

Микропроект 2

Корнилов Георгий, вариант 11

ФКН ПИ НИУ ВШЭ, группа БПИ-197

1 Текст задачи

Задача о магазине. В магазине работают три отдела, каждый отдел обслуживает один продавец. Покупатель, зайдя в магазин, делает покупки в произвольных отделах, и если в выбранном отделе продавец не свободен, покупатель становится в очередь и засыпает, пока продавец не освободится. Создать многопоточное приложение, моделирующее рабочий день магазина.

2 Методы решения задачи

Программа разделена на объекты покупателей и продавцов отделов магазина.

У объекта-покупателя есть очередь отделов магазина, по которой он должен пройти в определенном порядке. Покупатель по очереди посещает отделы магазина: на очередном отделе покупатель встает в очередь в отдел и ждет, пока его «обслужат» в этом отделе.

У объекта-продавца есть очередь, в которую вставать покупателя. На очередном шаге продавец берет первого покупателя в очереди, выводит текущее состояние очереди и обслуживает покупателя со случайной задержкой от 500 до 1500 миллисекунд.

После того, как все покупатели закончат свои покупки, отделы магазина закрываются.

3 Примеры работы программы

Примеры работы программы на ранних тестовых данных (скриншоты работы программы приложены по [ссылке](#)):

```
C:\Users\Mi\Desktop\architectures\microproject2>a.exe
Enter amount of buyers: 3
Buyer #1 goes in following order: 3
Buyer #2 goes in following order: 1
Buyer #3 goes in following order: 1
Buyer #1 goes to shop #3
Buyer #2 goes to shop #1
Buyer #3 goes to shop #1
Seller #3 processes client #1
Current condition
---
|3|
-/-
 1

Seller #1 processes client #2
Current condition
---
|1|
-/-
 2
 3

Seller #1 processes client #3
Current condition
---
|1|
-/-
 3
```

```

C:\Users\Mi\Desktop\architectures\microproject2>a.exe
Enter amount of buyers: 3
Buyer #1 goes in following order: 3
Buyer #2 goes in following order: 1
Buyer #3 goes in following order: 1
Buyer #1 goes to shop #3
Buyer #2 goes to shop #1
Buyer #3 goes to shop #1
Seller #3 processes client #1
Current condition
---
|3|
-/-
1

Seller #1 processes client #2
Current condition
---
|1|
-/-
2
3

Seller #1 processes client #3
Current condition
---
|1|
-/-
3

PS C:\Users\Mi\Desktop\architectures\microproject2> .\a.exe
Enter amount of buyers: -2
Amount of buyers should be positive number

```

4 Пути улучшения

Поскольку на моем компьютере не получилось поставить компиляторы, поддерживающие posix threads либо стандарт C++ 20, то при походе покупателя в отдел магазина необходимо передавать указатель на мьютекс.

Пример использования указателя на мьютекс в коде:

```

void seller::add_buyer(buyer* new_buyer, std::mutex* new_mutex)
{
    queue_mutex.lock();
    new_mutex->lock();
    current_buyers.push({ new_buyer, new_mutex });
    queue_mutex.unlock();
}

```

```

void seller::process_buyer(buyer* cur_buyer, std::mutex* cur_mutex)
{
    cout_mutex->lock();
    std::cout << "Seller_#" << getNumber()
               << "_processes_client_#"
               << cur_buyer->getNumber()
               << std::endl
               << "Current_condition"
               << std::endl
               << to_string()
               << std::endl;
    cout_mutex->unlock();

    std::uniform_int_distribution<int> range(500, 1500);
    int milliseconds = range(randomizer);
    std::this_thread::sleep_for(std::chrono::milliseconds(milliseconds));
    cur_mutex->unlock();
}

void buyer::run()
{
    while (!sellers_order.empty())
    {
        seller* current_seller = sellers_order.front();

        cout_mutex->lock();
        std::cout << "Buyer_#" << getNumber() << "_goes_to_shop_#" << current_seller->
        cout_mutex->unlock();

        std::mutex cur_mutex;
        sellers_order.pop();
        current_seller->add_buyer(this, &cur_mutex);

        cur_mutex.lock();
        cur_mutex.unlock();
    }
}

```

При использовании стандарта C++ 20 вместо мьютекса можно передавать ссылку на `std::latch` либо `std::barrier`. Тогда в методе `buyer::run` вместо лока-анлока мьютекса можно использовать метод `std::latch::wait`. Внутри же метода `seller::process_buyer` вместо анлока мьютекса можно использовать `latch::count_down`.

5 Источники информации

1. <http://www.softcraft.ru/edu/comparch/practice/thread/02-sync/>
2. <https://pro-prof.com/forums/topic/parallel-programming-paradigms>
3. http://rsuib.cc.rsu.ru/tutor/high_performance_computing/chapter3/page02.html
4. <https://www.modernescpp.com/index.php/latches-and-barriers>