

Московский Авиационный Институт  
(Национальный Исследовательский Университет)  
Институт №8 “Компьютерные науки и прикладная математика”  
Кафедра №806 “Вычислительная математика и программирование”

**Лабораторная работа №2 по курсу**  
**«Операционные системы»**

Группа: М8О-209БВ-24

Студент: Галич А.П

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_

Дата: 01.12.25

Москва, 2025

# Постановка задачи

## Вариант 7.

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

Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

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

## Общий метод и алгоритм решения

### Использованные системные вызовы:

- `time()` – получение текущего времени
- `clock()` – измерение процессорного времени выполнения
- `pthread_create()` - создание потоков для параллельного поиска
- `pthread_join()` - ожидание завершения всех потоков и получение результата
- `gettimeofday()` = измерение времени выполнения

### Алгоритм работы программы:

#### 1. Инициализация параметров и планирование потоков

- Парсинг аргументов командной строки: K, очки игрока 1, очки игрока 2, количество экспериментов, колтчество потоков
- Проверка корректности всех входных параметров
- Инициализация главного генератора сулчайных чисел
- Распсеределение экспериментов между потоками
- Создание массива стурктур для передачи данных в каждый поток

#### 2. Параллельное выполнение

- Каждый поток работает независимо со своей копией данных
- Инициализация потокобезопасности генератора слуучайных чисел (`rand_r`) с уникальным seed

- Цикл по назначенным экспериментам:
  - Сброс текущих очков к начальным значениям
  - Моделирование К бросков для каждого игрока
  - Сравнение итоговых результатов и обновление локальных счетчиков

### 3. Финальная обработка

- Ожидание завершения всех рабочих потоков
- Суммирование результатов из всех потоков
- Вычисляется вероятность побед и ничьих
- Форматированный вывод статистики и времени выполнения

## Код программы

**dice\_game.c**

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <time.h>
#include <unistd.h>
#include <sys/time.h>

typedef struct {
    int k;
    int player1_score;
    int player2_score;
    int experiments;
    int thread_id;

    long player1_wins;
    long player2_wins;
    long draws;
} thread_data_t;

long global_player1_wins = 0;
long global_player2_wins = 0;
long global_draws = 0;

int roll_dice(unsigned int* seed) {
    return (rand_r(seed) % 6 + 1) + (rand_r(seed) % 6 + 1);
}

void* simulate_games(void* arg) {
    thread_data_t* data = (thread_data_t*)arg;

    unsigned int seed = time(NULL) ^ data->thread_id;

    for (int exp = 0; exp < data->experiments; exp++) {
```

```

    int player1_current = data->player1_score;
    int player2_current = data->player2_score;

    for (int i = 0; i < data->k; i++) {
        player1_current += roll_dice(&seed);
        player2_current += roll_dice(&seed);
    }

    if (player1_current > player2_current) {
        data->player1_wins++;
    } else if (player2_current > player1_current) {
        data->player2_wins++;
    } else {
        data->draws++;
    }
}

return NULL;
}

void print_help() {
    printf("Исп: ./dice_game К раунд очки1 очки2 эксперименты потоки\nПример: ./dice_game\n5 3 25 30 1000000 4\n");
}

int main(int argc, char* argv[]) {
    if (argc != 7) {
        printf("Ошибка: Неверное количество параметров!\n");
        printf("Ожидается 6 параметров, получено: %d\n\n", argc - 1);
        print_help();
        return 1;
    }

    int k = atoi(argv[1]); // atoi преобразует строку в число
    int current_round = atoi(argv[2]);
    int player1_total = atoi(argv[3]);
    int player2_total = atoi(argv[4]);
    int total_experiments = atoi(argv[5]);
    int max_threads = atoi(argv[6]);

    if (k <= 0 || current_round <= 0 || player1_total < 0 || player2_total < 0 ||
        total_experiments <= 0 || max_threads <= 0) {
        printf("Ошибка: Все параметры должны быть положительными числами!\n");
        printf("Очки игроков могут быть 0 или больше.\n");
        return 1;
    }

    pthread_t threads[max_threads];
    thread_data_t thread_data[max_threads];

    int experiments_per_thread = total_experiments / max_threads;
    int remaining_experiments = total_experiments % max_threads;

```

```

struct timeval start_time, end_time;
gettimeofday(&start_time, NULL);

for (int i = 0; i < max_threads; i++) {
    thread_data[i].k = k;
    thread_data[i].player1_score = player1_total;
    thread_data[i].player2_score = player2_total;
    thread_data[i].experiments = experiments_per_thread;
    thread_data[i].thread_id = i;
    thread_data[i].player1_wins = 0;
    thread_data[i].player2_wins = 0;
    thread_data[i].draws = 0;

    if (i < remaining_experiments) {
        thread_data[i].experiments++;
    }

    if (pthread_create(&threads[i], NULL, simulate_games, &thread_data[i]) != 0) {
        perror("Ошибка при создании потока");
        return 1;
    }
}

for (int i = 0; i < max_threads; i++) {
    pthread_join(threads[i], NULL);
}

for (int i = 0; i < max_threads; i++) {
    global_player1_wins += thread_data[i].player1_wins;
    global_player2_wins += thread_data[i].player2_wins;
    global_draws += thread_data[i].draws;
}

gettimeofday(&end_time, NULL);
double execution_time = (end_time.tv_sec - start_time.tv_sec) +
    (end_time.tv_usec - start_time.tv_usec) / 1000000.0;

printf("=== РЕЗУЛЬТАТЫ ===\n");
printf("Количество потоков: %d\n", max_threads);
printf("Всего экспериментов: %d\n", total_experiments);
printf("Побед игрока 1: %ld (%.2f%%)\n",
    global_player1_wins,
    (double)global_player1_wins / total_experiments * 100);
printf("Побед игрока 2: %ld (%.2f%%)\n",
    global_player2_wins,
    (double)global_player2_wins / total_experiments * 100);
printf("Ничьих: %ld (%.2f%%)\n",
    global_draws,
    (double)global_draws / total_experiments * 100);
printf("Время выполнения: %.3f секунд\n", execution_time);

return 0;
}

```

# Протокол работы программы

## Тестирование:

```
kishaki@416:~/lab_OS2$ ./dice_game 10000 1 0 0 1000000 4
=== РЕЗУЛЬТАТЫ ===
Количество потоков: 4
Всего экспериментов: 1000000
Побед игрока 1: 499560 (49.96%)
Побед игрока 2: 499333 (49.93%)
Ничьих: 1107 (0.11%)
Время выполнения: 32.383 секунд
```

## Strace:

```
$ strace -f ./dice_game 1000 1 0 0 1000000 4
```

```
execve("./dice_game", [ "./dice_game", "10000", "1", "0", "0", "1000000", "4"],
0x7ffdc4a6db88 /* 29 vars */) = 0

brk(NULL)                               = 0x5c453ffe2000

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x70c865f98000

access("/etc/ld.so.preload", R_OK)       = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0

mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x70c865f8e000

close(3)                                 = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"... , 832)
= 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"... , 784,
64) = 784

fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"... , 784,
64) = 784

mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x70c865c00000

mmap(0x70c865c28000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x70c865c28000

mmap(0x70c865db0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x70c865db0000

mmap(0x70c865dff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1fe000) = 0x70c865dff000

mmap(0x70c865e05000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x70c865e05000

close(3)                                 = 0
```

```

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x70c865f8b000

arch_prctl(ARCH_SET_FS, 0x70c865f8b740) = 0

set_tid_address(0x70c865f8ba10) = 27451

set_robust_list(0x70c865f8ba20, 24) = 0

rseq(0x70c865f8c060, 0x20, 0, 0x53053053) = 0

mprotect(0x70c865dff000, 16384, PROT_READ) = 0

mprotect(0x5c4508f1a000, 4096, PROT_READ) = 0

mprotect(0x70c865fd0000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x70c865f8e000, 37043) = 0

rt_sigaction(SIGRT_1, {sa_handler=0x70c865c99530, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x70c865c45330}, NULL, 8)
= 0

rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x70c8653ff000

mprotect(0x70c865400000, 8388608, PROT_READ|PROT_WRITE) = 0

getrandom("\x66\xe1\x87\x05\x80\x8f\x65\xbc", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x5c453ffe2000

brk(0x5c4540003000) = 0x5c4540003000

rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|C
LONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x70c865bff990,
parent_tid=0x70c865bff990, exit_signal=0, stack=0x70c8653ff000, stack_size=0x7fff80,
tls=0x70c865bff6c0}strace: Process 27452 attached

=> {parent_tid=[27452]}, 88) = 27452

[pid 27452] rseq(0x70c865bffffe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 27451] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27452] <... rseq resumed> = 0

[pid 27451] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27452] set_robust_list(0x70c865bff9a0, 24 <unfinished ...>

[pid 27451] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

[pid 27452] <... set_robust_list resumed>) = 0

[pid 27451] <... mmap resumed> = 0x70c864bfe000

[pid 27452] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27451] mprotect(0x70c864bff000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 27452] <... rt_sigprocmask resumed>NULL, 8) = 0

```

```

[pid 27451] <... mprotect resumed>)      = 0

[pid 27451] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 27451]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_S
ETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x70c8653fe990,
parent_tid=0x70c8653fe990, exit_signal=0, stack=0x70c864bfe000, stack_size=0x7fff80,
tls=0x70c8653fe6c0}strace: Process 27453 attached

=> {parent_tid=[27453]}, 88) = 27453

[pid 27451] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27453] rseq(0x70c8653fefe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 27451] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27451] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

[pid 27453] <... rseq resumed>)          = 0

[pid 27451] <... mmap resumed>)          = 0x70c8643fd000

[pid 27453] set_robust_list(0x70c8653fe9a0, 24 <unfinished ...>

[pid 27451] mprotect(0x70c8643fe000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 27453] <... set_robust_list resumed>) = 0

[pid 27451] <... mprotect resumed>)      = 0

[pid 27453] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27451] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 27453] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27451] <... rt_sigprocmask resumed>[], 8) = 0

[pid 27451]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_S
ETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x70c864bfd990,
parent_tid=0x70c864bfd990, exit_signal=0, stack=0x70c8643fd000, stack_size=0x7fff80,
tls=0x70c864bfd6c0}strace: Process 27454 attached

=> {parent_tid=[27454]}, 88) = 27454

[pid 27454] rseq(0x70c864bfdfe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 27451] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27454] <... rseq resumed>)          = 0

[pid 27451] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27454] set_robust_list(0x70c864bfd9a0, 24 <unfinished ...>

[pid 27451] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

[pid 27454] <... set_robust_list resumed>) = 0

[pid 27451] <... mmap resumed>)          = 0x70c863bfc000

[pid 27451] mprotect(0x70c863bfd000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 27454] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

```



```

[pid 27451] <... mprotect resumed>)      = 0

[pid 27454] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27451] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 27451]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_S
ETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEAR_TID, child_tid=0x70c8643fc990,
parent_tid=0x70c8643fc990, exit_signal=0, stack=0x70c863bfc000, stack_size=0x7fff80,
tls=0x70c8643fc6c0}strace: Process 27455 attached

=> {parent_tid=[27455]}, 88) = 27455

[pid 27455] rseq(0x70c8643fcfe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 27451] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 27455] <... rseq resumed>)          = 0

[pid 27451] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27455] set_robust_list(0x70c8643fc9a0, 24 <unfinished ...>

[pid 27451] futex(0x70c865bff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 27452, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 27455] <... set_robust_list resumed>) = 0

[pid 27455] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

[pid 27455] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

[pid 27455] madvise(0x70c863bfc000, 8368128, MADV_DONTNEED) = 0

[pid 27455] exit(0)                        = ?

[pid 27455] +++ exited with 0 +++

[pid 27452] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

[pid 27452] madvise(0x70c8653ff000, 8368128, MADV_DONTNEED) = 0

[pid 27452] exit(0)                        = ?

[pid 27451] <... futex resumed>)          = 0

[pid 27453] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 27452] +++ exited with 0 +++

[pid 27451] futex(0x70c8653fe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 27453, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 27453] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 27453] madvise(0x70c864bfe000, 8368128, MADV_DONTNEED) = 0

[pid 27453] exit(0)                        = ?

[pid 27451] <... futex resumed>)          = 0

[pid 27453] +++ exited with 0 +++

[pid 27451] futex(0x70c864bfd990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 27454, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 27454] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

[pid 27454] madvise(0x70c8643fd000, 8368128, MADV_DONTNEED) = 0

```

```

[pid 27454] exit(0)                                = ?

[pid 27451] <... futex resumed>                     = 0

[pid 27454] +++ exited with 0 +++

fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x5), ...}) = 0

write(1, "===
\320\240\320\225\320\227\320\243\320\233\320\254\320\242\320\220\320\242\320\253 ===\n",
29=== РЕЗУЛЬТАТЫ ===

) = 29

write(1,
"\320\232\320\276\320\273\320\270\321\207\320\265\321\201\321\202\320\262\320\276
\320\277\320\276\321\202\320\276\320\272\320"... , 39Количество потоков: 4

) = 39

write(1, "\320\222\321\201\320\265\320\263\320\276
\321\215\320\272\321\201\320\277\320\265\321\200\320\270\320\274\320\265\320\275\321"... ,
47Всего экспериментов: 1000000

) = 47

write(1, "\320\237\320\276\320\261\320\265\320\264
\320\270\320\263\321\200\320\276\320\272\320\260 1: 49933"... , 43Побед игрока 1: 499330
(49.93%)

) = 43

write(1, "\320\237\320\276\320\261\320\265\320\264
\320\270\320\263\321\200\320\276\320\272\320\260 2: 49947"... , 43Побед игрока 2: 499473
(49.95%)

) = 43

write(1, "\320\235\320\270\321\207\321\214\320\270\321\205: 1197 (0.12%)\n", 27Ничьих:
1197 (0.12%)

) = 27

write(1, "\320\222\321\200\320\265\320\274\321\217
\320\262\321\213\320\277\320\276\320\273\320\275\320\265\320\275\320\270\321\217:"... ,
53Время выполнения: 32.595 секунд

) = 53

exit_group(0)                                       = ?

+++ exited with 0 +++

```

Таблица составленная по результатам 10000 1 0 0 1000000

Потоки	Время (с)	Ускорение	Эффективность
1	119.883	1	1
2	58.567	2.047	1.023
4	32.539	3.685	0.921
8	21.105	5.68	0.71

Эксперимент показал, что при увеличении числа потоков время выполнения уменьшается, но с определенного момента понижается производительность из-за увеличения накладных расходов на управление потоками

## **Вывод**

В ходе выполнения лабораторной работы освоены принципы многопоточного программирования с использованием библиотеки `pthreads`. Практические результаты показывают, что многопоточное программирование является эффективным инструментом для ускорения сложных задач, однако требует тщательного выбора количества потоков для достижения оптимального соотношения между производительностью и эффективностью использования системных ресурсов.