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 P33211
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, некешируемого) 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=(futex, 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 stap 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=cv
#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;
    char* start;
} WriteInMemoryProps;

typedef struct {
    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;
    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(char* startAddress) {
    FILE* urandom = fopen("/dev/urandom", "r");

    int offset = 0;
    int writePart = 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 = writePart;
        props->src = urandom;
        pthread_create(&(writeInMemoryThreads[i]), NULL, writeInMemory, props);
        startAddress += writePart;
        offset += writePart;
    }

    WriteInMemoryProps* lastThreadProps = malloc(sizeof(WriteInMemoryProps));
    lastThreadProps->start = startAddress;
    lastThreadProps->length = writePart + ALLOCATED_MEMORY_SIZE_MB * 1024 * 1024 %
WRITE_IN_MEMORY_THREADS_CNT;
    lastThreadProps->src = urandom;
    offset += lastThreadProps->length;
    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);
    startAddress -= offset;
    fclose(urandom);
}

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

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

void writeInFile(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);
    }
}

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();
    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();

    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 writeThreads[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(&writeThreads[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(writeThreads[i], NULL);
    }

    return 0;
}
```

Сбор статистики

Before allocation

```
mihail@mihail-W65-67SF:~$ pmap 11245
11245:   ./main
000055e6a60d6000      4K r---- main
000055e6a60d7000      8K r-x-- main
000055e6a60d9000      4K r---- main
000055e6a60da000      4K r---- main
000055e6a60db000      4K rw--- main
000055e6a60fd000    132K rw---   [ anon ]
00007f9ee33ab000     12K rw---   [ anon ]
00007f9ee33ae000    148K r---- libc-2.31.so
00007f9ee33d3000   1504K r-x-- libc-2.31.so
00007f9ee354b000    296K r---- libc-2.31.so
00007f9ee3595000      4K ----- libc-2.31.so
00007f9ee3596000     12K r---- libc-2.31.so
00007f9ee3599000     12K rw--- libc-2.31.so
00007f9ee359c000     16K rw---   [ anon ]
00007f9ee35a0000     28K r---- libpthread-2.31.so
00007f9ee35a7000     68K r-x-- libpthread-2.31.so
00007f9ee35b8000     20K r---- libpthread-2.31.so
00007f9ee35bd000      4K r---- libpthread-2.31.so
00007f9ee35be000      4K rw--- libpthread-2.31.so
00007f9ee35bf000     24K rw---   [ anon ]
00007f9ee35de000      4K r---- ld-2.31.so
00007f9ee35df000    140K r-x-- ld-2.31.so
00007f9ee3602000     32K r---- ld-2.31.so
00007f9ee360b000      4K r---- ld-2.31.so
00007f9ee360c000      4K rw--- ld-2.31.so
00007f9ee360d000      4K rw---   [ anon ]
00007ffcbfaca000    132K rw---   [ stack ]
00007ffcbfbfb000     12K r----   [ anon ]
00007ffcbfbfe000      4K r-x--   [ anon ]
ffffffffff600000      4K --x--   [ anon ]
 total             2648K
```

```
top - 16:10:02 up  1:24,  1 user,  load average: 0,30, 0,55, 0,60
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1,6 us,  0,8 sy,  0,0 ni, 96,7 id,  0,0 wa,  0,0 hi,  0,8 si,  0,0 st
MiB Mem :   7845,6 total,    692,3 free,   3859,5 used,   3293,8 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   3255,0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11245 mihail    20   0    2644    564    480 S   0,0   0,0   0:00.00 main
```

After allocation

```
mihail@mihail-W65-67SF:~$ pmap 11245
11245:   ./main
0000000033f26000 282624K rw---   [ anon ]
000055e6a60d6000      4K r---- main
000055e6a60d7000      8K r-x-- main
000055e6a60d9000      4K r---- main
000055e6a60da000      4K r---- main
000055e6a60db000      4K rw--- main
000055e6a60fd000    132K rw---   [ anon ]
00007f9ee33ab000     12K rw---   [ anon ]
00007f9ee33ae000    148K r---- libc-2.31.so
00007f9ee33d3000   1504K r-x-- libc-2.31.so
00007f9ee354b000    296K r---- libc-2.31.so
00007f9ee3595000      4K ----- libc-2.31.so
00007f9ee3596000     12K r---- libc-2.31.so
00007f9ee3599000     12K rw--- libc-2.31.so
00007f9ee359c000     16K rw---   [ anon ]
00007f9ee35a0000     28K r---- libpthread-2.31.so
00007f9ee35a7000     68K r-x-- libpthread-2.31.so
00007f9ee35b8000     20K r---- libpthread-2.31.so
00007f9ee35bd000      4K r---- libpthread-2.31.so
00007f9ee35be000      4K rw--- libpthread-2.31.so
00007f9ee35bf000     24K rw---   [ anon ]
00007f9ee35de000      4K r---- ld-2.31.so
00007f9ee35df000    140K r-x-- ld-2.31.so
00007f9ee3602000     32K r---- ld-2.31.so
00007f9ee360b000      4K r---- ld-2.31.so
00007f9ee360c000      4K rw--- ld-2.31.so
00007f9ee360d000      4K rw---   [ anon ]
00007ffcbfaca000    132K rw---   [ stack ]
00007ffcbfbfb000     12K r----   [ anon ]
00007ffcbfbfe000      4K r-x--   [ anon ]
ffffffffff600000      4K --x--   [ anon ]
 total           285272K
```

```
top - 16:10:22 up  1:24,  1 user,  load average: 0,43, 0,56, 0,60
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1,7 us,  0,8 sy,  0,0 ni, 97,5 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   7845,6 total,    613,4 free,   3854,8 used,   3377,4 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   3176,4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11245 mihail    20   0  285268    564    480 S   0,0   0,0   0:00.00 main
```

After filling

```
mihail@mihail-W65-67SF:~$ pmap 11245
11245:   ./main
0000000033f26000 282624K rw---   [ anon ]
000055e6a60d6000      4K r---- main
000055e6a60d7000      8K r-x-- main
000055e6a60d9000      4K r---- main
000055e6a60da000      4K r---- main
000055e6a60db000      4K rw--- main
000055e6a60fd000    132K rw---   [ anon ]
00007f9eb87b9000      4K -----   [ anon ]
00007f9eb87ba000   8192K rw---   [ anon ]
00007f9eb8fba000      4K -----   [ anon ]
00007f9eb8fbb000   8192K rw---   [ anon ]
00007f9eb97bb000      4K -----   [ anon ]
00007f9eb97bc000   8192K rw---   [ anon ]
00007f9eb9fbc000      4K -----   [ anon ]
00007f9eb9fbd000   8192K rw---   [ anon ]
00007f9edc000000    132K rw---   [ anon ]
00007f9edc021000  65404K -----   [ anon ]
00007f9ee33ab000     12K rw---   [ anon ]
00007f9ee33ae000    148K r---- libc-2.31.so
00007f9ee33d3000   1504K r-x-- libc-2.31.so
00007f9ee354b000    296K r---- libc-2.31.so
00007f9ee3595000      4K ----- libc-2.31.so
00007f9ee3596000     12K r---- libc-2.31.so
00007f9ee3599000     12K rw--- libc-2.31.so
00007f9ee359c000     16K rw---   [ anon ]
00007f9ee35a0000     28K r---- libpthread-2.31.so
00007f9ee35a7000     68K r-x-- libpthread-2.31.so
00007f9ee35b8000     20K r---- libpthread-2.31.so
00007f9ee35bd000      4K r---- libpthread-2.31.so
00007f9ee35be000      4K rw--- libpthread-2.31.so
```

```
top - 16:10:40 up  1:24,  1 user,  load average: 0,38, 0,54, 0,59
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,8 us,  0,8 sy,  0,0 ni, 98,3 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   7845,6 total,    269,0 free,   4135,1 used,   3441,5 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   2832,4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11245 mihail    20   0  383588 284296   1500 S   0,0   3,5   0:03.52 main
```

After deallocation

```
mihail@mihail-W65-67SF:~$ pmap 11245
11245:   ./main
000055e6a60d6000      4K r---- main
000055e6a60d7000      8K r-x-- main
000055e6a60d9000      4K r---- main
000055e6a60da000      4K r---- main
000055e6a60db000      4K rw--- main
000055e6a60fd000    132K rw---   [ anon ]
00007f9eb87b9000      4K -----   [ anon ]
00007f9eb87ba000   8192K rw---   [ anon ]
00007f9eb8fba000      4K -----   [ anon ]
00007f9eb8fbb000   8192K rw---   [ anon ]
00007f9eb97bb000      4K -----   [ anon ]
00007f9eb97bc000   8192K rw---   [ anon ]
00007f9eb9fbc000      4K -----   [ anon ]
00007f9eb9fbd000   8192K rw---   [ anon ]
00007f9edc000000    132K rw---   [ anon ]
00007f9edc021000  65404K -----   [ anon ]
00007f9ee33ab000     12K rw---   [ anon ]
00007f9ee33ae000    148K r---- libc-2.31.so
00007f9ee33d3000   1504K r-x-- libc-2.31.so
00007f9ee354b000    296K r---- libc-2.31.so
00007f9ee3595000      4K ----- libc-2.31.so
00007f9ee3596000     12K r---- libc-2.31.so
00007f9ee3599000     12K rw--- libc-2.31.so
00007f9ee359c000     16K rw---   [ anon ]
00007f9ee35a0000     28K r---- libpthread-2.31.so
00007f9ee35a7000     68K r-x-- libpthread-2.31.so
```

```
top - 16:11:24 up  1:25,  1 user,  load average: 0,22, 0,48, 0,57
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,6 us,  0,1 sy,  0,0 ni, 99,2 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :   7845,6 total,    481,9 free,   3837,5 used,   3526,1 buff/cache
MiB Swap:   2048,0 total,   2048,0 free,      0,0 used.   3045,7 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11245 mihail    20   0  100964   1672   1500 S   0,0   0,0   0:03.54 main
```

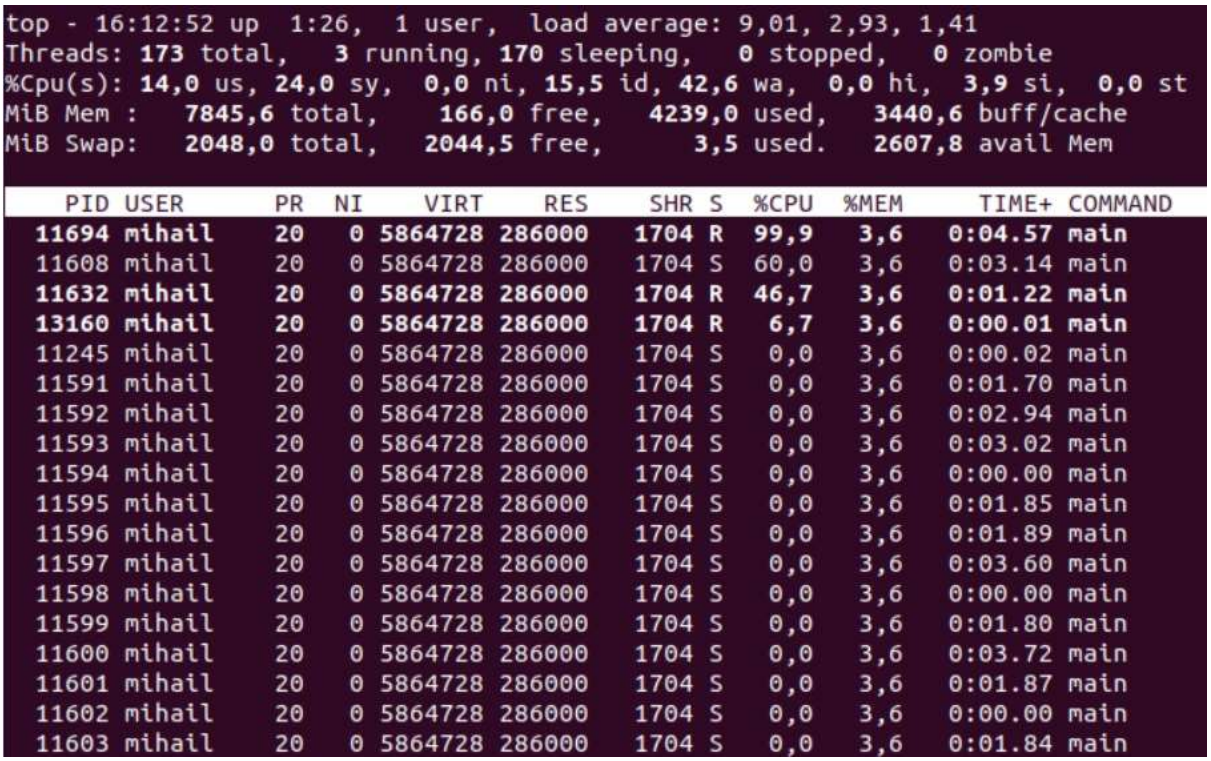
Strace:

execve("./main", ["./main"], 0x7ffd3fb2c7e0 / 60 vars */) = 0*
brk(NULL) = 0x5560cc4f1000
arch_prctl(0x3001 / ARCH_??? */, 0x7ffed3f2e1f0) = -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) = 0x7f160e0c2000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\201\0\0\0\0\0\0"..., 832) = 832
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0O\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) = 0x7f160e0c0000
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0O\305\3743\364B\2216\244\224\306@\261\23\327o"..., 68, 824) = 68
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f160e09d000
mmap(0x7f160e0a4000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f160e0a4000
.....
write(1, "Before memory allocation.", 25Before memory allocation.) = 25
read(0,
"\n", 1024) = 1
mmap(0x33f2678f, 289406976, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x33f26000
write(1, "After memory allocation.", 24After memory allocation.) = 24
read(0,
"\n", 1024) = 1
openat(AT_FDCWD, "/dev/urandom", O_RDONLY) = 3
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f160d6a7000
mprotect(0x7f160d6a8000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f160dea6fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[13866], tls=0x7f160dea7700, child_tidptr=0x7f160dea79d0) = 13866
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f160cea6000
mprotect(0x7f160cea7000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f160d6a5fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[13867], tls=0x7f160d6a6700, child_tidptr=0x7f160d6a69d0) = 13867

```
.....
.....
futex(0x7f160dea79d0, FUTEX_WAIT, 13866, NULL) = 0
futex(0x7f160d6a69d0, FUTEX_WAIT, 13867, NULL) = 0
futex(0x7f160cea59d0, FUTEX_WAIT, 13868, NULL) = 0
futex(0x7f1607fff9d0, FUTEX_WAIT, 13869, NULL) = 0
futex(0x7f16077fe9d0, FUTEX_WAIT, 13870, NULL) = 0
munmap(0x7f160d6a7000, 8392704)         = 0
futex(0x7f1606ffd9d0, FUTEX_WAIT, 13871, NULL) = 0
munmap(0x7f160cea6000, 8392704)         = 0
futex(0x7f16067fc9d0, FUTEX_WAIT, 13872, NULL) = 0
munmap(0x7f160c6a5000, 8392704)         = 0
futex(0x7f1605ffb9d0, FUTEX_WAIT, 13873, NULL) = 0
munmap(0x7f16077ff000, 8392704)         = 0
futex(0x7f16057fa9d0, FUTEX_WAIT, 13874, NULL) = 0
munmap(0x7f1606ffe000, 8392704)         = 0
futex(0x7f1604ff99d0, FUTEX_WAIT, 13875, NULL) = 0
munmap(0x7f16067fd000, 8392704)         = 0
futex(0x7f16047f89d0, FUTEX_WAIT, 13876, NULL) = 0
munmap(0x7f1605ffc000, 8392704)         = 0
.....
close(3)                                = 0
write(1, "After memory filling.", 21After memory filling.)   = 21
read(0,
"\n", 1024)                             = 1
munmap(0x33f26000, 289406976)           = 0
write(1, "After memory deallocation.", 26After memory deallocation.) = 26
read(0,
"\n", 1024)                             = 1
mmap(0x33f2678f, 289406976, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x33f26000
clone(child_stack=0x7f15e37b5fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
SVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTIDWaiting on cv
file_0.bin
, parent_tid=[13947], tls=0x7f15e37b6700, child_tidptr=0x7f15e37b69d0) = 13947
clone(child_stack=0x7f15e3fb6fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
SVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTIDWaiting on cv
file_1.bin
, parent_tid=[13948], tls=0x7f15e3fb7700, child_tidptr=0x7f15e3fb79d0) = 13948
clone(child_stack=0x7f15e47b7fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
SVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTIDWaiting on cv
file_2.bin
, parent_tid=[13949], tls=0x7f15e47b8700, child_tidptr=0x7f15e47b89d0) = 13949
clone(child_stack=0x7f15e4fb8fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
```

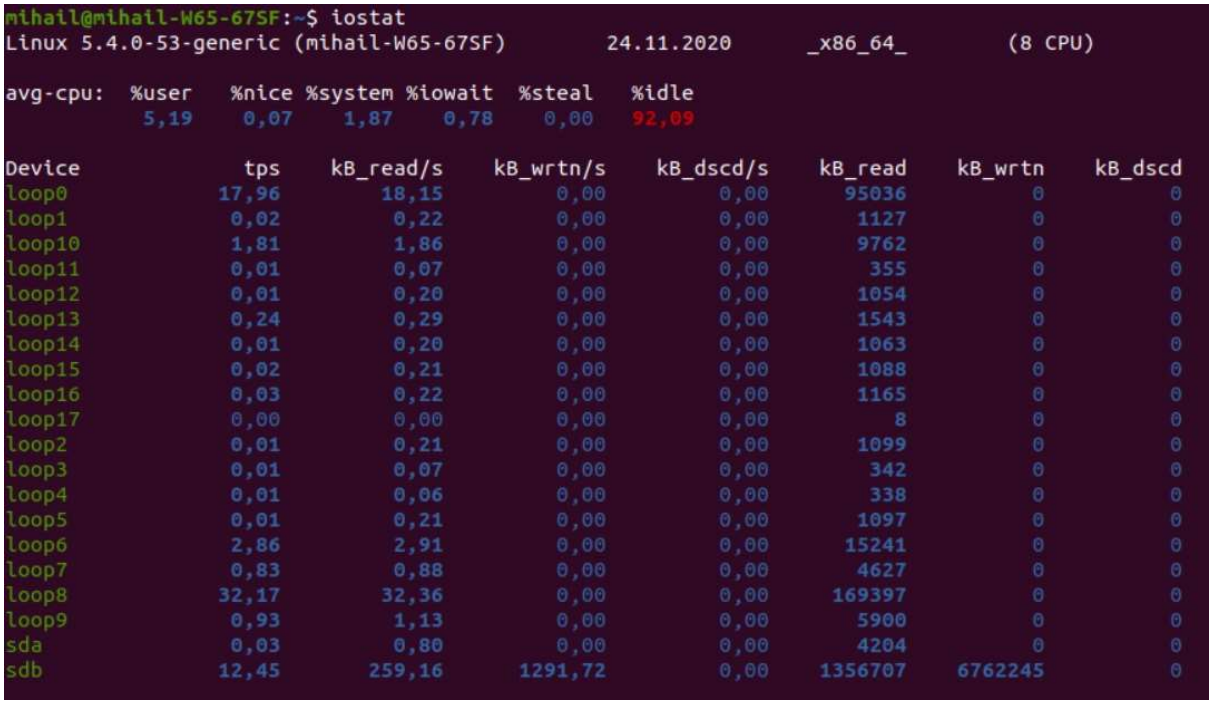
SVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTIDWaiting on cv file_3.bin
, parent_tid=[13950], tls=0x7f15e4fb9700, child_tidptr=0x7f15e4fb99d0) = 13950
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f160d6a7000
mprotect(0x7f160d6a8000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f160dea6fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTIDWaiting on cv file_4.bin
, parent_tid=[13951], tls=0x7f160dea7700, child_tidptr=0x7f160dea79d0) = 13951
.....
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f15b2fee000
mprotect(0x7f15b2fef000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f15b37edfb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[14090], tls=0x7f15b37ee700, child_tidptr=0x7f15b37ee9d0) = 14090
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f15adfe4000
mprotect(0x7f15adfe5000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7f15ae7e3fb0,
flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
parent_tid=[14099], tls=0x7f15ae7e4700, child_tidptr=0x7f15ae7e49d0) = 14099

Загрузка процессора во время выполнения:



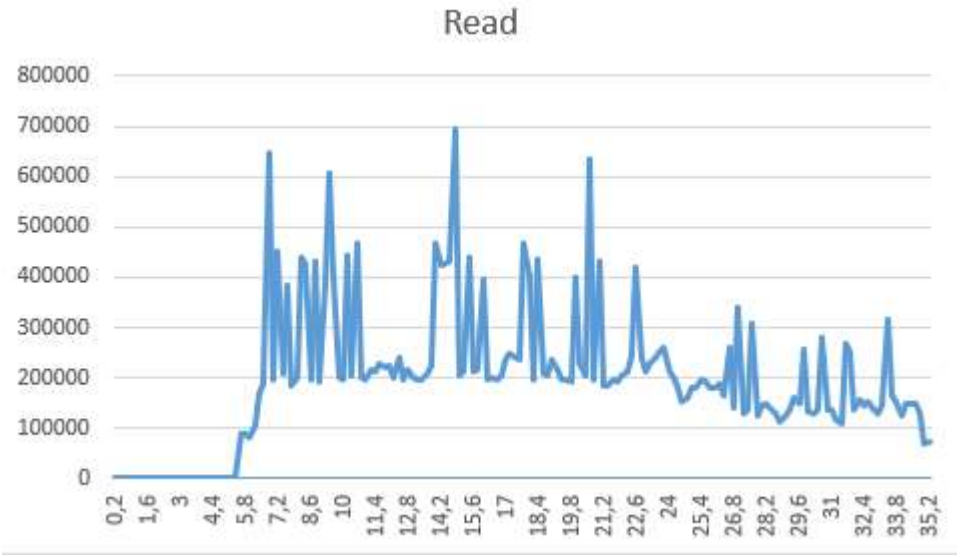
```
top - 16:12:52 up  1:26,  1 user,  load average: 9,01, 2,93, 1,41
Threads: 173 total,   3 running, 170 sleeping,   0 stopped,   0 zombie
%Cpu(s): 14,0 us, 24,0 sy,  0,0 ni, 15,5 id, 42,6 wa,  0,0 hi,  3,9 si,  0,0 st
MiB Mem :   7845,6 total,    166,0 free,   4239,0 used,   3440,6 buff/cache
MiB Swap:   2048,0 total,   2044,5 free,      3,5 used.   2607,8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  11694 mihail    20   0 5864728 286000   1704 R  99,9   3,6   0:04.57 main
  11608 mihail    20   0 5864728 286000   1704 S  60,0   3,6   0:03.14 main
  11632 mihail    20   0 5864728 286000   1704 R  46,7   3,6   0:01.22 main
  13160 mihail    20   0 5864728 286000   1704 R   6,7   3,6   0:00.01 main
  11245 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:00.02 main
  11591 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.70 main
  11592 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:02.94 main
  11593 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:03.02 main
  11594 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:00.00 main
  11595 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.85 main
  11596 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.89 main
  11597 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:03.60 main
  11598 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:00.00 main
  11599 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.80 main
  11600 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:03.72 main
  11601 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.87 main
  11602 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:00.00 main
  11603 mihail    20   0 5864728 286000   1704 S   0,0   3,6   0:01.84 main
```

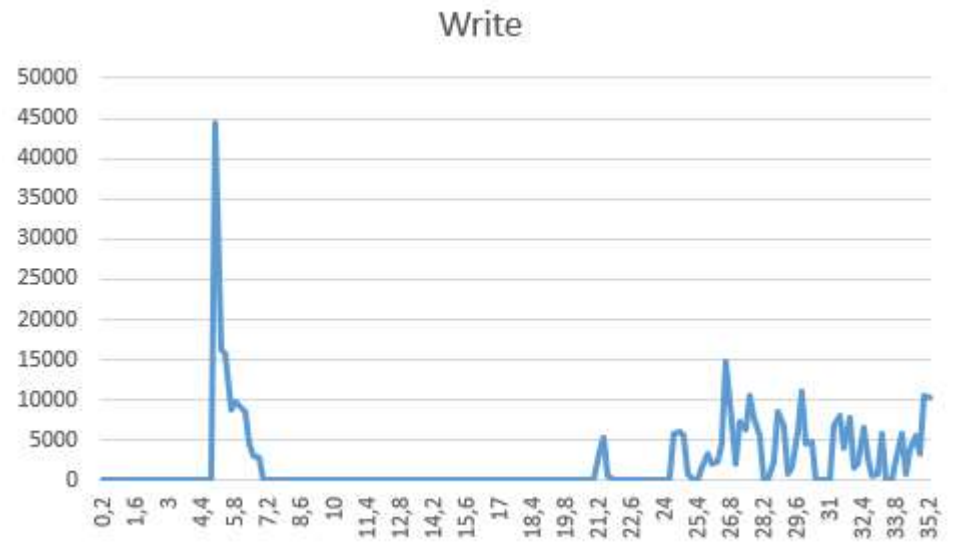
Статистика по вводу-выводу:

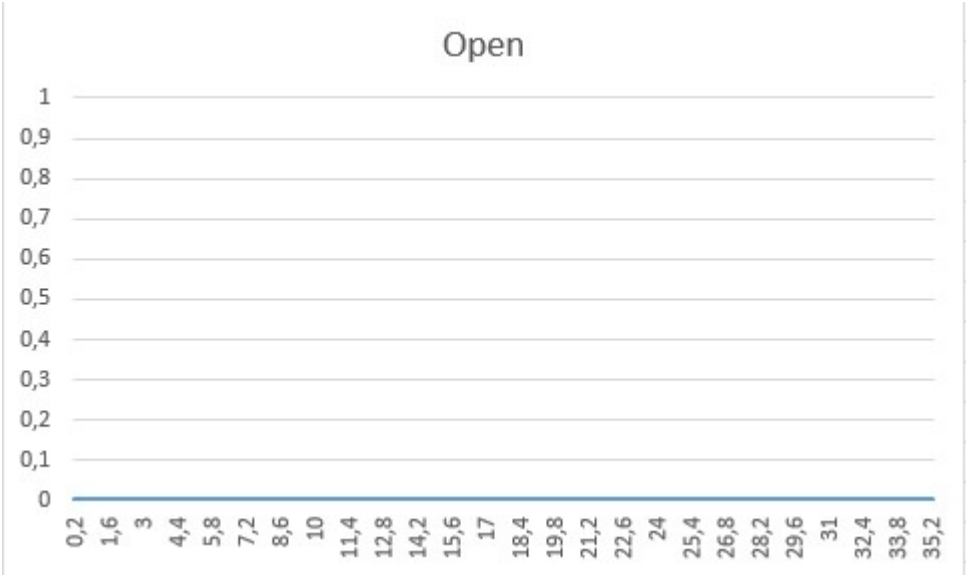


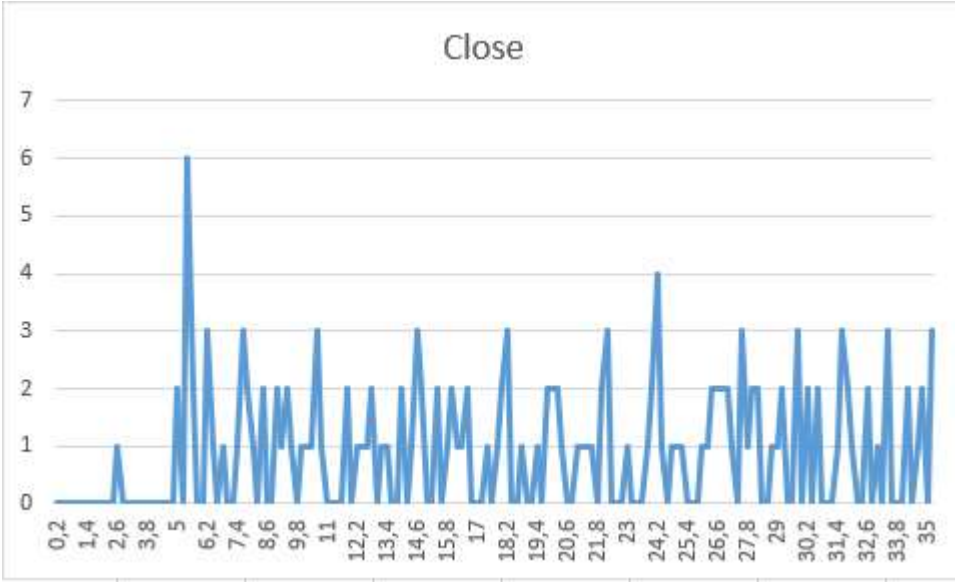
```
mihail@mihail-W65-67SF:~$ iostat
Linux 5.4.0-53-generic (mihail-W65-67SF)        24.11.2020      _x86_64_        (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5,19    0,07    1,87    0,78    0,00   92,09

Device             tps    kB_read/s    kB_wrtn/s    kB_dscd/s    kB_read    kB_wrtn    kB_dscd
loop0            17,96        18,15         0,00         0,00      95036          0          0
loop1             0,02         0,22         0,00         0,00       1127          0          0
loop10            1,81         1,86         0,00         0,00       9762          0          0
loop11            0,01         0,07         0,00         0,00        355          0          0
loop12            0,01         0,20         0,00         0,00       1054          0          0
loop13            0,24         0,29         0,00         0,00       1543          0          0
loop14            0,01         0,20         0,00         0,00       1063          0          0
loop15            0,02         0,21         0,00         0,00       1088          0          0
loop16            0,03         0,22         0,00         0,00       1165          0          0
loop17            0,00         0,00         0,00         0,00          8          0          0
loop2             0,01         0,21         0,00         0,00       1099          0          0
loop3             0,01         0,07         0,00         0,00        342          0          0
loop4             0,01         0,06         0,00         0,00        338          0          0
loop5             0,01         0,21         0,00         0,00       1097          0          0
loop6             2,86         2,91         0,00         0,00      15241          0          0
loop7             0,83         0,88         0,00         0,00       4627          0          0
loop8            32,17        32,36         0,00         0,00     169397          0          0
loop9             0,93         1,13         0,00         0,00       5900          0          0
sda               0,03         0,80         0,00         0,00       4204          0          0
sdb              12,45       259,16      1291,72         0,00    1356707    6762245          0
```

Графики системных вызовов, полученные с помощью stap.






Open
1
0,9
0,8
0,7
0,6
0,5
0,4
0,3
0,2
0,1
0
0,2
1,6
3
4,4
5,8
7,2
8,6
10
11,4
12,8
14,2
15,6
17
18,4
19,8
21,2
22,6
24
25,4
26,8
28,2
29,6
31
32,4
33,8
35,2


Close
7
6
5
4
3
2
1
0
0,2
1,4
2,6
3,8
5
6,2
7,4
8,6
9,8
11
12,2
13,4
14,6
15,8
17
18,2
19,4
20,6
21,8
23
24,2
25,4
26,6
27,8
29
30,2
31,4
32,6
33,8
35

Вывод

В процессе работы мы разработали многопоточную программу на языку 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.