

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-214Б-23

Студент: Заваротный А. А.

Преподаватель: Бахарев В. Д.

Оценка: _____

Дата: 28.10.24

Москва, 2024

Постановка задачи

Вариант 20.

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. Найти в большом целочисленном массиве минимальный и максимальный элементы.

Общий метод и алгоритм решения

Использованные системные вызовы:

- `pthread_t threads[num_threads];` - объявляет массив потоков.
- `pthread_create(&threads[i], NULL, Find_min_max, &thread_data_array[i]);` - создаёт новый поток.
- `pthread_join(threads[i], NULL);` - ожидание завершения конкретного потока.

Решение:

1. Объявляю функцию, которая будет выполняться внутри потока.
2. Считываю количество потоков и размер массива.
3. Заполняю массив чисел случайным образом, используя функцию `rand()`.
4. Инициализирую массив потоков, массив данных потоков и переменные для максимального и минимального элементов.
5. В цикле заполняю данные потока и запускаю поток.
6. В цикле жду завершения каждого потока и сравниваю его ответ с текущим.
7. Вывожу ответ.

Код программы

client.c

```
#include <stdint.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>
#include <ctype.h>
#include <time.h>
#include <stdio.h>
#include <pthread.h>

typedef struct {
    int *array;
    int ar_max;
    int ar_min;
    int start;
    int end;
} thread_data;

void* Find_min_max(void* arg) {
    thread_data *data = (thread_data *)arg;
    int *array = data->array;
    int start = data->start, end = data->end;
    int ar_max = array[start], ar_min = array[start];
    for (size_t i = start; i < end; i++) {
        if (array[i] > ar_max) {
```

```

        ar_max = array[i];
    }
    else if (array[i] < ar_min) {
        ar_min = array[i];
    }
}
data->ar_max = ar_max;
data->ar_min = ar_min;
pthread_exit(NULL);
return NULL;
}

int main(int argc, char **argv) {

    int ind_array = atoi(argv[2]);
    int *array = (int*) malloc(ind_array * sizeof (int));
    if (array == NULL) {
        const char msg[] = "error: failed to allocate memory\n";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }
    srand(time(NULL));
    for (int i = 0; i < ind_array; i++) {
        array[i] = rand() % 2000 - 1000;
        //printf("%d ", array[i]);
    }
    //printf("\n");
    int num_of_threads = atoi(argv[1]);

    pthread_t threads[num_of_threads];
    thread_data thread_data_array[num_of_threads];

    for (int i = 0; i < num_of_threads; i++) {
        thread_data array[i].array = array;
        thread_data_array[i].start = ind_array / num_of_threads * i;
        if (i == num_of_threads - 1) {
            thread_data_array[i].end = ind_array;
        }
        else {
            thread_data_array[i].end = thread_data_array[i].start + ind_array /
num_of_threads;
        }
        if (pthread_create(&threads[i], NULL, Find_min_max, &thread_data_array[i]) !=
0) {
            const char msg[] = "error: failed to create thread\n";
            write(STDERR_FILENO, msg, sizeof(msg));
            exit(EXIT_FAILURE);
        }
    }

    int arr_mx = -40000, arr_mn = 40000;
    for (int i = 0; i < num_of_threads; i++) {
        pthread_join(threads[i], NULL);
        if (arr_mx < thread_data_array[i].ar_max) {
            arr_mx = thread_data_array[i].ar_max;
        }
        else if (arr_mn > thread_data_array[i].ar_min) {
            arr_mn = thread_data_array[i].ar_min;
        }
    }
    printf("Min = %d\nMax = %d\n", arr_mn, arr_mx);
    free(array);
    return 0;
}

```

Протокол работы программы

Тестирование:

```
$ cc -o client client.c -lm -pthread
```

```
$ ./client 2 50
```

```
Min = -801
```

```
Max = 859
```

Число потоков	Время исполнения (мс)	Ускорение	Эффективность
1	8	1	1
2	7	1,14	0,57
3	5	1,6	0,53
4	8	1	0,25
5	12	0,66	0,13
6	8	1	0,16

```
ind_array = 50000000
```

Число потоков	Время исполнения (мс)	Ускорение	Эффективность
1	1815	1	1
2	1490	1,21	0,6
3	1454	1,25	0,41
4	1436	1,26	0,31
5	1429	1,27	0,25
6	1419	1,28	0,21

```
Num_Points = 500000000
```

Число потоков	Время исполнения (мс)	Ускорение	Эффективность
1	16396	1	1
2	15877	1,03	0,52
3	15005	1,09	0,36
4	13751	1,19	0,29
5	13566	1,21	0,24
6	13460	1,22	0,2

Strace:

```
$ strace -f ./client 2 50
```

```
execve("./client", ["/client", "2", "50"], 0x7fff4a47f088 /* 26 vars */) = 0
```

```
brk(NULL) = 0x562b3565f000
```

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =  
0x7fa9188ad000
```

```
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=20071, ...}) = 0

mmap(NULL, 20071, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa9188a8000

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\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) = 0x7fa918696000

mmap(0x7fa9186be000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fa9186be000

mmap(0x7fa918846000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x7fa918846000

mmap(0x7fa918895000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7fa918895000

mmap(0x7fa91889b000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fa91889b000

close(3) = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fa918693000

arch_prctl(ARCH_SET_FS, 0x7fa918693740) = 0

set_tid_address(0x7fa918693a10) = 58965

set_robust_list(0x7fa918693a20, 24) = 0

rseq(0x7fa918694060, 0x20, 0, 0x53053053) = 0

mprotect(0x7fa918895000, 16384, PROT_READ) = 0

mprotect(0x562b1b5ea000, 4096, PROT_READ) = 0

mprotect(0x7fa9188e5000, 8192, PROT_READ) = 0

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

munmap(0x7fa9188a8000, 20071) = 0

getrandom("\xa9\xae\x34\x4b\x7a\x80\x70\x27", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x562b3565f000

brk(0x562b35680000) = 0x562b35680000

```

```

rt_sigaction(SIGRT_1, {sa_handler=0x7fa91872f520, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7fa9186db320},
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) =
0x7fa917e92000

mprotect(0x7fa917e93000, 8388608, PROT_READ|PROT_WRITE) = 0

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

clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM
|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7fa918692990,
parent_tid=0x7fa918692990, exit_signal=0, stack=0x7fa917e92000, stack_size=0x7fff80,
tls=0x7fa9186926c0}strace: Process 58966 attached

=> {parent_tid=[58966]}, 88) = 58966

[pid 58966] rseq(0x7fa918692fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 58965] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 58966] <... rseq resumed>) = 0

[pid 58965] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 58966] set_robust_list(0x7fa9186929a0, 24 <unfinished ...>

[pid 58965] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

[pid 58966] <... set_robust_list resumed>) = 0

[pid 58965] <... mmap resumed>) = 0x7fa917691000

[pid 58966] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 58965] mprotect(0x7fa917692000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 58966] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 58965] <... mprotect resumed>) = 0

[pid 58966] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 58965] futex(0x7fa9188e7a58, FUTEX_WAIT_PRIVATE, 2, NULL <unfinished ...>

[pid 58966] <... openat resumed>) = 3

[pid 58966] fstat(3, {st_mode=S_IFREG|0644, st_size=20071, ...}) = 0

[pid 58966] mmap(NULL, 20071, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fa9188a8000

[pid 58966] close(3) = 0

[pid 58966] mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fa90f691000

[pid 58966] munmap(0x7fa90f691000, 9891840) = 0

[pid 58966] munmap(0x7fa914000000, 57217024) = 0

[pid 58966] mprotect(0x7fa910000000, 135168, PROT_READ|PROT_WRITE) = 0

```

```
[pid 58966] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1",
O_RDONLY|O_CLOEXEC) = 3

[pid 58966] read(3,
"\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 58966] fstat(3, {st_mode=S_IFREG|0644, st_size=183024, ...}) = 0

[pid 58966] mmap(NULL, 185256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7fa917663000

[pid 58966] mmap(0x7fa917667000, 147456, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x7fa917667000

[pid 58966] mmap(0x7fa91768b000, 16384, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7fa91768b000

[pid 58966] mmap(0x7fa91768f000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x7fa91768f000

[pid 58966] close(3) = 0

[pid 58966] mprotect(0x7fa91768f000, 4096, PROT_READ) = 0

[pid 58966] futex(0x7fa9188e7a58, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 58965] <... futex resumed> = 0

[pid 58966] <... futex resumed> = 1

[pid 58965] futex(0x7fa9188e7a58, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 58966] munmap(0x7fa9188a8000, 20071 <unfinished ...>

[pid 58965] <... futex resumed> = 0

[pid 58966] <... munmap resumed> = 0

[pid 58965] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 58966] futex(0x7fa917690230, FUTEX_WAKE_PRIVATE, 2147483647 <unfinished ...>

[pid 58965] <... rt_sigprocmask resumed>[], 8) = 0

[pid 58966] <... futex resumed> = 0

[pid 58965]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM
|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7fa917e91990,
parent_tid=0x7fa917e91990, exit_signal=0, stack=0x7fa917691000, stack_size=0x7fff80,
tls=0x7fa917e916c0} <unfinished ...>

[pid 58966] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

strace: Process 58967 attached

[pid 58965] <... clone3 resumed> => {parent_tid=[58967]}, 88) = 58967

[pid 58966] madvise(0x7fa917e92000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 58965] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 58967] rseq(0x7fa917e91fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 58965] <... rt_sigprocmask resumed>NULL, 8) = 0
```

```

[pid 58966] <... madvise resumed>)          = 0

[pid 58965] futex(0x7fa918692990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 58966,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 58967] <... rseq resumed>)              = 0

[pid 58966] exit(0 <unfinished ...>

[pid 58967] set_robust_list(0x7fa917e919a0, 24 <unfinished ...>

[pid 58966] <... exit resumed>)              = ?

[pid 58967] <... set_robust_list resumed>) = 0

[pid 58966] +++ exited with 0 +++

[pid 58965] <... futex resumed>)              = 0

[pid 58967] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 58965] futex(0x7fa917e91990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 58967,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 58967] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 58967] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

[pid 58967] madvise(0x7fa917691000, 8368128, MADV_DONTNEED) = 0

[pid 58967] exit(0)                          = ?

[pid 58965] <... futex resumed>)              = 0

[pid 58967] +++ exited with 0 +++

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

write(1, "Min = -992\n", 11Min = -992

)

= 11

write(1, "Max = 941\n", 10Max = 941

)

= 10

exit_group(0)                                = ?

+++ exited with 0 +++

```

Вывод

В результате выполнения лабораторной работы удалось познакомиться с многопоточным программированием. Программа успешно реализует задачу обработки данных в многопоточном режиме с использованием стандартных средств операционной системы. Были изучены базовые системные вызовы для создания и управления потоками.