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

# Мета

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

# Завдання для самостійної підготовки

1. 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].

1. 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.

# Довідковий матеріал

Тут наведено лише мінімальну інформацію, достатню хіба що для того, щоби зрозуміти, про що йде мова. Решту інформації необхідно здобути з джерел, названих вище.

Спочатку створення потоків у Linux здійснювали за допомогою виклику clone(), який багато у чому подібний до fork(), але дозволяє призначити, які ресурси є спільними між батьківським процесом і нащадком. В результаті насправді з’являвся не потік, а новий процес.

Для підтримки потоків у ядрі Linux спочатку була розроблена бібліотека LinuxThreads, а пізніше — NPTL. Слід зазначити, що на відміну від Solaris і Windows, де процеси фактично є контейнерами для потоків, потоки у Linux все одно залишаються “полегшеними процесами”, але програмно вони виглядають так (або майже так) як повинні виглядати потоки POSIX.

Програмний інтерфейс роботи з потоками — це бібліотека pthread. У цій роботі ви застосуєте цю бібліотеку для створення і завершення потоків і дослідите, як це працює. Ця бібліотека реалізована для різних операційних систем, у тому числі і для Windows. Тому навички, які будуть отримані вами у ході виконання цієї роботи, можна буде застосовувати для програмування під багатьма різними операційними системами.

Функції, які вам необхідно застосовувати (а отже, необхідно прочитати їх документацію):

pthread\_create()

pthread\_exit() (зверніть увагу на особливості застосування цієї функції у початковому потоці, тобто у функції main()) pthread\_join() pthread\_cancel()

А також, можливо, і деякі інші.

# 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()).

Усі отримані результати оформіть у вигляді протоколу. Під час здачі роботи продемонструйте викладачеві роботу розроблених вами програм і їх код.