Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

Студент: Меркулов Ф. А.

Группа: М8О-207Б-21

Вариант: 7

Преподаватель: Черемисинов Максим Леонидович

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

https://github.com/WhatTheMUCK/OSi

**Постановка задачи**

**Цель работы**

Целью является приобретение практических навыков в:

* Управление потоками в ОС
* Обеспечение синхронизации между потоками

**Задание**

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы.

Так же необходимо уметь продемонстрировать количество потоков, используемое вашей

программой с помощью стандартных средств операционной системы.

В отчете привести исследование зависимости ускорения и эффективности алгоритма от входящих

данных и количества потоков. Получившиеся результаты необходимо объяснить.

7 вариант) Два человека играют в кости. Правила игры следующие: каждый игрок делает бросок 2-ух костей K раз; побеждает тот, кто выбросил суммарно большее количество очков. Задача

программы экспериментально определить шансы на победу каждого из игроков. На вход

программе подается K, какой сейчас тур, сколько очков суммарно у каждого из игроков и

количество экспериментов, которые должна произвести программа

**Общие сведения о программе**

Программа компилируется из файла main.cpp. Также используется заголовочные файлы: stdio.h, stdlib.h, iostream, pthread.h

В программе используются следующие системные вызовы:

1. pthread\_create() – создаёт новый поток
2. pthread\_join(pthread\_t THREAD\_ID, void \*\* DATA) - ожидает завершения потока обозначенного THREAD\_ID

**Общий метод и алгоритм решения**

С помощью ключа запуска передаётся необходимое количество потоков. С консоли программа считывает: количество очков первого игрока, количество очков второго игрока, тур, количество бросков костей и общее количество экспериментов. На каждый поток передаётся количество экспериментов равное [общее количество экспериментов / количество потоков], но так как количество экспериментов на потоках – целое число, то последний поток обрабатывает [(общее количество экспериментов / количество потоков) + (общее количество экспериментов % количество потоков)]. На каждом потоке происходит экспериментальный подсчёт количества побед первого и второго игроков, с помощью потокобезопасной функции rand\_r(&seed), где seed – id потока. То есть для каждого отдельного эксперимента, для каждого игрока программа моделирую бросок 2 кубиков заданное количество раз и выясняет кто победил в этом эксперименте: первый или второй игрок. Так все потоки подсчитывают общее количество побед первого и второго игроков за общее количество экспериментов и для нахождения вероятности победы, мы просто делим число побед игрока на общее число экспериментов.

**Исходный код**

#include <stdio.h>

#include <stdlib.h>

#include <iostream>

#include <pthread.h>

//#include <time.h>

using namespace std;

typedef struct thread\_data{

int counter\_first;

int counter\_second;

int plays;

int tour;

int experements;

int wins\_first;

int wins\_second;

} thread\_data;

void\* thread\_func(void \*arg){

thread\_data \*tdata = (thread\_data \*)arg;

unsigned int seed;

seed = pthread\_self();

int first;

int second;

for (int counter1 = 0; counter1 < tdata->experements; counter1++){

first = tdata->counter\_first;

second = tdata->counter\_second;

for (int counter2 = 0; counter2 < tdata->plays - tdata->tour + 1; counter2++){

first += rand\_r(&seed)%6 + 1;

first += rand\_r(&seed)%6 + 1;

second += rand\_r(&seed)%6 + 1;

second += rand\_r(&seed)%6 + 1;

}

if (first > second)

tdata->wins\_first++;

if (second > first)

tdata->wins\_second++;

}

return 0;

}

int main(int argc, char \*argv[]){

if (argc != 2){

cerr << "error: Program must have only 1 key\n";

} else {

//clock\_t start\_time, end\_time;

//float timer;

//start\_time = clock();

int NUMBER\_THREADS = atoi(argv[1]);

//cout << "#Количество потоков = " << NUMBER\_THREADS << "\n";

int counter\_first, counter\_second, plays, tour, experements, count\_of\_experements\_for\_one\_thread;

float wins\_first = 0, wins\_second = 0;

cout << "Введите:\n 1)Количество очков первого игрока: ";

cin >> counter\_first;

cout << " 2)Количество очков второго игрока: ";

cin >> counter\_second;

cout << " 3)Номер данного тура: ";

cin >> tour;

cout << " 4)Количество бросков костей: ";

cin >> plays;

cout << " 5)Количество экспериментов: ";

cin >> experements;

count\_of\_experements\_for\_one\_thread = experements / NUMBER\_THREADS;

//cout << "#Количество эксперементов для одного потока = " << count\_of\_experements\_for\_one\_thread << "\n";

thread\_data tdata[NUMBER\_THREADS];

for (int i = 0; i < NUMBER\_THREADS; i++){

tdata[i].counter\_first = counter\_first;

tdata[i].counter\_second = counter\_second;

tdata[i].plays = plays;

tdata[i].tour = tour;

if (i == NUMBER\_THREADS - 1)

tdata[i].experements = count\_of\_experements\_for\_one\_thread + experements % NUMBER\_THREADS;

else

tdata[i].experements = count\_of\_experements\_for\_one\_thread;

tdata[i].wins\_first = 0;

tdata[i].wins\_second = 0;

}

pthread\_t thread[NUMBER\_THREADS];

for (int i = 0; i < NUMBER\_THREADS; i++){

if(pthread\_create(&thread[i], NULL, thread\_func, &tdata[i]) != 0){

cerr << "error: Cannot create thread # " << i << "\n";

break;

}

}

for (int i = 0; i < NUMBER\_THREADS; i++){

if(pthread\_join(thread[i], NULL) != 0){

cerr << "error: Cannot join thread # " << i << "\n";

break;

}

thread\_data \*result = &tdata[i];

wins\_first += result->wins\_first;

wins\_second += result->wins\_second;

}

cout << "Вероятность победы первого игрока: " << wins\_first/experements << "\n";

cout << "Вероятность победы второго игрока: " << wins\_second/experements << "\n";

//end\_time = clock()`;

//timer = end\_time - start\_time;

//cout << "Time: " << timer / CLOCKS\_PER\_SEC << "\n";

}

}

**Демонстрация работы программы**

papik@papik-VirtualBox:~/OSlaba3/src$ g++ main.cpp -pthread

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 1

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493947

Вероятность победы второго игрока: 0.494133

real 0m9,223s

user 0m3,959s

sys 0m0,004s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 2

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.494142

Вероятность победы второго игрока: 0.493887

real 0m7,612s

user 0m3,973s

sys 0m0,008s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 4

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493423

Вероятность победы второго игрока: 0.49443

real 0m6,246s

user 0m4,009s

sys 0m0,000s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 8

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493825

Вероятность победы второго игрока: 0.494213

real 0m7,180s

user 0m4,044s

sys 0m0,016s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 16

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493358

Вероятность победы второго игрока: 0.494585

real 0m7,105s

user 0m4,101s

sys 0m0,000s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 32

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.49463

Вероятность победы второго игрока: 0.493577

real 0m6,184s

user 0m4,038s

sys 0m0,032s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 64

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493978

Вероятность победы второго игрока: 0.494026

real 0m6,331s

user 0m4,047s

sys 0m0,016s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 128

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493533

Вероятность победы второго игрока: 0.494505

real 0m7,406s

user 0m4,030s

sys 0m0,025s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 256

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493946

Вероятность победы второго игрока: 0.494211

real 0m6,926s

user 0m4,015s

sys 0m0,045s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 512

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493667

Вероятность победы второго игрока: 0.494381

real 0m6,545s

user 0m4,051s

sys 0m0,134s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 1024

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.494016

Вероятность победы второго игрока: 0.493905

real 0m6,105s

user 0m4,058s

sys 0m0,196s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 2048

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493373

Вероятность победы второго игрока: 0.494668

real 0m5,780s

user 0m4,070s

sys 0m0,270s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 4096

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.494171

Вероятность победы второго игрока: 0.493933

real 0m5,940s

user 0m4,147s

sys 0m0,419s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 8192

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.494255

Вероятность победы второго игрока: 0.494281

real 0m6,389s

user 0m4,060s

sys 0m0,971s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 16384

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.493272

Вероятность победы второго игрока: 0.49494

real 0m7,199s

user 0m4,368s

sys 0m1,774s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 32743

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.494032

Вероятность победы второго игрока: 0.494283

real 0m7,957s

user 0m4,562s

sys 0m3,158s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 32744

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

error: Cannot create thread # 32743

error: Cannot join thread # 32743

Вероятность победы первого игрока: 0.485222

Вероятность победы второго игрока: 0.485605

real 0m8,007s

user 0m4,416s

sys 0m2,817s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 1

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498066

Вероятность победы второго игрока: 0.498231

real 0m52,258s

user 0m39,736s

sys 0m0,021s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 2

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.49792

Вероятность победы второго игрока: 0.498345

real 0m30,638s

user 0m46,704s

sys 0m0,004s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 4

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.497825

Вероятность победы второго игрока: 0.498467

real 0m19,680s

user 0m55,021s

sys 0m0,066s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 8

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.49797

Вероятность победы второго игрока: 0.498293

real 0m16,326s

user 0m42,723s

sys 0m0,029s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 16

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498341

Вероятность победы второго игрока: 0.497958

real 0m20,185s

user 0m55,672s

sys 0m0,126s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 32

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.497621

Вероятность победы второго игрока: 0.49866

real 0m19,375s

user 0m55,745s

sys 0m0,037s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 64

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498141

Вероятность победы второго игрока: 0.498197

real 0m18,086s

user 0m42,307s

sys 0m0,161s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 128

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498084

Вероятность победы второго игрока: 0.498245

real 0m18,642s

user 0m50,554s

sys 0m0,141s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 256

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.497831

Вероятность победы второго игрока: 0.4985

real 0m16,791s

user 0m42,192s

sys 0m0,085s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 512

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.497892

Вероятность победы второго игрока: 0.498375

real 0m19,951s

user 0m50,306s

sys 0m0,500s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 1024

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498171

Вероятность победы второго игрока: 0.498102

real 0m16,931s

user 0m42,810s

sys 0m0,314s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 2056

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498943

Вероятность победы второго игрока: 0.497317

real 0m19,895s

user 0m52,960s

sys 0m0,646s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 4098

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498041

Вероятность победы второго игрока: 0.498297

real 0m17,164s

user 0m43,676s

sys 0m0,709s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 8192

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498507

Вероятность победы второго игрока: 0.497815

real 0m17,989s

user 0m44,302s

sys 0m1,153s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 16384

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.497763

Вероятность победы второго игрока: 0.498604

real 0m22,351s

user 0m50,272s

sys 0m2,683s

papik@papik-VirtualBox:~/OSlaba3/src$ time ./a.out 32743

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 1000

5)Количество экспериментов: 1000000

Вероятность победы первого игрока: 0.498149

Вероятность победы второго игрока: 0.498053

real 0m21,841s

user 0m48,352s

sys 0m5,645s

papik@papik-VirtualBox:~/OSlaba3/src$ strace -e trace=process ./a.out 64

execve("./a.out", ["./a.out", "64"], 0x7fff81061428 /\* 49 vars \*/) = 0

arch\_prctl(0x3001 /\* ARCH\_??? \*/, 0x7ffc5c1d1a90) = -1 EINVAL (Недопустимый аргумент)

arch\_prctl(ARCH\_SET\_FS, 0x7f729d207740) = 0

Введите:

1)Количество очков первого игрока: 0

2)Количество очков второго игрока: 0

3)Номер данного тура: 1

4)Количество бросков костей: 100

5)Количество экспериментов: 1000000

clone(child\_stack=0x7f729d205fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232453], tls=0x7f729d206700, child\_tidptr=0x7f729d2069d0) = 232453

clone(child\_stack=0x7f729ca04fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232454], tls=0x7f729ca05700, child\_tidptr=0x7f729ca059d0) = 232454

clone(child\_stack=0x7f729c203fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232455], tls=0x7f729c204700, child\_tidptr=0x7f729c2049d0) = 232455

clone(child\_stack=0x7f729ba02fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232456], tls=0x7f729ba03700, child\_tidptr=0x7f729ba039d0) = 232456

clone(child\_stack=0x7f729b201fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232457], tls=0x7f729b202700, child\_tidptr=0x7f729b2029d0) = 232457

clone(child\_stack=0x7f729aa00fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232458], tls=0x7f729aa01700, child\_tidptr=0x7f729aa019d0) = 232458

clone(child\_stack=0x7f729a1fffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232459], tls=0x7f729a200700, child\_tidptr=0x7f729a2009d0) = 232459

clone(child\_stack=0x7f72999fefb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232460], tls=0x7f72999ff700, child\_tidptr=0x7f72999ff9d0) = 232460

clone(child\_stack=0x7f72991fdfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232461], tls=0x7f72991fe700, child\_tidptr=0x7f72991fe9d0) = 232461

clone(child\_stack=0x7f72989fcfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232462], tls=0x7f72989fd700, child\_tidptr=0x7f72989fd9d0) = 232462

clone(child\_stack=0x7f72981fbfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232463], tls=0x7f72981fc700, child\_tidptr=0x7f72981fc9d0) = 232463

clone(child\_stack=0x7f72979fafb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232464], tls=0x7f72979fb700, child\_tidptr=0x7f72979fb9d0) = 232464

clone(child\_stack=0x7f72971f9fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232465], tls=0x7f72971fa700, child\_tidptr=0x7f72971fa9d0) = 232465

clone(child\_stack=0x7f72969f8fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232466], tls=0x7f72969f9700, child\_tidptr=0x7f72969f99d0) = 232466

clone(child\_stack=0x7f72961f7fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232467], tls=0x7f72961f8700, child\_tidptr=0x7f72961f89d0) = 232467

clone(child\_stack=0x7f72959f6fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232468], tls=0x7f72959f7700, child\_tidptr=0x7f72959f79d0) = 232468

clone(child\_stack=0x7f72951f5fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232469], tls=0x7f72951f6700, child\_tidptr=0x7f72951f69d0) = 232469

clone(child\_stack=0x7f72949f4fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232470], tls=0x7f72949f5700, child\_tidptr=0x7f72949f59d0) = 232470

clone(child\_stack=0x7f72941f3fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232471], tls=0x7f72941f4700, child\_tidptr=0x7f72941f49d0) = 232471

clone(child\_stack=0x7f72939f2fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232472], tls=0x7f72939f3700, child\_tidptr=0x7f72939f39d0) = 232472

clone(child\_stack=0x7f72931f1fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232473], tls=0x7f72931f2700, child\_tidptr=0x7f72931f29d0) = 232473

clone(child\_stack=0x7f72929f0fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232474], tls=0x7f72929f1700, child\_tidptr=0x7f72929f19d0) = 232474

clone(child\_stack=0x7f72921effb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232475], tls=0x7f72921f0700, child\_tidptr=0x7f72921f09d0) = 232475

clone(child\_stack=0x7f72919eefb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232476], tls=0x7f72919ef700, child\_tidptr=0x7f72919ef9d0) = 232476

clone(child\_stack=0x7f72911edfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232477], tls=0x7f72911ee700, child\_tidptr=0x7f72911ee9d0) = 232477

clone(child\_stack=0x7f72909ecfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232478], tls=0x7f72909ed700, child\_tidptr=0x7f72909ed9d0) = 232478

clone(child\_stack=0x7f72901ebfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232479], tls=0x7f72901ec700, child\_tidptr=0x7f72901ec9d0) = 232479

clone(child\_stack=0x7f728f9eafb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232480], tls=0x7f728f9eb700, child\_tidptr=0x7f728f9eb9d0) = 232480

clone(child\_stack=0x7f728f1e9fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232481], tls=0x7f728f1ea700, child\_tidptr=0x7f728f1ea9d0) = 232481

clone(child\_stack=0x7f728e9e8fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232482], tls=0x7f728e9e9700, child\_tidptr=0x7f728e9e99d0) = 232482

clone(child\_stack=0x7f728e1e7fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232483], tls=0x7f728e1e8700, child\_tidptr=0x7f728e1e89d0) = 232483

clone(child\_stack=0x7f728d9e6fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232484], tls=0x7f728d9e7700, child\_tidptr=0x7f728d9e79d0) = 232484

clone(child\_stack=0x7f728d1e5fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232485], tls=0x7f728d1e6700, child\_tidptr=0x7f728d1e69d0) = 232485

clone(child\_stack=0x7f728c9e4fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232486], tls=0x7f728c9e5700, child\_tidptr=0x7f728c9e59d0) = 232486

clone(child\_stack=0x7f728c1e3fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232487], tls=0x7f728c1e4700, child\_tidptr=0x7f728c1e49d0) = 232487

clone(child\_stack=0x7f728b9e2fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232488], tls=0x7f728b9e3700, child\_tidptr=0x7f728b9e39d0) = 232488

clone(child\_stack=0x7f728b1e1fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232489], tls=0x7f728b1e2700, child\_tidptr=0x7f728b1e29d0) = 232489

clone(child\_stack=0x7f728a9e0fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232490], tls=0x7f728a9e1700, child\_tidptr=0x7f728a9e19d0) = 232490

clone(child\_stack=0x7f728a1dffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232491], tls=0x7f728a1e0700, child\_tidptr=0x7f728a1e09d0) = 232491

clone(child\_stack=0x7f72899defb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232492], tls=0x7f72899df700, child\_tidptr=0x7f72899df9d0) = 232492

clone(child\_stack=0x7f72891ddfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232493], tls=0x7f72891de700, child\_tidptr=0x7f72891de9d0) = 232493

clone(child\_stack=0x7f72889dcfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232494], tls=0x7f72889dd700, child\_tidptr=0x7f72889dd9d0) = 232494

clone(child\_stack=0x7f72881dbfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232495], tls=0x7f72881dc700, child\_tidptr=0x7f72881dc9d0) = 232495

clone(child\_stack=0x7f72879dafb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232496], tls=0x7f72879db700, child\_tidptr=0x7f72879db9d0) = 232496

clone(child\_stack=0x7f72871d9fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232497], tls=0x7f72871da700, child\_tidptr=0x7f72871da9d0) = 232497

clone(child\_stack=0x7f72869d8fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232498], tls=0x7f72869d9700, child\_tidptr=0x7f72869d99d0) = 232498

clone(child\_stack=0x7f72861d7fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232499], tls=0x7f72861d8700, child\_tidptr=0x7f72861d89d0) = 232499

clone(child\_stack=0x7f72859d6fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232500], tls=0x7f72859d7700, child\_tidptr=0x7f72859d79d0) = 232500

clone(child\_stack=0x7f72851d5fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232501], tls=0x7f72851d6700, child\_tidptr=0x7f72851d69d0) = 232501

clone(child\_stack=0x7f72849d4fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232502], tls=0x7f72849d5700, child\_tidptr=0x7f72849d59d0) = 232502

clone(child\_stack=0x7f72841d3fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232503], tls=0x7f72841d4700, child\_tidptr=0x7f72841d49d0) = 232503

clone(child\_stack=0x7f72839d2fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232504], tls=0x7f72839d3700, child\_tidptr=0x7f72839d39d0) = 232504

clone(child\_stack=0x7f72831d1fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232505], tls=0x7f72831d2700, child\_tidptr=0x7f72831d29d0) = 232505

clone(child\_stack=0x7f72829d0fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232506], tls=0x7f72829d1700, child\_tidptr=0x7f72829d19d0) = 232506

clone(child\_stack=0x7f72821cffb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232507], tls=0x7f72821d0700, child\_tidptr=0x7f72821d09d0) = 232507

clone(child\_stack=0x7f72819cefb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232508], tls=0x7f72819cf700, child\_tidptr=0x7f72819cf9d0) = 232508

clone(child\_stack=0x7f72811cdfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232509], tls=0x7f72811ce700, child\_tidptr=0x7f72811ce9d0) = 232509

clone(child\_stack=0x7f72809ccfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232510], tls=0x7f72809cd700, child\_tidptr=0x7f72809cd9d0) = 232510

clone(child\_stack=0x7f72801cbfb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232511], tls=0x7f72801cc700, child\_tidptr=0x7f72801cc9d0) = 232511

clone(child\_stack=0x7f727f9cafb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232512], tls=0x7f727f9cb700, child\_tidptr=0x7f727f9cb9d0) = 232512

clone(child\_stack=0x7f727f1c9fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232513], tls=0x7f727f1ca700, child\_tidptr=0x7f727f1ca9d0) = 232513

clone(child\_stack=0x7f727e9c8fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232514], tls=0x7f727e9c9700, child\_tidptr=0x7f727e9c99d0) = 232514

clone(child\_stack=0x7f727e1c7fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232515], tls=0x7f727e1c8700, child\_tidptr=0x7f727e1c89d0) = 232515

clone(child\_stack=0x7f727d9c6fb0, flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, parent\_tid=[232516], tls=0x7f727d9c7700, child\_tidptr=0x7f727d9c79d0) = 232516

Вероятность победы первого игрока: 0.494142

Вероятность победы второго игрока: 0.493845

exit\_group(0) = ?

+++ exited with 0 +++

**Выводы**

За время выполнения лабораторной работы я научился управлять потоками в ОС, а также разобрался с обеспечение синхронизации между потоками.