

Московский авиационный институт
(национальный исследовательский университет)

Институт: «Информационные технологии и прикладная
математика»

Кафедра: 806 «Вычислительная математика и
программирование»

Лабораторная работа №1
по курсу «Операционные системы»

УПРАВЛЕНИЕ ПОТОКАМИ В ОС

Студент: Лагуткина Мария Сергеевна

Группа: М8О-206Б-19

Преподаватель: Соколов А. А.

Дата:

Оценка:

Москва, 2020

1 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 обработке 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.

Вариант 14. Есть колода из 52 карт, рассчитать экспериментально (метод Монте-Карло) вероятность того, что сверху лежат две одинаковых карты. Количество раундов подается с ключом

2 Общий метод и алгоритм решения

Так как в задании более точно не указано по какому именно признаку карты одинаковы, то карты будут считаться одинаковыми, если у них равные значения, но разные масти. Вероятность того, что две карты будут равны считается экспериментально и задача сводится к нахождению отношения всех благоприятных исходов на всевозможные. Аргументом к программе подается количество потоков. В каждом потоке с помощью генерации псевдослучайных чисел вычисляются 2 значения и 2 масти, если условие задачи выполняется, увеличивается значение глобальной переменной. После завершения работы созданных потоков считается вероятность.

3 Основные файлы программы

os3.c

```
1 | include <pthread.h>
2 | #include <stdio.h>
3 | #include <stdlib.h>
4 | #include <string.h>
```

```

5
6 int count_true;
7
8 int get_rand_range_int(const int min, const int max) {
9     return rand() % (max - min + 1) + min;
10 }
11
12 void* tread_count(void* arg){
13     int num1 = get_rand_range_int(1,13);
14     int num2 = get_rand_range_int(1,13);
15     int suit1 = get_rand_range_int(1,4);
16     int suit2 = get_rand_range_int(1,4);
17     if((num1 == num2) && (suit1 != suit2)){
18         count_true++;
19     }
20     pthread_exit(NULL);
21 }
22
23 int main(int argc, char *argv[]){
24     int number_of_treads = 0;
25     char string_number_of_treads[10];
26     if (argc == 1) {
27         printf("Enter number of treads: \n");
28         scanf("%d",&number_of_treads);
29     } else {
30         strcpy(string_number_of_treads, argv[1]);
31         number_of_treads = atoi(string_number_of_treads);
32     }
33     pthread_t threads[number_of_treads];
34     count_true = 0;
35     int status;
36     int status_addr;
37     for(int i =0; i<number_of_treads; ++i){
38         status = pthread_create(&threads[i], NULL, tread_count, (void *)i);
39         if (status != 0) {
40             printf("main error: can't create thread, status = %d\n", status);
41             exit(-1);
42         }
43     }
44     for(int i =0; i<number_of_treads; ++i){
45         status = pthread_join(threads[i],NULL);
46         if (status != 0) {
47             printf("main error: can't join thread, status = %d\n", status);
48             exit(-2);
49         }
50     }
51     double probability = (double) count_true / number_of_treads;
52     printf("%f\n", probability);
53     return 0;

```

4 Пример работы

```
maria@maria-MS-7817:~/project/os2$ ./a.out 100
0.020000
maria@maria-MS-7817:~/project/os2$ ./a.out 1000
0.059000
maria@maria-MS-7817:~/project/os2$ ./a.out 10000
0.060900
maria@maria-MS-7817:~/project/os2$ ./a.out 10
0.100000
maria@maria-MS-7817:~/project/os2$ ./a.out 1
0.000000
maria@maria-MS-7817:~/project/os2$ ./a.out 5
0.200000
```

5 Вывод

В этой лабораторной я узнала как работать с потоками, опробовала системные вызовы для создания и ожидания завершения работы потоков. Также было интересно почитать про метод Монте-Карло.