Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Департамент программной инженерии

Отчет к домашнему заданию По дисциплине «Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-195 Гуницкий Р.Я.

Москва 2020

ОГЛАВЛЕНИЕ

1. УСЛОВИЕ

Задача о каннибалах. Племя из n дикарей ест вместе из большого горшка, который вмещает m кусков тушеного миссионера. Когда дикарь хочет обедать, он ест из горшка один кусок, если только горшок не пуст, иначе дикарь будит повара и ждет, пока тот не наполнит горшок. Повар, сварив обед, засыпает. Создать многопоточное приложение, моделирующее обед дикарей. При решении задачи пользоваться семафорами.

2. РЕШЕНИЕ

Перед началом работы программы пользователь вводит N - количество дикарей в племени и M - количество кусков тушеного миссионера, которое вмещает в себя горшок. В цикле создается массив потоков имитирующих действия дикарей и создается один поток, имитирующий работу повара. Входя в свою пещеру (занимает от 0 до 5 секунд) дикарь начинает искать себе еду в горшке (в консоль выводится сообщение «[Savage {{id}}] Looking for his portion») и если в горшке есть порция для него, то он берет свою порцию (уменьшает значение семафора через функцию **sem_wait()**) и начинает трапезу (сопровождается сообщением в консоль «[Savage {{id}}] Eating»). Дикарь может есть от 1 до 3 секунд, после чего он заканчивает трапезу и выходит из пещеры (в консоль выводится сообщение «[Savage {{id}}] Finished eating»).

В случае если поиск еды в горшке не дал результатов и на данный момент времени дикарь единственный, кто не ест, то он идет к повару и начинает его будить, активируя условную переменную cv (действие сопровождается сообщением в консоль «[Savage {{id}}] Wakes up the cook»). Как только повар просыпается, он сразу уведомляет пользователя об этом сообщением «[Cook] I woke up» и начинает готовить еду. Через некоторое время (от 0,5 до 1,5 секунд) повар наполняет горшок полностью и говорит об этом пользователю, выводя сообщение в консоль «[Cook] Prepared {{ countOfPortions }} more portions» и используя функцию **sem_post_multiple()** увеличивает значение семафора на countOfPortions и идет спать, выводя сообщение «[Cook] I gonna sleep».

В случае если поиск еды не дал результатов и дикарь не единственный, кто остался на некоторое время без еды, то он просто ждет пока вопрос с едой решится, выводя при этом сообщение в консоль «[Savage {{id}}] Wait food». Как только повар наполняет горшок новой едой, дикарь либо начинает есть, либо, если ему так и не хватило еды, будит повара еще раз и снова ждет приготовления новых порций.

Как только все дикари закончили свою трапезу, в консоль выводится сообщение «{{N}} savage finished eating», чтобы убедиться в том, что все дикари поели.

3. ТЕСТИРОВАНИЕ

3.1. Ввод некорректных данных

```
Enter count of savages:0
Incorrect input!
Enter number again:30000
Incorrect input!
Enter number again:-1
Incorrect input!
Enter number again:100
Enter start count of portions:-10
Incorrect input!
Enter number again:0
Incorrect input!
Enter number again:3000
Incorrect input!
Enter number again:20
[Savage 98]  Looking for his portion
[Savage 98]     Eating
[Savage 49]     Looking for his portion
[Savage 49]     Eating
[Savage 82]     Looking for his portion
[Savage 82]     Eating
[Savage 33]     Looking for his portion
[Savage 33]     Eating
[Savage 64]     Looking for his portion
[Savage 64]     Eating
[Savage 15]     Looking for his portion
[Savage 15]     Eating
[Savage 97]     Looking for his portion
[Savage 97]     Eating
[Savage 48]     Looking for his portion
```

Рисунок 1. – Обработка некорректных входных данных.

3.2. Ввод корректных входных данных

```
Enter count of savages:12
Enter start count of portions:5
[Savage 12]        Looking for his portion
[Savage 12]     Eating
[Savage 10]     Looking for his portion
[Savage 10]     Eating
[Savage 7]      Looking for his portion
[Savage 7]      Eating
[Savage 12]     Finished eating
[Savage 11]     Looking for his portion
[Savage 11]     Eating
[Savage 8]      Looking for his portion
[Savage 8]      Eating
```

Рисунок 2. – Пришедшие дикари съедают всю еду.

```
[Savage 5]      Looking for his portion
[Savage 5]      I don't have my food...
[Savage 5]      Wakes up the cook
[Cook]          I woke up
[Savage 9]      Looking for his portion
[Savage 9]      I don't have my food...
[Savage 9]      Wait food
[Savage 6]      Looking for his portion
[Savage 6]      I don't have my food...
[Savage 6]      Wait food
[Savage 3]      Looking for his portion
[Savage 3]      I don't have my food...
[Savage 3]      Wait food
```

Рисунок 3. – Первый дикарь без еды будит повара, все остальные ждут.

```
[Cook]          Prepared 5 more portions
[Savage 9]      Eating
[Savage 3]      Eating
[Savage 5]      Eating
[Savage 6]      Eating
[Cook]          I gonna sleep
[Savage 4]      Looking for his portion
[Savage 4]      Eating
```

Рисунок 4. – Повар заполнил горшок едой и ушел спать, дикари едят новую еду.

```
[Savage 1]      Looking for his portion
[Savage 1]      I don't have my food...
[Savage 1]      Wakes up the cook
[Cook]          I woke up
[Savage 8]      Finished eating
[Savage 11]     Finished eating
[Savage 7]      Finished eating
[Savage 10]     Finished eating
[Savage 2]      Looking for his portion
[Savage 2]      I don't have my food...
[Savage 2]      Wait food
[Cook]          Prepared 5 more portions
[Cook]          I gonna sleep
[Savage 1]      Eating
[Savage 2]      Eating
[Savage 5]      Finished eating
[Savage 3]      Finished eating
[Savage 6]      Finished eating
[Savage 9]      Finished eating
[Savage 4]      Finished eating
[Savage 2]      Finished eating
[Savage 1]      Finished eating
12 savage finished eating
Process finished with exit code 0
```

Рисунок 5. – Повтор алгоритма, вывод количества накормленных дикарей и завершение программы.

4. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

5. Cppreference (2020) «Документация по C++: std::condition_variable::wait» (https://en.cppreference.com/w/cpp/thread/condition_variable/wait).
6. Cppreference (2020) «Документация по C++: std::mutex» (https://ru.cppreference.com/w/cpp/thread/mutex).
7. Docs Microsoft (2020) «Creating Threads» (https://docs.microsoft.com/en-us/windows/win32/procthread/creating-threads)
8. Легалов А.И.(2020) «Архитектура параллельных вычислительных систем. Многопоточность » (http://softcraft.ru/edu/comparch/lect/07-parthread/)
9. Легалов А.И.(2020) «Многопоточность. Простая многопоточная программа. Основные функции» (http://softcraft.ru/edu/comparch/practice/thread/01-simple/).
10. Легалов А.И.(2020) «Многопоточность. Синхронизация потоков. Методы синхронизации» (http://softcraft.ru/edu/comparch/practice/thread/02-sync/).
11. Хабр «Такие удивительные семафоры» (https://habr.com/ru/post/261273/)
12. Learn.info «Семафоры: введение» (https://learnc.info/c/pthreads_semaphores.html)