

Міністерство освіти і науки України
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

ОПЕРАЦІЙНІ СИСТЕМИ

Комп'ютерний практикум

Робота № 7

Варіант 8

Виконав:

Студент групи ФБ-95

Прохоренко Я.Ю.

Основи роботи з потоками у Linux з використанням бібліотеки pthread

Мета: Оволодіння практичними навичками роботи з потоками POSIX у Linux з використанням бібліотеки pthread.

Operating System: Ubuntu 20.10

Kernel: Linux 5.8.0-43-generic

Architecture: x86-64

Дата виконання: 19.02.2021

```
#include <iostream>
#include <pthread.h>
#include <string>
#include <unistd.h>

using namespace std;

string parent[] = { "I", "draw", "ten", "lines", "of", "text", "right", "now", "and", "here" };
string child[] = { "I'm", "a", "child", "and", "I", "am", "drawing", "10", "lines", "now" };

void *functionChild(void*)
{
    for(int i = 0; i < 10; i++)
    {
        cout << child[i] << "\n";
    }

    return 0;
}

void *functionParent(void*)
{
    pthread_t child_thread;
    pthread_create(&child_thread, NULL, *functionChild, NULL);
    sleep(1);
    for(int i = 0; i < 10; i++)
    {
        cout << parent[i] << "\n";
    }
    return 0;
}
```

```

int main()
{
    pthread_t parent_thread;
    pthread_create(&parent_thread, NULL, *functionParent, NULL);
    sleep(8);
    return 0;
}

```



```

augfitzr@ubuntu:~$ g++ -std=c++11 lab7.cpp -o lab7 -lpthread
augfitzr@ubuntu:~$ ./lab7
I'm
a
child
and
i
am
drawing
10
lines
now
I
draw
ten
lines
of
text
right
now
and
here

```

Sleep необхідно було викликати для того щоб потоки встигли виконати свої задачі, до завершення main потоку.

```

#include <iostream>
#include <pthread.h>
#include <string>
#include <unistd.h>

```

```
using namespace std;
```

```
string parent[] = { "I", "draw", "ten", "lines", "of", "text", "right", "now", "and", "here"};  
string child[] = { "I'm", "a", "child", "and", "i", "am", "drawing", "10", "lines", "now" };
```

```
void *functionChild(void*)  
{  
    for(int i = 0; i < 10; i++)  
    {  
        cout << child[i] << "\n";  
    }  
  
    return 0;  
}
```

```
void *functionParent(void*)  
{  
    pthread_t child_thread;  
    pthread_create(&child_thread, NULL, *functionChild, NULL);  
    pthread_join(child_thread, NULL);  
    for(int i = 0; i < 10; i++)  
    {  
        cout << parent[i] << "\n";  
    }  
  
    return 0;  
}
```

```
int main()  
{  
    pthread_t parent_thread;  
    pthread_create(&parent_thread, NULL, *functionParent, NULL);  
    pthread_join(parent_thread, NULL);  
    return 0;  
}
```

```
augfitzr@ubuntu:~$ g++ -std=c++11 lab7.cpp -o lab7 -lpthread
augfitzr@ubuntu:~$ ./lab7
I`m
a
child
and
i
am
drawing
10
lines
now
I
draw
ten
lines
of
text
right
now
and
here
```

Pthread_join очікує завершення потоку і тому «сон» використовувати не потрібно, «рух» починається одразу як потік завершить свою роботу.

```
#include <iostream>
#include <pthread.h>
#include <string>
#include <unistd.h>

using namespace std;

#define count_threads 4

void* output_text_line(void* tl)
{
    string *getted = (string*)tl;
    for(int i = 0; i < 2; i++)
    {
        cout << getted[i] << "\n";
    }

    return NULL;
}
```

```

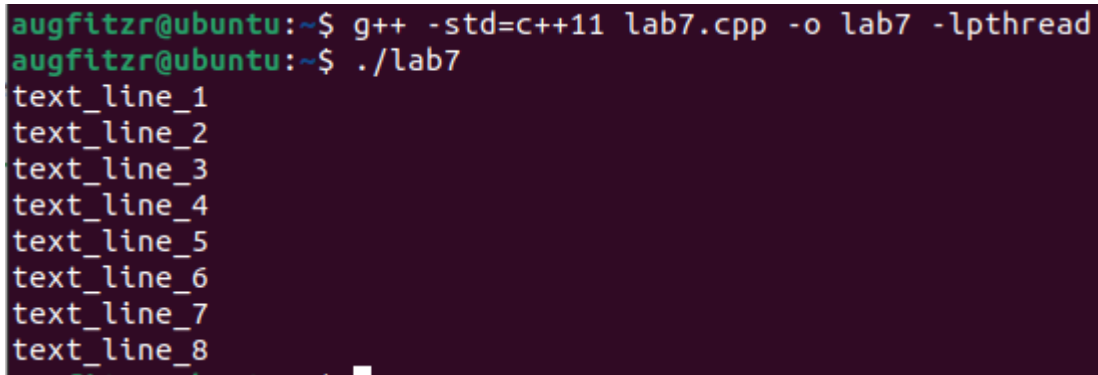
int main()
{
    string text_lines[] = {
        "text_line_1", "text_line_2", "text_line_3", "text_line_4", "text_line_5", "text_line_6", "text_line_7", "text_line_8"
    };

    pthread_t threads[count_threads];

    for(int i = 0; i < count_threads; i++)
    {
        pthread_create(&threads[i], NULL, output_text_line, (void*)(text_lines+(i*2)));
        pthread_join(threads[i], NULL);
    }

    return 0;
}

```



```

augfitzr@ubuntu:~$ g++ -std=c++11 lab7.cpp -o lab7 -lpthread
augfitzr@ubuntu:~$ ./lab7
text_line_1
text_line_2
text_line_3
text_line_4
text_line_5
text_line_6
text_line_7
text_line_8

```

```

#include <iostream>
#include <pthread.h>
#include <string>
#include <unistd.h>

using namespace std;

string text[] = {"asd", "vxc", "thd", "324", "xcv"};

void* functionChild(void*)
{
    for(int i = 0; i < 5; i++)

```

```

    {
        sleep(1);
        cout << text[i] << "\n";
    }
    return 0;
}

```

```

void* functionParent(void*)

```

```

{
    pthread_t child_thread;
    pthread_create(&child_thread, NULL, functionChild, NULL);
    sleep(3);
    if(pthread_cancel(child_thread) == 0)
    {
        cout << "Child thread cancelled\n";
    }

    return 0;
}

```

```

int main()

```

```

{
    pthread_t parent_thread;
    pthread_create(&parent_thread, NULL, functionParent, NULL);
    pthread_join(parent_thread, NULL);

    return 0;
}

```

```

augfitzr@ubuntu:~$ g++ -std=c++11 lab7.cpp -o lab7 -lpthread
augfitzr@ubuntu:~$ ./lab7
asd
vxc
Child thread cancelled

```

```

#include <iostream>
#include <pthread.h>
#include <string>

```

```

#include <unistd.h>

using namespace std;

string text[] = {"asd", "vxc", "thd", "324", "xcv"};

void func(void*)
{
    cout << "Something\n";
}

void* functionChild(void*)
{
    pthread_cleanup_push(func, NULL);
    for(int i = 0; i < 5; i++)
    {
        sleep(1);
        cout << text[i] << "\n";
    }
    pthread_cleanup_pop(1);
    return 0;
}

void* functionParent(void*)
{
    pthread_t child_thread;
    pthread_create(&child_thread, NULL, functionChild, NULL);
    sleep(3);
    pthread_cancel(child_thread);
    pthread_join(child_thread, NULL);

    return 0;
}

int main()
{
    pthread_t parent_thread;
    pthread_create(&parent_thread, NULL, functionParent, NULL);
    pthread_join(parent_thread, NULL);
}

```



```
    return 0;  
}
```

```
augfitzr@ubuntu:~$ g++ -std=c++11 lab7.cpp -o lab7 -lpthread  
augfitzr@ubuntu:~$ ./lab7  
asd  
vxc  
Something
```

Висновки: за допомогою потоків ми можемо виконувати якісь дії паралельно, а не одна за одною, передаючи різні параметри та коректно обробляючи.