

Міністерство освіти і науки України

НТУУ «Київський політехнічний інститут»

Фізико-технічний інститут

ОПЕРАЦІЙНІ СИСТЕМИ

**Комп’ютерний практикум №7**

Варіант №11

**Виконав:**

Студент 2 курсу ФТІ

Групи ФІ-92

Поночевний Назар Юрійович

**Перевірив:**

Ільїн Костянтин Іванович

Київ – 2021

**Робота №7. Основи роботи з потоками у Linux з використанням бібліотеки pthread**

Варіант №11

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

**Завдання для самостійної підготовки**

1. Ознайомитись з документацією і прикладами використання бібіліотеки pthread:

● man pages;

● книги з числа рекомендованих, зокрема [1, розд. 2.2], [5, п. 3.8.4];

● для першого знайомства — POSIX Threads, матеріал з Вікіпедії [8] (російською мовою, бо на поточний момент стаття українською мовою є вкрай неповною);

● дуже непогана стаття [9];

● простий приклад застосування pthread [10];

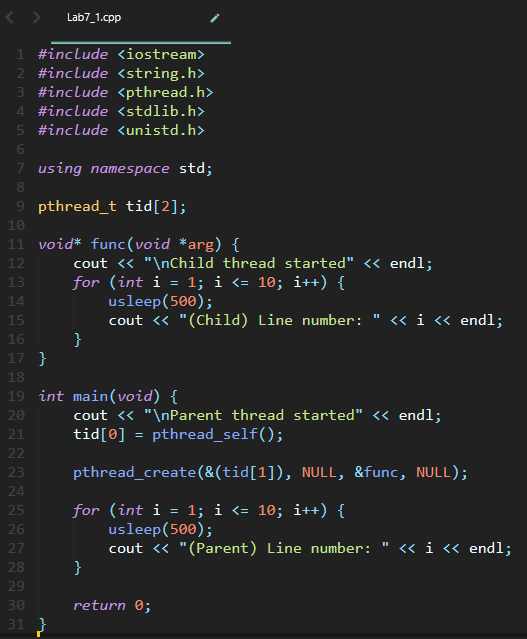
● велика книга [11];

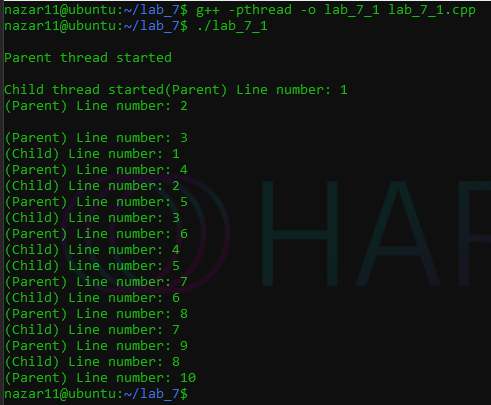
● авторитетне першоджерело [12].

2. Якщо не робили попередню роботу, то перевірити, чи встановлений у вашій системі Linux компілятор С/С++ (g++). Якщо ні, встановіть за допомогою менеджера пакетів.

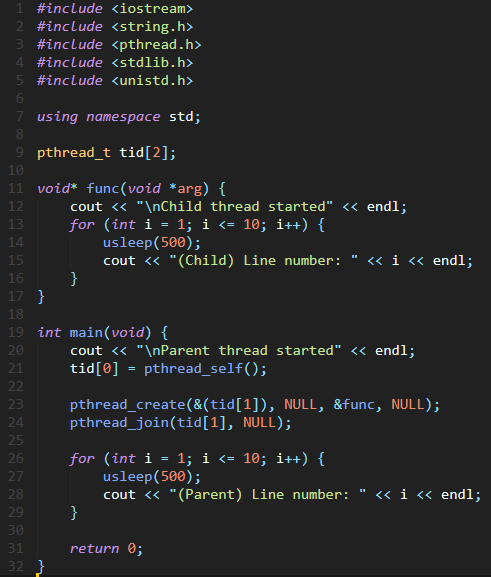
**Завдання до виконання**

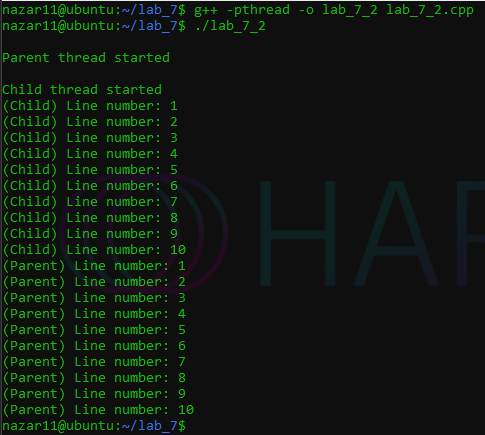
1. Створення потоку. Напишіть програму, що створює потік. Застосуйте атрибути за умовчанням. Батьківський і дочірній потоки мають роздрукувати по десять рядків тексту.



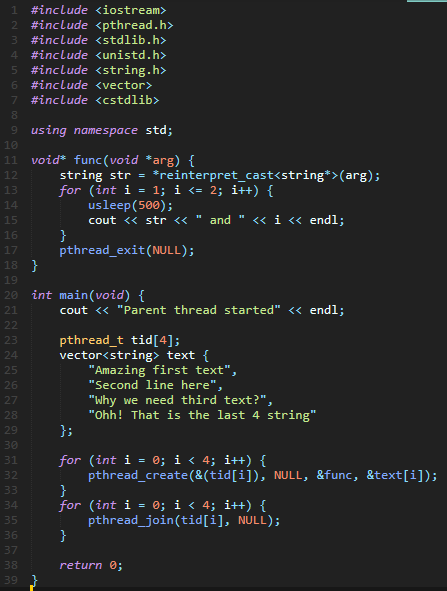


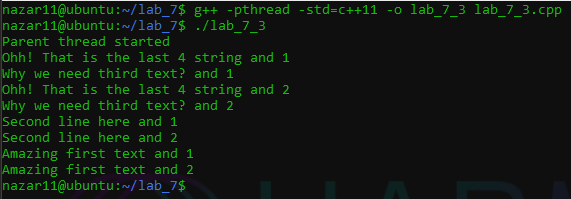
2. Очікування потоку. Модифікуйте програму п. 1 так, щоби батьківський потік здійснював роздрукування після завершення дочірнього (функція pthread\_join()).



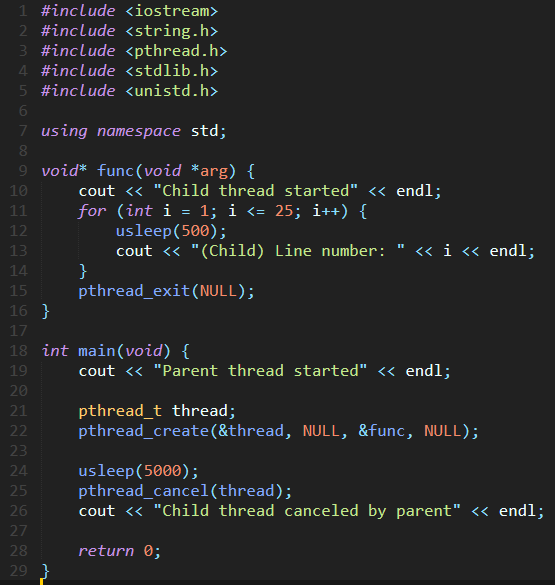


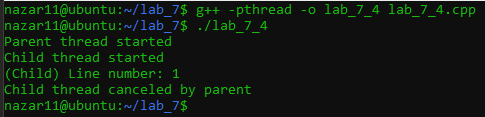
3. Параметри потоку. Напишіть програму, що створює чотири потоки, що виконують одну й ту саму функцію. Ця функція має роздруковувати послідовність текстових рядків, переданих як параметр. Кожний зі створених потоків має роздруковувати різні послідовності рядків.



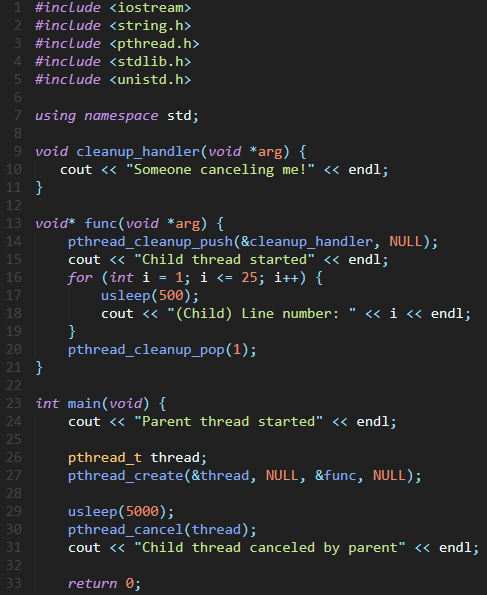


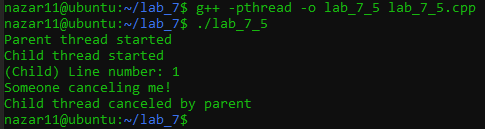
4. Примусове завершення потоку. Дочірній потік має роздруковувати текст на екран. Через дві секунди після створення дочірнього потоку, батіківський потік має перервати його (функція pthread\_cancel()).





5. Обробка завершення потоку. Модифікуйте програму п. 4 так, щоби дочірній потік перед завершенням роздруковував повідомлення про це (pthread\_cleanup\_push()).





**Висновок**

ОС Linux надає гарну підтримку потоків та бібліотеки для роботи з ними. Працюючи з потоками з Windows можу сказати, що всі необхідні методи, такі як створення, передача параметрів, очікування та безпечне переривання дуже схоже реалізовані. Звичайно, зараз вже є більш потужні бібліотеки, такі як <thread>, наприклад, але основні методи працюють за однаковими принципами.