

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки і комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Звіт
про виконання лабораторної роботи № 5
“Керування процесами і потоками”

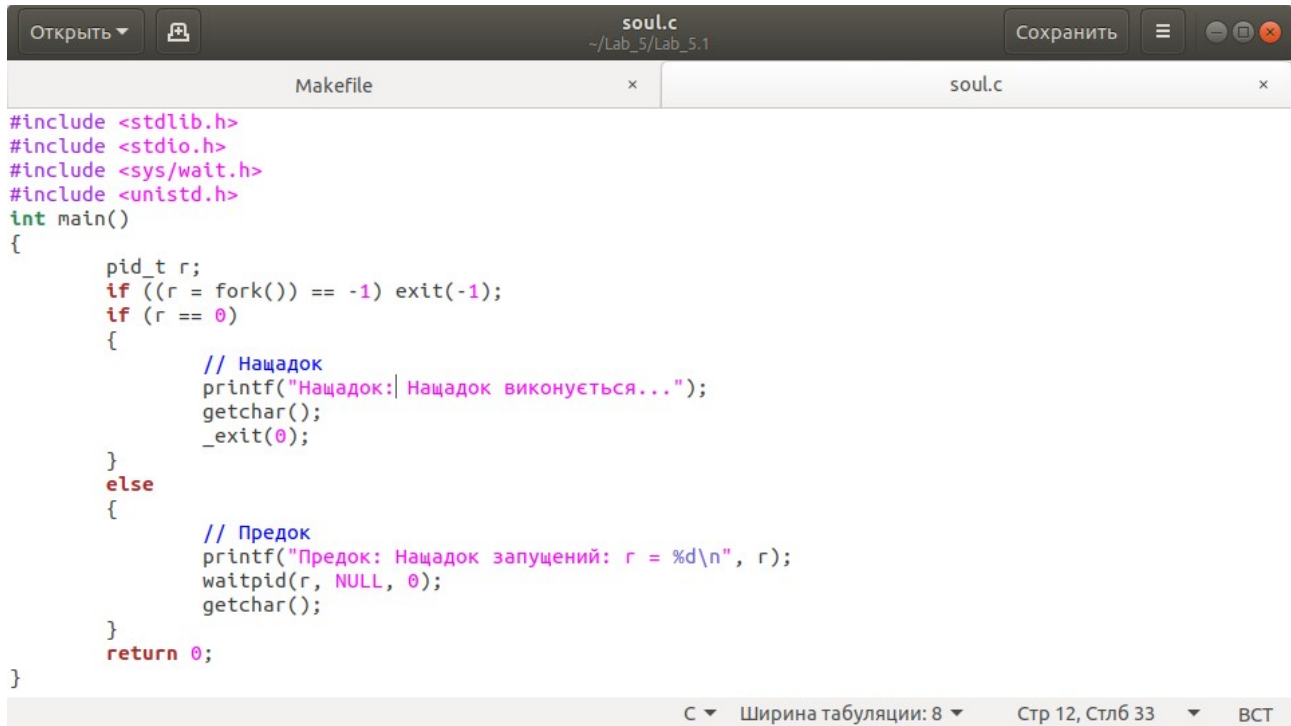
Виконав
студент групи ФЕІ-23
Дребот А. С.
Перевірів
ас. Сінкевич О. О.

Львів - 2019

1. Напишіть функцію, виклик якої приведе до знищення всіх процесів-зомбі, створених поточним процесом.

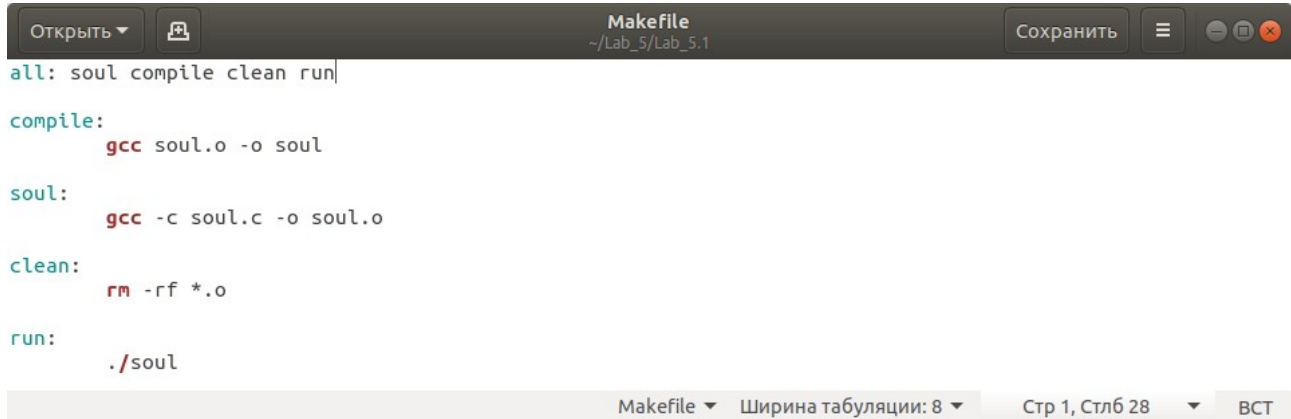
Код:

soul.c



```
#include <stdlib.h>
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
int main()
{
    pid_t r;
    if ((r = fork()) == -1) exit(-1);
    if (r == 0)
    {
        // Нащадок
        printf("Нащадок: Нащадок виконується...");
        getchar();
        _exit(0);
    }
    else
    {
        // Предок
        printf("Предок: Нащадок запущений: r = %d\n", r);
        waitpid(r, NULL, 0);
        getchar();
    }
    return 0;
}
```

Makefile



```
all: soul compile clean run

compile:
    gcc soul.o -o soul

soul:
    gcc -c soul.c -o soul.o

clean:
    rm -rf *.o

run:
    ./soul
```

Тестування:

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.1
Файл Правка Вид Поиск Терминал Справка
andrew@andrew-VirtualBox:~$ cd /home/andrew/Lab_5/Lab_5.1
andrew@andrew-VirtualBox:~/Lab_5/Lab_5.1$ make
gcc -c soul.c -o soul.o
gcc soul.o -o soul
rm -rf *.o
./soul
Предок: Нащадок запущений: r = 17947
Нащадок: Нащадок виконується...
```

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.1
Файл Правка Вид Поиск Терминал Вкладки Справка
andrew@andrew-VirtualBox: ~/Lab_5/Lab... x andrew@andrew-VirtualBox: ~/Lab_5/Lab... x
root      16990      2  0 17:30 ?          00:00:00 [kworker/1:1-eve]
root      17056      2  0 17:30 ?          00:00:00 [kworker/2:1-eve]
andrew    17266    1351  0 17:34 ?          00:00:10 /usr/bin/nautilus --gapplication
root      17296      2  0 17:35 ?          00:00:00 [kworker/1:0-eve]
root      17375      2  0 17:38 ?          00:00:00 [kworker/0:0-rcu]
root      17388      2  0 17:38 ?          00:00:00 [kworker/3:0-eve]
root      17571      2  0 17:43 ?          00:00:00 [kworker/u8:2-ev]
root      17572      2  0 17:43 ?          00:00:00 [kworker/2:0-eve]
root      17735      2  0 17:49 ?          00:00:00 [kworker/u8:0-ev]
root      17863      2  0 17:54 ?          00:00:00 [kworker/2:2-eve]
root      17864      2  0 17:54 ?          00:00:00 [kworker/3:2-cgr]
root      17865      2  0 17:54 ?          00:00:00 [kworker/3:3-eve]
root      17868      2  0 17:54 ?          00:00:00 [kworker/3:4-eve]
root      17875      2  0 17:54 ?          00:00:00 [kworker/1:2-eve]
root      17876      2  0 17:54 ?          00:00:00 [kworker/0:1-eve]
andrew    17902    1351  0 17:54 ?          00:00:01 /usr/lib/gnome-terminal/gnome-te
andrew    17917    17902  0 17:54 pts/0        00:00:00 bash
andrew    17937    17917  0 17:55 pts/0        00:00:00 make
andrew    17946    17937  0 17:55 pts/0        00:00:00 ./soul
andrew    17947    17946  0 17:55 pts/0        00:00:00 ./soul
andrew    18077    17902  0 17:58 pts/1        00:00:00 bash
andrew    18085    1555 22 17:58 tty2        00:00:06 /snap/chromium/937/usr/lib/chrom
andrew    18212    18085  0 17:58 tty2        00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18214    18212  0 17:58 tty2        00:00:00 /snap/chromium/937/usr/lib/chrom
```

Як бачимо наш процес запущено (./soul)

Зупиняємо роботу процесу і знову перевіряємо стан всіх процесів.

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.1
Файл Правка Вид Поиск Терминал Вкладки Справка
andrew@andrew-VirtualBox: ~/Lab_5/Lab... x andrew@andrew-VirtualBox: ~/Lab_5/Lab... x
andrew@andrew-VirtualBox:~$ cd /home/andrew/Lab_5/Lab_5.1
andrew@andrew-VirtualBox:~/Lab_5/Lab_5.1$ make
gcc -c soul.c -o soul.o
gcc soul.o -o soul
rm -rf *.o
./soul
Предок: Нащадок запущений: r = 17947
Нащадок: Нащадок виконується...

andrew@andrew-VirtualBox:~/Lab_5/Lab_5.1$
```

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.1
Файл Правка Вид Поиск Терминал Вкладки Справка
andrew@andrew-VirtualBox: ~/Lab_5/Lab... x andrew@andrew-VirtualBox: ~/Lab_5/Lab... x
root      17388      2  0 17:38 ?      00:00:00 [kworker/3:0-eve]
root      17571      2  0 17:43 ?      00:00:00 [kworker/u8:2-ev]
root      17735      2  0 17:49 ?      00:00:00 [kworker/u8:0-ev]
root      17863      2  0 17:54 ?      00:00:00 [kworker/2:2-eve]
root      17864      2  0 17:54 ?      00:00:00 [kworker/3:2-cgr]
root      17875      2  0 17:54 ?      00:00:00 [kworker/1:2-eve]
root      17876      2  0 17:54 ?      00:00:00 [kworker/0:1-eve]
andrew    17902    1351  0 17:54 ?      00:00:02 /usr/lib/gnome-terminal/gnome-te
andrew    17917    17902  0 17:54 pts/0      00:00:00 bash
andrew    18077    17902  0 17:58 pts/1      00:00:00 bash
andrew    18085    1555  7 17:58 tty2      00:00:06 /snap/chromium/937/usr/lib/chrom
andrew    18212    18085  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18214    18212  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18231    18085  1 17:58 tty2      00:00:01 /snap/chromium/937/usr/lib/chrom
andrew    18237    18085  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18378    18214  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18546    18214  3 17:58 tty2      00:00:02 /snap/chromium/937/usr/lib/chrom
andrew    18557    18214  2 17:58 tty2      00:00:02 /snap/chromium/937/usr/lib/chrom
andrew    18570    18214  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18578    18214  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18588    18214  3 17:58 tty2      00:00:02 /snap/chromium/937/usr/lib/chrom
andrew    18600    18214  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
andrew    18611    18214  0 17:58 tty2      00:00:00 /snap/chromium/937/usr/lib/chrom
root      18654      1  0 17:59 ?      00:00:00 /lib/systemd/systemd-hostnamed
```

Процес – зомбі був видалений.

2. Розробіть простий командний інтерпретатор для Linux і Windows.

Код:

soul.c

```
Открыть  soul.c  Сохранить
~/Lab_5/Lab_5.2

#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
#include <unistd.h>
#include <signal.h>
#include <string.h>

int main()
{
    pid_t pid;
    char command[256];
    struct sigaction act = { 0 };
    sigaction(SIGINT, &act, 0);
    while (1)
    {
        printf(">");
        gets(command);
        char* argv[20];
        int beakground;
        char* token;
        token = strtok(command, " ");
        int i = 0;
        for(;token!=NULL;i++){
            argv[i] = token;
            token = strtok(NULL, " ");
        }
        if(!strcmp(argv[i-1], "&")){
            beakground = 1;
            argv[i-1] = NULL;
        }
        else{
            beakground = 0;
            argv[i] = NULL;
        }
        if((pid=fork())== -1) exit(1);
        if(pid==0){
            execvp(argv[0],argv);
            exit(1);
        }
        if(!beakground) wait(NULL);
        if (strcmp(command, "exit") == 0) exit(0);
    }
    return 0;
}
```

С Ширина табуляції: 8 Стр 41, Стлб 34 ВСТ

Makefile

```
Открыть  Makefile  Сохранить
~/Lab_5/Lab_5.2

all: soul compile clean run

compile:
    gcc soul.o -o soul

soul:
    gcc -c soul.c -o soul.o

clean:
    rm -rf *.o

run:
    ./soul
```

Makefile Ширина табуляції: 8 Стр 1, Стлб 1 ВСТ

Тестування:

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.2
Файл Правка Вид Поиск Терминал Справка
andrew@andrew-VirtualBox:~$ cd /home/andrew/Lab_5/Lab_5.2
andrew@andrew-VirtualBox:~/Lab_5/Lab_5.2$ make
gcc -c soul.c -o soul.o
soul.c: In function 'main':
soul.c:17:3: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
    gets(command);
    ^~~~~
    fgets
gcc soul.o -o soul
soul.o: In function `main':
soul.c:(.text+0x64): warning: the `gets' function is dangerous and should not be used.
rm -rf *.o
./soul
>ls
Makefile  photo  program  soul  soul.c
>./program
fact(5) = 120
average(5, 10) = 7.500000
>./program &
>fact(5) = 120
average(5, 10) = 7.500000
█
```


Файл Правка Вид Поиск Терминал Справка

andrew@andrew-VirtualBox:~\$ cd /home/andrew/Lab_5/Lab_5.2

andrew@andrew-VirtualBox:~/Lab_5/Lab_5.2\$ make

gcc -c soul.c -o soul.o

soul.c: In function 'main':

soul.c:17:3: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]

gets(command);

^~~~~

fgets

gcc soul.o -o soul

soul.o: In function 'main':

soul.c:(.text+0x64): warning: the 'gets' function is dangerous and should not be used.

rm -rf *.o

./soul

>ls

Makefile photo program soul soul.c

>./program

fact(5) = 120

average(5, 10) = 7.500000

>./program &

>fact(5) = 120

average(5, 10) = 7.500000

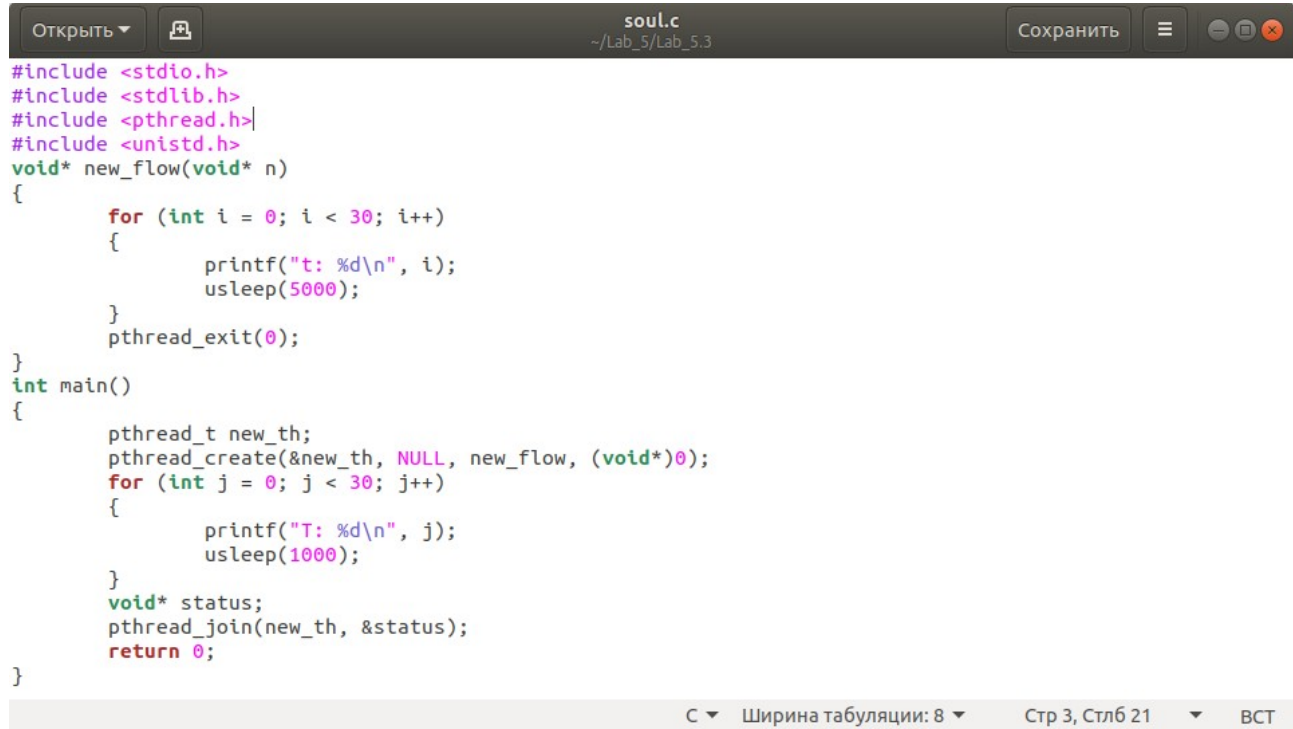
>exit

andrew@andrew-VirtualBox:~/Lab_5/Lab_5.2\$

3. Розробіть застосування для Linux і Windows, що реалізує паралельне виконання коду двома потоками.

Код:

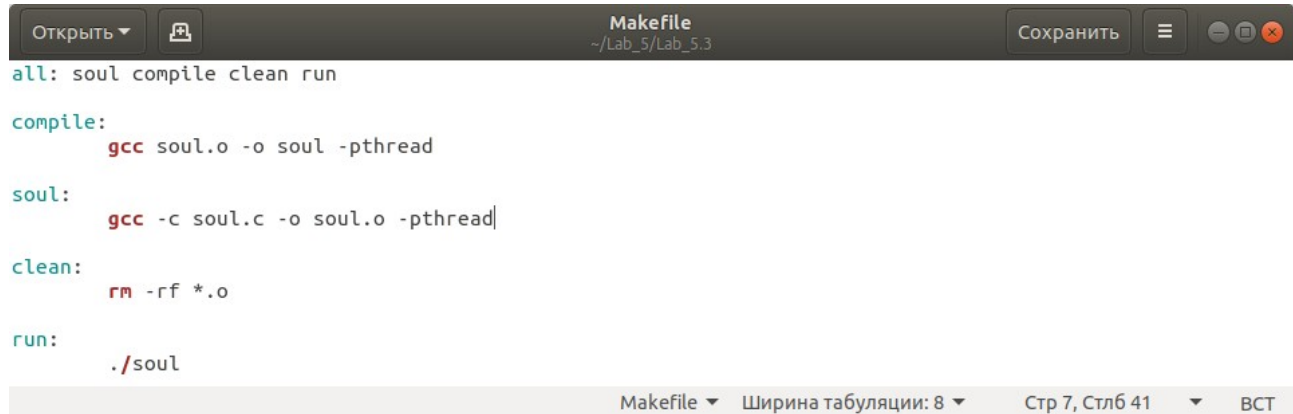
soul.c



```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <unistd.h>
void* new_flow(void* n)
{
    for (int i = 0; i < 30; i++)
    {
        printf("t: %d\n", i);
        usleep(5000);
    }
    pthread_exit(0);
}
int main()
{
    pthread_t new_th;
    pthread_create(&new_th, NULL, new_flow, (void*)0);
    for (int j = 0; j < 30; j++)
    {
        printf("T: %d\n", j);
        usleep(1000);
    }
    void* status;
    pthread_join(new_th, &status);
    return 0;
}
```

С Ширина табуляції: 8 Стр 3, Стлб 21 ВСТ

Makefile



```
all: soul compile clean run

compile:
    gcc soul.o -o soul -pthread

soul:
    gcc -c soul.c -o soul.o -pthread

clean:
    rm -rf *.o

run:
    ./soul
```

Makefile Ширина табуляції: 8 Стр 7, Стлб 41 ВСТ

Тестування:

```
andrew@andrew-VirtualBox: ~/Lab_5/Lab_5.3
Файл Правка Вид Поиск Терминал Справка
andrew@andrew-VirtualBox:~/Lab_5/Lab_5.3$ cd /home/andrew/Lab_5/Lab_5.3
andrew@andrew-VirtualBox:~/Lab_5/Lab_5.3$ make
gcc -c soul.c -o soul.o -pthread
gcc soul.o -o soul -pthread
rm -rf *.o
./soul
T: 0
t: 0
T: 1
T: 2
T: 3
T: 4
t: 1
T: 5
T: 6
T: 7
T: 8
T: 9
t: 2
T: 10
T: 11
T: 12
T: 13
t: 3
T: 14
T: 15
T: 16
T: 17
T: 18
t: 4
T: 19
T: 20
T: 21
T: 22
T: 23
t: 5
T: 24
```

Без приєднання потоків, потік t завершується після завершення T (потік T завешує весь процес). Коли потоки приєднані, то потік t виконується після потоку T, до свого власного завершення.

Висновок: Виконавши дану лабораторну роботу я оволодів практичними навичками застосування програмних інтерфейсів ОС для керування процесами та потоками.