

**Микропроект №2.**  
**Задача о каннибалах**

Ястребов Игорь Андреевич

НИУ ВШЭ

Факультет Компьютерных наук

Группа БПИ197

## Текст задания

### Вариант 5

*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.

## Решение задачи

В решении задачи использовались семафоры. Когда каннибал хочет есть, используется метод `wait()`. Он ждет, когда поле `v` будет больше 0, после этого декрементирует `v`. Каждый раз, когда дикарь хочет есть, также проверяется, есть ли куски мяса в горшке. Если нет, то вызывается метод `cook()`. При вызове этого метода  $M$  раз вызывается метод `post()` семафора, который инкрементирует поле `v` семафора.

# Тесты задачи

Скриншот на вызове программы ./a 10 12 (не будет необходимости готовить дополнительные куски мяса):

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1256]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Lenovo\Desktop\assembly\homework\assemblyHomework\microProject2>a 10 12
The cooking has started
Done cooking    12
Cannibal number 0 has started eating
Cannibal number 0 has finished eating
Cannibal number 1 has started eating
Cannibal number 1 has finished eating
Cannibal number 2 has started eating
Cannibal number 2 has finished eating
Cannibal number 3 has started eating
Cannibal number 3 has finished eating
Cannibal number 4 has started eating
Cannibal number 4 has finished eating
Cannibal number 5 has started eating
Cannibal number 5 has finished eating
Cannibal number 6 has started eating
Cannibal number 6 has finished eating
Cannibal number 7 has started eating
Cannibal number 7 has finished eating
Cannibal number 8 has started eating
Cannibal number 8 has finished eating
Cannibal number 9 has started eating
Cannibal number 9 has finished eating

C:\Users\Lenovo\Desktop\assembly\homework\assemblyHomework\microProject2>
```

Скриншот на вызове программы ./a 10 3 (придется дополнительно готовить мясо):

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18363.1256]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Lenovo\Desktop\assembly\homework\assemblyHomework\microProject2>a 10 3
The cooking has started
Done cooking    3
Cannibal number 0 has started eating
Cannibal number 0 has finished eating
Cannibal number 1 has started eating
Cannibal number 1 has finished eating
Cannibal number 2 has started eating
Cannibal number 2 has finished eating
The cooking has started
Done cooking    3
Cannibal number 3 has started eating
Cannibal number 3 has finished eating
Cannibal number 4 has started eating
Cannibal number 4 has finished eating
Cannibal number 5 has started eating
Cannibal number 5 has finished eating
The cooking has started
Done cooking    3
Cannibal number 6 has started eating
Cannibal number 6 has finished eating
Cannibal number 7 has started eating
Cannibal number 7 has finished eating
Cannibal number 8 has started eating
Cannibal number 8 has finished eating
The cooking has started
Done cooking    3
Cannibal number 9 has started eating
Cannibal number 9 has finished eating

C:\Users\Lenovo\Desktop\assembly\homework\assemblyHomework\microProject2>
```

# Текст программы

```
#include <iostream>
#include <vector>
#include <string>
#include <thread>
#include <mutex>
#include <chrono>
using namespace std;

int N;
int M;

class semaphore {
private:
    mutex m;
    int v;
public:
    int getV() { return v; }
    semaphore(int val = 0) {
        v = val;
    }
    void post() {
        m.lock();
        v++;
        m.unlock();
    }
    void wait() {
        while (1) {
            m.lock();
            if (v > 0) {
                --v;
                m.unlock();
                return;
            }
            m.unlock();
        }
    }
    bool isEmpty() { return v == 0; }
};

bool check(string line) {
    for (int i = 0; i < line.length(); i++)
        if (line[i] < '0' || line[i] > '9')
            return false;
    return true;
}

semaphore s(0);
```

```

void cook() {
    cout << "The cooking has started" << endl;
    for (int i = 0; i < M; i++)
        s.post();
    cout << "Done cooking\t" << s.getV() << endl;
}

void eat(int i) {
    if (s.isEmpty())
        cook();
    cout << "Cannbial number " + to_string(i) + " has started eating\t" << endl;
    s.wait();
    cout << "Cannbial number " + to_string(i) + " has finished eating\t" << endl;
}

int main(int argc, char* argv[]) {
    if (argc != 3) {
        cout << "The first argument has to be N, the second one has to be M" << endl;
        return 0;
    }

    if (!check(argv[1]) || !check(argv[2])) {
        cout << "N and M have to be natural numbers" << endl;
        return 0;
    }

    N = stoi(argv[1]);
    M = stoi(argv[2]);
    for (int i = 0; i < N; i++) {
        thread t(eat, i);
        t.join();
    }
    return 0;
}

```