

Отчет о написании программы

Описание полученного задания

Условие из ТЗ:

18. Задача о болтунах. N болтунов имеют телефоны, ждут звонков и звонят друг другу, чтобы побеседовать. Если телефон занят, болтун будет звонить, пока ему кто-нибудь не ответит. Побеседовав, болтун не унимается и или ждет звонка или звонит на другой номер. Создать многопоточное приложение, моделирующее поведение болтунов. Для решения задачи использовать **мьютексы**.

Дополнение разработчика:

У каждого болтуна есть одинаковые городские телефонные справочники 2010го года (ведь после этого года ими никто больше не пользовался 😞), в котором сдержатся N ($N \geq 2$) номеров. Болтунам скучно, и чтобы развлечься, они звонят по номерам из их книг от 0 до N. Конечно, каждый из них знает свой собственный номер телефона и не будет звонить сам себе. Болтун не может одновременно разговаривать с несколькими людьми. Разговор двух болтунов длится 1 секунду. Работа приложения заканчивается, когда каждый болтун обзвонил всех человек из телефонного справочника.

Модель

Взаимодействующие равные – модель, в которой исключен не занимающийся непосредственными вычислениями управляющий поток. Распределение работ в таком приложении либо фиксировано заранее, либо динамически определяется во время выполнения. Одним из распространенных способов динамического распределения работ является «портфель задач». Портфель задач, как правило, реализуется с помощью разделяемой переменной, доступ к которой в один момент времени имеет только один процесс.

Используется именно эта модель, поскольку в задаче каждый болтун выполняет свою задачу без дальнейшей обработки другими болтунами, то есть их работа не зависит от других болтунов. Болтун – самостоятельная единица задачи, поведение которой влияет только на возможность связаться с ней или с её собеседником. Для реализации использовались мьютексы и геттер, чтобы не было смешения и «гонок» потоков.

Общие сведения

- Команды:
 - `<int>` - работа с указанным N количеством болтунов;
 - `<отсутствие параметров>` - работа со сгенерированным количеством.
- Количество строк «звонков» для N болтунов равно $N*(N-1)$.
- Обработка ошибок
 - При некорректном вводе параметров программа сообщает об этом в консоли и завершает работу.

Спецификации разработки

- Операционная система: Ubuntu (64-bit)
- Архитектура: x86-64
- RAM: 15GB
- Средство сборки – CLion(v2021.2.2)
- Примененные библиотеки на C++
 - `iostream`
 - `stdlib`
 - `utility`
 - `thread`
 - `mutex`
 - `vector`
 - `ctime`

Структура проекта

- `main.cpp` – точка входа;
- `Caller.cpp` – файл с описанием болтуна из условия задачи;
- `Caller.h` – заголовочный файл с описанием болтуна из условия задачи;
- `CMakeLists.txt` – набор инструкций для работы с программой

Характеристики проекта

- Количество программных файлов (`.cpp/.h` включая `main` класс) – 3
- Размер используемых файлов (`.cpp/.h`) - ~8.0Кб
- Размер проекта - ~ 4.30Мб
- Время работы для N болтунов (рассматривается среднее значение) :

• 10 элементов	• 9.124 с
• 100 элементов	• 100.412 с
• 1000 элементов	• 1014.66 с
• 10000 элементов	• Более 167 минут

Дополнительные характеристики:

- Модульная структура
- Обработка ошибок при работе с параметрами
- Документация кода
- Возможность генерации данных (количества болтунов)
- Определение времени выполнения программы