

Отчет о домашнем задании по «Архитектуре вычислительных систем» #5

Студент: Кунин Илья, БПИ203

1. Вариант задания

16. Задача о картинной галерее. Вахтер следит за тем, чтобы в картинной галерее одновременно было не более 50 посетителей. Для обозрения представлены 5 картин. Посетитель ходит от картины к картине, и если на картину любуются более чем десять посетителей, он стоит в стороне и ждет, пока число желающих увидеть картину не станет меньше. Посетитель может покинуть галерею. Создать многопоточное приложение, моделирующее работу картинной галереи.

2. Компиляция и запуск программы

```
g++ -pthread -std=c++17 -o gallery gallery.cpp  
./gallery
```

3. Содержание архива

Файл **gallery.cpp** – исходные тексты программы

Файл **gallery** – исполняемый файл

Файл **отчет.pdf** - этот файл

4. Интерпретация задания

Изначально в картинную галерею заходят 50 человек. Далее все происходит по циклам. В каждом цикле каждый посетитель пытается подойти к какой-то из картин (попытки подойти происходят по очереди). Только после двух циклов посетители начинают расходиться – кто-то уходит сразу, кто-то остается.

- Возможно, у картины уже будут стоять 10 человек, тогда у посетителя не получится к ней подойти, и он останется ждать следующего цикла у текущей картины.
- Возможно, посетитель захочет остаться у текущей картины.
- Возможно, на первом цикле окажутся посетители, которые не смогли подойти ни к одной из картин.

Программа заканчивается, когда все посетители покинули картинную галерею. Переходы посетителей генерируется рандомным образом по нормальному распределению и с использованием вихря Мерсенна.

5. Настройка программы

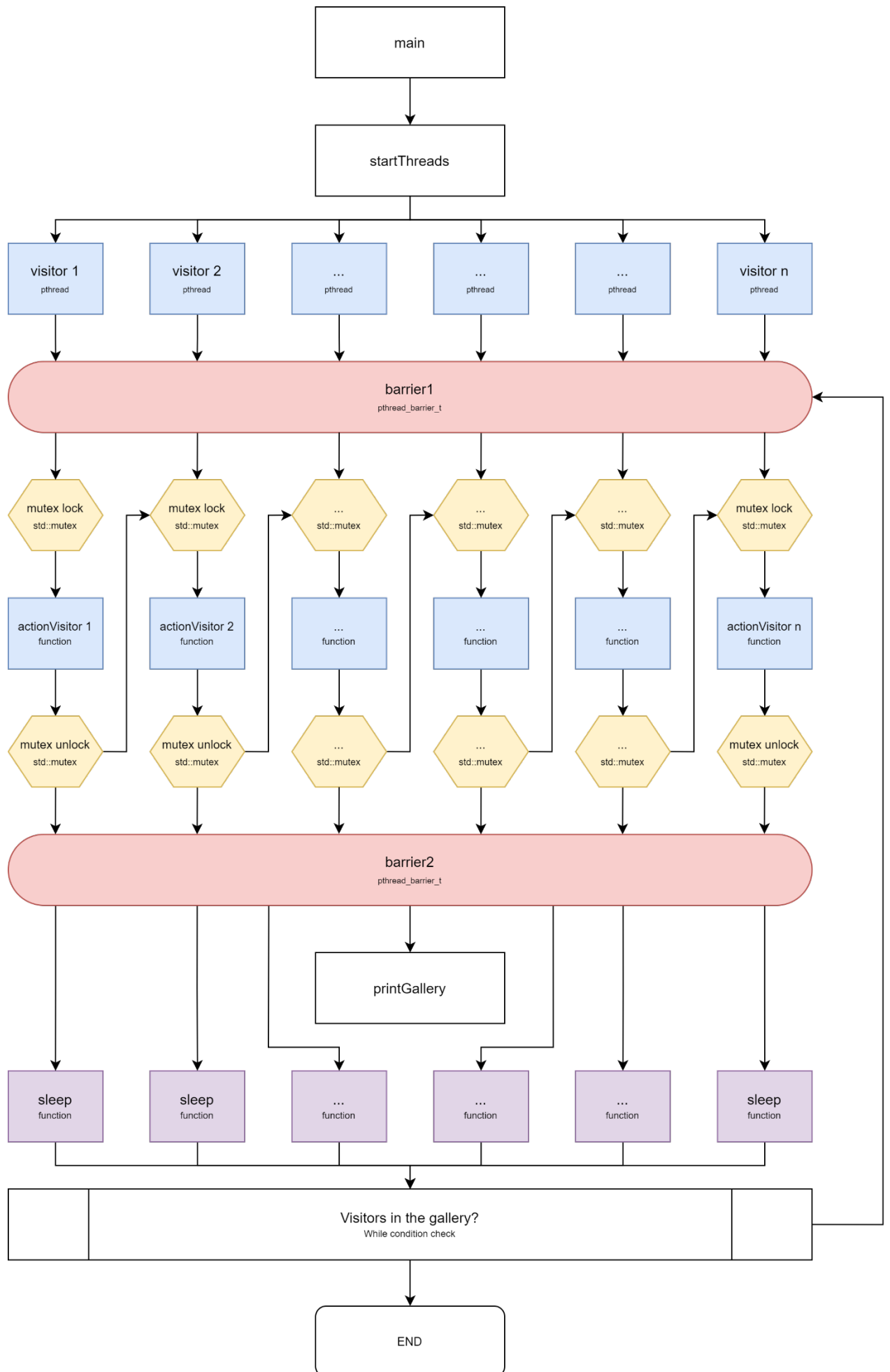
Константы настраиваются для моделирования различных ситуаций посещаемости картинной галереи. Менять их можно в исходном файле .cpp, после чего заново скомпилировать исполняемый файл. Далее подробно про каждую из констант.

- **artworks** – количество экспозиций выставленных в картинной галерее
- **visitors** – изначальное количество посетителей картинной галереи
- **maximum_viewers** – максимальное количество наблюдателей за одной экспозицией
- **sleep_seconds** – технический параметр программы, обозначающий количество секунд перерыва между циклами для удобства наблюдения за выводом программы
- **probability_to_leave** – вероятность, что посетитель уйдет на текущем цикле, варьируется от 0 до 100, обозначает процент вероятности.
- **dont_leave_for_cycles** – количество первых циклов, во время которых ни один посетитель не уйдет из галереи
- **threads** – количество создаваемых потоков, в данной реализации всегда равно количеству посетителей. **НЕ стоит менять данный параметр**, это повлечет за собой некорректную работу программы. Вынесен в блок с константами для удобства написания программы.

```
const size_t artworks = 9;           // 9
const size_t visitors = 50;          // 50
const size_t maximum_viewers = 10;   // 10
const size_t sleep_seconds = 1;      // 1
const size_t probability_to_leave = 20; // 20
const size_t dont_leave_for_cycles = 2; // 2
const size_t threads = visitors;
```

6. Модель многопоточной программы

Каждый посетитель является отдельным потоком. На каждом цикле все посетители синхронизируется два раза с помощью `pthread_barrier_t`: перед выполнением действий и перед выводом финальной информации по циклу. Также перед перемещением пользователя между картинами выполняется блокировка массива данных с помощью `mutex lock`. Все потоки остаются активными, пока в галерее есть хотя бы один посетитель. Подробнее рассмотреть построенную модель можно на блок-схеме.



7. Источники информации, использованные в модели

- [Запуск нескольких потоков](#)
 - [Синхронизация с помощью барьеров](#)
 - [Блокировка с помощью mutex](#)
 - [Блокировка с помощью mutex 2](#)
 - [Получение id текущего потока](#)
-
- [Многопоточность в C++](#)
 - [Многопоточность в C++ 2](#)
 - [Компиляция pthread в linux](#)