

Операционные системы
ИДЗ 2. Отчёт
Вариант 35
Работа на 7 баллов

Фролов-Буканов Виктор Дмитриевич БПИ-228

21 апреля 2024

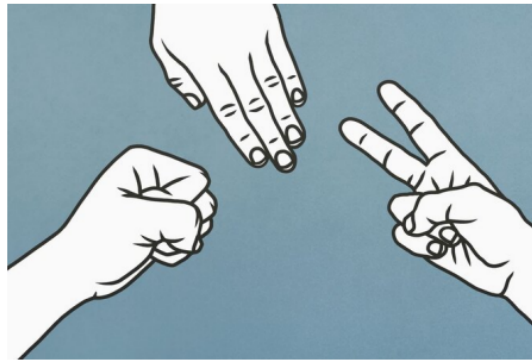
1 Условие задачи

35. «Камень, ножницы, бумага» 1 — однокруговая система.

N студентов, изнывающих от скуки на лекции по операционным системам решили организовать однокруговой турнир в игру «Камень, ножницы, бумага». Каждый с каждым при этом играет только один раз. За победу дается два очка, за ничью — одно, за поражение — ноль.

Требуется создать многопроцессное приложение, моделирующее турнир.

Каждый студент — отдельный процесс. Генерация камня, ножниц и бумаги в каждом процессе формируется случайно.



2 Работа с командной строкой / запуск программы

Все программы написаны на языке C, так что их компиляция осуществляется с помощью команды `gcc -o <имя_выходного_файла> <имя_исходного_файла>`. Каждая программа в командной строке принимает ровно один параметр - целое положительное число, задающее число процессов (то есть число игроков). В случае невалидного ввода (если введено отрицательное число или ноль, а также если введена строка вместо числа), то программа закончит свое выполнение досрочно, уведомив об этом пользователя следующей строкой

```
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps f
Invalid input
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps -9
Invalid input
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps 0
Invalid input
```

3 Схема решения задачи на 4-5 баллов

Синхронизация процессов осуществляется с использованием именованных семафоров. Программа декомпозирована на 2 header-файла (*types.h* с определением типов данных, используемых в программе (Game, Player), а также *utilities.h* с определениями вспомогательных функций). При запуске программы основной поток создаёт п дочерних потоков, являющихся игроками, при этом порождающий поток назовём судьёй, так как он составляет турнирную сетку и назначает матчи, а также подводит подсчёт очков. Перед началом турнира надо дождаться, пока все игроки вступят в игру. Это достигается посредством использования семафоров *sem_player* и *sem_judge*. Процессы не продолжают свое выполнение, пока значение *sem_player* не установится в 0, что достигается только при условии, когда все игроки вступили в турнир. Далее процесс-судья должен сформировать список участников и турнирную сетку, после чего начнутся непосредственно сами игры. Для отслеживания этого состояния используется семафор *sem_judge*. Дочерние процессы будут ждать, пока значение этого семафора не установится в ноль (эта логика достигается посредством использования метода *void wait(sem_t *sem, int value)*, который блокирует выполнение процесса до тех пор, пока переданный семафор не примет значение value. Как только семафор *sem_judge* устанавливается в ноль, начинаются игры, которые контролирует процесс-судья. Открытая игра обозначается состоянием семафора *sem_match* 1, закрытая - 0. В зависимости от состояния игры и запущенных игроков, выполняется нужный участок кода в нужных дочерних процессах, обеспечивающих верную эмуляцию турнирного процесса.

4 Схема решения задачи на 6-7 баллов

Аналогична схеме решения на 4-5, за исключение того, что теперь для синхронизации работы процессов используются неименованные семафоры

5 Результаты работы программы на 4-7 баллов

Так как условие в реализациях на 4-5 и на 6-7 отличаются лишь типом используемых семафоров, а сам алгоритм решения задачи и структура программы остаются неизменной, то результат работы программы приведу один на эти два пункта

```
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps 2
Player with id 1 joined the game
Player with id 2 joined the game
All players ready, the tournament is starting...

Tournament grid:
1 vs 2

[Match] 1 vs 2
Player 1 throws paper
Player 2 throws paper
    Player 1 finishes the tournament
    Player 2 finishes the tournament
[Draw] (paper vs paper) -> player 1 vs player 2
Tournament is over
Results:
Player 1 earned 1 points
Player 2 earned 1 points
```

```
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps 2
Player with id 1 joined the game
Player with id 2 joined the game
All players ready, the tournament is starting...

Tournament grid:
1 vs 2

[Match] 1 vs 2
Player 1 throws rock
Player 2 throws rock
    Player 1 finishes the tournament
    Player 2 finishes the tournament
[Draw] (rock vs rock) -> player 1 vs player 2
Tournament is over
Results:
Player 1 earned 1 points
Player 2 earned 1 points
```

Пример прерывания до завершения программы (досрочное завершение эмуляции турнира)

```
frolovbuk@LAPTOP-K0RFAB50:~/ihws/ihw2/5points$ ./rps 3
Player with id 1 joined the game
Player with id 2 joined the game
Player with id 3 joined the game
All players ready, the tournament is starting...

Tournament grid:
1 vs 2
1 vs 3
2 vs 3

[Match] 1 vs 2
Player 1 throws scissors
Player 2 throws scissors
[Win] (scissors vs paper) -> player 2 vs player 1
[Match] 1 vs 3
Player 1 throws paper
Player 3 throws scissors
    Player 1 finishes the tournament
[Win] (scissors vs paper) -> player 3 vs player 1
[Match] 2 vs 3
Player 3 throws paper
Player 2 throws paper
    Player 3 finishes the tournament
    Player 2 finishes the tournament
[Draw] (paper vs paper) -> player 2 vs player 3
Tournament is over
Results:
Player 1 earned 0 points
Player 2 earned 3 points
Player 3 earned 3 points
```