ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной

инженерии

КОНСОЛЬНОЕ ПРИЛОЖЕНИЕ МОДЕЛИРУЮЩЕЕ ПОВЕДЕНИЕ КУРИЛЬЩИКОВ И ПОСРЕДНИКА

Вариант № 6

Пояснительная записка

Исполнитель
Студент группы БПИ199
_____/Галанов А. С./
«___»_____2020 г.

Москва 2020

Оглавление

Текст задния

Есть три процесса-курильщика и один процесс-посредник. Курильщик непрерывно скручивает сигареты и курит их. Чтобы скрутить сигарету, нужны табак, бумага и спички. У одного процесса курильщика есть табак, у второго – бумага, а у третьего – спички. Посредник кладет на стол по два разных случайных компонента. Тот процесс курильщик, у которого есть третий компонент, забирает компоненты со стола, скручивает сигарету и курит. Посредник дожидается, пока курильщик закончит, затем процесс повторяется. Создать многопоточное приложение, моделирующее поведение курильщиков и посредника. При решении задачи использовать семафоры.

Описание программы

- **Сценарий взаимодействия курильщиков и посредника**
- Программа начинается с того, что посредник достает случайным образом 2 компонента, относительно которых и будет выбран курильщик, кто начнет курить сигарету.
- После того как посредник положил компоненты на стол соответствующий курильщик берет их со стола и начинает курить. Сам же посредник начинает ждать курильщика.
- Когда курильщик докурил сигарету он кричит что хочет еще, данный крик понимает только посредник. После крика курильщик начинает ждать нужные ему компоненты на столе.
- Из – за крика посредник просыпается и начинает по новой искать следующие компоненты.
- **Протокол взаимодействия курильщиков и посредника**

1. Программа начинается с того, что инициализируются потоки-курильщики (используется стандартная библиотека thread[2]) со своими номерами, именами и компонентом, который у них есть изначально. Для вывода курильщика используется следующий формат: “Курильщик-{номер курильщика}({имя курильщика})”. После инициализации курильщиков, они начинают ожидать нужные им компоненты на столе для сигареты. Момент инициализации характеризуется строкой: “И так давайте покурим! - говорят курильщики”
2. Далее посредник случайным образом выбирает, случайные компоненты, которые кладет на стол (посредник у объекта стола устанавливает соответствующие поля-компоненты). Данное событие характеризуется строкой: “Посредник положил на стол {компонент a} и {компонент b} и решил чуть-чуть вздремнуть”.
3. После того как посредник положил компоненты на стол он начинает ждать, когда курильщик, кто их забрал, закончит курить сигарету. (Реализовано с помощью собственного класса семафора[3][4]. Пока семафор равен 0, то посредник ждет, иначе продолжить).

4. Каждый курильщик в самом начале проверяет есть ли нужные ему компоненты для сигареты (Курильщик в бесконечном цикле обращается к объекту стола, чтобы разузнать есть ли компоненты на столе):
    a. Если их нет, то курильщик ждет, повторяя свои обращения.
    b. Если есть, то курильщик забирает компоненты со стола (изменяя соответствующие поля-компоненты у объекта стола) и начинает делать, а потом курить саму сигарету. Данное событие характеризуется строкой: “Курильщик-{номер курильщика}({имя курильщика}) забрал {компонент a} {компонент b} со стола, сделал сигарету и начал курить”
5. Когда курильщик докуривает сигарету он кричит, от чего просыпается посредник (Курильщик изменяет семафор, таким образом сообщая посреднику, что он закончил курить). После крика курильщик возвращается на шаг 4. Данное событие характеризуется строкой: “Курильщик-{номер курильщика} ({Имя курильщика}) докурил сигарету и начал ждать”.
6. В момент крика просыпается посредник (так как семафор изменился, посредник продолжает работу) и начинает по-новому процесс. (процесс взаимодействия курильщиков и посредника реализован в цикле, количество повторений которого задает пользователь одним из параметров для программы). Данное событие характеризуется строкой: “Посредник вздрогнул и быстро сунул руки в карманы”.
7. Процесс взаимодействия по новой начинается с шага 2.
8. После окончания процесса взаимодействия, очищаются потоки курильщики, а также и семафор. Данное событие характеризуется строкой: “У посредника кончились компоненты для сигарет, и он ушел в магазин…”

- **Описание входных данных**
    - **Формат входных данных для консоли**

В качестве входных данных пользователь в командной строке указывает следующие аргументы:

1. Количество повторений процесса курения.

2. Полный путь к файлу для отчета.

Параметрическая строка имеет вид: **“{первый аргумент} {второй аргумент}”**
Пример: **“10 C:\MyFiles\report.txt”**

- **Требования к входным данным**

1. Количество повторений процесса курения может быть числом, лежащим в следующем отрезке [1;1000].
2. Путь к файлу не должен содержать пробелы.

- **Описание выходных данных**

В качестве выходных данных создается текстовый документ, на каждой строке которого отражено соответствующее событие, произошедшее по мере выполнения программы.

Пояснительная записка написана на основании инструкции [1].

Список использованной литературы

1. Инструкция по составлению пояснительной записки [Электронный ресурс]. //URL: http://www.softcraft.ru/edu/comparch/tasks/mp02/ (Дата обращения: 8.12.2020, режим доступа: свободный)
2. Документация к библиотеке thread записки [Электронный ресурс]. //URL: https://en.cppreference.com/w/cpp/thread (Дата обращения: 10.12.2020, режим доступа: свободный)
3. Обучающая статья по семафорам [Электронный ресурс]. //URL: https://habr.com/ru/post/476940/ (Дата обращения: 10.12.2020, режим доступа: свободный)
4. Пример программы с использованием семафора [Электронный ресурс]. //URL: https://github.com/andron23/Semaphor (Дата обращения: 10.12.2020, режим доступа: свободный)