



Міністерство освіти і науки України
Національний технічний університет
України
«Київський політехнічний інститут»

Лабораторна робота №2
СТВОРЕННЯ ПАРАЛЕЛЬНИХ ПОТОКІВ ОПЕРАЦІЙНОЇ
СИСТЕМИ LINUX ТА НАЙПРОСТІША СИНХРОНІЗАЦІЯ

Виконав студент групи: КВ-04

ПІБ: Ільчук Олександр

Київ 2022

Постановка задачі

1. Опрацювати всі надані лектором приклади коду паралельних потоків по темі «Засоби взаємодії паралельних потоків операційної системи Linux», що знаходяться в директорії **01_Creation**, тобто:

- - вміти запускати всі ці приклади і отримувати результати на захисті лабораторної роботи;
- - знати які структури даних та функції взаємодії паралельних потоків описані в коді кожного прикладу та як вони працюють, а також вміти це пояснити на захисті лабораторної роботи;
- - бути готовими до виконання модифікацій будь-яких з цих прикладів на захисті лабораторної роботи.

2. Написати програму, яка виводить на друк таблицю значень трьох функцій при паралельній реалізації обчислення значень кожної функції за допомогою трьох паралельних потоків згідно наведених нижче вимог.

3. Кожна із трьох заданих функцій $f_1(x)$, $f_2(x)$, $f_3(x)$ обчислює свої значення при зміні значень аргументу x_i ($i=0, 1, \dots, n$; $n>0$) в межах діапазону, заданого дійсними числами a та b ($b>a$); тобто значення x_i обчислюються за формулою $x_i=a+ih$, де $h=(b-a)/n$.

4. Отримані дійсні значення x_i та $f_1(x_i)$, $f_2(x_i)$, $f_3(x_i)$ вивести на екран у вигляді таблиці заданої форми (табл. 1).

5. Номери функцій $f_1(x)$, $f_2(x)$, $f_3(x)$, а також значення a , b , n визначаються за варіантом завдання (табл. 2).

6. Виконати налагодження написаної програми.

Вимоги до програми

1. Програма повинна правильно розв'язувати поставлену задачу при вхідних даних a , b , n ($a < b$, $n \leq 10$).

2. В програмі не дозволяється використовувати масиви ані для збереження обчислених значень функцій, ані для будь-яких інших цілей.

3. В заголовку надрукованої таблиці мають вказуватися назви функції відповідно до варіанта, наприклад, $SIN(x)$, $ABS(x+7)*5$, а не $f_1(x)$, $f_2(x)$, $f_3(x)$.

4. Обчислення значень математичних функцій $f_1(x)$, $f_2(x)$, $f_3(x)$ мусить відбуватися з врахуванням області допустимих значень **в рамках трьох паралельних потоків**.

5. Створення та запуск усіх трьох потоків, що обчислюють значення математичних функцій $f_1(x)$, $f_2(x)$, $f_3(x)$, повинні бути виконані у головній програмі (головному, четвертому, потоці). **При створенні потоків їм повинні бути передані значення a , b , n через аргумент потоку.**

6. Виведення результуючої таблиці заданої форми (табл. 1) на екран повинно бути виконано у головному (четвертому) потоці багатопоточної програми.

7. Алгоритм кожного з трьох паралельних потоків, що обчислюють значення математичних функцій $f_1(x)$, $f_2(x)$, $f_3(x)$, повинен бути реалізований у вигляді циклу, що обчислює задану кількість значень функції згідно заданих значень a , b , n .

7. Передавання значень функцій $f_1(x)$, $f_2(x)$, $f_3(x)$, що обчислюються у трьох потоках, до головного потоку виконувати після отримання кожного нового значення функцій через глобальні змінні.

8. Для синхронізації обчислення значень функцій $f_1(x)$, $f_2(x)$, $f_3(x)$ у трьох потоках та виведення на екран нового рядка таблиці після отримання нових значень функцій у головному потоці дозволяється використовувати тільки конструкцію бар'єра (*pthread_barrier_t*) та затримки (функція *usleep()*).

Зміст звіту

1. Постановка задачі, вимоги до програми та конкретний варіант завдання.
2. Текст програми.
3. Тести для налагодження і результати налагодження, отримані на комп'ютері.
4. Побудована таблиця значень функцій $f_1(x)$, $f_2(x)$, $f_3(x)$.

№ п/п	Номери функцій $f_1(x)$, $f_2(x)$, $f_3(x)$	a	b	n
7	2, 5, 11	$-\pi$	π	8

Функції:

$$f_1(x) = \cos^2(x) + \sin(x)$$

$$f_2(x) = \sin^2(x) * (1 + \cos(x))$$

$$f_3(x) = \sin(x) * (1 + \cos^2(x))$$

Код програми:

Main.c

```
#include <stdio.h>
#include <pthread.h>
#include <unistd.h>
#include <pair.h>
#include <pthread_barrier.h>
#include "Constants.h"
#include "barrier.h"
#include "functions.h"
```

```
#define InitArgs(args) \
{ \
```



```

    pair pair_args = * (pair*) args; \
    pair boundaries = pair_first(pair, pair_args); \
    pair otherArgs = pair_second(pair, pair_args); \
    float a = pair_first(float, boundaries); \
    float b = pair_second(float, boundaries); \
    int n = pair_first(int, otherArgs); \
    float h = pair_second(float, otherArgs);

float CalculateX(float a, float h, float i) { return a + i *
h; }
void *function1(void *args) {
    ArgsHandler(args)
    for (int i = 0; i <= n; i++) {
        float x = CalculateX(a, h, i);
        function1Result = powf(cosf(x), 2) + sinf(x);
        pthread_barrier_wait(GetBarrierPointer());
    }
    return 0;
}
void *function2(void *args) {
    ArgsHandler(args)
    for (int i = 0; i <= n; i++) {
        float x = CalculateX(a, h, i);
        function2Result = powf(sinf(x), 2) * (1 + cosf(x));
        pthread_barrier_wait(GetBarrierPointer());
    }
    return 0;
}
void *function3(void *args) {
    ArgsHandler(args)
    for (int i = 0; i <= n; i++) {
        float x = CalculateX(a, h, i);
        function3Result = sinf(x) * (1 + powf(cosf(x), 2));
        pthread_barrier_wait(GetBarrierPointer());
    }
    return 0;
}
float GetFunction1Result() { return function1Result; }
float GetFunction2Result() { return function2Result; }
float GetFunction3Result() { return function3Result; }

```

Functions.h

```

#ifndef functions_h
#define functions_h
float CalculateX(float a, float h, float i);
void *function1(void *args);
void *function2(void *args);
void *function3(void *args);
float GetFunction1Result();

```

```
a := -π    b := π    n := 8
```

```
h :=  $\frac{b-a}{n}$  = 0.7854
```

```
x (i) := a + i · h
```

```
f1 (x) := cos (x)2 + sin (x)
```

```
f2 (x) := (sin (x))2 · (1 + cos (x))
```

```
f3 (x) := sin (x) · (1 + cos (x)2)
```

i := 0	x := x (i) = -3.1416	f1 (x) = 1	f2 (x) = 0	f3 (x) = -6.4622 · 10 ⁻¹⁵
i := 1	x := x (i) = -2.3562	f1 (x) = -0.2071	f2 (x) = 0.1464	f3 (x) = -1.0607
i := 2	x := x (i) = -1.5708	f1 (x) = -1	f2 (x) = 1	f3 (x) = -1
i := 3	x := x (i) = -0.7854	f1 (x) = -0.2071	f2 (x) = 0.8536	f3 (x) = -1.0607
i := 4	x := x (i) = 1.0658 · 10 ⁻¹⁴	f1 (x) = 1	f2 (x) = 2 · 10 ⁻²⁸	f3 (x) = 2.1316 · 10 ⁻¹⁴
i := 5	x := x (i) = 0.7854	f1 (x) = 1.2071	f2 (x) = 0.8536	f3 (x) = 1.0607
i := 6	x := x (i) = 1.5708	f1 (x) = 1	f2 (x) = 1	f3 (x) = 1
i := 7	x := x (i) = 2.3562	f1 (x) = 1.2071	f2 (x) = 0.1464	f3 (x) = 1.0607
i := 8	x := x (i) = 3.1416	f1 (x) = 1	f2 (x) = 0	f3 (x) = -4.2388 · 10 ⁻¹⁴

```
float GetFunction2Result();
float GetFunction3Result();
#endif /* functions_h */
```

Constants.h

```
#endif /* functions_h */
#ifndef Constants_h
#define Constants_h
#include <math.h>
const float A = -M_PI;
const float B = M_PI;
const int N = 8;
const float H = (B - A) / N;
```

barrier.h

```
#endif /* Constants_h */
```

```
#ifndef barrier_h
#define barrier_h
void* GetBarrierPointer();
#endif /* barrier_h */
```

barrier.c

```
#include <pthread_barrier.h>
pthread_barrier_t barrier;
void* GetBarrierPointer() { return &barrier; }
```

Тести: Expected

```
a := -pi      b := pi      n := 8
```

```
h := (b - a) / n = 0.7854
```

```
x(i) := a + i * h
```

```
f1(x) := cos(x)^2 + sin(x)
```

```
f2(x) := (sin(x))^2 * (1 + cos(x))
```

```
f3(x) := sin(x) * (1 + cos(x)^2)
```

i := 0	x := x(i) = -3.1416	f1(x) = 1	f2(x) = 0	f3(x) = -6.4622 · 10 ⁻¹⁵
i := 1	x := x(i) = -2.3562	f1(x) = -0.2071	f2(x) = 0.1464	f3(x) = -1.0607
i := 2	x := x(i) = -1.5708	f1(x) = -1	f2(x) = 1	f3(x) = -1
i := 3	x := x(i) = -0.7854	f1(x) = -0.2071	f2(x) = 0.8536	f3(x) = -1.0607
i := 4	x := x(i) = 0.000000	f1(x) = 1	f2(x) = 0.000000	f3(x) = 0.000000
i := 5	x := x(i) = 0.7854	f1(x) = 1.2071	f2(x) = 0.8536	f3(x) = 1.0607
i := 6	x := x(i) = 1.5708	f1(x) = 1	f2(x) = 1	f3(x) = 1
i := 7	x := x(i) = 2.3562	f1(x) = 1.2071	f2(x) = 0.1464	f3(x) = 1.0607
i := 8	x := x(i) = 3.1416	f1(x) = 1	f2(x) = 0	f3(x) = -6.4622 · 10 ⁻¹⁵

Actual

index	x	cos^2(x)+sin(x)	sin^2(x)*(1+cos(x))	sin(x)*(1+cos^2(x))
0	-3.141593	1.000000	0.000000	0.000000
1	-2.356194	-0.207107	0.146447	-1.060660
2	-1.570796	-1.000000	1.000000	-1.000000
3	-0.785398	-0.207107	0.853553	-1.060660
4	0.000000	1.000000	0.000000	0.000000
5	0.785398	1.207107	0.853553	1.060660
6	1.570796	1.000000	1.000000	1.000000
7	2.356195	1.207107	0.146446	1.060660
8	3.141593	1.000000	0.000000	-0.000000

Program ended with exit code: 0