Национальный исследовательский университет «Высшая школа экономики»

Домашнее задание №5

Архитектура вычислительных систем

Пояснительная записка

Исполнитель

студент группы БПИ202

Галюта Ева Валерьевна

2021 г.

Задание

Вариант: 6

Задача о курильщиках. (С++)

1. Есть три процесса-курильщика и один процесс посредник.
2. 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.
3. 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.
4. Создать многопоточное приложение, моделирующее поведение курильщиков и посредника.
5. При решении задачи использованы семафоры. (описание приведено на 3 листе)
6. Используется модель производители и потребители. (описание приведено на 4 листе)
7. Формат ввода (ограничения по кол-ву запросов ( 1 <= x <= 30) - важна корректность работы и реализация задачи, а не скорость, ограничения сделаны для того, чтобы пользователь не ждал долго пока программа проработает)

* ./threads\_hw5 -f <input.txt> - ввод из файла

input file:

1. Первая строка: кол-во итераций программы (сколько раз посредник даст материалы)
2. Далее: «a b», где a и b номера элементов (1- табак, 2 – бумага, 3 - спички)

* ./threads\_hw5 -n <number of iterations> - ввод с использованием рандомного генератора

1. Компиляция С++

* просьба использовать 11 и выше версию стандарта языка, а также (-phtread ), чтобы подключить pthreads библиотеку
* пример (g++): $ g++ -std=c++17 -pthread main.cpp

Семафор

Семафор - примитив синхронизации работы процессов и потоков, в основе которого лежит счётчик, над которым можно делать операции: увеличение и уменьшение значения на единицу, при этом операция уменьшения для нулевого значения счётчика представляется блокирующейся.

Используется для синхронизации параллельно функционирующих задач, для защиты передачи данных через разделяемую память, для защиты критических секций, а также для управления доступом к аппаратному обеспечению.

Чуть проще говоря, семафор помогает ограничить доступ потоков к объекту синхронизации по их количеству.

Первоначально семафор инициализируется и ему передается количество потоков, которые к нему могут обратиться. Дальше при любом обращении к ресурсу его счетчик уменьшается. Когда счетчик понизится до 0 к ресурсу обратиться уже нельзя. При отсоединении потока от семафора его счетчик увеличивается, что позволяет иным потокам обратиться к нему. Когда счетчик равен 0, семафор считается сброшенным.

Источники:

http://netcode.ru/cpp/?artID=2356 (кратко, наглядно)

https://learnc.info/c/pthreads\_semaphores.html (хороший ресурс с примерами)

http://www.codenet.ru/progr/cpp/7/3.php (один из лучших ресурсов по объяснениям)

https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D1%84%D0%BE%D1%80\_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) (самое полное определение семафора)

Модель производители и потребители.

Производители и потребители – это парадигма взаимодействующих неравноправных потоков. Одни потоки «производят» данные, другие их «потребляют». Часто такие потоки организуются в конвейер, через который проходит информация. Каждый поток конвейера потребляет выход своего предшественника и производит входные данные для своего последователя.

Производители — это потоки, производящие данные, потребители — это потоки, потребляющие данные. Если производитель быстро обрабатывает данные, но скорость обработки потребителя низкая, то производитель должен обязательно дождаться (что очень важно), пока потребитель закончит обработку, перед тем как продолжить отдавать данные.

Важно:

Основа модели в ожидании потоков, производители ожидают окончания работы потребителя.

Источники:

https://russianblogs.com/article/1939946469/

https://www.codetd.com/ru/article/10284797

http://vsyromyatnikov.blogspot.com/2014/11/blog-post\_21.html

Характеристики программы

Интерфейсные модули: 0

Модули реализации: 1

Общий размер исходных тестов: 6.327 Кб

Общий размер результатов тестов: 0 Кб (вывод в консоль)

Размер исполняемого кода: 5.964 Кб

Результаты тестов (в секундах)

|  |  |  |
| --- | --- | --- |
| Количество элементов | Ввод из файла | Генерация элементов |
| 5 (test\_1\_5.txt) | 35.009 | 35.021 |
| 10 (test\_2\_10.txt) | 70.018 | 70.058 |
| 20 (test\_3\_15.txt) | 105.025 | 105.073 |
| 20 (test\_4\_20.txt) | 140.022 | 133.135 |
| 30 (test\_5\_30.txt) | 210.034 | 210.202 |

Отмечу, что данная таблица приведена ради интереса, так как в программе присутствует  sleep() для моделирования условий задачи, что в свою очередь влияет на время работы. Тем более данный код написан не для измерения скорости, а для моделирования работы потоков приложения по условиям задачи.

Заключение

Разработали консольное приложение на языке С++ с использованием библиотеки POSIX Threads языка программирования C. Разобрались в идее семафоров, а также в модели «производители и потребители».