

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

Звіт
Лабораторної роботи № 8
З предмету «Операційні системи»

Тема: «Програмування керуванням процесами в ОС Unix»

Виконала:
Студентка групи AI-205
Шаповалова Вікторія

Перевірили:
Блажко А. А.
Дрозд М. А.

Одеса 2021

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

Вимоги до оформлення протоколу виконання лабораторної роботи

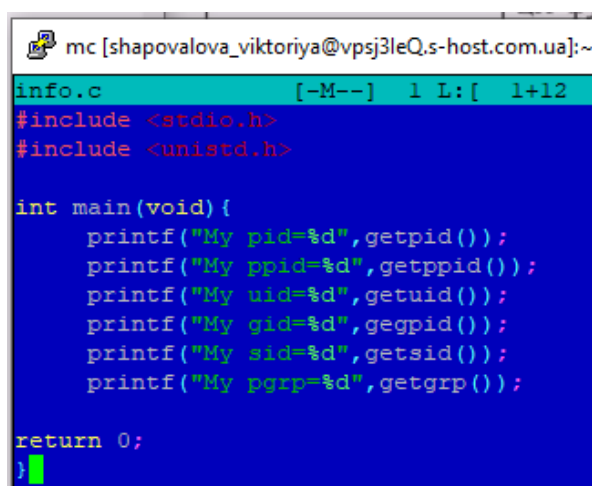
Протокол у електронному вигляді повинен мати наступну структуру

- 1) титульний аркуш з назвою дисципліни, теми лабораторної роботи, групи та ПІБ виконавця;
- 2) аркуш із завданням до лабораторної роботи;
- 3) аркуші з результатами виконання пунктів завдань:
 - пункт із завданням;
 - скріншот частини екрану з командами (з кольорами тексту білий фон/чорний тон);
 - скріншот частини екрану з результатом виконання команди;
- 4) аркуш з висновками:
 - перелік завдань, які були найскладнішими.

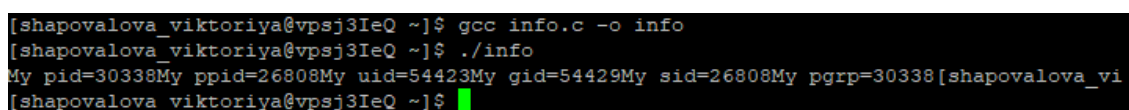
Завдання 1. Перегляд інформації про процес

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.



```
mc [shapovalova_viktoriya@vpsj3leQ.s-host.com.ua]:~  
info.c      [-M--]  1 L:[  1+12  
#include <stdio.h>  
#include <unistd.h>  
  
int main(void) {  
    printf("My pid=%d",getpid());  
    printf("My ppid=%d",getppid());  
    printf("My uid=%d",getuid());  
    printf("My gid=%d",getgid());  
    printf("My sid=%d",getsid());  
    printf("My pgrp=%d",getpgrp());  
  
    return 0;  
}
```



```
[shapovalova_viktoriya@vpsj3leQ ~]$ gcc info.c -o info  
[shapovalova_viktoriya@vpsj3leQ ~]$ ./info  
My pid=30338My ppid=26808My uid=54423My gid=54429My sid=26808My pgrp=30338[shapovalova_v  
[shapovalova_viktoriya@vpsj3leQ ~]$
```

Завдання 2. Стандартне створення процесу

Створіть С-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.

```
mc [shapovalova_viktoriya@vpsj3IeQ.s-host.com.ua]:~
2.c      [----] 32 L:[ 1+ 8  9/ 23] *(169 / 618b) 0032 0x020
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern char ** environ;

int main(void)
{
    pid_t pid = fork();
    char* echo_argsl[] = {"echo", "Child of Shapovalova", NULL};
    if (pid == 0)
    {
        printf ("«I am child: pid =%d\n", getpid());
    }
    else
    {
        printf ("«I am parent: pid = %d\n", getpid());
        execve("/bin/echo", echo_argsl, environ);
        fprintf(stderr, "Error!");
        return 1;
    }
}

return 0;
}
```

```
[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 2.c -o 2
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./2
«I am parent: pid = 28329
«I am child: pid =28330
Child of Shapovalova
[shapovalova_viktoriya@vpsj3IeQ ~]$
```

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.

```
mc [shapovalova_viktoriya@vpsj3IeQ.s-host.com.ua]:~
3.c      [-----] 1 L:[ 1+11 12/ 12] *(312 / 312b) <EOF>
#include <signal.h>
#include <stdio.h>
static void sig_usr(int signo){
    if (signo == SIGUSR1)
        printf("Process of Shapovalova got signal\n ");
    }
int main(void){
    if(signal(SIGUSR1,sig_usr) == SIG_ERR)
        fprintf(stderr, "Error!");
    for ( ; ; )
        pause();
}
```

```
[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 3.c -o 3
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./3
```

```
mc [shapovalova_viktoriya@vpsj3IeQ.s-host.com.ua]:~
[sh] shapovalova_viktoriya@vpsj3IeQ:~
3.c login as: shapovalova_viktoriya
#i shapovalova_viktoriya@91.219.60.189's password:
com Last login: Sat May 22 19:21:02 2021 from 213.231.10.137.pool.breezein.net
[sh] shapovalova_viktoriya@vpsj3IeQ ~]$ ps -u shapovalova_viktoriya -o pid,stat,cmd
    PID STAT CMD
[sh] 4928 S    sshd: shapovalova_viktoriya@pts/1
    " 4929 Ss   -bash
143 5258 T    /usr/bin/mc -P /tmp/mc-shapovalova_viktoriya/mc.pwd.4929
145 5260 Ss+  bash -rcfile .bashrc
196 5299 T    /bin/sh /tmp/mc-shapovalova_viktoriya/mcextOCHM30
1.s 5300 T    vi /home/shapovalova_viktoriya/3.c
2 5336 S+   /usr/bin/mc -P /tmp/mc-shapovalova_viktoriya/mc.pwd.4929
214 5338 Ss   bash -rcfile .bashrc
290 5824 S+   ./3
2.c 6028 S    sshd: shapovalova_viktoriya@pts/5
    " 6029 Ss   -bash
-ba 6150 R+   ps -u shapovalova_viktoriya -o pid,stat,cmd
13340 S    ping localhost
[sh] 13484 S    ping localhost
3.c 15305 RN   sh shap.sh
#i 17161 R    sh shap2.sh
com 17194 R    sh shap3.sh
[sh] shapovalova_viktoriya@vpsj3IeQ ~]$
```

3.2 Створіть С-програму, яка надсилає сигнал SIGUSR2 процесу, запущеному в попередньому пункту завдання.

Запустіть створену С-програму та проаналізуйте повідомлення, які виводить перша програма.

```
mc [shapovalova_viktoriya@vpsj3leQ.s-host.com.ua]:~  
4.c [-M--] 11 L:[ 1+ 7 8/ 14] *(88  
#include <signal.h>  
#include <stdio.h>  
  
pid_t pid = 18329;  
  
int main(void)  
{  
    if(!kill(pid, SIGUSR1))  
        printf("Send signal to pid =%d !", pid);  
    else  
        fprintf(stderr, "Error!\n");  
    return 0;  
}
```

```
Error!  
[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 4.c -o 4  
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./4  
Send signal to pid =18329 !  
[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 4.c -o 4  
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./4  
Send signal to pid =18329 !  
[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 4.c -o 4  
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./4  
Send signal to pid =18329 !  
[shapovalova_viktoriya@vpsj3IeQ ~]$
```

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.
 $n=2+1=3$;

```
mc [shapovalova_viktoriya@vpsj3leQ.s-host.com.ua]:~  
41.c [----] 54 L:[ 1+15 16/ 19] *(362 / 412b) 0034 0x022  
#include <stdio.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
int main(void)  
{ int i=0;  
    pid_t pid = fork();  
    if (pid!=0)  
    {  
        printf("I am parent pid=%d\n", getpid());  
        sleep(12);  
        _exit(0);  
    }  
    else{  
        for(i =0; i<21; i++)  
            {printf("Parent of Shapovalova ppid=%d\n", getppid());  
              sleep(1);}}  
    return 0;}
```

```

[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 4l.c -o 4l
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./4l
I am parent pid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
Parent of Shapovalova ppid=22542
[shapovalova_viktoriya@vpsj3IeQ ~]$ Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1
Parent of Shapovalova ppid=1

```

Завдання 5. Створення процесу-зомбі

Створіть С-програму, в якій процес-нащадок несподівано завершується раніше процесу-батька, перетворюється на зомбі, виводячи в результаті повідомлення, наприклад, «I am Zombie-process of Ivanov», за шаблоном як в попередньому завданні.

Запустіть програму у фоновому режимі, а в окремому терміналі вивчіть вміст таблиці процесів і зробіть відповідні висновки.

```

mc [shapovalova_viktoriya@vpsj3IeQ.s-host.com.ua]:~
5.c [-----] 14 L: [ 1+ 2 3/ 23] * (56 / 399b) 011
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>

int main(void)
{
    int i;
    pid_t pid = fork();
    if (pid == 0)
    {
        sleep(12);
        printf("I am zombie-process of Shapovalova\n");
        _exit(0);
    }
    else
    {
        for(i=0; i<21; i++)
        {
            printf("I am parent\n");
            sleep(1);
        }
    }
    return 0;
}

```

```

[shapovalova_viktoriya@vpsj3IeQ ~]$ gcc 5.c -o 5
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./5
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am zombie-process of Shapovalova
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
I am parent
[shapovalova_viktoriya@vpsj3IeQ ~]$

```

```

[shapovalova_viktoriya@vpsj3IeQ ~]$ ps -u shapovalova_viktoriya -o pid,stat,cmd
  PID STAT CMD
 13340 S   ping localhost
 13484 S   ping localhost
 15305 RN  sh shap.sh
 17161 R   sh shap2.sh
 17194 R   sh shap3.sh
 23775 S   sshd: shapovalova_viktoriya@pts/1
 23777 Ss  -bash
 23806 S+  /usr/bin/mc -P /tmp/mc-shapovalova_viktoriya/mc.pwd.23777
 23809 Ss+  bash -rcfile .bashrc
 25123 S   sshd: shapovalova_viktoriya@pts/10
 25124 Ss  -bash
 26146 R+  ps -u shapovalova_viktoriya -o pid,stat,cmd
[shapovalova_viktoriya@vpsj3IeQ ~]$

```

Завдання 6. Попередження створення процесу-зомбі

Створіть С-програму, в якій процес-нащадок завершується раніше процесу-батька, але ця подія контролюється процесом-батьком.

Процес-нащадок повинен виводити повідомлення, наприклад, «Child of Ivanov is finished», за шаблоном як в попередньому завданні.

Процес-батько повинен очікувати ($3 \cdot n$) секунд.

Значення n - n – номер команди студента + номер студента в команді.

Запустіть програму у фоновому режимі, а в окремому терміналі вивчіть вміст таблиці процесів.

```
mc [shapovalova_viktoriya@vpsj3IeQ.s-host.com.ua]:~  
6.c [----] 1 L:[ 1+26 27/ 27] *(408 / 408b)  
#include <stdio.h>  
#include <sys/types.h>  
#include <unistd.h>  
#include <sys/wait.h>  
  
void sighandler( int sig)  
{  
    printf("Signal handler for signal=%d\n", sig);  
    wait(0);  
}  
  
int main(void)  
{  
    int i;  
    signal(SIGCHLD, &sighandler);  
    pid_t pid=fork();  
    if(pid==0)  
    {  
        sleep(12);  
        printf("Child of Shapovalova is finished\n");  
        _exit(0);  
    }  
    else{  
        printf(" I am parent\n");  
        sleep(30);  
    }  
    return 0;  
}
```

```
[shapovalova_viktoriya@vpsj3IeQ ~]$ ./6  
 I am parent  
Child of Shapovalova is finished  
Signal handler for signal=17  
[shapovalova_viktoriya@vpsj3IeQ ~]$
```

```
Signal handler for signal=17  
[shapovalova_viktoriya@vpsj3IeQ ~]$ ps -u shapovalova_viktoriya -o pid,stat,cmd  
  PID STAT  CMD  
13340 S      ping localhost  
13484 S      ping localhost  
15305 RN     sh shap.sh  
17161 R      sh shap2.sh  
17194 R      sh shap3.sh  
25123 S      sshd: shapovalova_viktoriya@pts/10  
25124 Ss     -bash  
28905 S      sshd: shapovalova_viktoriya@pts/1  
28906 Ss     -bash  
28975 S+     /usr/bin/mc -P /tmp/mc-shapovalova_viktoriya/mc.pwd.28906  
28977 Ss+    bash -rcfile .bashrc  
31134 R+     ps -u shapovalova_viktoriya -o pid,stat,cmd  
[shapovalova_viktoriya@vpsj3IeQ ~]$
```

Висновок: в ході роботи ми придбали навичок в управлінні процесами в ОС Unix на рівні мови програмування C. Найскладнішими були останні завдання.