

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота №8
З дисципліни "Операційні системи"
Тема : «Програмування керуванням процесами в ОС Unix»

Виконав
Ст.гр. AI-204
Костецький Б.В.
Перевірили:
Блажко О.А.

Одеса 2021

Мета роботи: отримання навичок в управлінні процесами в ОС Unix на рівні мови програмування C.

Завдання до виконання

1.Перегляд інформації про процес

Створіть C-програму, яка виводить на екран таку інформацію:

- ідентифікатор групи процесів лідера сесії;
- ідентифікатор групи процесів, до якої належить процес;
- ідентифікатор процесу, що викликав цю функцію;
- ідентифікатор батьківського процесу;
- ідентифікатор користувача процесу, який викликав цю функцію;
- ідентифікатор групи користувача процесу, який викликав цю функцію.

2.Стандартне створення процесу

Створіть C-програму, яка створює процес-нащадок, породжуючи процес та замінюючи образ процесу. У програмі процес-батько повинен видати повідомлення типу «Parent of Ivanov», а процес-нащадок повинен видати повідомлення типу «Child of Ivanov» через виклик команди echo, де замість слова Ivanov в повідомленні повинно бути ваше прізвище в транслітерації.

3.Обмін сигналами між процесами

3.1 Створіть C-програму, в якій процес очікує отримання сигналу SIGUSR2 та виводить повідомлення типу «Process of Ivanov got signal» після отримання сигналу, де замість слова Ivanov в повідомленні повинно бути ваше прізвище в транслітерації. Запустіть створену C-програму.

3.2 Створіть C-програму, яка надсилає сигнал SIGUSR2 процесу, запущеному в попередньому пункту завдання. Запустіть створену C-програму та проаналізуйте повідомлення, які виводить перша програма. Завершіть процес, запущеному в попередньому пункту завдання.

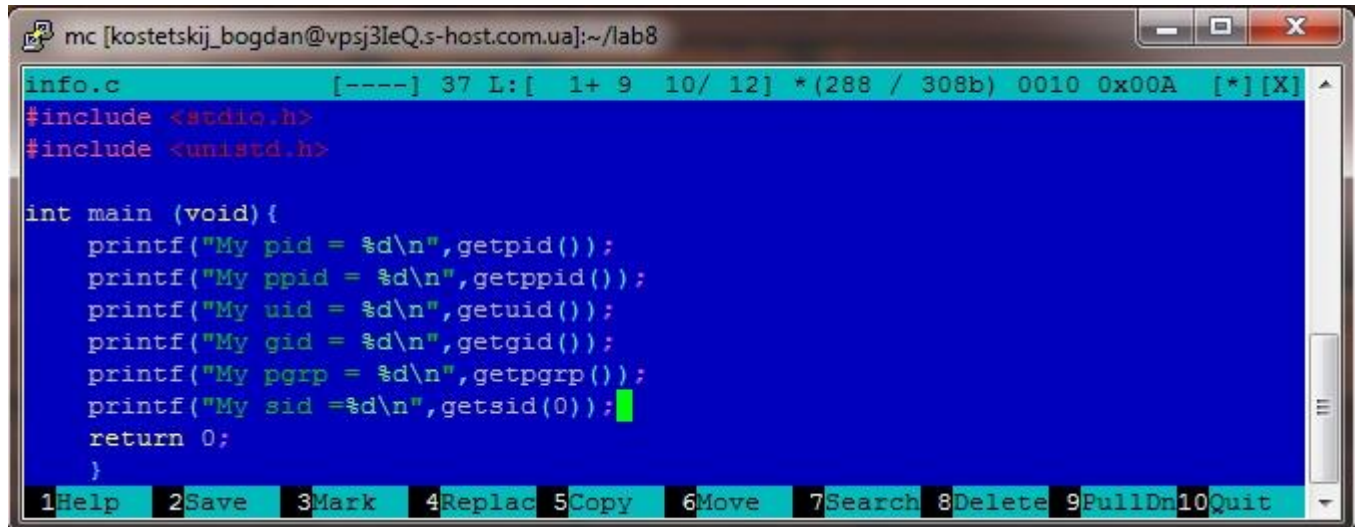
4.Створення процесу-сироти

Створіть C-програму, в якій процес-батько несподівано завершується раніше процесу-нащадку. Процес-батько повинен очікувати завершення $n+1$ секунд. Процес-нащадок повинен в циклі $(2*n+1)$ раз із затримкою в 1 секунду виводити повідомлення, наприклад, «Parent of Ivanov», за шаблоном як в попередньому завданні, і додатково виводити PPID процесу-батька. Значення n – номер команди студента + номер студента в команді. Перевірте роботу програми, вивчіть вміст таблиці процесів і зробіть відповідні висновки.

Хід роботи

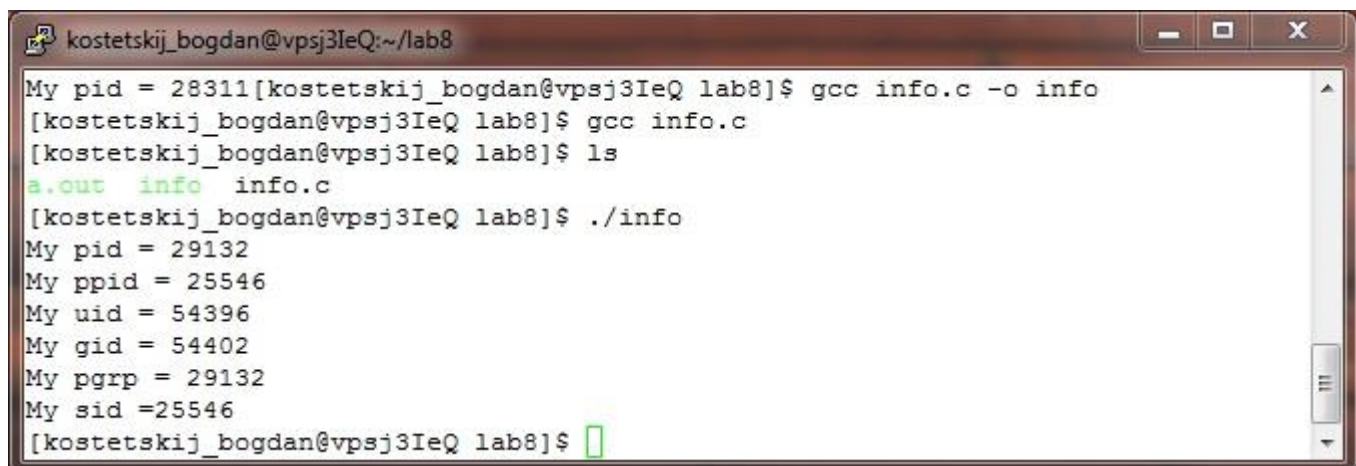
1. Перегляд інформації про процес

Створили С-програму, яка виводить на екран таку інформацію



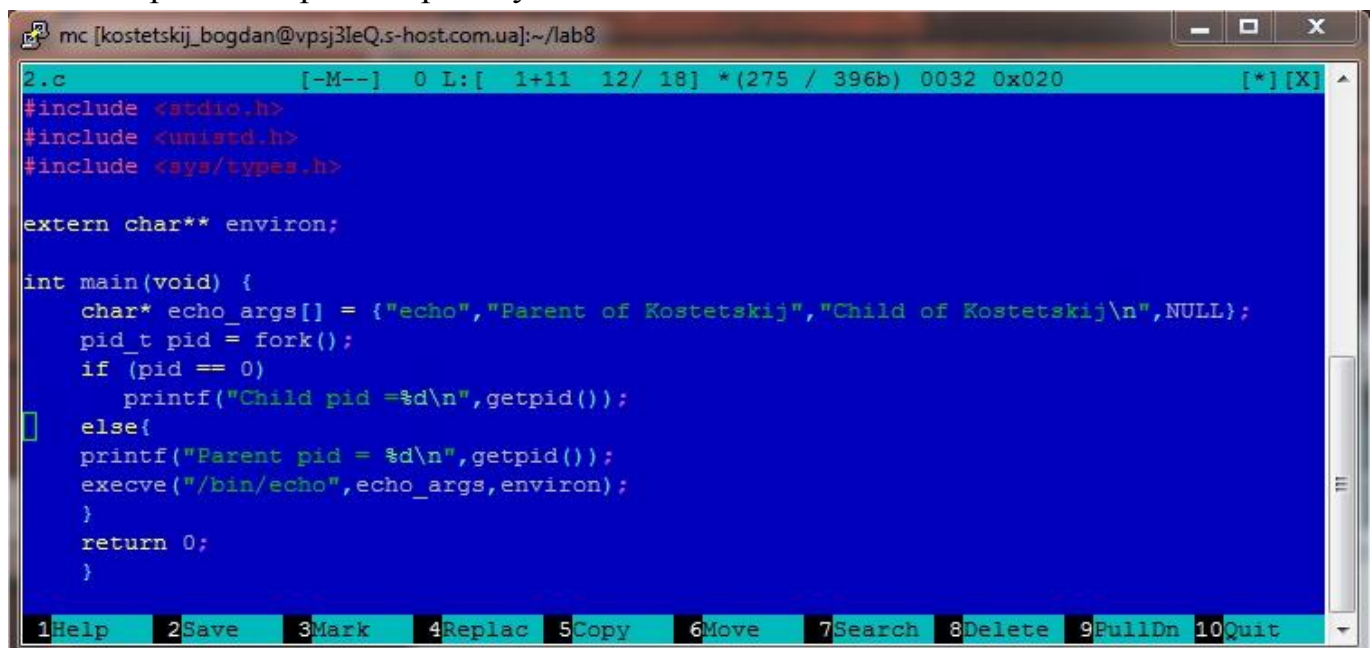
```
mc [kostetskij_bogdan@vpsj3IeQ.s-host.com.ua]:~/lab8
info.c      [----] 37 L:[ 1+ 9 10/ 12] *(288 / 308b) 0010 0x00A  [*][X]
#include <stdio.h>
#include <unistd.h>

int main (void){
    printf("My pid = %d\n",getpid());
    printf("My ppid = %d\n",getppid());
    printf("My uid = %d\n",getuid());
    printf("My gid = %d\n",getgid());
    printf("My pgrp = %d\n",getpgrp());
    printf("My sid =%d\n",getsid(0));
    return 0;
}
```



```
kostetskij_bogdan@vpsj3IeQ:~/lab8
My pid = 28311[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc info.c -o info
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc info.c
[kostetskij_bogdan@vpsj3IeQ lab8]$ ls
a.out  info  info.c
[kostetskij_bogdan@vpsj3IeQ lab8]$ ./info
My pid = 29132
My ppid = 25546
My uid = 54396
My gid = 54402
My pgrp = 29132
My sid =25546
[kostetskij_bogdan@vpsj3IeQ lab8]$
```

2. Стандартне створення процесу



```
mc [kostetskij_bogdan@vpsj3IeQ.s-host.com.ua]:~/lab8
2.c         [-M--]  0 L:[ 1+11 12/ 18] *(275 / 396b) 0032 0x020  [*][X]
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern char** environ;

int main(void) {
    char* echo_args[] = {"echo","Parent of Kostetskij","Child of Kostetskij\n",NULL};
    pid_t pid = fork();
    if (pid == 0)
        printf("Child pid =%d\n",getpid());
    else{
        printf("Parent pid = %d\n",getpid());
        execve("/bin/echo",echo_args,environ);
    }
    return 0;
}
```

```
kostetskij_bogdan@vpsj3IeQ:~/lab8
login as: kostetskij_bogdan
kostetskij_bogdan@91.219.60.189's password:
Last login: Mon Apr 19 11:09:48 2021 from 109.122.7.76
[kostetskij_bogdan@vpsj3IeQ ~]$ pushd lab8
~/lab8 ~
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc 2.c -o 2
[kostetskij_bogdan@vpsj3IeQ lab8]$ ./2
Parent pid = 32665
Child pid =32666
Child of Kostetskij

Child of Kostetskij

[kostetskij_bogdan@vpsj3IeQ lab8]$
```

3.Обмін сигналами між процесами

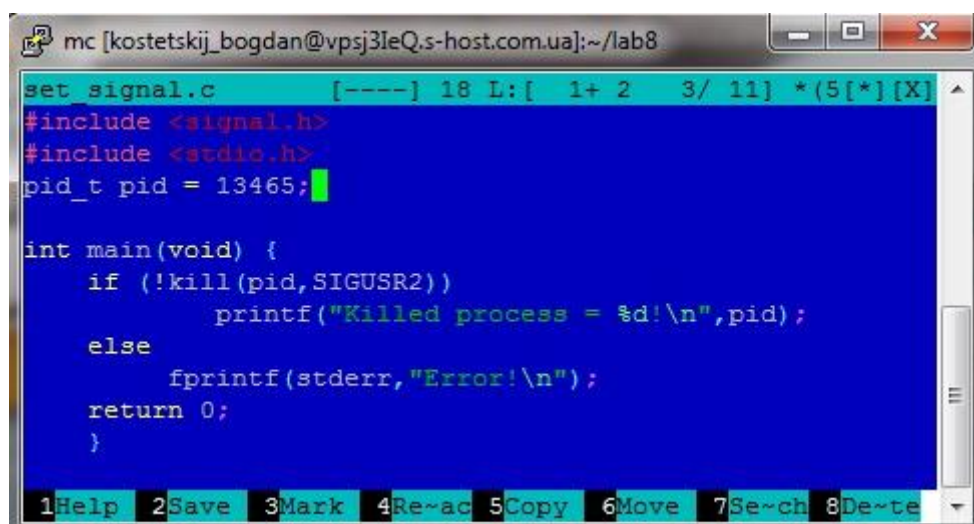
3.1

```
mc [kostetskij_bogdan@vpsj3IeQ.s-host.com.ua]:~/lab8
get_signal.c  [-----]  1 L:[ 1+14 15/ 15] *(299 / 299b) <EOF>  [*][X]
#include <signal.h>
#include <stdio.h>

static void sig_usr(int signo){
    if (signo == SIGUSR2)
        printf("Process of Kostetskij got signal!\n");
}

int main (void)
{
    if (signal(SIGUSR2,sig_usr) == SIG_ERR)
        fprintf(stderr,"Error!\n");
    for( ; ; )
        pause();
    return 0;
}
```

```
kostetskij_bogdan@vpsj3IeQ:~/lab8
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc get_signal.c -o get_signal
[kostetskij_bogdan@vpsj3IeQ lab8]$ ./get_signal
```

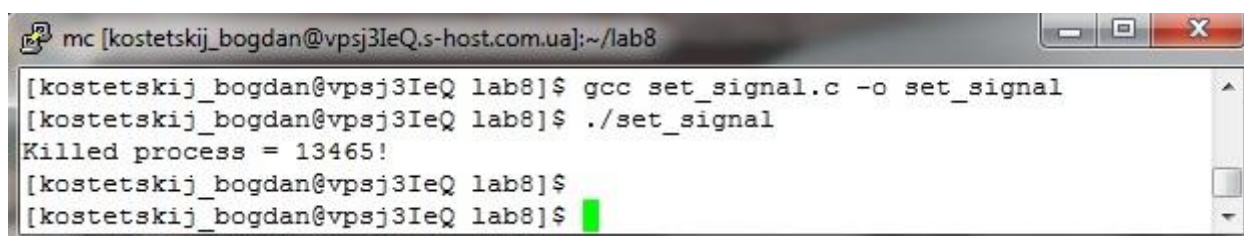


```

set_signal.c  [-----] 18 L:[ 1+ 2 3/ 11] *(5[*][X]
#include <signal.h>
#include <stdio.h>
pid_t pid = 13465;

int main(void) {
    if (!kill(pid, SIGUSR2))
        printf("Killed process = %d!\n", pid);
    else
        fprintf(stderr, "Error!\n");
    return 0;
}
1Help 2Save 3Mark 4Re~ac 5Copy 6Move 7Se~ch 8De~te

```

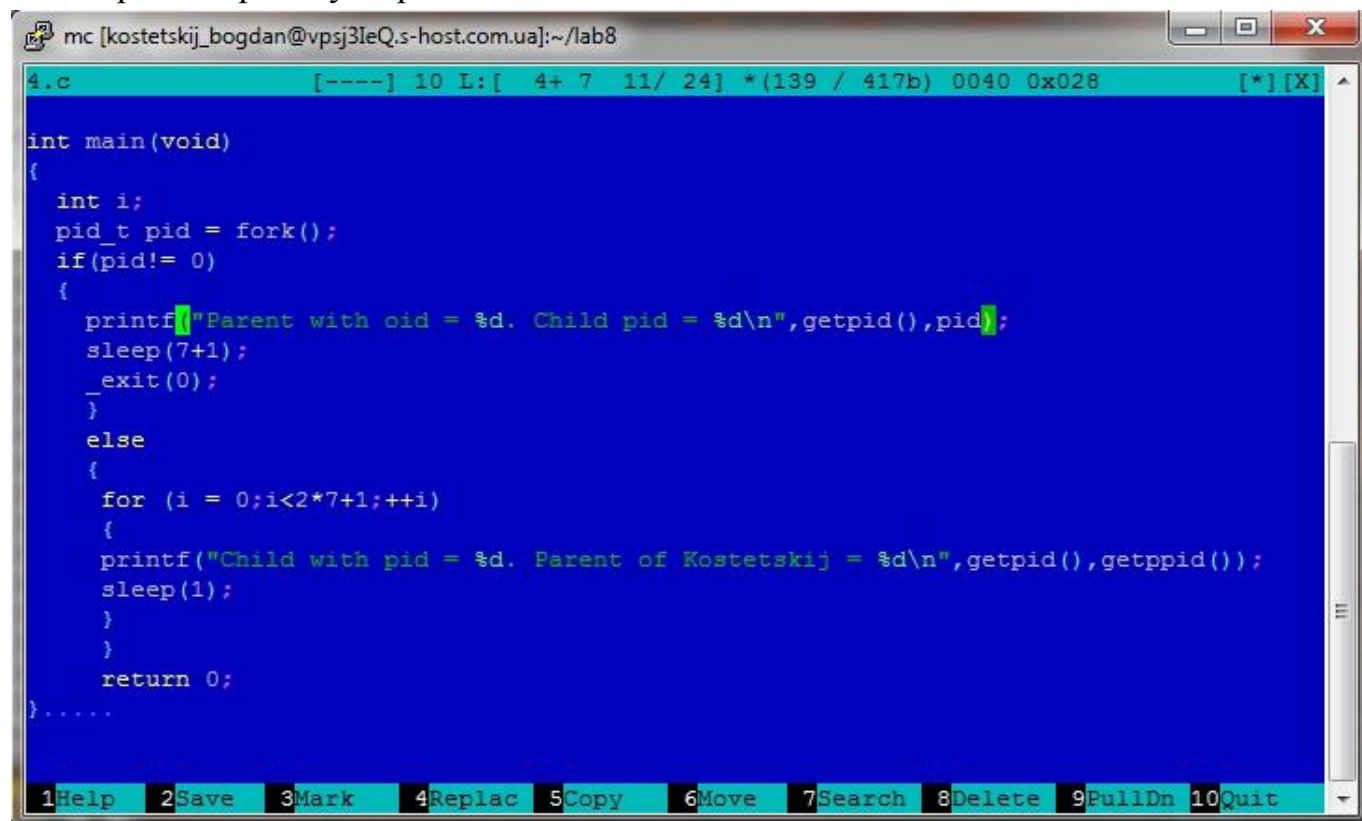


```

mc [kostetskij_bogdan@vpsj3IeQ.s-host.com.ua]:~/lab8
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc set_signal.c -o set_signal
[kostetskij_bogdan@vpsj3IeQ lab8]$ ./set_signal
Killed process = 13465!
[kostetskij_bogdan@vpsj3IeQ lab8]$
[kostetskij_bogdan@vpsj3IeQ lab8]$

```

4. Створення процесу-сироти



```

4.c  [-----] 10 L:[ 4+ 7 11/ 24] *(139 / 417b) 0040 0x028 [*][X]
int main(void)
{
    int i;
    pid_t pid = fork();
    if(pid != 0)
    {
        printf("Parent with oid = %d. Child pid = %d\n", getpid(), pid);
        sleep(7+1);
        _exit(0);
    }
    else
    {
        for (i = 0; i < 2*7+1; ++i)
        {
            printf("Child with pid = %d. Parent of Kostetskij = %d\n", getpid(), getppid());
            sleep(1);
        }
        return 0;
    }
}.....
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

```



```
kostetskij_bogdan@vpsj3IeQ:~/lab8
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc 4.c -o 4
/tmp/cc4IsDNb.o: In function `main':
4.c:(.text+0x2c): undefined reference to `print'
collect2: error: ld returned 1 exit status
[kostetskij_bogdan@vpsj3IeQ lab8]$ gcc 4.c -o 4
[kostetskij_bogdan@vpsj3IeQ lab8]$ ./4
Parent with oid = 20519. Child pid = 20520
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
Child with pid = 20520. Parent of Kostetskij = 20519
[kostetskij_bogdan@vpsj3IeQ lab8]$ Child with pid = 20520. Parent of Koste
```

Висновки: під час виконання лабораторної роботи було отримано навичок в управлінні процесами в ОС Unix на рівні мови програмування C.