ ~]$ ]
```

**Вывод:**

В данной лабораторной мы получили навыки в управлении процессами в ОС Unix на уровне языка программирования С