

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 керування процесами в ОС Unix».

Виконала:
студентка групи AI-204
Томчук Вікторія Борисівна

Перевірив:
Блажко О.А.

Одеса-2021

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

2. Завдання

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.

Завдання 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.

Завдання 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 пункті завдання. Запустіть створену C-програму та проаналізуйте повідомлення, які виводить перша програма. Завершіть процес, запущеному в попередньому пункті завдання.

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.

Завдання 5 Створення процесу-зомбі

Створіть C-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.

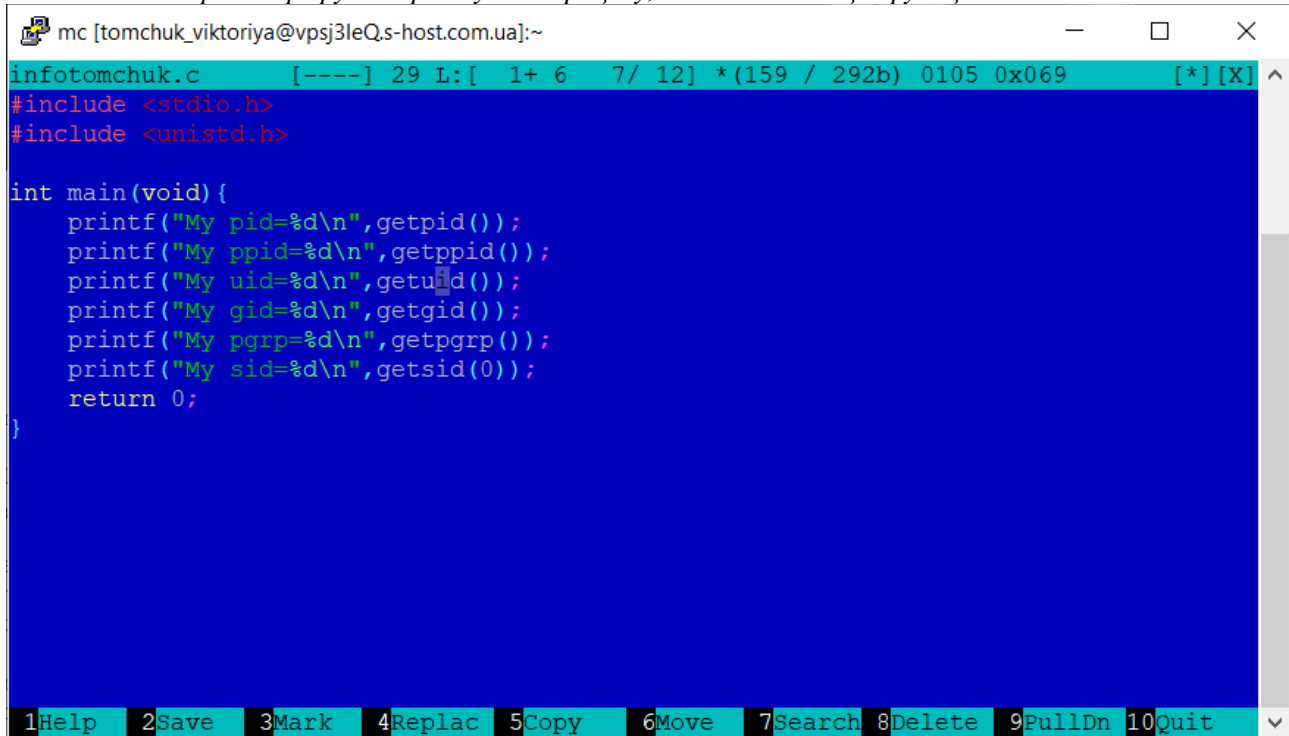
Завдання 6 Попередження створення процесу-зомбі

Створіть C-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 $(3*n)$ секунд. Значення n - n – номер команди студента + номер студента в команді. Запустіть програму у фоновому режимі, а в окремому терміналі вивчіть вміст таблиці процесів і зробіть відповідні висновки.

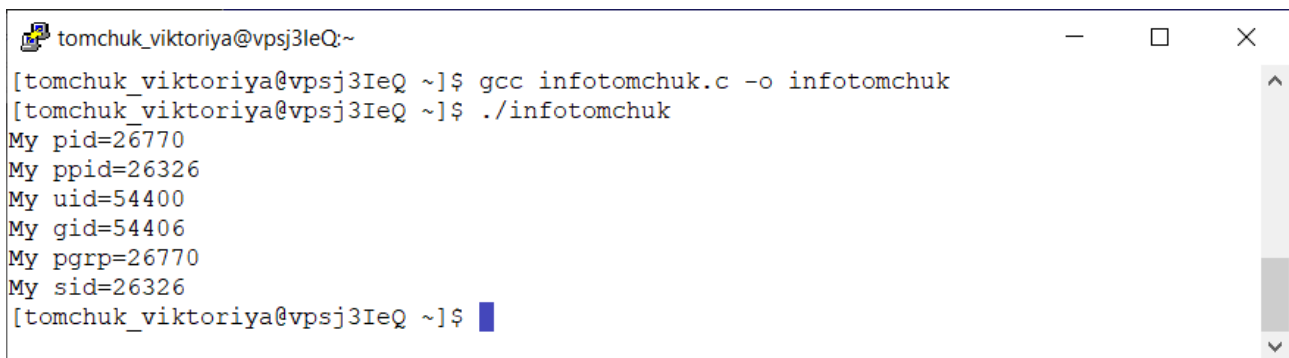
Завдання 1. Перегляд інформації про процес

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.

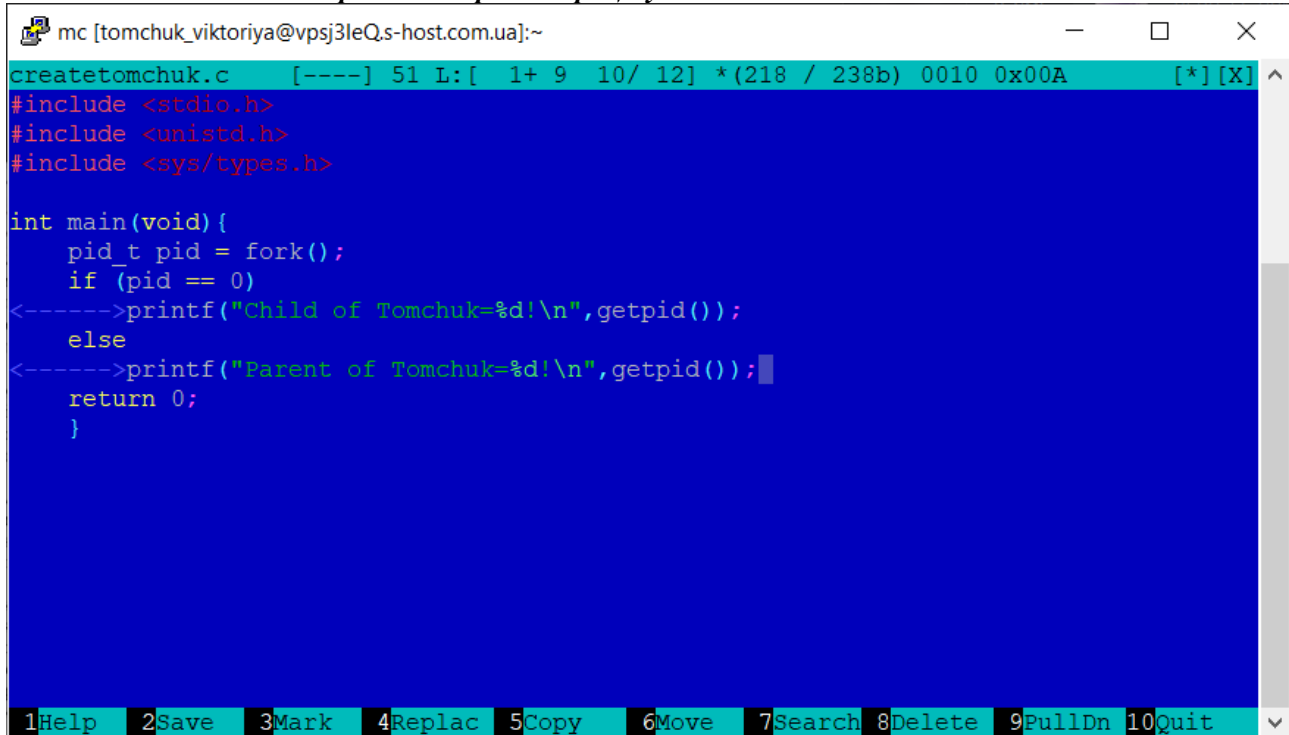


```
mc [tomchuk_viktoriya@vpsj3leQ.s-host.com.ua]:~  
infotomchuk.c  [----] 29 L:[ 1+ 6 7/ 12] *(159 / 292b) 0105 0x069 [*][X] ^  
#include <stdio.h>  
#include <unistd.h>  
  
int main(void) {  
    printf("My pid=%d\n", getpid());  
    printf("My ppid=%d\n", getppid());  
    printf("My uid=%d\n", getuid());  
    printf("My gid=%d\n", getgid());  
    printf("My pgrp=%d\n", getpgrp());  
    printf("My sid=%d\n", getsid(0));  
    return 0;  
}
```



```
tomchuk_viktoriya@vpsj3leQ:~  
[tomchuk_viktoriya@vpsj3leQ ~]$ gcc infotomchuk.c -o infotomchuk  
[tomchuk_viktoriya@vpsj3leQ ~]$ ./infotomchuk  
My pid=26770  
My ppid=26326  
My uid=54400  
My gid=54406  
My pgrp=26770  
My sid=26326  
[tomchuk_viktoriya@vpsj3leQ ~]$
```

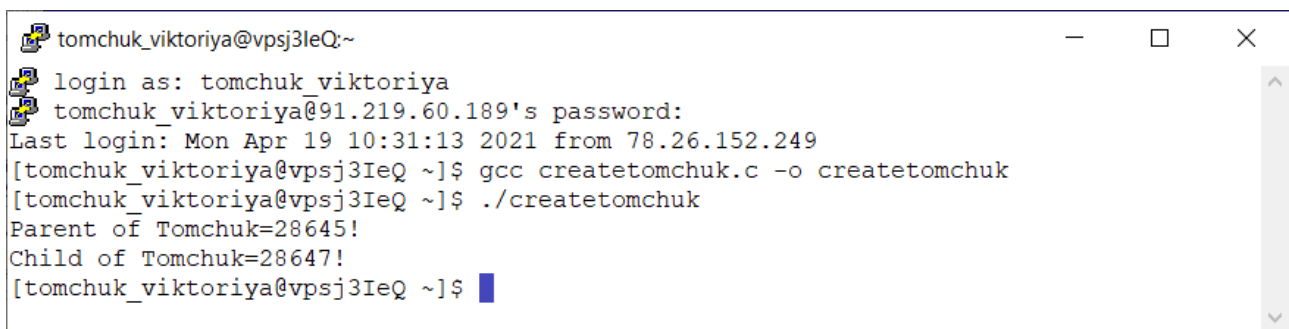
Завдання 2. Стандартне створення процесу



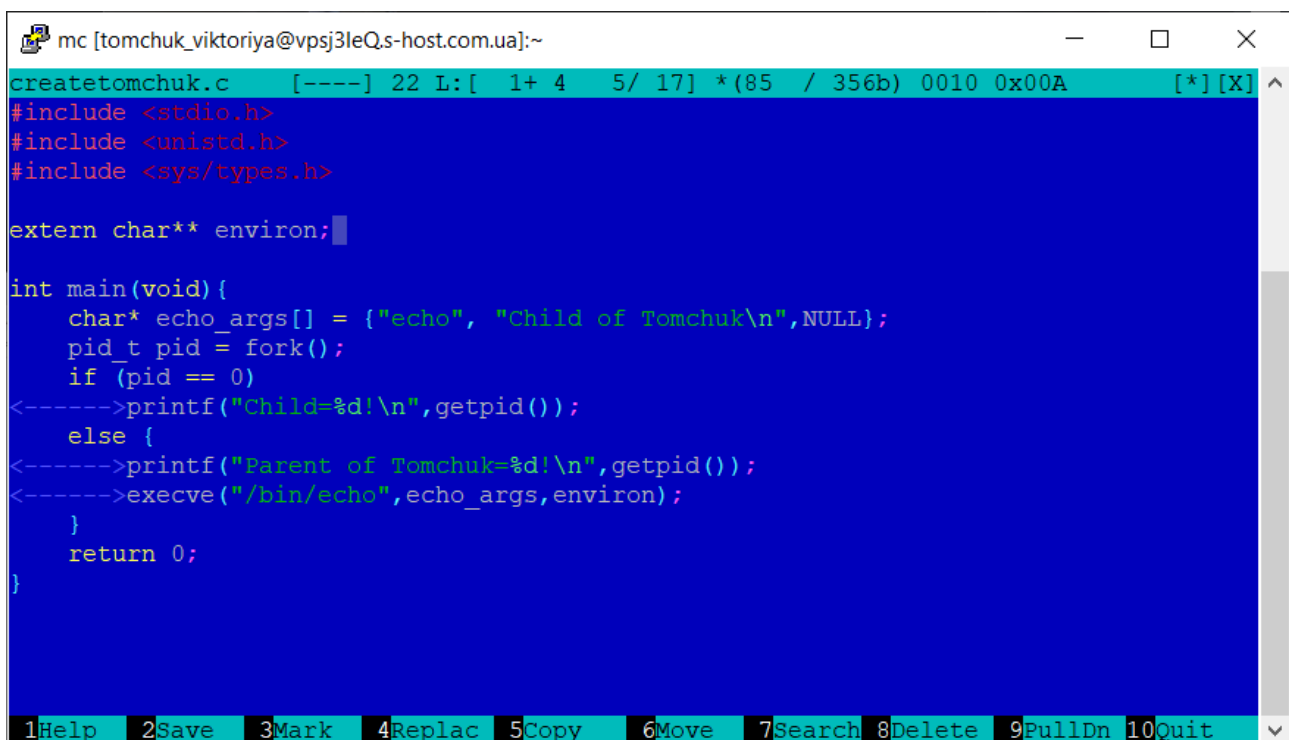
```
mc [tomchuk_viktoriya@vpsj3leQ.s-host.com.ua]:~
createtomchuk.c [----] 51 L:[ 1+ 9 10/ 12] *(218 / 238b) 0010 0x00A [*][X] ^
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

int main(void){
    pid_t pid = fork();
    if (pid == 0)
<----->printf("Child of Tomchuk=%d!\n",getpid());
    else
<----->printf("Parent of Tomchuk=%d!\n",getpid());
    return 0;
}
```

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit



```
tomchuk_viktoriya@vpsj3leQ:~
login as: tomchuk_viktoriya
tomchuk_viktoriya@91.219.60.189's password:
Last login: Mon Apr 19 10:31:13 2021 from 78.26.152.249
[tomchuk_viktoriya@vpsj3leQ ~]$ gcc createtomchuk.c -o createtomchuk
[tomchuk_viktoriya@vpsj3leQ ~]$ ./createtomchuk
Parent of Tomchuk=28645!
Child of Tomchuk=28647!
[tomchuk_viktoriya@vpsj3leQ ~]$
```



```
mc [tomchuk_viktoriya@vpsj3leQ.s-host.com.ua]:~
createtomchuk.c [----] 22 L:[ 1+ 4 5/ 17] *(85 / 356b) 0010 0x00A [*][X] ^
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern char** environ;

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

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit

```
tomchuk_viktoriya@vpsj3IeQ:~  
login as: tomchuk_viktoriya  
tomchuk_viktoriya@91.219.60.189's password:  
Last login: Mon Apr 19 10:44:37 2021 from 78.26.152.249  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc createtomchuk.c -o createtomchuk  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ./createtomchuk  
Parent of Tomchuk=30305!  
Child=30306!  
Child of Tomchuk  
  
[tomchuk_viktoriya@vpsj3IeQ ~]$
```

Завдання 3. Обмін сигналами між процесами

3.1.

```
mc [tomchuk_viktoriya@vpsj3IeQs-host.com.ua]:~  
signaltomchuk.c [----] 24 L:[ 1+ 4 5/ 13] *(97 / 268b) 0041 0x029 [*][X]  
#include <signal.h>  
#include <stdio.h>  
  
static void sig_usr(int signo) {  
    if (signo == SIGUSR2)  
<----->printf("Process of Tomchuk got signal!\n");  
}  
  
int main(void) {  
    if (signal(SIGUSR2, sig_usr) == SIG_ERR)  
<----->    fprintf(stderr, "Error!\n");  
    for ( ; ; )  
<----->pause();  
}
```

```
tomchuk_viktoriya@vpsj3IeQ:~  
login as: tomchuk_viktoriya  
tomchuk_viktoriya@91.219.60.189's password:  
Last login: Mon Apr 19 10:57:39 2021 from 78.26.152.249  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc signaltomchuk.c -o signaltomchuk  
signaltomchuk.c: In function 'main':  
signaltomchuk.c:13:2: error: expected declaration or statement at end of input  
    pause();  
    ^  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc signaltomchuk.c -o signaltomchuk  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ./signaltomchuk
```

```
tomchuk_viktoriya@vpsj3IeQ:~  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ps -u tomchuk_viktoriya -o pid,stat,cmd  
PID STAT CMD  
625 S+ ./signaltomchuk  
648 S sshd: tomchuk_viktoriya@pts/55  
649 Ss -bash  
814 R+ ps -u tomchuk_viktoriya -o pid,stat,cmd  
23214 S sshd: tomchuk_viktoriya@pts/14  
23215 Ss -bash  
23502 S+ /usr/bin/mc -P /tmp/mc-tomchuk_viktoriya/mc.pwd.23215  
23504 Ss+ bash -rcfile .bashrc  
24443 S sshd: tomchuk_viktoriya@pts/42  
24444 Ss+ -bash  
25410 S sshd: tomchuk_viktoriya@pts/49  
25411 Ss+ -bash  
25751 S sshd: tomchuk_viktoriya@pts/52  
25752 Ss -bash  
25779 T /usr/bin/mc -P /tmp/mc-tomchuk_viktoriya/mc.pwd.25752 infotomchuk.c  
25781 Ss+ bash -rcfile .bashrc  
25818 T /bin/sh /tmp/mc-tomchuk_viktoriya/mcextOEZQ10  
25819 T vi /home/tomchuk_viktoriya/infotomchuk.c  
26325 S sshd: tomchuk_viktoriya@pts/56  
26326 Ss+ -bash  
26849 S+ /usr/bin/mc -P /tmp/mc-tomchuk_viktoriya/mc.pwd.25752  
26851 Ss+ bash -rcfile .bashrc  
28570 S sshd: tomchuk_viktoriya@pts/70  
28571 Ss+ -bash  
30258 S sshd: tomchuk_viktoriya@pts/79  
30259 Ss+ -bash  
32324 S sshd: tomchuk_viktoriya@pts/89  
32333 Ss -bash
```

3.2.

```
mc [tomchuk_viktoriya@vpsj3IeQs-host.com.ua]:~  
killsignal.c [----] 1 L:[ 1+12 13/ 14] *(187 / 188b) 0010 0x00A [*][X]  
#include <signal.h>  
#include <stdio.h>  
  
pid_t pid = 1228;  
  
int main(void) {  
<----->if(!kill(pid.SIGUSR2))  
<-----><----->printf("Got signal of Tomchuk\n");  
<----->else  
<-----><----->fprintf(stderr,"Error!\n");  
<----->return 0;  
<----->  
}  
1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn 10Quit
```

```
tomchuk_viktoriya@vpsj3IeQ:~  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc killsignal.c -o killsignal  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ./killsignal  
Got signal of Tomchuk  
[tomchuk_viktoriya@vpsj3IeQ ~]$
```

```
tomchuk_viktoriya@vpsj3IeQ:~  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc signaltomchuk.c -o signaltomchuk  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ./signaltomchuk  
Process of Tomchuk got signal!  
Terminated
```

Завдання 4. Створення процесу-сироти

```
mc [tomchuk_viktoriya@vpsj3IeQ.s-host.com.ua]:~  
sirotatomchuk.c [----] 20 L:[ 1+11 12/ 21] *(291 / 392b) 0041 0x029 [*][X] ^  
clude <stdio.h>  
clude <unistd.h>  
clude <sys/types.h>  
  
main(void) {  
    int i;  
    pid_t pid = fork();  
    if (pid == 0) {  
        printf("Tomchuk child pid=%d\n",getpid());  
        for (i=0;i<23;i++) {  
            printf("Child of Tomchuk pid=%d\n, My parent=%d\n",getpid(),getppid());  
            sleep(10);  
        }  
    }  
    else {  
        printf("Parent of Tomchuk=%d\n",getpid());  
        sleep(1);  
        _exit(0);  
    }  
    return 0;  
}
```

```
tomchuk_viktoriya@vpsj3IeQ:~  
[tomchuk_viktoriya@vpsj3IeQ ~]$ gcc sirotatomchuk.c -o sirotatomchuk  
[tomchuk_viktoriya@vpsj3IeQ ~]$ ./sirotatomchuk  
Parent of Tomchuk=5987  
Tomchuk child pid=5988  
Child of Tomchuk pid=5988  
, My parent=5987  
[tomchuk_viktoriya@vpsj3IeQ ~]$ Child of Tomchuk pid=5988  
, My parent=1  
Child of Tomchuk pid=5988  
, My parent=1  
[tomchuk_viktoriya@vpsj3IeQ ~]$
```

Висновок: були отримані навички в управлінні процесами в ОС Unix на рівні мови програмування C.