МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет

«Харківський політехнічний інститут»

Навчально-науковий інститут комп’ютерного моделювання, прикладної фізики та математики

Кафедра динаміки та міцності машин

Звіт

про виконання

Комп’ютерного практикуму №10

по курсу «Комп'ютерні мережі та розподілені обчислення»

з теми "Потоки"

Виконав: студент групи ІКМ-220Б

Пономаренко О.В.

Викладач: доцент кафедри ДММ

Трубаєв О.І.

Харків 2022

**Завдання 1.**

Написати код, у якому є глобальна змінна int data = 0 та два потоки (один у циклі збільшує змінну на 1, інший потік зменшує на 1).

1й потік закінчує свою роботу, якщо data == 1000,

2й потік закінчує свою роботу, якщо data == -1000.

Який потік першим закінчить свою роботу?

При зупиненні одного потоку, інший також завершує роботу.

*Розв'язок.*

Спочатку підключаємо всі необхідні бібліотеки. Далі створюємо змінну data з початковим значенням 0. Створюємо дві функції: одна в циклі додає 1, інша віднімає. Умова закінчення для кожної з них – data набуває значення +-1000 (рис. 1).

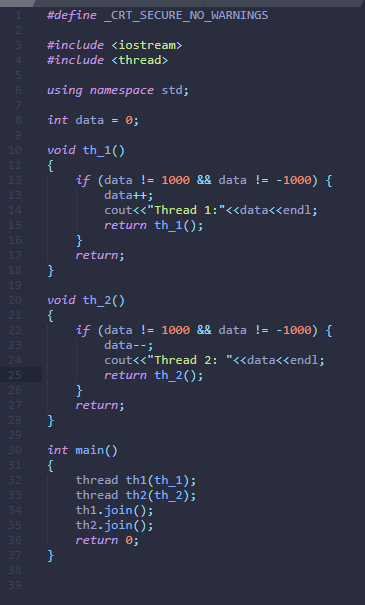


Рисунок 1

В main запускаємо ці дві функції в два потоки, і під час роботи потоків виводимо змінну data і бачимо в консолі текст, який повідомляє нам, який поток зараз виконався. Останнє число буде або 1000, або -1000.

Оцінимо результат роботи програми (рис. 2). Можемо побачити, що другий потік виконувався швидше, тому останнє число -1000.

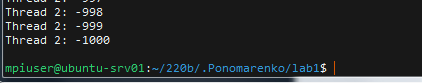


Рисунок 2

**Завдання 2.**

Написати код, у якому є список std::list <int> та два потоки.

В одному потоці у списку знаходяться числа, які згенеровано за допомогою генератору випадкових чисел. В іншому потоці ці числа виводяться на екран.

Організувати роботу потоків так, щоб взаємодія зі списком була коректною. Для організації роботи потоків використовувати м'ютекси.

*Розв'язок.*

Знову ж таки спочатку додаємо необхідні бібліотеки. Створюємо змінну n(довжина нашого списку) та створюємо сам список х. Створюємо дві функції: одна з яких додає рандомні значення в список, а інша в циклі виводить ці значення по черзі. Ми додаємо м'ютекс на кожній ітерації для того, щоб поки один потік не завершить роботу, другий потік не міг взаємодіяти з даними в першому потоці, і навпаки. (рис. 3)

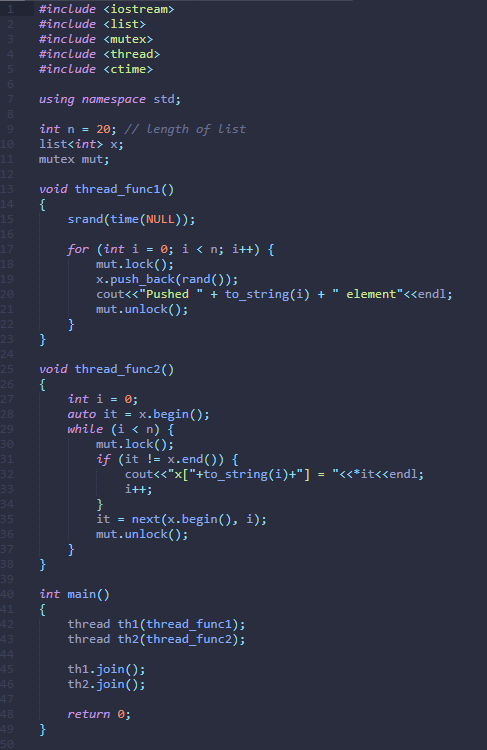


Рисунок 3

В main просто запускаємо ці функції в два потоки і чекаємо на завершення роботи. Результат можна побачити на рис. 4.

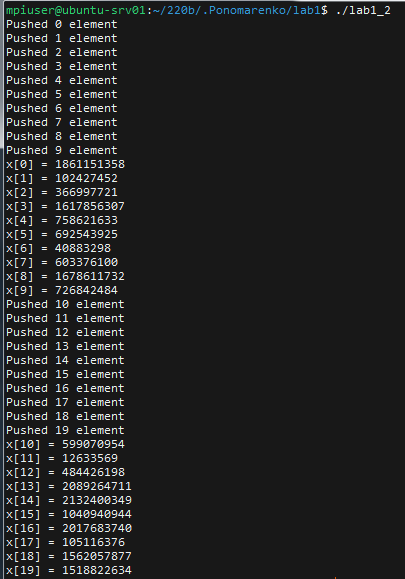


Рисунок 4

**Завдання 3.**

Розробити програму, що емулює діяльність Філософів, якщо відомо, що:

1. Є 5 філософів, 5 тарілок, 5 виделок
2. Філософ може знаходитись у двох станах:

* Думає
* Їсть

1. Для того щоб їсти, необхідні дві виделки
2. Філософ може їсти деякий час (випадкове число), перш ніж почне думати. Якщо у філософа є дві вільні виделки – він їсть.
3. В початковий момент всі Філософи думають.

Програма виводить на екран поточний стан Філософів.

*Розв'язок.*

Знову ж таки спочатку додаємо необхідні бібліотеки, задаємо значення змінних для вилок, тарілок і філософів, а також функцію для обмеження величини рандомних значень від х до у включно. Також створимо клас “філософ”. (рис. 5) Кожен філософ знає загальну кількість філософів, знає те, що він зараз робить, у кожного свій потік, і кожен знає свій номер. Також кожен бачить кількість виделок та тарілок.



Рисунок 5

На рисунку 6 можна побачити другу частину класу філософа, де прописаний алгоритм роботи його дій, та саму функцію main, яка створює задану кількість філософів (в нашому випадку 5).



Рисунок 6

Нижче, на 7 рисунку показаний вивід роботи программи.

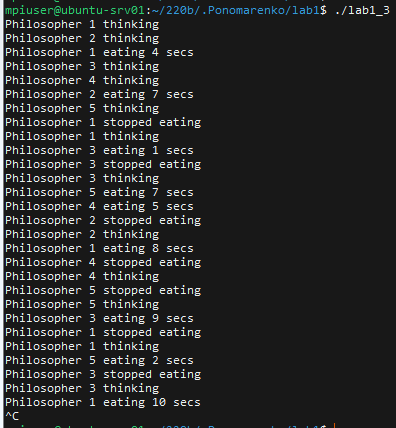


Рисунок 7