МІНІСТЕРСТВО ОСВІТИ Й НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ
Інститут комп’ютерних систем
Кафедра інформаційних систем

Лабораторна робота № 8
За дисциплиною: ”Операційні системи”
Тема:
«Програмування керуванням процесами в ОС Unix»

Виконала:
Студентка групи АІ-205
Алєксєєва А. О.
Перевірили:
Блажко О.А.
Дрозд М.О.

Одеса 2021

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

Завдання

Завдання 1 Перегляд інформації про процес

Створіть С-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.

```
zavd.c            [----]  0 L:[  1+ 0   1/ 12] *(0   / 288b) 0035 0x023  [*][X
#include <stdio.h>
#include <unistd.h>
int main(void) {
pid_t pid = getpid();
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(pid));
return 0;
}
```

```
[alekseeva_alina@vpsj3IeQ ~]$ gcc -o zavd2 zavd.c
[alekseeva_alina@vpsj3IeQ ~]$ ./zavd2
My pid=19928
My ppid=18394
My uid=54424
My gid=54430
My pgrp=19928
My sid=18394
```

Завдання 2 Стандартне створення процесу

Створіть С-програму, яка створює процес-нащадок, породжуючи процес та замінюючи образ процесу. У програмі:

1) процес-батько повинен видати повідомлення типу «Parent of Ivanov», використовуючи стандартний потік виводу, або помилок;

2) процес-нащадок повинен видати повідомлення типу «Child of Ivanov», але вже через виклик команди echo з функції execve, де замість слова Ivanov в повідомленні повинно бути ваше прізвище в транслітерації.

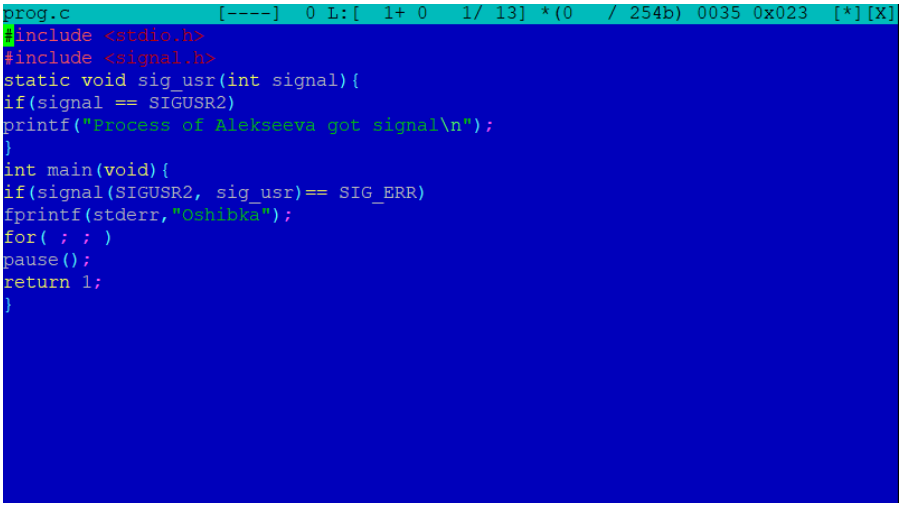
```
inform1.c          [----]  0 L:[  1+ 0   1/ 16] *(0   /
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
extern char** environ;
int main(void)
 {
 char* echo_args[] = {"echo","I am Echo\n",NULL};
 pid_t pid = fork ();
 if(pid == 0)
 printf("Child of Alekseeva: pid=%d\n",getpid());
 else{
 printf("Parent of Alekseeva: pid=%d\n",getpid());
 execve("/bin/echo",echo_args,environ);
 }
 return 0;
 }
```

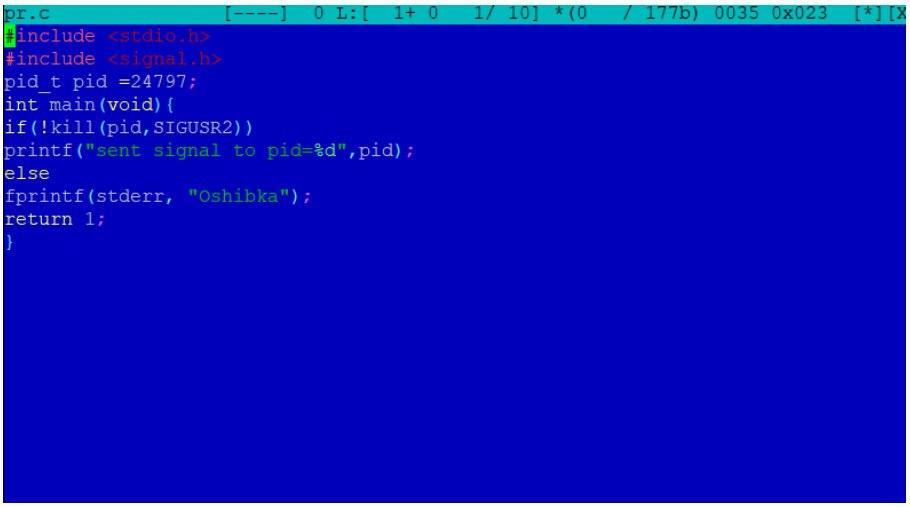
```
[alekseeva_alina@vpsj3IeQ ~]$ gcc -o inform2 inform1.c
[alekseeva_alina@vpsj3IeQ ~]$ ./inform2
Parent of Alekseeva: pid=22187
Child of Alekseeva: pid=22188
I am Echo
```

Завдання 3 Обмін сигналами між процесами

3.1 Створіть С-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 С-програму.

3.2 Створіть С-програму, яка надсилає сигнал SIGUSR2 процесу, запущеному в попередньому пункту завдання. Запустіть створену С-програму та проаналізуйте повідомлення, які виводить першапрограма. Завершіть процес, запущений в попередньому пункті завдання.





Завдання 4 Створення процесу-сироти

Створіть С-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.

```
sirotka.c          [-M--]  1 L:[  1+20  21/ 21] *(336 / 336b) <EOF>        [*]
#include <stdio.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(),pid);
sleep(10);
_exit(0);
}
else{
for(i=0;i<19;i++){
printf("I am child with pid=%d. My parent id =%d\n",getpid(),getppid());
sleep(1);
}
}
return 0;
}
```

```
[alekseeva_alina@vpsj3IeQ ~]$ gcc -o sirotka2 sirotka.c
[alekseeva_alina@vpsj3IeQ ~]$ ./sirotka2
I am parent with pid=24118. My child pid =24119
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
I am child with pid=24119. My parent id =24118
```

Висновок: в ході цієї лабораторної роботи ми отримали навички в управлінні процесами в ОС Unix на рівні мови програмування С.