

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота №1

з дисципліни «Операційні Системи»

на тему: «Створення потоків і їх синхронізація в режимі користувача»

Виконав:
студент гр.ПЗ1911
Сафонов Д. Є.
Прийняв:
Андрющенко В.О.

Дніпро, 2021

Тема. Створення потоків і їх синхронізація в режимі користувача.

Завдання. Розробити програму, яка створює два або більше потоки керування (системний виклик `CreateThread`), що отримують доступ до спільних ресурсів за принципом взаемовиключного доступу. Для синхронізації застосовувати системні структури `CRITICAL_SECTION`. Характеристики ресурсів і спосіб їх використання потоками відповідає індивідуальному завданню.

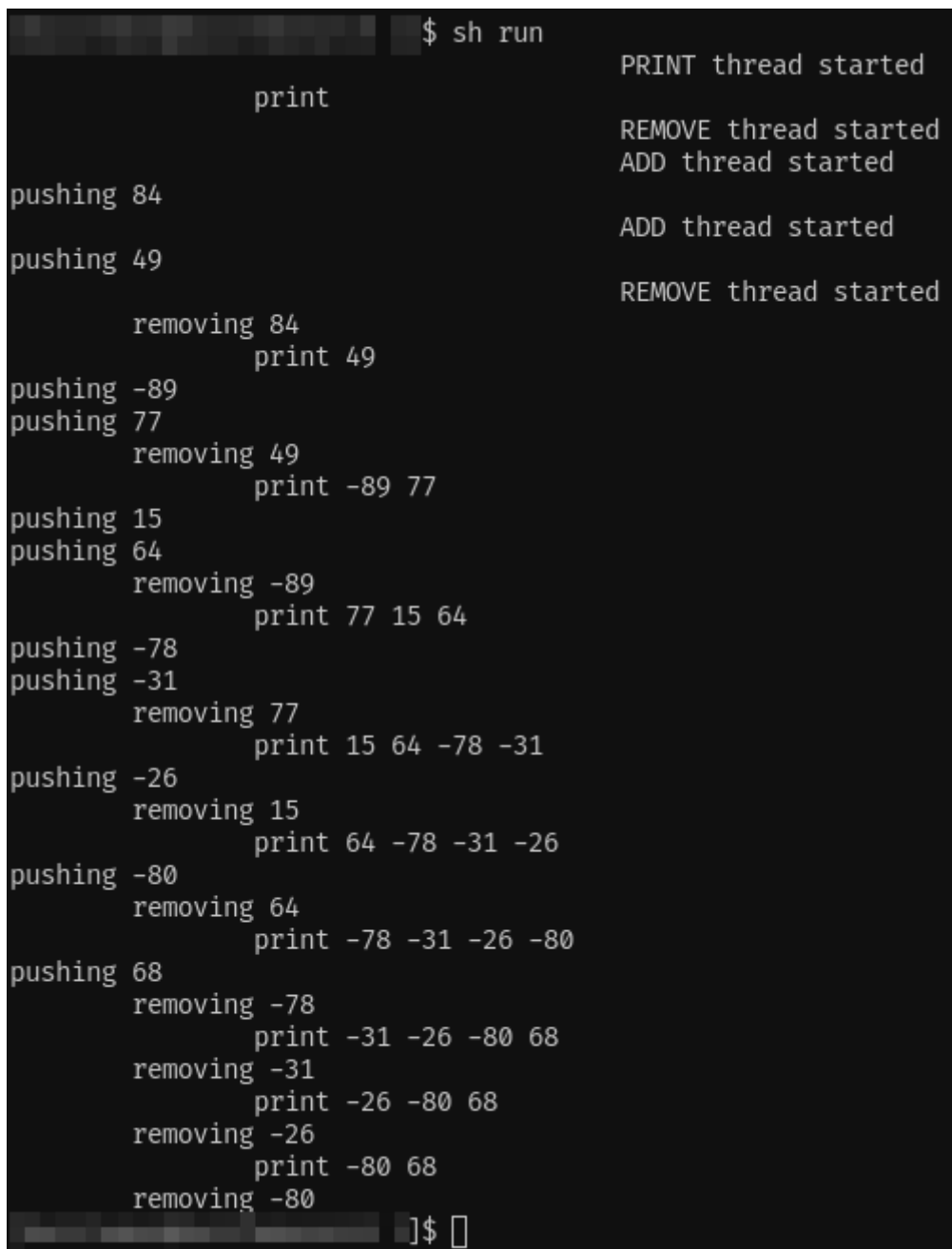
Варіант 11.

Характеристики ресурсу(3): Черга на основі масиву структурованих об'єктів фіксованого розміру.

Спосіб використання ресурсу(1): Потоки першого виду додають інформацію, потоки другого виду видаляють інформацію.

Текст програми. [github](#)

Результати виконання програми.



```
$ sh run
PRINT thread started
print
REMOVE thread started
ADD thread started
pushing 84
ADD thread started
pushing 49
REMOVE thread started
removing 84
print 49
pushing -89
pushing 77
removing 49
print -89 77
pushing 15
pushing 64
removing -89
print 77 15 64
pushing -78
pushing -31
removing 77
print 15 64 -78 -31
pushing -26
removing 15
print 64 -78 -31 -26
pushing -80
removing 64
print -78 -31 -26 -80
pushing 68
removing -78
print -31 -26 -80 68
removing -31
print -26 -80 68
removing -26
print -80 68
removing -80
]$
```

Рисунок 1: З використанням критичної секції

```
$ sh run
print
ADD thread started
REMOVE thread started
PRINT thread started

pushing 3
pushing -74
ADD thread started

removing 3
print -74
REMOVE thread started

pushing 59
pushing -21
removing -74
print 59 -21

pushing 65
pushing -42
removing 59
print -21 65 -42

pushing -70
pushing -64
removing -21
print 65 pushing -72

-42 -70 -64
pushing 86
removing 65
pushing -27
print -42 -70 -64 -72 86 -27

pushing -20
removing -42
pushing -79
print -70 -64 pushing 3

-72 86 -27 -20 -79
removing -70
pushing -18
pushing -57
print -64 -72 86 -27 -20 -79 3 -18 -57

removing -64
pushing 1
print -72 86 -27 -20 -79 3 -18 -57 1

removing -72
print 86 -27 -20 -79 3 -18 -57 1

removing 86
]$
```

Рисунок 2: Без використання критичної секції

Аналіз результатів та висновки.

У сучасному світі багатопотоковість програмного забезпечення є дуже важливим тому, що сучасні процесори мають багато фізичних ядер, а також по декілька потоків на кожне ядро(крім того операційні системи мають планувальники задач можуть розподіляти віртуальні потоки між фізичними, тож програмісту не треба про це турбуватися).

В наведеній програмі можна побачити, що без використання критичної секції декілька потоків можуть звертатися до об'єктів одночасно і таким чином коли один з потоків виконує складні маніпуляції над об'єктом інші потоки можуть користуватися цим потоком, коли він не є готовим.