

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
В.Н.КАРАЗІНА
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ ТА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ

ЛАБОРАТОРНА РОБОТА № 8
З ДИСЦИПЛІНИ «ОПЕРАЦІЙНІ СИСТЕМИ»

ТЕМА «ПОТОКИ. СИНХРОНІЗАЦІЯ 2»

Виконав студент 3курсу, групи
КС31

спеціальності

122 – Комп'ютерні науки

Касьяненко Микита Михайлович

Прийняв:

доцен кафедри шт. ін.

і прогр. забезп. к.н.т.

О.Є. Споров ____

Харків 2022

Завдання 1

Існує потік-виробник та потік-споживач даних. Виробник генерує ціле псевдовипадкове число із заданого діапазону. Споживач забирає це число. За допомогою семафорів синхронізуйте їхню роботу так, щоб споживач не міг намагатися отримати ще не створені числа, а виробник не міг згенерувати більше чисел, ніж може отримати споживач. Через заданий проміжок часу потоки скасовуються головним потоком і програма завершує свою роботу.

Для синхронізації потоків розмістимо між потоком-виробником та потоком-споживачем буфер заданого при старті програми розміру. Виробник може поміщати

згенеровані числа у буфер, а споживач може забирати числа з буфера. Якщо споживач забирає число, його виключають з буфера. Необхідно забезпечити кілька вимог:

1. коли споживач або виробник працює з буфером, то інші потоки повинні чекати, аж поки він завершить свою роботу;
2. коли виробник намагається помістити об'єкт у буфер, а буфер вже заповнений, то виробник повинен дочекатись, поки в буфері з'явиться місце;
3. коли споживач намагається забрати об'єкт із буфера, а буфер ще порожній, то споживач повинен дочекатись, доки в буфері з'явиться об'єкт.

Відповідь

```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-8/task-1
~/Pr/university-works/o/laboratory-8/task-1 P main +3 !8 ?54 gcc main.c -o main ✓
~/Pr/university-works/o/laboratory-8/task-1 P main +3 !8 ?54 ./main 10 ✓

0 0 0 0 0 0 0 0 0 0
Maker: 10
Consumer: 10
Maker: 1
Consumer: 1
0 0 0 0 0 0 0 0 0 0
Maker: 3
Consumer: 3
0 0 0 0 0 0 0 0 0 0
Maker: 7
Consumer: 7
0 0 0 0 0 0 0 0 0 0
Maker: 7
Consumer: 7
0 0 0 0 0 0 0 0 0 0
Maker: 7
Consumer: 7
0 0 0 0 0 0 0 0 0 0
Maker: 7
Consumer: 7
7 0 0 0 0 0 0 0 0 0 Consumer: 7
0
Maker: 8
0 0 0 0 0 0 0 0 0 0 Consumer: 8
0
Maker: 8
```

```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-8/task-1

7 0 0 0 0 0 0 0 0 0 Consumer: 7
0
Maker: 8
0 0 0 0 0 0 0 0 0 0 Consumer: 8
0
Maker: 8
Consumer: 8
0 0 0 0 0 0 0 0 0 0
Maker: 1
0 0 0 0 0 0 0 0 0 0
Consumer: 1
Maker: 7
0 0 0 0 0 0 0 0 0 0
Consumer: 7
Maker: 7
0 0 0 0 0 0 0 0 0 0
Consumer: 7
0 0 0 0 Maker: 3
0 0 0 0 0
Consumer: 3
Maker: 5
0 0 0 0 0 0 0 0 0 0
Consumer: 5
0 0 0 0 0 0 0 0 0 0
Maker: 5
Consumer: 5

~/Pr/university-works/o/laboratory-8/task-1 P main +3 !8 ?54 15s
```

Завдання 2

Існує потік-виробник та потік-споживач даних. Потік-виробник із досить великим періодом генерує псевдовипадкове число, зберігає його в глобальній змінній та сповіщає про це потік-споживач. Потік-споживач отримує сповіщення, забирає число із глобальної змінної,

зануляючи її, та відразу виводить їх у стандартний потік виведення.

Поки даних немає, потік-споживач чекає на повідомлення. Синхронізуйте роботу потоків за допомогою умовних

змінних. Через заданий проміжок часу потоки скасовуються головним потоком і програма завершує свою роботу.

Відповідь

```
mykyta@senku:~/Projects/university-works/operating-systems/laboratory-8/task-2
~/Pr/university-works/o/laboratory-8/task-2 main +3 18 754 gcc main.c -o main
main.c: In function 'produce':
main.c:12:9: warning: implicit declaration of function 'srand' [-Wimplicit-function-declaration]
   12 |         srand(time(NULL));
      |         ^~~~~
main.c:14:28: warning: implicit declaration of function 'rand' [-Wimplicit-function-declaration]
   14 |         int val = (rand() % 10) + 1;
      |                        ^~~~~
main.c:21:17: warning: implicit declaration of function 'sleep' [-Wimplicit-function-declaration]
   21 |         sleep(2);
      |         ^~~~~
~/Pr/university-works/o/laboratory-8/task-2 main +3 18 754 ./main
Threds will process 15 sec
Maker: 2
Consumer: 2
Maker: 6
Consumer: 6
Maker: 9
Consumer: 9
Maker: 8
Consumer: 8
Maker: 3
Consumer: 3
Maker: 8
Consumer: 8
Maker: 3
Consumer: 3
Maker: 1
Consumer: 1
~/Pr/university-works/o/laboratory-8/task-2 main +3 18 754 [15s]
```