

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)
Факультет Программной инженерии и компьютерной техники

Лабораторная работа №1
“Операционные системы”

Выполнили студенты
Группы Р33211
Просолович М.А.
Тайц Ю.М.
Преподаватель
Покид Александр Владимирович

Санкт-Петербург 2020

Задание

Лабораторная работа №1

Вариант:

A=276;B=0x33F2678F;C=mmap;D=77;E=37;F=block;G=120;H=random;I=91;J=max;K=cv

Разработать программу на языке C, которая осуществляет следующие действия

- Создает область памяти размером A мегабайт, начинающихся с адреса B (если возможно) при помощи C=(malloc, mmap) заполненную случайными числами /dev/urandom в D потоков. Используя системные средства мониторинга определите адрес начала в адресном пространстве процесса и характеристики выделенных участков памяти. Замеры виртуальной/физической памяти необходимо снять:
 1. До аллокации
 2. После аллокации
 3. После заполнения участка данными
 4. После деаллокации
- Записывает область памяти в файлы одинакового размера E мегабайт с использованием F=(блочного, кешируемого) обращения к диску. Размер блока ввода-вывода G байт. Преподаватель выдает в качестве задания последовательность записи/чтения блоков H=(последовательный, заданный или случайный)
- Генерацию данных и запись осуществлять в бесконечном цикле.
- В отдельных I потоках осуществлять чтение данных из файлов и подсчитывать агрегированные характеристики данных - J=(сумму, среднее значение, максимальное, минимальное значение).
- Чтение и запись данных в/из файла должна быть защищена примитивами синхронизации K=(mutex, cv, sem, flock).
- По заданию преподавателя изменить приоритеты потоков и описать изменения в характеристиках программы.

Для запуска программы возможно использовать операционную систему Windows 10 или Debian/Ubuntu в виртуальном окружении.

Измерить значения затраченного процессорного времени на выполнение программы и на операции ввода-вывода используя системные утилиты.

Отследить трассу системных вызовов.

Используя star построить графики системных характеристик.

Код программы

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

//A=276;B=0x33F2678F;C=mmap;D=77;E=37;F=block;G=120;H=random;I=91;J=max;K=c
v
#define ALLOCATED_MEMORY_SIZE_MB 276
#define START_ADDRESS 0x33F2678F
#define WRITE_IN_MEMORY_THREADS_CNT 77
#define FILES_SIZE_MB 37
#define IO_BUFFER_SIZE_B 120
#define READ_FILES_THREADS_CNT 91

#define INT_MIN -2147483648

typedef struct {
    FILE* src;
    int length;
    unsigned char* start;
} WriteInMemoryProps;

typedef struct {
    unsigned char* src;
    char* fileName;
    pthread_mutex_t* mutex;
    pthread_cond_t* cv;
} WriteInFileProps;

typedef struct {
    char* fileName;
    pthread_mutex_t* mutex;
    pthread_cond_t* cv;
} ReadFileProps;

void* writeInMemory(void* props) {
    WriteInMemoryProps* data = props;
    unsigned char* start = data->start;
    int length = data->length;
    FILE* src = data->src;
    size_t i = 0;
    while (i < length) {
        i += fread(start + i, 1, length - i, src);
    }
    return NULL;
}

//writing data from /dev/urandom in memory beginning from start address
void writeInMemoryFromUrandom(unsigned char* startAddress) {
    FILE* urandom = fopen("/dev/urandom", "r");

    int wrn_part = ALLOCATED_MEMORY_SIZE_MB * 1024 * 1024 /
WRITE_IN_MEMORY_THREADS_CNT;

    pthread_t writeInMemoryThreads[WRITE_IN_MEMORY_THREADS_CNT];
```

```

    for (int i = 0; i < WRITE_IN_MEMORY_THREADS_CNT - 1; i++) {
        WriteInMemoryProps* props = malloc(sizeof(WriteInMemoryProps));
        props->start = startAddress;
        props->length = wrn_part;
        props->src = urandom;
        pthread_create(&(writeInMemoryThreads[i]), NULL, writeInMemory,
props);
        startAddress += wrn_part;
    }

    WriteInMemoryProps* lastThreadProps =
malloc(sizeof(WriteInMemoryProps));
    lastThreadProps->start = startAddress;
    lastThreadProps->length = wrn_part + ALLOCATED_MEMORY_SIZE_MB * 1024 *
1024 % WRITE_IN_MEMORY_THREADS_CNT;
    lastThreadProps->src = urandom;
    pthread_create(&(writeInMemoryThreads[WRITE_IN_MEMORY_THREADS_CNT -
1]), NULL, writeInMemory, lastThreadProps);

    for (int i = 0; i < WRITE_IN_MEMORY_THREADS_CNT; i++)
        pthread_join(writeInMemoryThreads[i], NULL);

    fclose(urandom);
}

_Noreturn void* infiniteGenerating(void* startAddress) {
    while (1) {
        writeInMemoryFromUrandom(startAddress);
    }
}

double generateRandom() {
    return (double)rand() / (double)RAND_MAX ;
}

void writeInFile(unsigned char* src, char* fileName, pthread_mutex_t*
mutex, pthread_cond_t* cv) {
    pthread_mutex_lock(mutex);
    FILE* file = fopen(fileName, "wb");
    int file_size = FILES_SIZE_MB * 1024 * 1024;
    for (int j = 0; j < file_size / IO_BUFFER_SIZE_B; j++) {
        int blockNumber = (int) (generateRandom() * file_size /
IO_BUFFER_SIZE_B);
        int blockSize = IO_BUFFER_SIZE_B;
        if (blockNumber == (file_size / IO_BUFFER_SIZE_B) && file_size %
IO_BUFFER_SIZE_B != 0) {
            //this is a last block, and we should adjust block size
            blockSize = file_size % IO_BUFFER_SIZE_B;
        }
        fwrite(src + blockNumber * IO_BUFFER_SIZE_B, 1, blockSize, file);
    }
    fclose(file);
    printf("Data generated for %s\n", fileName);
    pthread_cond_broadcast(cv);
    pthread_mutex_unlock(mutex);
}

_Noreturn void* writeFromMemoryToFile(void* props) {
    WriteInFileProps* args = props;
    while (1) {
        writeInFile(args->src, args->fileName, args->mutex, args->cv);
    }
}

```

```

    }
}

int find_max(int* begin, int length) {
    int max = INT_MIN;
    for (int* i = begin; i < begin + length; i++) {
        int val = *i;
        if (val > max)
            max = val;
    }
    return max;
}

_Noreturn void* readFile(void* props) {
    ReadFileProps* args = props;
    char* fileName = args->fileName;
    while (1) {
        int max = INT_MIN;
        pthread_mutex_lock(args->mutex);
        printf("Waiting on cv %s \n", fileName);
        pthread_cond_wait(args->cv, args->mutex);
        FILE* file = fopen(fileName, "rb");
        unsigned char buf[IO_BUFFER_SIZE_B];
        int file_size = FILES_SIZE_MB * 1024 * 1024;
        for (int i = 0; i < file_size / IO_BUFFER_SIZE_B; i++) {
            int blockNumber = (int) (generateRandom() * file_size /
IO_BUFFER_SIZE_B);
            int blockSize = IO_BUFFER_SIZE_B;
            if (blockNumber == (file_size / IO_BUFFER_SIZE_B) && file_size
% IO_BUFFER_SIZE_B != 0) {
                //this is a last block, and we should adjust block size
                blockSize = file_size % IO_BUFFER_SIZE_B;
            }
            fseek(file, blockNumber * IO_BUFFER_SIZE_B, SEEK_SET);
            fread(&buf, 1, blockSize, file);
            int localMax = find_max((int* ) &buf[0], IO_BUFFER_SIZE_B / 4);
            if (localMax > max)
                max = localMax;
        }
        printf("Max in %s: %d\n", fileName, max);
        fclose(file);
        pthread_mutex_unlock(args->mutex);
    }
}

unsigned char* allocate_memory() {
    return mmap((void* ) START_ADDRESS, ALLOCATED_MEMORY_SIZE_MB * 1024 *
1024, PROT_READ | PROT_WRITE,
                MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
}

void allocateDeallocateMemory() {
    printf("Before memory allocation.");
    getchar();
    unsigned char* allocatedStartAddress = allocate_memory();
    printf("After memory allocation.");
    getchar();
    writeInMemoryFromUrandom(allocatedStartAddress);
    printf("After memory filling.");
    getchar();
    munmap(allocatedStartAddress, ALLOCATED_MEMORY_SIZE_MB * 1024 * 1024);
}

```

```

        printf("After memory deallocation.");
        getchar();
    }

    char* fileName;
    char* getFileNameByNumber(int fileNumber) {
        fileName = malloc(sizeof(char) * 11);
        snprintf(fileName, 11, "file_%d.bin", fileNumber);
        return fileName;
    }

    int main() {
        allocateDeallocateMemory();

        unsigned char* allocatedStartAddress = allocate_memory();

        const unsigned int filesCnt = ALLOCATED_MEMORY_SIZE_MB / FILES_SIZE_MB
+ 1;
        pthread_mutex_t mutexes[filesCnt];
        pthread_cond_t cvs[filesCnt];
        for (int i = 0; i < filesCnt; i++) {
            pthread_mutex_init(&(mutexes[i]), NULL);
            pthread_cond_init(&(cvs[i]), NULL);
        }

        pthread_t readThreads[READ_FILES_THREADS_CNT];
        int fileNumber = 0;
        for (int i = 0; i < READ_FILES_THREADS_CNT; i++) {
            if (fileNumber == filesCnt)
                fileNumber = 0;
            ReadFileProps* props = malloc(sizeof(ReadFileProps));
            props->fileName = getFileNameByNumber(fileNumber);
            props->mutex = &(mutexes[fileNumber]);
            props->cv = &(cvs[fileNumber]);
            pthread_create(&readThreads[i], NULL, readFile, props);
            fileNumber++;
        }

        pthread_t writeInMemory;
        pthread_create(&writeInMemory, NULL, infiniteGenerating,
allocatedStartAddress);

        pthread_t fileThreads[filesCnt];
        for (int i = 0; i < filesCnt; i++) {
            WriteInFileProps* props = malloc(sizeof(WriteInFileProps));
            props->src = allocatedStartAddress;
            props->fileName = getFileNameByNumber(i);
            props->mutex = &(mutexes[i]);
            props->cv = &(cvs[i]);
            pthread_create(&fileThreads[i], NULL, writeFromMemoryToFile,
props);
        }

        for (int i = 0; i < READ_FILES_THREADS_CNT; i++)
            pthread_join(readThreads[i], NULL);
        pthread_join(writeInMemory, NULL);
        for (int i = 0; i < filesCnt; i++) {
            pthread_join(fileThreads[i], NULL);
        }
    }

```

```
return 0;
}
```

Сбор статистики

Before allocation

```
mihail@mihail-W65-67SF:~$ pmap -x 11085
11085: ./main
Address          Kbytes      RSS      Dirty  Mode  Mapping
000055b9e3baa000      4         4        0 r---- main
000055b9e3bab000      8         8        0 r-x-- main
000055b9e3bad000      4         4        0 r---- main
000055b9e3bae000      4         4        4 r---- main
000055b9e3baf000      4         4        4 rw--- main
000055b9e4ce4000    132         4        4 rw--- [ anon ]
00007f8e78b70000     12         8        8 rw--- [ anon ]
00007f8e78b73000    148        144        0 r---- libc-2.31.so
00007f8e78b98000   1504        788        0 r-x-- libc-2.31.so
00007f8e78d10000    296        124        0 r---- libc-2.31.so
00007f8e78d5a000      4          0        0 ----- libc-2.31.so
00007f8e78d5b000     12         12       12 r---- libc-2.31.so
00007f8e78d5e000     12         12       12 rw--- libc-2.31.so
00007f8e78d61000     16         16       16 rw--- [ anon ]
00007f8e78d65000     28         28        0 r---- libpthread-2.31.so
00007f8e78d6c000     68         68        0 r-x-- libpthread-2.31.so
00007f8e78d7d000     20          0        0 r---- libpthread-2.31.so
00007f8e78d82000      4         4        4 r---- libpthread-2.31.so
00007f8e78d83000      4         4        4 rw--- libpthread-2.31.so
00007f8e78d84000     24         12       12 rw--- [ anon ]
00007f8e78da3000      4         4        0 r---- ld-2.31.so
00007f8e78da4000    140        140        0 r-x-- ld-2.31.so
00007f8e78dc7000     32         32        0 r---- ld-2.31.so
00007f8e78dd0000      4         4        4 r---- ld-2.31.so
00007f8e78dd1000      4         4        4 rw--- ld-2.31.so
00007f8e78dd2000      4         4        4 rw--- [ anon ]
00007ffceb8ff000    132         16       16 rw--- [ stack ]
00007ffceb9da000     12          0        0 r---- [ anon ]
00007ffceb9dd000      4         4        0 r-x-- [ anon ]
ffffffffffff600000      4          0        0 --x-- [ anon ]
-----
total kB          2648      1456      108
```



```
top - 00:23:02 up 1:04, 1 user, load average: 0,10, 0,27, 0,47
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,2 us, 0,0 sy, 0,0 ni, 99,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 7845,6 total, 1316,4 free, 3423,6 used, 3105,6 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 3837,5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11085	mihaill	20	0	2644	564	480	S	0,0	0,0	0:00.00	main

After allocation

```
mihaill@mihaill-W65-675F:~$ pmap -x 11085
11085: ./main
Address          Kbytes      RSS      Dirty Mode  Mapping
00000000033f26000 282624        0         0 rw--- [ anon ]
000055b9e3baa000      4          4         0 r---- main
000055b9e3bab000      8          8         0 r-x-- main
000055b9e3bad000      4          4         0 r---- main
000055b9e3bae000      4          4         4 r---- main
000055b9e3baf000      4          4         4 rw--- main
000055b9e4ce4000    132          4         4 rw--- [ anon ]
00007f8e78b70000     12          8         8 rw--- [ anon ]
00007f8e78b73000    148        144         0 r---- libc-2.31.so
00007f8e78b98000   1504        848         0 r-x-- libc-2.31.so
00007f8e78d10000    296        124         0 r---- libc-2.31.so
00007f8e78d5a000      4          0         0 ----- libc-2.31.so
00007f8e78d5b000     12         12        12 r---- libc-2.31.so
00007f8e78d5e000     12         12        12 rw--- libc-2.31.so
00007f8e78d61000     16         16        16 rw--- [ anon ]
00007f8e78d65000     28         28         0 r---- libpthread-2.31.so
00007f8e78d6c000     68         68         0 r-x-- libpthread-2.31.so
00007f8e78d7d000     20          0         0 r---- libpthread-2.31.so
00007f8e78d82000      4          4         4 r---- libpthread-2.31.so
00007f8e78d83000      4          4         4 rw--- libpthread-2.31.so
00007f8e78d84000     24         12        12 rw--- [ anon ]
00007f8e78da3000      4          4         0 r---- ld-2.31.so
00007f8e78da4000    140        140         0 r-x-- ld-2.31.so
00007f8e78dc7000     32         32         0 r---- ld-2.31.so
00007f8e78dd0000      4          4         4 r---- ld-2.31.so
00007f8e78dd1000      4          4         4 rw--- ld-2.31.so
00007f8e78dd2000      4          4         4 rw--- [ anon ]
00007ffceb8ff000    132         16        16 rw--- [ stack ]
00007ffceb9da000     12          0         0 r---- [ anon ]
00007ffceb9dd000      4          4         0 r-x-- [ anon ]
fffffffffff60000      4          0         0 --x-- [ anon ]
-----
total kB          285272    1516    108
```



```

mihaill@mihaill-W65-675F:~$ pmap -x 11085
11085:  ./main
Address      Kbytes      RSS      Dirty Mode  Mapping
0000000033f26000 282624      0         0 rw--- [ anon ]
000055b9e3baa000      4          4         0 r---- main
000055b9e3bab000      8          8         0 r-x-- main
000055b9e3bad000      4          4         0 r---- main
000055b9e3bae000      4          4         4 r---- main
000055b9e3baf000      4          4         4 rw--- main
000055b9e4ce4000     132         4         4 rw--- [ anon ]
00007f8e78b70000      12          8         8 rw--- [ anon ]
00007f8e78b73000     148        144         0 r---- libc-2.31.so
00007f8e78b98000    1504        848         0 r-x-- libc-2.31.so
00007f8e78d10000     296        124         0 r---- libc-2.31.so
00007f8e78d5a000       4          0         0 ----- libc-2.31.so
00007f8e78d5b000      12         12        12 r---- libc-2.31.so
00007f8e78d5e000      12         12        12 rw--- libc-2.31.so
00007f8e78d61000      16         16        16 rw--- [ anon ]
00007f8e78d65000      28         28         0 r---- libpthread-2.31.so
00007f8e78d6c000      68         68         0 r-x-- libpthread-2.31.so
00007f8e78d7d000      20          0         0 r---- libpthread-2.31.so
00007f8e78d82000       4          4         4 r---- libpthread-2.31.so
00007f8e78d83000       4          4         4 rw--- libpthread-2.31.so
00007f8e78d84000      24         12        12 rw--- [ anon ]
00007f8e78da3000       4          4         0 r---- ld-2.31.so
00007f8e78da4000     140        140         0 r-x-- ld-2.31.so
00007f8e78dc7000      32         32         0 r---- ld-2.31.so
00007f8e78dd0000       4          4         4 r---- ld-2.31.so
00007f8e78dd1000       4          4         4 rw--- ld-2.31.so
00007f8e78dd2000       4          4         4 rw--- [ anon ]
00007ffceb8ff000     132         16        16 rw--- [ stack ]
00007ffceb9da000      12          0         0 r---- [ anon ]
00007ffceb9dd000       4          4         0 r-x-- [ anon ]
fffffffffff6000000      4          0         0 --x-- [ anon ]
-----
total kB      285272      1516      108

```

After filling

```

mihail@mihail-W65-67SF:~$ pmap -x 11085
11085:  ./main
Address          Kbytes      RSS      Dirty Mode  Mapping
0000000033f26000 282624    282624    282624 rw---  [ anon ]
000055b9e3baa000      4         4         0 r----  main
000055b9e3bab000      8         8         0 r-x--  main
000055b9e3bad000      4         4         0 r----  main
000055b9e3bae000      4         4         4 r----  main
000055b9e3baf000      4         4         4 rw---  main
000055b9e4ce4000    132        28        28 rw---  [ anon ]
00007f8e4dfbc000      4         0         0 -----  [ anon ]
00007f8e4dfbd000    8192         8         8 rw---  [ anon ]
00007f8e4e7bd000      4         0         0 -----  [ anon ]
00007f8e4e7be000    8192         8         8 rw---  [ anon ]
00007f8e4efbe000      4         0         0 -----  [ anon ]
00007f8e4efbf000    8192         8         8 rw---  [ anon ]
00007f8e4f7bf000      4         0         0 -----  [ anon ]
00007f8e4f7c0000    8192         8         8 rw---  [ anon ]
00007f8e70000000    132         8         8 rw---  [ anon ]
00007f8e70021000   65404         0         0 -----  [ anon ]
00007f8e78b70000     12         8         8 rw---  [ anon ]
00007f8e78b73000    148        144         0 r----  libc-2.31.so
00007f8e78b98000    1504        976         0 r-x--  libc-2.31.so
00007f8e78d10000    296        124         0 r----  libc-2.31.so
00007f8e78d5a000      4         0         0 -----  libc-2.31.so
00007f8e78d5b000     12         12        12 r----  libc-2.31.so
00007f8e78d5e000     12         12        12 rw---  libc-2.31.so
00007f8e78d61000     16         16        16 rw---  [ anon ]
00007f8e78d65000     28         28         0 r----  libpthread-2.31.so

```

```

top - 00:25:01 up 1:06, 1 user, load average: 0,18, 0,28, 0,45
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,2 us, 0,1 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 7845,6 total, 1015,7 free, 3723,5 used, 3106,4 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 3538,7 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11085	mihail	20	0	383588	284332	1536	S	0,0	3,5	0:03.57	main

After deallocation


```
mihail@mihail-W65-67SF:~$ pmap -x 11085
```

```
11085: ./main
```

Address	Kbytes	RSS	Dirty	Mode	Mapping
000055b9e3baa000	4	4	0	r----	main
000055b9e3bab000	8	8	0	r-x--	main
000055b9e3bad000	4	4	0	r----	main
000055b9e3bae000	4	4	4	r----	main
000055b9e3baf000	4	4	4	rw---	main
000055b9e4ce4000	132	28	28	rw---	[anon]
00007f8e4dfbc000	4	0	0	-----	[anon]
00007f8e4dfbd000	8192	8	8	rw---	[anon]
00007f8e4e7bd000	4	0	0	-----	[anon]
00007f8e4e7be000	8192	8	8	rw---	[anon]
00007f8e4efbe000	4	0	0	-----	[anon]
00007f8e4efbf000	8192	8	8	rw---	[anon]
00007f8e4f7bf000	4	0	0	-----	[anon]
00007f8e4f7c0000	8192	8	8	rw---	[anon]
00007f8e70000000	132	8	8	rw---	[anon]
00007f8e70021000	65404	0	0	-----	[anon]
00007f8e78b70000	12	8	8	rw---	[anon]
00007f8e78b73000	148	144	0	r----	libc-2.31.so
00007f8e78b98000	1504	976	0	r-x--	libc-2.31.so
00007f8e78d10000	296	124	0	r----	libc-2.31.so
00007f8e78d5a000	4	0	0	-----	libc-2.31.so
00007f8e78d5b000	12	12	12	r----	libc-2.31.so
00007f8e78d5e000	12	12	12	rw---	libc-2.31.so
00007f8e78d61000	16	16	16	rw---	[anon]
00007f8e78d65000	28	28	0	r----	libpthread-2.31.so
00007f8e78d6c000	68	68	0	r-x--	libpthread-2.31.so
00007f8e78d7d000	20	0	0	r----	libpthread-2.31.so
00007f8e78d82000	4	4	4	r----	libpthread-2.31.so
00007f8e78d83000	4	4	4	rw---	libpthread-2.31.so
00007f8e78d84000	24	12	12	rw---	[anon]
00007f8e78da3000	4	4	0	r----	ld-2.31.so

```
top - 00:25:11 up 1:06, 1 user, load average: 0,22, 0,28, 0,45
Tasks: 1 total, 0 running, 1 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1,7 us, 0,0 sy, 0,0 ni, 98,3 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 7845,6 total, 1290,2 free, 3447,7 used, 3107,7 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 3813,6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11085	mihail	20	0	100964	1708	1536	S	0,0	0,0	0:03.58	main

Strace:

```
execve("./main", ["/main"], 0x7ffff5e1ed40 /* 60 vars */) = 0
brk(NULL) = 0x557655443000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffd5fa6670) = -1 EINVAL (Invalid argument)
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=98934, ...}) = 0
mmap(NULL, 98934, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f9920025000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0"..., 832) = 832
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\00\305\3743\364B\2216\244\224\306@\261\23\327o"..., 68, 824) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f9920023000
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\00\305\3743\364B\2216\244\224\306@\261\23\327o"..., 68, 824) = 68
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f9920000000
mmap(0x7f9920007000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f9920007000
mmap(0x7f9920018000, 20480, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f9920018000
mmap(0x7f992001d000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7f992001d000
mmap(0x7f992001f000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f992001f000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\360q\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"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\02\0\0\0300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 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"..., 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\02\0\0\0300\4\0\0\0\3\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0363\377?\332\200\270\27\304d\245n\355Y\377\t\334"..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f991fe0e000
mprotect(0x7f991fe33000, 1847296, PROT_NONE) = 0
mmap(0x7f991fe33000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f991fe33000
mmap(0x7f991ffab000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f991ffab000
mmap(0x7f991fff6000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f991fff6000
mmap(0x7f991fffc000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f991fffc000
close(3) = 0
```

```

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f991fe0b000
arch_prctl(ARCH_SET_FS, 0x7f991fe0b740) = 0
mprotect(0x7f991fff6000, 12288, PROT_READ) = 0
mprotect(0x7f992001d000, 4096, PROT_READ) = 0
mprotect(0x557654d93000, 4096, PROT_READ) = 0
mprotect(0x7f992006b000, 4096, PROT_READ) = 0
munmap(0x7f9920025000, 98934) = 0
set_tid_address(0x7f991fe0ba10) = 16150
set_robust_list(0x7f991fe0ba20, 24) = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f9920007bf0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO,
sa_restorer=0x7f99200153c0}, NULL, 8) = 0
clone(child_stack=0x7f991ee07fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM
|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[16153],
tls=0x7f991ee08700, child_tidptr=0x7f991ee089d0) = 16153
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f991de07000
mprotect(0x7f991de08000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f991e606fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM
|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[16154],
tls=0x7f991e607700, child_tidptr=0x7f991e6079d0) = 16154
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f991d606000
mprotect(0x7f991d607000, 8388608, PROT_READ|PROT_WRITE) = 0
.....
futex(0x7f991fe0a9d0, FUTEX_WAIT, 16151, NULL) = 0
futex(0x7f991f6099d0, FUTEX_WAIT, 16152, NULL) = 0
futex(0x7f991ee089d0, FUTEX_WAIT, 16153, NULL) = 0
futex(0x7f991e6079d0, FUTEX_WAIT, 16154, NULL) = 0
futex(0x7f991de069d0, FUTEX_WAIT, 16155, NULL) = 0
munmap(0x7f991f60a000, 8392704) = 0
futex(0x7f991d6059d0, FUTEX_WAIT, 16156, NULL) = 0
munmap(0x7f991ee09000, 8392704) = 0
futex(0x7f991ce049d0, FUTEX_WAIT, 16157, NULL) = 0
munmap(0x7f991e608000, 8392704) = 0
futex(0x7f9917fff9d0, FUTEX_WAIT, 16158, NULL) = 0
munmap(0x7f991de07000, 8392704) = 0
futex(0x7f99177fe9d0, FUTEX_WAIT, 16159, NULL) = 0
munmap(0x7f991d606000, 8392704) = 0
futex(0x7f9916ffd9d0, FUTEX_WAIT, 16160, NULL) = 0
.....

```


Загрузка процессора во время выполнения:

```
top - 00:27:06 up 1:08, 1 user, load average: 6,40, 1,73, 0,91
Threads: 178 total, 3 running, 175 sleeping, 0 stopped, 0 zombie
%Cpu(s): 18,4 us, 33,8 sy, 0,0 ni, 11,4 id, 35,6 wa, 0,0 hi, 0,7 si, 0,0 st
MiB Mem : 7845,6 total, 1043,9 free, 3489,9 used, 3311,8 buff/cache
MiB Swap: 2048,0 total, 2048,0 free, 0,0 used. 3848,9 avail Mem
```

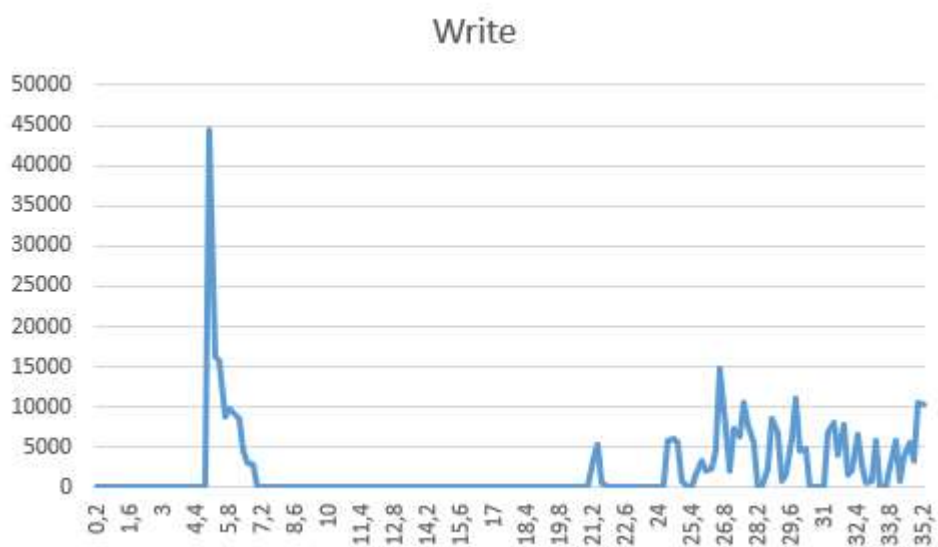
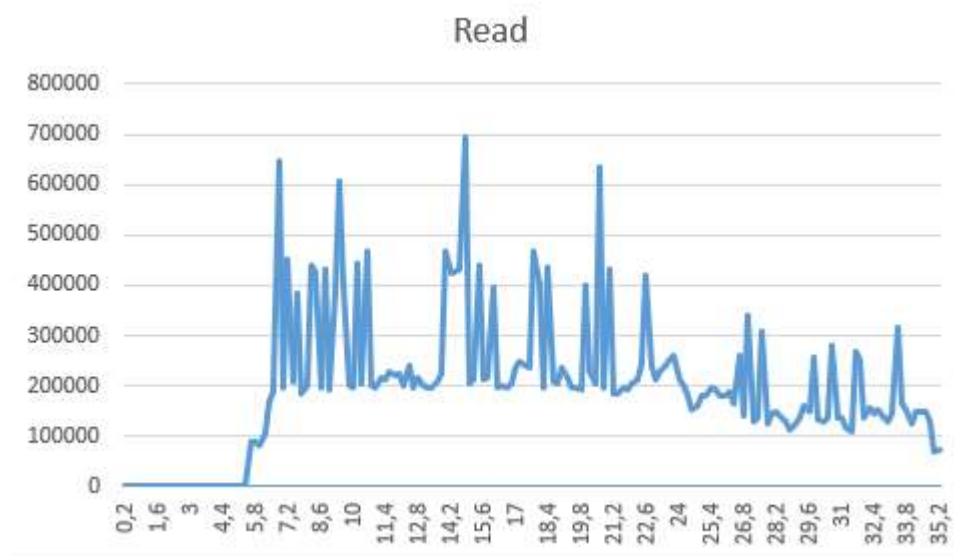
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11634	mihail	20	0	5864728	285688	1660	S	52,5	3,6	0:03.32	main
11575	mihail	20	0	5864728	285688	1660	S	42,9	3,6	0:01.29	main
11598	mihail	20	0	5864728	285688	1660	S	40,9	3,6	0:01.23	main
11583	mihail	20	0	5864728	285688	1660	S	36,2	3,6	0:03.12	main
11610	mihail	20	0	5864728	285688	1660	S	27,9	3,6	0:02.99	main
11618	mihail	20	0	5864728	285688	1660	R	19,9	3,6	0:02.29	main
11567	mihail	20	0	5864728	285688	1660	R	19,6	3,6	0:00.59	main
11650	mihail	20	0	5864728	285688	1660	D	16,6	3,6	0:03.54	main
11652	mihail	20	0	5864728	285688	1660	D	15,9	3,6	0:02.66	main
11657	mihail	20	0	5864728	285688	1660	D	14,0	3,6	0:07.25	main
11654	mihail	20	0	5864728	285688	1660	D	10,0	3,6	0:02.64	main
11606	mihail	20	0	5864728	285688	1660	S	5,0	3,6	0:01.40	main
11649	mihail	20	0	5864728	285688	1660	D	4,3	3,6	0:06.68	main
11659	mihail	20	0	5864728	285688	1660	D	2,0	3,6	0:03.09	main
12211	mihail	20	0	5864728	285688	1660	R	1,3	3,6	0:00.04	main
11648	mihail	20	0	5864728	285688	1660	S	0,3	3,6	0:00.02	main
11085	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:00.03	main
11557	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:00.00	main
11558	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:01.63	main
11559	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:01.61	main
11560	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:01.60	main
11561	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:01.79	main
11562	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:02.42	main
11563	mihail	20	0	5864728	285688	1660	S	0,0	3,6	0:00.00	main

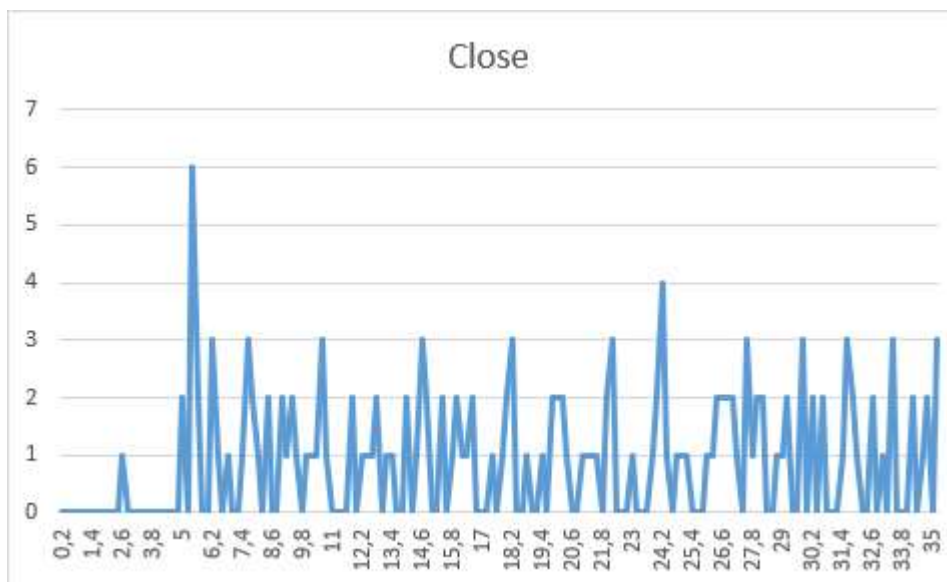
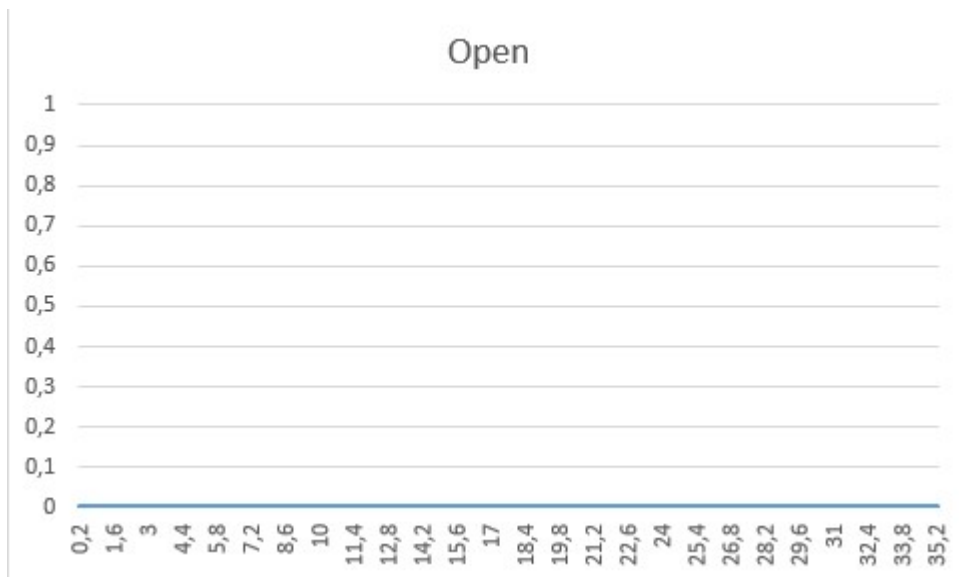
Статистика по вводу-выводу:

```
mihail@mihail-W65-67SF:~$ iostat
```

Linux 5.4.0-53-generic (mihail-W65-67SF)		24.11.2020		_x86_64_		(8 CPU)	
avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle	
	4,09	0,09	2,39	2,59	0,00	90,84	
Device	tps	kB_read/s	kB_wrtn/s	kB_dscd/s	kB_read	kB_wrtn	kB_dscd
loop0	39,19	39,43	0,00	0,00	167095	0	0
loop1	22,32	22,56	0,00	0,00	95611	0	0
loop10	2,23	2,30	0,00	0,00	9759	0	0
loop11	0,01	0,08	0,00	0,00	338	0	0
loop12	0,02	0,25	0,00	0,00	1054	0	0
loop13	0,01	0,25	0,00	0,00	1072	0	0
loop14	0,02	0,26	0,00	0,00	1088	0	0
loop15	0,02	0,26	0,00	0,00	1097	0	0
loop16	3,51	3,58	0,00	0,00	15190	0	0
loop17	0,00	0,00	0,00	0,00	8	0	0
loop2	0,01	0,25	0,00	0,00	1063	0	0
loop3	1,02	1,09	0,00	0,00	4627	0	0
loop4	1,08	1,32	0,00	0,00	5603	0	0
loop5	0,27	0,34	0,00	0,00	1442	0	0
loop6	0,01	0,08	0,00	0,00	355	0	0
loop7	0,01	0,08	0,00	0,00	342	0	0
loop8	1,15	1,39	0,00	0,00	5900	0	0
loop9	0,03	0,27	0,00	0,00	1157	0	0
sda	0,04	0,99	0,00	0,00	4204	0	0
sdb	19,41	362,50	3063,19	0,00	1536247	12981681	0

Графики системных вызовов, полученные с помощью star.





Вывод

В процессе работы мы разработали многопоточную программу на языке C, поработали с примитивами синхронизации. Поставили виртуальную машину на Windows и узнали основные команды для мониторинга системы в Linux. Написали скрипт для `systemtap`, с помощью которого построили графики системных характеристик.