

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»

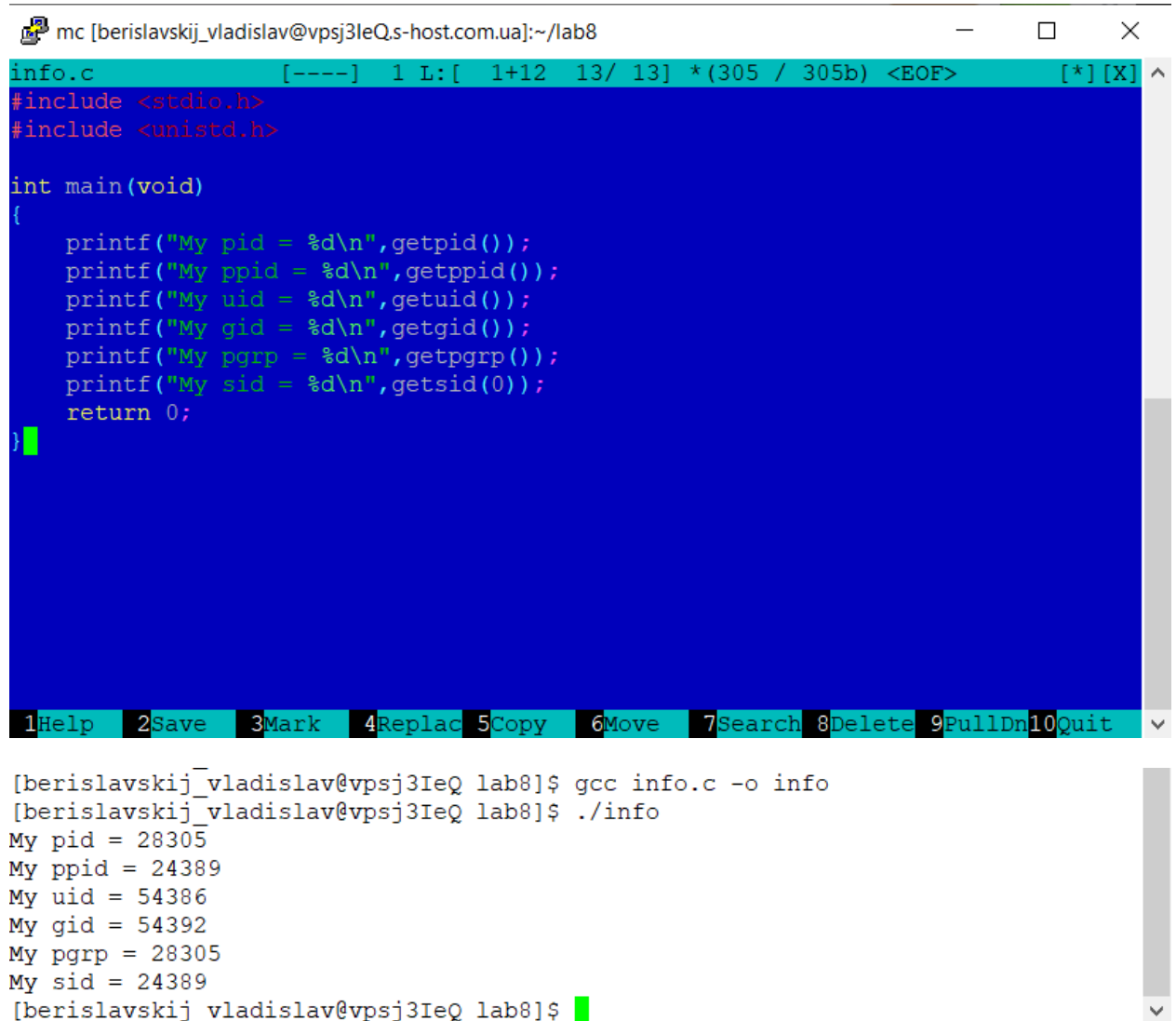
Виконав:
ст. гр. AI-204
Бериславський В.Р

Перевірив:
Блажко О. А.
Дрозд М.О.

Одеса – 2021

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

1



The image shows a code editor window titled 'mc [berislavskij_vladislav@vpsj3IeQ.s-host.com.ua]:~/lab8'. The editor contains a C program named 'info.c' with the following code:

```
info.c [----] 1 L:[ 1+12 13/ 13] *(305 / 305b) <EOF> [*] [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;
}
```

Below the editor is a terminal window showing the compilation and execution of the program:

```
[berislavskij_vladislav@vpsj3IeQ lab8]$ gcc info.c -o info
[berislavskij_vladislav@vpsj3IeQ lab8]$ ./info
My pid = 28305
My ppid = 24389
My uid = 54386
My gid = 54392
My pgrp = 28305
My sid = 24389
[berislavskij_vladislav@vpsj3IeQ lab8]$
```

```

mc [berislavskij_vladislav@vpsj3leQs-host.com.ua]:~/lab8
create.c [-M--] 34 L: [ 1+17 18/ 22] *(370 / 450b) 0044 0x02C [*] [X] ^
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern char** environ;

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

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit v

[berislavskij_vladislav@vpsj3leQ lab8]$ gcc create.c -o create
[berislavskij_vladislav@vpsj3leQ lab8]$ ./create
Parent pid = 4290
Child pid = 4291
Child of Berislavskij

```

The image shows two terminal windows side-by-side. The left window shows a login session for 'berislavskij_vladislav' on 'vpsj3leQ' at 'lab8'. The user enters their password and is prompted for a command. They enter 'gcc get_signal.c -o get_signal', which results in a 'fatal error: no input files' and 'compilation terminated'. The right window shows the same user logging in, then running 'ps -u'. This displays a table of processes for the user 'berisla+'.

```

berislavskij_vladislav@vpsj3leQ:~/lab8
login as: berislavskij_vladislav
berislavskij_vladislav@91.219.60.189's password:
Last login: Mon Apr 19 11:57:57 2021 from 188.163.100.178
[berislavskij_vladislav@vpsj3leQ ~]$ gcc get_signal.c -o get_signal
gcc: error: get_signal.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
[berislavskij_vladislav@vpsj3leQ ~]$ pushd lab8
~/lab8 ~
[berislavskij_vladislav@vpsj3leQ lab8]$ gcc get_signal.c -o get_signal
[berislavskij_vladislav@vpsj3leQ lab8]$ ./get_signal

berislavskij_vladislav@vpsj3leQ:~
login as: berislavskij_vladislav
berislavskij_vladislav@91.219.60.189's password:
Last login: Mon Apr 19 12:03:49 2021 from 188.163.100.178
[berislavskij_vladislav@vpsj3leQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
berisla+  7426  0.0  0.1 115548 2132 pts/13  Ss   12:03   0:00 -bash
berisla+  7597  0.0  0.0  4212    348 pts/13  S+   12:04   0:00 ./get_signal
berisla+  7635  0.0  0.1 115548 2056 pts/33  Ss   12:05   0:00 -bash
berisla+  7671  0.0  0.0 155476 1880 pts/33  R+   12:05   0:00 ps -u
[berislavskij_vladislav@vpsj3leQ ~]$

```

The image shows a code editor window titled 'mc [berislavskij_vladislav@vpsj3leQ.s-host.com.ua]:~/lab8'. It displays the source code for 'set_signal.c'. The code includes headers for signal, stdio, and errno. It defines a pid_t variable 'pid' with the value 7597. The main function calls 'kill(pid, SIGUSR1)' and prints a message if successful, or an error message if it fails. The editor has a status bar at the bottom with menu items: 1Help, 2Save, 3Mark, 4Replac, 5Copy, 6Move, 7Search, 8Delete, 9PullDn, 10Quit.

```

set_signal.c  [----]  1 L:[ 1+13 14/ 14] *(226 / 226b) <EOF>  [*] [X]
#include <signal.h>
#include <stdio.h>
#include <errno.h>

pid_t pid = 7597;

int main(void)
{
    if (!kill(pid, SIGUSR1))
<----->printf("Process with pid = %d killed!\n", pid);
    else.
<----->fprintf(stderr, "Error!");
    return 0;
}

```

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

```
mc [berislavskij_vladislav@vpsj3IeQs-host.com.ua]:~/lab8
get_signal.c [----] 9 L:[ 1+15 16/ 17] *(275 / 277b) 0010 0x00A [*][X] ^
#include <signal.h>
#include <stdio.h>

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

int main (void)
{
    if (signal(SIGUSR1, sig_usr) == SIG_ERR)
<----->fprintf(stderr, "Error!\n");
    for(;;)
<----->pause();
return 0;
}

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit
```

```
berislavskij_vladislav@vpsj3IeQ:~/lab8
login as: berislavskij_vladislav
berislavskij_vladislav@91.219.60.189's password:
Last login: Mon Apr 19 12:03:49 2021 from 188.163.100.178
[berislavskij_vladislav@vpsj3IeQ ~]$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
berisla+  7426  0.0  0.1 115548 2132 pts/13  Ss   12:03   0:00 -bash
berisla+  7597  0.0  0.0  4212    348 pts/13  S+   12:04   0:00 ./get_signal
berisla+  7635  0.0  0.1 115548 2056 pts/33  Ss   12:05   0:00 -bash
berisla+  7671  0.0  0.0 155476 1880 pts/33  R+   12:05   0:00 ps -u
[berislavskij_vladislav@vpsj3IeQ ~]$ mc

[berislavskij_vladislav@vpsj3IeQ lab8]$ gcc set_signal.c -o set_signal
[berislavskij_vladislav@vpsj3IeQ lab8]$ ./set_signal
Process with pid = 7597 killed!
[berislavskij_vladislav@vpsj3IeQ lab8]$
```

```

mc [berislavskij_vladislav@vpsj3IeQ.s-host.com.ua]:~/lab8
sirota.c [----] 15 L: [ 4+17 21/ 25] *(359 / 401b) 0041 0x029 [*] [X] ^

int main(void)
{
    int i;
    pid_t pid = fork();
    if(pid == 0)
    {
<----->printf("Child pid = %d\n",getpid());
<----->for (i=0; i<7; i++)
<----->{
<----->    printf("I am child. My parent id = %d\n", getppid());
<----->    sleep(1);
<----->}
    }
    else
    {
        printf("Parent pid = %d\n",getpid());
        sleep(4);
        _exit(0);
    }
    return 0;
}

1Help 2Save 3Mark 4Replac 5Copy 6Move 7Search 8Delete 9PullDn10Quit v

[berislavskij_vladislav@vpsj3IeQ lab8]$ gcc sirota.c -o sirota
[berislavskij_vladislav@vpsj3IeQ lab8]$ ./sirota
Parent pid = 2538
Child pid = 2539
I am child. My parent id = 2538
I am child. My parent id = 2538
I am child. My parent id = 2538
I am child. My parent id = 2538
[berislavskij_vladislav@vpsj3IeQ lab8]$ I am child. My parent id = 1
I am child. My parent id = 1
I am child. My parent id = 1

```

Висновок : у ході виконання роботи була проведена робота з управління процесів ОС UNIX за допомогою мови С. Найскладнішим було 3 завдання з процесами, так як потребувало деякої уважності та обережності.