Лабораторная работа №3 по курсу "Операционные системы"

Студент группы: М80-207Б-21, Крючков Артемий Владимирович\ Контакты: artemkr2003@mail.ru\ Работа выполнена: 17.09.2022\ Преподаватель: Миронов Евгений Сергеевич

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.

Вариант 2

Отсортировать массив целых чисел при помощи параллельного алгоритма быстрой сортировки

Методы и алгоритмы решения

```c++
# include <pthread.h>

# include <stdlib.h>

# include <sys/time.h>

# include <time.h>

# include <unistd.h>

# include

# include

using namespace std;

class QS { public: int* array_t; int size_t; int thread_t; QS** threads_t;
```

```
 QS(int* array, int size, QS** threads, int thread_id)
{
    array_t = array;
    size_t = size;
    threads_t = threads;
    thread_t = thread_id;
    threads[thread_id] = this;
}
~QS()
{
    threads_t[thread_t] = NULL;
}
```

};

void qsort(int* array, const unsigned int size) { if (size <= 50) { for (unsigned int i = 1; i < size; i++) { int temp = array[i]; unsigned int j = i;

```
         while (j > 0 && temp < array[j - 1]) {
            array[j] = array[j - 1];
            j--;
        }

        array[j] = temp;
    }
} else {
    int pivot = array[size / 2];
    int* left = array;
    int* right = array + size - 1;

    while (true) {
        while ((left <= right) && (*left < pivot))
            left++;

        while ((left <= right) && (*right > pivot))
            right--;

        if (left > right)
            break;

        int temp = *left;
        *left = *right;
        left++;
        *right = temp;
        right--;
    }

    qsort(array, right - array + 1);
    qsort(left, array + size - left);
}
```

}

void* qsort_thread(void* obj) { qsort(((QS*)obj)->array_t, ((QS*)obj)->size_t); delete ((QS*)obj); return NULL; }

int main(int argc, char** argv) { int maxCountElements = 2000; int maxCountThreads = 12;

```
 printf("Введите кол-во элементов массива: ");
scanf("%d", &maxCountElements);
printf("Введите кол-во потоков: ");
scanf("%d", &maxCountThreads);

if (maxCountThreads < 1)
    maxCountThreads = 1;

int* array = new int[maxCountElements];
int len = maxCountElements / maxCountThreads;

srand(clock());

QS** threads = new QS*[maxCountThreads];

struct timeval tv;
gettimeofday(&tv, NULL);
double time_mil = (tv.tv_sec) * 1000. + (tv.tv_usec) / 1000.;




for (int i = 0; i < maxCountElements; i++) {
    array[i] = rand() % 2000;
}

for (int i = 0, ai = 0; i < maxCountThreads; i++, ai += len) {
    threads[i] = NULL;
    pthread_t t;

    int size = len + (i == (maxCountThreads - 1) ? (maxCountElements % maxCountThreads) : 0);

    pthread_create(&t, 0, qsort_thread, new QS(&array[ai], size, threads, i));
}

int i = 0;
while (i < maxCountThreads) {
    if (threads[i])
        continue;
    i++;
}


if (maxCountThreads > 1) {
    qsort(array, maxCountElements);
}
```

```
gettimeofday(&tv, NULL);
double time_mil_ = (tv.tv_sec) * 1000. + (tv.tv_usec) / 1000.;
cout << "\nSorted in " << (time_mil_ - time_mil) << " ms";

int last = 0;
for (int i = 0; i < maxCountElements; i++) {
    if (array[i] < last) {
        cout << "\n\nArray isn't Sorted";
        delete[] array;
        delete[] threads;
        return 0;
    }

    last = array[i];
}

cout << "\n\nArray Sorted\n";

delete[] array;
delete[] threads;

return 0;
```

}

Выполнение программы

```txt
Введите кол-во элементов массива: 10000000
Введите кол-во потоков: 1

Sorted in 961.037 ms

Array Sorted
```

```
 Введите кол-во элементов массива: 10000000
Введите кол-во потоков: 12

Sorted in 822.158 ms

Array Sorted
```

Strace

```
 strace ./main
```

```
execve("./main", ["./main"], 0x7ffd9a55eee0 /* 70 vars */) = 0
brk(NULL)                               = 0x557cd6009000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffe296af40) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=158379, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 158379, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fdc74cc7000
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=19198496, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc74cc5000
mmap(NULL, 2320384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdc74a8e000
mmap(0x7fdc74b27000, 1138688, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x99000) = (
mmap(0x7fdc74c3d000, 487424, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1af000) = 0x7fdc74c3(
mmap(0x7fdc74cb4000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x225000) = (
mmap(0x7fdc74cc2000, 10240, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdc
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=944600, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 946368, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdc749a6000
mmap(0x7fdc749b4000, 499712, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x
mmap(0x7fdc74a2e000, 385024, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x88000) = 0x7fdc74a2e(
mmap(0x7fdc74a8c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe5000) = 0x
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=571848, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 127304, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdc74986000
mmap(0x7fdc74989000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7
mmap(0x7fdc749a0000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7fdc749a00(
mmap(0x7fdc749a4000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x
close(3)                                = 0
openat(AT_FDCWD, "/usr/lib/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P4\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
newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=1953472, ...}, AT_EMPTY_PATH) = 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, 1994384, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fdc7479f000
mmap(0x7fdc747c1000, 1421312, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = (
mmap(0x7fdc7491c000, 356352, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x17d000) = 0x7fdc7491(
mmap(0x7fdc74973000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d4000) = (
mmap(0x7fdc74979000, 52880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdc
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc7479d000
arch_prctl(ARCH_SET_FS, 0x7fdc7479e200) = 0
set_tid_address(0x7fdc7479e4d0)         = 5348
set_robust_list(0x7fdc7479e4e0, 24)     = 0
rseq(0x7fdc7479eb20, 0x20, 0, 0x53053053) = 0
mprotect(0x7fdc74973000, 16384, PROT_READ) = 0
```

```
mprotect(0x7fdc749a4000, 4096, PROT_READ) = 0
mprotect(0x7fdc74a8c000, 4096, PROT_READ) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc7479b000
mprotect(0x7fdc74cb4000, 53248, PROT_READ) = 0
mprotect(0x557cd566d000, 4096, PROT_READ) = 0
mprotect(0x7fdc74d1f000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7fdc74cc7000, 158379)          = 0
getrandom("\x66\x5f\xa4\xfe\xae\x95\x13\x6d", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x557cd6009000
brk(0x557cd602a000)                     = 0x557cd602a000
futex(0x7fdc74cc26bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}, AT_EMPTY_PATH) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\273-\320\262\3
read(0, 10000000
"10000000\n", 1024)             = 9
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\273-\320\262\3
read(0, 12
"12\n", 1024)                   = 3
mmap(NULL, 40001536, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdc72175000
clock_gettime(CLOCK_PROCESS_CPUTIME_ID, {tv_sec=0, tv_nsec=2441263}) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7fdc74822d00, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTA
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc71974000
mprotect(0x7fdc71975000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc71173000
mprotect(0x7fdc71174000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc70972000
mprotect(0x7fdc70973000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc70171000
mprotect(0x7fdc70172000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6f970000
mprotect(0x7fdc6f971000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6f16f000
mprotect(0x7fdc6f170000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6e96e000
mprotect(0x7fdc6e96f000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6e16d000
mprotect(0x7fdc6e16e000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6d96c000
mprotect(0x7fdc6d96d000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6d16b000
mprotect(0x7fdc6d16c000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6c96a000
mprotect(0x7fdc6c96b000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fdc6c169000
mprotect(0x7fdc6c16a000, 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|CLON
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
write(1, "\n", 1
)                       = 1
write(1, "Sorted in 811.767 ms\n\nArray Sort"..., 35Sorted in 811.767 ms

Array Sorted
) = 35
munmap(0x7fdc72175000, 40001536)        = 0
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Недопустимая операция смещения)
exit_group(0)                           = ?
+++ exited with 0 +++
```