Федеральное государственное автономное образовательное учреждение высшего образования

«**Санкт-Петербургский национальный исследовательский**

университет информационных технологий, механики и оптики»

Факультет ПИиКТ

Дисциплина: Операционные системы

Лабораторная работа № 1

Выполнил: Гурин Евгений Иванович

Преподаватель: Покид Александр Владимирович

Группа: P33122

Санкт-Петербург 2020г.

Вариант

A=224;
B=0x4B7A441D;
C=mmap;
D=25;E=144;
F=nocache;
G=126;
H=random;
I=55;
J=max;
K=futex

Задание

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.

Исходный код

```
#include <pthread.h>
#include <math.h>
#include <stdlib.h>
#include <malloc.h>
#include <stdio.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <stdint.h>
#include <stdatomic.h>
#include <linux/futex.h>
#include <syscall.h>
#include <string.h>

#define DEBUG_THR 0
#define DEBUG_THR_IO 0
#define DEBUG_THR_FILE_IO 0
#define DEBUG_THR_AGG_IO 0
#define DEBUG_THR_NUMBERS 0
#define MEM_ALLOC_ACTION_STOP 1

#define INFTY_LOOP 1
#define BYTES_IN_MB 1024*1024
#define MEM_SIZE 224 * BYTES_IN_MB
#define START_MEM_CELL_P 0x4B7A441D
#define MEM_THR_AMOUNT 25
#define FILE_SIZE 144 * BYTES_IN_MB
#define FILE_THR_AMOUNT 5
#define IO_BLOCK_SIZE 126
#define IO_BLOCK_ALIGN_SIZE 512
#define AGG_THR_AMOUNT 55
#define RANDOM_FNAME "/dev/urandom"

unsigned char * START_MEM_PTR;


// B=0x4B7A441D;C=mmap;D=25;E=144;F=nocache;G=126;H=random;I=55;J=max;K=futex

typedef struct thread_info  {
    pthread_t thread_id;
    int thread_number;
    unsigned char *start;
    size_t size;
    int fd;
    int block;
} thread_info;

thread_info * mem_thr;
thread_info * file_thr;
thread_info * file_max_thr;


size_t min(size_t a, size_t b) {
    return a < b ? a : b;
}

size_t max(size_t a, size_t b) {
    return a > b ? a : b;
```

```
}

int futex_wait(int * ptr, int value) {
    return syscall(SYS_futex,  ptr, FUTEX_WAIT, value, NULL, NULL, 0);
}

int futex_wake(int * ptr, int value) {
    return syscall(SYS_futex,  ptr, FUTEX_WAKE, value, NULL, NULL, 0);
}


void * fillMemBlock(void * args) {
    thread_info * attrs = ((thread_info *) args);
    unsigned char * start = (attrs->start);
    size_t size = (attrs->size);
    int thread_number = (attrs->thread_number);
    int fd = (attrs->fd);

    if (DEBUG_THR) printf("\nTHR %2d MEM start: %p size: %d\n", thread_number, (void
*)start, (int)size);

    double min_size = (long)size / (double)IO_BLOCK_SIZE;
    int parts = ceil(min_size);

    if (DEBUG_THR) printf("THR %2d MEM parts: %d min_size: %f\n", thread_number,
(int)parts, min_size);

    size_t filled = 0;

    for (int i = 0; i < parts; ++i){
        size_t size_part = min(IO_BLOCK_SIZE, size - filled);
        unsigned char * start_part = start + filled;
        int bytes = read(fd, start_part, size_part);

        if (bytes == -1) {
            if (DEBUG_THR_IO) printf("THR %2d MEM read failed\n", thread_number);
            continue;
        }

        filled += size_part;

        if (DEBUG_THR_IO) printf("THR %2d MEM read part: %d/%d start: %p size_part:
%d bytes: %d\n", thread_number, i, parts, (void *) start_part, (int)size_part,
bytes);
        if (DEBUG_THR_IO) printf("THR %2d MEM save in memory\n", thread_number);

        if (!DEBUG_THR_NUMBERS) continue;
        printf("THR %2d MEM NUMBERS ", thread_number);
        for (int i = 0; i < size_part; ++i) printf("%d ", start[i] );
        printf("\n");

    }

    if (DEBUG_THR) printf("\nTHR %2d MEM END\n", thread_number);

    pthread_exit(0);
}


void allocateMem() {
    if (DEBUG_THR) printf("TRY ALLOC MEM with mmap ptr: %p\n", (void *)
START_MEM_CELL_P);
```

```
    START_MEM_PTR = (unsigned char *) mmap((unsigned char *) START_MEM_CELL_P,
MEM_SIZE, PROT_READ | PROT_WRITE, MAP_SHARED | MAP_ANONYMOUS, -1, 0);
    if (START_MEM_PTR == MAP_FAILED) {
        printf("ERROR ALLOC MEM with mmap\n");
        exit(-1);
    }

    if (DEBUG_THR || MEM_ALLOC_ACTION_STOP) printf("ALLOC MEM with mmap ptr: %p\n",
(void *) START_MEM_PTR);
}


void deallocateMem() {
    if (DEBUG_THR) printf("TRY FREE MEM with ptr: %p\n", (void *) START_MEM_PTR);

    munmap(START_MEM_PTR, MEM_SIZE);
    free(mem_thr);
    free(file_thr);
    free(file_max_thr);
}


void startMemFillThreads() {
    mem_thr = (thread_info *) malloc(sizeof(thread_info) * (MEM_THR_AMOUNT + 1));
    size_t part_size = ceil( (double)MEM_SIZE/MEM_THR_AMOUNT);

    if (DEBUG_THR) printf("MEM_SIZE: %d\n", (int) MEM_SIZE);
    if (DEBUG_THR) printf("MEM_THR_AMOUNT: %d\n", (int) MEM_THR_AMOUNT);
    if (DEBUG_THR) printf("ONE_PART_SIZE: %d\n", (int) part_size);
    if (DEBUG_THR) printf("\n");

    size_t filled_size = 0;
    int fd = open(RANDOM_FNAME, O_RDONLY);


    int i = 0;
    while(filled_size < MEM_SIZE) {
        size_t size = min(part_size, MEM_SIZE - filled_size);

        mem_thr[i].fd = fd;
        mem_thr[i].size = size;
        mem_thr[i].start = START_MEM_PTR + filled_size;
        mem_thr[i].thread_number = i;

        if (DEBUG_THR) printf("Create THR %2d filled_size: %15d size: %d\n",i,  (int)
filled_size, (int) size);
        pthread_create(&mem_thr[i].thread_id, NULL, fillMemBlock, &mem_thr[i]);
        i++;
        filled_size += size;

    }
}


void joinMemFillThreads() {
    void * res;

    for (int i = 0; i < MEM_THR_AMOUNT; ++i) {
        pthread_join(mem_thr[i].thread_id, &res);
    }
}
```

```
void pauseRun(char * msg) {
    if (!MEM_ALLOC_ACTION_STOP) return;
    printf("%s", msg);
    getchar();
}


void * fillFile(void * args) {
    thread_info * attrs = ((thread_info *) args);
    int thread_number = (attrs->thread_number);

    if (DEBUG_THR) printf("\nTHR %2d FILE start\n", thread_number);

    char file_name[2];
    sprintf(file_name, "%d", thread_number);
    int fd = open(file_name, O_CREAT | O_TRUNC | __O_DIRECT | O_WRONLY, S_IRWXU |
S_IRWXG );

    size_t filled = 0;
    const size_t MAX_OFFSET = MEM_SIZE - IO_BLOCK_SIZE + 1;

    if (DEBUG_THR) printf("THR %2d FILE open: %s fd: %d MAX_OFFSET: %ld\n",
thread_number, file_name, fd, MAX_OFFSET);

    while (filled < FILE_SIZE) {
        size_t offset = random() % MAX_OFFSET;
        unsigned char * start = START_MEM_PTR + offset;

        size_t size_part = min(IO_BLOCK_SIZE, FILE_SIZE - filled);
        void * aligned_start = memalign(IO_BLOCK_ALIGN_SIZE, size_part);
        memcpy(aligned_start, start, size_part);

        int bytes = write(fd, aligned_start, IO_BLOCK_ALIGN_SIZE);

        filled += bytes;

        if (bytes == -1) {
            printf("THR %2d FILE write ERROR\n", thread_number);
            continue;
        }

        if (DEBUG_THR_FILE_IO) printf(
            "THR %2d FILE write: %p filed: %10ld / %10d\nsize_part: %ld bytes
written: %d offset: %ld\n",
            thread_number, (void *) start, filled, FILE_SIZE, size_part, bytes,
offset
        );
    }

    if (DEBUG_THR) printf("THR %2d FILE END\n", thread_number);

    file_thr[thread_number].block = 1;
    futex_wake(&file_thr[thread_number].block, 1);
    pthread_exit(0);
}


void createFillFilesThreads() {
    file_thr = (thread_info *) malloc(sizeof(thread_info) * FILE_THR_AMOUNT);
```

```
    for (int i = 0; i < FILE_THR_AMOUNT; ++i) {
        file_thr[i].thread_number = i;
        file_thr[i].block = 0;
        if (DEBUG_THR) printf("Create THR %2d FILE\n", i);
        pthread_create(&file_thr[i].thread_id, NULL, fillFile, &file_thr[i]);
    }

    for (int i = 0; i < FILE_THR_AMOUNT; ++i) {
       futex_wake(&file_thr[i].block, 1);
    }
}


void wakeFillFilesThreads() {
    for (int i = 0; i < FILE_THR_AMOUNT; ++i) {
        futex_wake(&file_thr[i].block, 1);
    }
}


void joinFillFilesThreads() {
    void * res;

    for (int i = 0; i < FILE_THR_AMOUNT; ++i) {
        pthread_join(file_thr[i].thread_id, &res);
    }
}


unsigned char getMaxFromCharArray(unsigned char * array, size_t size, int debug) {
    unsigned char max_c = array[0];

    if (debug) printf("GET MAX FROM NUMBERS max: %d\n", max_c);

    for (int i = 0; i < size; ++i) {
        if (debug) printf("%5d", array[i]);
        max_c = max((int) max_c, (int) array[i]);
    }
    if (debug) printf("\n");

    return max_c;
}


void * countFileMaxNumber(void * args) {
    thread_info * attrs = ((thread_info *) args);
    int thread_number = (attrs->thread_number);
    int file_number = (attrs->fd);

    futex_wait(&file_thr[file_number].block, 0);

    if (DEBUG_THR) printf("THR %2d MAX for FILE %d start\n", thread_number,
file_number);

    char file_name[2];
    sprintf(file_name, "%d", file_number);
    int fd = open(file_name, O_RDONLY);

    unsigned char max_c = 0;
    int bytes;
    unsigned char buffer[IO_BLOCK_SIZE];
```

```
    do {
        bytes = read(fd, buffer, IO_BLOCK_SIZE);
        max_c = max(max_c, getMaxFromCharArray(buffer, IO_BLOCK_SIZE,
DEBUG_THR_AGG_IO));
        if (DEBUG_THR_AGG_IO) printf("THR %2d MAX for FILE %d read part bytes: %d
max: %d\n", thread_number, file_number, bytes, max_c);
    } while (bytes > 0);

    if (DEBUG_THR) printf("THR %2d MAX for FILE %d END max: %d\n", thread_number,
file_number, max_c);

    futex_wake(&file_thr[file_number].block, 1);
    pthread_exit(0);
}


void createMaxCountFilesThreads() {
    file_max_thr = (thread_info *) malloc(sizeof(thread_info) * AGG_THR_AMOUNT);

    for (int i = 0; i < AGG_THR_AMOUNT; ++i) {
        int random_file_number = random() % FILE_THR_AMOUNT;
        file_max_thr[i].thread_number = i;
        file_max_thr[i].fd = random_file_number;

        if (DEBUG_THR) printf("Create THR %2d MAX for FILE %d\n", i,
random_file_number);
        pthread_create(&file_max_thr[i].thread_id, NULL, countFileMaxNumber,
&file_max_thr[i]);
    }
}


void joinMaxCountFilesThreads() {
    void * res;

    for (int i = 0; i < AGG_THR_AMOUNT; ++i) {
        pthread_join(file_max_thr[i].thread_id, &res);
    }
}


int main(int argc, char const *argv[]) {
    do {
        pauseRun("\nBEFORE ALLOC (Enter)\n");
        allocateMem();
        pauseRun("\nAFTER ALLOC (Enter)\n");

        startMemFillThreads();
        joinMemFillThreads();
        pauseRun("\nAFTER MEM FILL (Enter)\n");

        createFillFilesThreads();
        createMaxCountFilesThreads();

        wakeFillFilesThreads();
        joinFillFilesThreads();
        joinMaxCountFilesThreads();

        deallocateMem();
        pauseRun("\nAFTER DEALLOC MEM (Enter)\n");

    } while (INFTY_LOOP);
```

```
    return 0;
}
```

Замеры виртуальной и физической памяти (ps -u)

	виртуальной	физической
До аллокации	10 408	1 124
После аллокации	239 784	1 124
После заполнения участка данными	340 248	231 624
После деаллокации	1 028 376	3 288

Вывод top (загрузка системных ресурсов)

```
top - 23:56:20 up  2:19,  3 users,  load average: 4.80, 2.16, 0.85
Tasks:  95 total,   1 running,  52 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us, 26.5 sy,  0.0 ni,  0.0 id, 13.2 wa,  0.0 hi, 60.3 si,  0.0 st
KiB Mem :  1344648 total,   691196 free,   234236 used,   419216 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   737540 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1969 egurin    20   0 1061368 342852 231324 S 39.0 25.5   0:57.01 lab1
```

Замеры затраченного времени

```
egurin@egurin:~/os/s264449/lab1$ time ./lab1

real    16m33.329s
user    0m9.592s
sys     6m13.883s
```

Карта процесса

```
egurin@egurin:~/os/s264449/lab1$ cat /proc/$(pgrep lab1)/maps
4b7a4000-597a4000 rw-s 00000000 00:05 23891                              /dev/zero (deleted)
560d74e29000-560d74e2b000 r-xp 00000000 fd:00 133236                     /home/egurin/os/s264449/lab1/lab1
560d7502b000-560d7502c000 r--p 00002000 fd:00 133236                     /home/egurin/os/s264449/lab1/lab1
560d7502c000-560d7502d000 rw-p 00003000 fd:00 133236                     /home/egurin/os/s264449/lab1/lab1
560d76124000-560d76145000 rw-p 00000000 00:00 0                          [heap]
7f0f04000000-7f0f056cf000 rw-p 00000000 00:00 0
7f0f056cf000-7f0f08000000 ---p 00000000 00:00 0
7f0f0c000000-7f0f0e04b000 rw-p 00000000 00:00 0
7f0f0e04b000-7f0f10000000 ---p 00000000 00:00 0
7f0f10000000-7f0f117a0000 rw-p 00000000 00:00 0
7f0f117a0000-7f0f14000000 ---p 00000000 00:00 0
7f0f14000000-7f0f160e6000 rw-p 00000000 00:00 0
7f0f160e6000-7f0f18000000 ---p 00000000 00:00 0
7f0f1afde000-7f0f1afdf000 ---p 00000000 00:00 0
7f0f1afdf000-7f0f1b7df000 rw-p 00000000 00:00 0
7f0f1b7df000-7f0f1b7e0000 ---p 00000000 00:00 0
7f0f1b7e0000-7f0f1bfe0000 rw-p 00000000 00:00 0
7f0f1bfe0000-7f0f1bfe1000 ---p 00000000 00:00 0
7f0f1bfe1000-7f0f1c7e1000 rw-p 00000000 00:00 0
7f0f1c7e1000-7f0f1c7e2000 ---p 00000000 00:00 0
7f0f1c7e2000-7f0f1cfe2000 rw-p 00000000 00:00 0
7f0f1cfe2000-7f0f1cfe3000 ---p 00000000 00:00 0
7f0f1cfe3000-7f0f1d7e3000 rw-p 00000000 00:00 0
7f0f1d7e3000-7f0f1d7e4000 ---p 00000000 00:00 0
7f0f1d7e4000-7f0f1dfe4000 rw-p 00000000 00:00 0
7f0f1dfe4000-7f0f1dfe5000 ---p 00000000 00:00 0
7f0f1dfe5000-7f0f1e7e5000 rw-p 00000000 00:00 0
7f0f1e7e5000-7f0f1e7e6000 ---p 00000000 00:00 0
7f0f1e7e6000-7f0f1efe6000 rw-p 00000000 00:00 0
7f0f1efe6000-7f0f1efe7000 ---p 00000000 00:00 0
7f0f1efe7000-7f0f1f7e7000 rw-p 00000000 00:00 0
7f0f1f7e7000-7f0f1f7e8000 ---p 00000000 00:00 0
7f0f1f7e8000-7f0f1ffe8000 rw-p 00000000 00:00 0
7f0f1ffe8000-7f0f1ffe9000 ---p 00000000 00:00 0
7f0f1ffe9000-7f0f207e9000 rw-p 00000000 00:00 0
7f0f207e9000-7f0f207ea000 ---p 00000000 00:00 0
7f0f207ea000-7f0f20fea000 rw-p 00000000 00:00 0
```

```
7f0f20fea000-7f0f20feb000 ---p 00000000 00:00 0
7f0f20feb000-7f0f217eb000 rw-p 00000000 00:00 0
7f0f217eb000-7f0f217ec000 ---p 00000000 00:00 0
7f0f217ec000-7f0f21fec000 rw-p 00000000 00:00 0
7f0f21fec000-7f0f21fed000 ---p 00000000 00:00 0
7f0f21fed000-7f0f227ed000 rw-p 00000000 00:00 0
7f0f227ed000-7f0f227ee000 ---p 00000000 00:00 0
7f0f227ee000-7f0f22fee000 rw-p 00000000 00:00 0
7f0f22fee000-7f0f22fef000 ---p 00000000 00:00 0
7f0f22fef000-7f0f237ef000 rw-p 00000000 00:00 0
7f0f237ef000-7f0f237f0000 ---p 00000000 00:00 0
7f0f237f0000-7f0f23ff0000 rw-p 00000000 00:00 0
7f0f23ff0000-7f0f23ff1000 ---p 00000000 00:00 0
7f0f23ff1000-7f0f247f1000 rw-p 00000000 00:00 0
7f0f247f1000-7f0f247f2000 ---p 00000000 00:00 0
7f0f247f2000-7f0f24ff2000 rw-p 00000000 00:00 0
7f0f24ff2000-7f0f24ff3000 ---p 00000000 00:00 0
7f0f24ff3000-7f0f257f3000 rw-p 00000000 00:00 0
7f0f257f3000-7f0f257f4000 ---p 00000000 00:00 0
7f0f257f4000-7f0f25ff4000 rw-p 00000000 00:00 0
7f0f25ff4000-7f0f25ff5000 ---p 00000000 00:00 0
7f0f25ff5000-7f0f267f5000 rw-p 00000000 00:00 0
7f0f267f5000-7f0f267f6000 ---p 00000000 00:00 0
7f0f267f6000-7f0f26ff6000 rw-p 00000000 00:00 0
7f0f26ff6000-7f0f26ff7000 ---p 00000000 00:00 0
7f0f26ff7000-7f0f277f7000 rw-p 00000000 00:00 0
7f0f277f7000-7f0f277f8000 ---p 00000000 00:00 0
7f0f277f8000-7f0f27ff8000 rw-p 00000000 00:00 0
7f0f27ff8000-7f0f27ff9000 ---p 00000000 00:00 0
7f0f27ff9000-7f0f287f9000 rw-p 00000000 00:00 0
7f0f287f9000-7f0f287fa000 ---p 00000000 00:00 0
7f0f287fa000-7f0f28ffa000 rw-p 00000000 00:00 0
7f0f28ffa000-7f0f28ffb000 ---p 00000000 00:00 0
7f0f28ffb000-7f0f297fb000 rw-p 00000000 00:00 0
7f0f297fb000-7f0f297fc000 ---p 00000000 00:00 0
7f0f297fc000-7f0f29ffc000 rw-p 00000000 00:00 0
7f0f29ffc000-7f0f29ffd000 ---p 00000000 00:00 0
7f0f29ffd000-7f0f2a7fd000 rw-p 00000000 00:00 0
7f0f2a7fd000-7f0f2a7fe000 ---p 00000000 00:00 0
7f0f2a7fe000-7f0f2affe000 rw-p 00000000 00:00 0
7f0f2affe000-7f0f2afff000 ---p 00000000 00:00 0
7f0f2afff000-7f0f2b7ff000 rw-p 00000000 00:00 0
7f0f2b7ff000-7f0f2b800000 ---p 00000000 00:00 0
7f0f2b800000-7f0f2c000000 rw-p 00000000 00:00 0
7f0f2c000000-7f0f2dbc3000 rw-p 00000000 00:00 0
7f0f2dbc3000-7f0f30000000 ---p 00000000 00:00 0
7f0f30200000-7f0f30201000 ---p 00000000 00:00 0
7f0f30201000-7f0f30a01000 rw-p 00000000 00:00 0
7f0f30a01000-7f0f30a18000 r-xp 00000000 fd:00 133043                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7f0f30a18000-7f0f30c17000 ---p 00017000 fd:00 133043                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7f0f30c17000-7f0f30c18000 r--p 00016000 fd:00 133043                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7f0f30c18000-7f0f30c19000 rw-p 00017000 fd:00 133043                     /lib/x86_64-linux-gnu/libgcc_s.so.1
7f0f30c19000-7f0f30c1a000 ---p 00000000 00:00 0
7f0f30c1a000-7f0f3141a000 rw-p 00000000 00:00 0
7f0f3141a000-7f0f3141b000 ---p 00000000 00:00 0
7f0f3141b000-7f0f31c1b000 rw-p 00000000 00:00 0
7f0f31c1b000-7f0f31c1c000 ---p 00000000 00:00 0
7f0f31c1c000-7f0f3241c000 rw-p 00000000 00:00 0
7f0f3241c000-7f0f3241d000 ---p 00000000 00:00 0
7f0f3241d000-7f0f32c1d000 rw-p 00000000 00:00 0
7f0f32c1d000-7f0f32c1e000 ---p 00000000 00:00 0
7f0f32c1e000-7f0f3341e000 rw-p 00000000 00:00 0
7f0f3341e000-7f0f3341f000 ---p 00000000 00:00 0
7f0f3341f000-7f0f33c1f000 rw-p 00000000 00:00 0
7f0f33c1f000-7f0f33c20000 ---p 00000000 00:00 0
7f0f33c20000-7f0f34420000 rw-p 00000000 00:00 0
7f0f34420000-7f0f34421000 ---p 00000000 00:00 0
7f0f34421000-7f0f34c21000 rw-p 00000000 00:00 0
7f0f34c21000-7f0f34c22000 ---p 00000000 00:00 0
7f0f34c22000-7f0f35422000 rw-p 00000000 00:00 0
7f0f35422000-7f0f35423000 ---p 00000000 00:00 0
7f0f35423000-7f0f35c23000 rw-p 00000000 00:00 0
7f0f35c23000-7f0f35c24000 ---p 00000000 00:00 0
7f0f35c24000-7f0f36424000 rw-p 00000000 00:00 0
7f0f36424000-7f0f36425000 ---p 00000000 00:00 0
7f0f36425000-7f0f36c25000 rw-p 00000000 00:00 0
7f0f36c25000-7f0f36c26000 ---p 00000000 00:00 0
7f0f36c26000-7f0f37426000 rw-p 00000000 00:00 0
7f0f37426000-7f0f37427000 ---p 00000000 00:00 0
7f0f37427000-7f0f37c27000 rw-p 00000000 00:00 0
7f0f37c27000-7f0f37c28000 ---p 00000000 00:00 0
7f0f37c28000-7f0f38428000 rw-p 00000000 00:00 0
7f0f38428000-7f0f38429000 ---p 00000000 00:00 0
7f0f38429000-7f0f38c29000 rw-p 00000000 00:00 0
7f0f38c29000-7f0f38c2a000 ---p 00000000 00:00 0
7f0f38c2a000-7f0f3942a000 rw-p 00000000 00:00 0
7f0f3942a000-7f0f3942b000 ---p 00000000 00:00 0
7f0f3942b000-7f0f39c2b000 rw-p 00000000 00:00 0
7f0f39c2b000-7f0f39c2c000 ---p 00000000 00:00 0
```

```
7f0f39c2c000-7f0f3a42c000 rw-p 00000000 00:00 0
7f0f3a42c000-7f0f3a42d000 ---p 00000000 00:00 0
7f0f3a42d000-7f0f3ac2d000 rw-p 00000000 00:00 0
7f0f3ac2d000-7f0f3ac2e000 ---p 00000000 00:00 0
7f0f3ac2e000-7f0f3b42e000 rw-p 00000000 00:00 0
7f0f3b42e000-7f0f3b42f000 ---p 00000000 00:00 0
7f0f3b42f000-7f0f3bc2f000 rw-p 00000000 00:00 0
7f0f3bc2f000-7f0f3bc30000 ---p 00000000 00:00 0
7f0f3bc30000-7f0f3c430000 rw-p 00000000 00:00 0
7f0f3c430000-7f0f3c431000 ---p 00000000 00:00 0
7f0f3c431000-7f0f3cc31000 rw-p 00000000 00:00 0
7f0f3cc31000-7f0f3cc32000 ---p 00000000 00:00 0
7f0f3cc32000-7f0f3d432000 rw-p 00000000 00:00 0
7f0f3d432000-7f0f3d619000 r-xp 00000000 fd:00 181387                     /lib/x86_64-linux-gnu/libc-2.27.so
7f0f3d619000-7f0f3d819000 ---p 001e7000 fd:00 181387                     /lib/x86_64-linux-gnu/libc-2.27.so
7f0f3d819000-7f0f3d81d000 r--p 001e7000 fd:00 181387                     /lib/x86_64-linux-gnu/libc-2.27.so
7f0f3d81d000-7f0f3d81f000 rw-p 001eb000 fd:00 181387                     /lib/x86_64-linux-gnu/libc-2.27.so
7f0f3d81f000-7f0f3d823000 rw-p 00000000 00:00 0
7f0f3d823000-7f0f3d83d000 r-xp 00000000 fd:00 181402                     /lib/x86_64-linux-gnu/libpthread-2.27.so
7f0f3d83d000-7f0f3da3c000 ---p 0001a000 fd:00 181402                     /lib/x86_64-linux-gnu/libpthread-2.27.so
7f0f3da3c000-7f0f3da3d000 r--p 00019000 fd:00 181402                     /lib/x86_64-linux-gnu/libpthread-2.27.so
7f0f3da3d000-7f0f3da3e000 rw-p 0001a000 fd:00 181402                     /lib/x86_64-linux-gnu/libpthread-2.27.so
7f0f3da3e000-7f0f3da42000 rw-p 00000000 00:00 0
7f0f3da42000-7f0f3dbdf000 r-xp 00000000 fd:00 181391                     /lib/x86_64-linux-gnu/libm-2.27.so
7f0f3dbdf000-7f0f3ddde000 ---p 0019d000 fd:00 181391                     /lib/x86_64-linux-gnu/libm-2.27.so
7f0f3ddde000-7f0f3dddf000 r--p 0019c000 fd:00 181391                     /lib/x86_64-linux-gnu/libm-2.27.so
7f0f3dddf000-7f0f3dde0000 rw-p 0019d000 fd:00 181391                     /lib/x86_64-linux-gnu/libm-2.27.so
7f0f3dde0000-7f0f3de09000 r-xp 00000000 fd:00 181383                     /lib/x86_64-linux-gnu/ld-2.27.so
7f0f3dffd000-7f0f3e002000 rw-p 00000000 00:00 0
7f0f3e009000-7f0f3e00a000 r--p 00029000 fd:00 181383                     /lib/x86_64-linux-gnu/ld-2.27.so
7f0f3e00a000-7f0f3e00b000 rw-p 0002a000 fd:00 181383                     /lib/x86_64-linux-gnu/ld-2.27.so
7f0f3e00b000-7f0f3e00c000 rw-p 00000000 00:00 0
7ffd1ee53000-7ffd1ee74000 rw-p 00000000 00:00 0                          [stack]
7ffd1efec000-7ffd1efef000 r--p 00000000 00:00 0                          [vvar]
7ffd1efef000-7ffd1eff1000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]
```

Трассировка системных вызовов

sudo strace -c -f ./lab1

```
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
 91.88 21706.834596   153949182       141           futex
  6.70 1582.003164        1073   1474560           write
  1.41  334.175334           5  67774724           read
  0.01    2.054362          11    183937           mprotect
  0.00    0.058858         878        67           openat
  0.00    0.035156         386        91           munmap
  0.00    0.000757           9        85           madvise
  0.00    0.000209           2       111           mmap
  0.00    0.000083           1        86           set_robust_list
  0.00    0.000064           1        85           clone
  0.00    0.000007           1         6         6 access
  0.00    0.000005           1         6           fstat
  0.00    0.000004           1         6           close
  0.00    0.000000           0         3           brk
  0.00    0.000000           0         2           rt_sigaction
  0.00    0.000000           0         1           rt_sigprocmask
  0.00    0.000000           0         1           execve
  0.00    0.000000           0         1           arch_prctl
  0.00    0.000000           0         1           set_tid_address
  0.00    0.000000           0         1           prlimit64
------ ----------- ----------- --------- --------- ----------------
100.00 23625.162599              69433915         6 total
```

sudo strace -f ./lab1

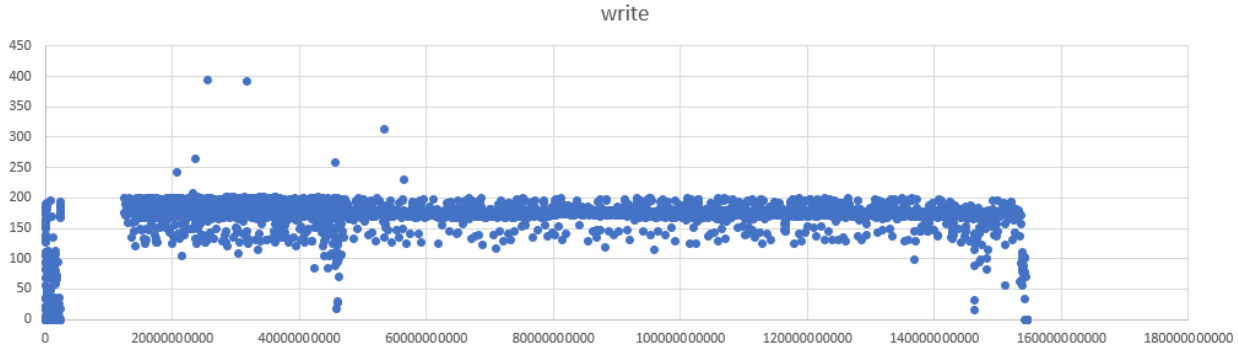
```
execve("./lab1", ["./lab1"], 0x7ffc31aa70c8 /* 14 vars */) = 0
brk(NULL)                               = 0x55baac8e2000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=25756, ...}) = 0
mmap(NULL, 25756, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1b79ef2000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.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\200\272\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=1700792, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1b79ef0000
mmap(NULL, 3789144, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1b79932000
mprotect(0x7f1b79acf000, 2093056, PROT_NONE) = 0
mmap(0x7f1b79cce000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x
19c000) = 0x7f1b79cce000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
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\0000b\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=144976, ...}) = 0
mmap(NULL, 2221184, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1b79713000
mprotect(0x7f1b7972d000, 2093056, PROT_NONE) = 0
mmap(0x7f1b7992c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x
19000) = 0x7f1b7992c000
mmap(0x7f1b7992e000, 13440, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1,
0) = 0x7f1b7992e000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
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\20\35\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0
mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f1b79322000
mprotect(0x7f1b79509000, 2097152, PROT_NONE) = 0
mmap(0x7f1b79709000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0
x1e7000) = 0x7f1b79709000
mmap(0x7f1b7970f000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1,
0) = 0x7f1b7970f000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1b79eed000
arch_prctl(ARCH_SET_FS, 0x7f1b79eed740) = 0
mprotect(0x7f1b79709000, 16384, PROT_READ) = 0
mprotect(0x7f1b7992c000, 4096, PROT_READ) = 0
mprotect(0x7f1b79cce000, 4096, PROT_READ) = 0
mprotect(0x55baac294000, 4096, PROT_READ) = 0
mprotect(0x7f1b79ef9000, 4096, PROT_READ) = 0
```

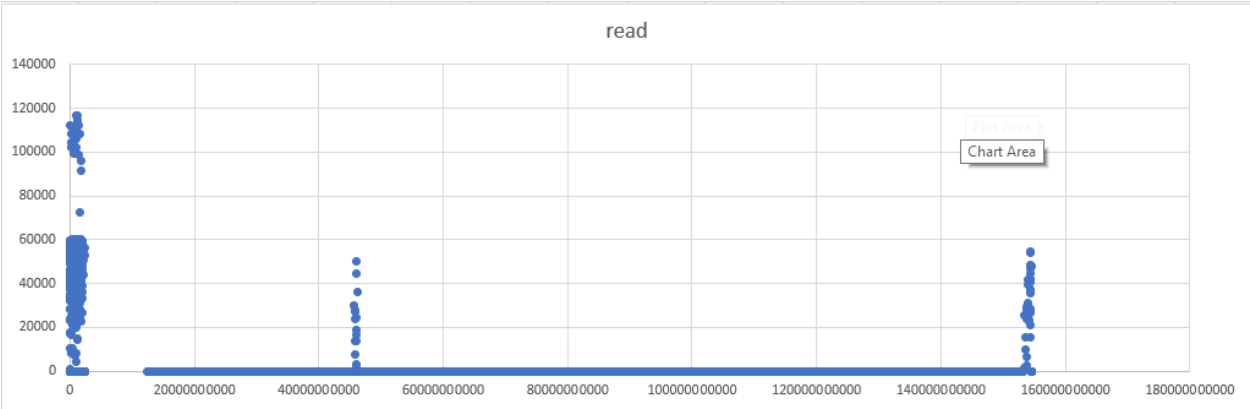
```
[pid  1647] <... read resumed> "\326Df(\202\354\353\254\353W\236\267\36D\4r~\212@\321\2552o
\34\2768\307^d+x\313"..., 126) = 126
[pid  1646] <... read resumed> ">\342\212]\241tM\264\275q\322A\n\372\2773\242=@\267\3\377\3
20\353\367\"\354\240\200\237c-"..., 126) = 126
[pid  1645] set_robust_list(0x7f1b7731d9e0, 24strace: Process 1644 attached
strace: Process 1653 attached
strace: Process 1652 attached
 <unfinished ...>
[pid  1651] set_robust_list(0x7f1b743179e0, 24 <unfinished ...>
[pid  1650] set_robust_list(0x7f1b74b189e0, 24 <unfinished ...>
[pid  1649] <... set_robust_list resumed> ) = 0
[pid  1648] <... read resumed> "\222\372\310\324\275]\32M\0\375_\213o<\347[x\202\352\2\263n
\3230H\252#E\202N\256\231"..., 126) = 126
[pid  1647] read(3,  <unfinished ...>
[pid  1646] read(3,  <unfinished ...>
[pid  1645] <... set_robust_list resumed> ) = 0
[pid  1644] set_robust_list(0x7f1b77b1e9e0, 24 <unfinished ...>
[pid  1640] <... clone resumed> child_stack=0x7f1b6e30afb0, flags=CLONE_VM|CLONE_FS|CLONE_F
LES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_
CLEARTID, parent_tidptr=0x7f1b6e30b9d0, tls=0x7f1b6e30b700, child_tidptr=0x7f1b6e30b9d0) =
1663
strace: Process 1656 attached
strace: Process 1655 attached
strace: Process 1654 attached
[pid  1653] set_robust_list(0x7f1b733159e0, 24 <unfinished ...>
[pid  1652] set_robust_list(0x7f1b73b169e0, 24 <unfinished ...>
[pid  1651] <... set_robust_list resumed> ) = 0
[pid  1650] <... set_robust_list resumed> ) = 0
[pid  1649] read(3,  <unfinished ...>
[pid  1648] read(3,  <unfinished ...>
[pid  1647] <... read resumed> "\231\17\224]\327\375\240\362\4|BX\335k>Z2\366q^\334\334\306
\361^\243\2761\357&\35&"..., 126) = 126
[pid  1646] <... read resumed> "Ru\r\37t!\357\214Rf4<\256\266\23\35\213\351\v\2\325n\331\22
\331+\364\33\2773\232\25"..., 126) = 126
[pid  1645] read(3,  <unfinished ...>
[pid  1644] <... set_robust_list resumed> ) = 0
strace: Process 1643 attached
[pid  1640] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0strace
: Process 1661 attached
strace: Process 1660 attached
strace: Process 1659 attached
strace: Process 1658 attached
strace: Process 1657 attached
 <unfinished ...>
[pid  1656] set_robust_list(0x7f1b71b129e0, 24 <unfinished ...>
[pid  1655] set_robust_list(0x7f1b723139e0, 24 <unfinished ...>
[pid  1654] set_robust_list(0x7f1b72b149e0, 24 <unfinished ...>
```

```
[pid  1665] read(3, "\235R\375\300\322\256\307f5O\335\t\371rN\213\251-\0\312\322\33(\366\35
3\210\221\220\221U3\240"..., 126) = 126
[pid  1665] read(3, "\n\241\367\376\247>\261\217?\0\323\265\363\255\200\226T\251\336\2\3760
\303l\211\325\243*\247D\331&"..., 126) = 126
[pid  1665] read(3, "k[\1\263!@\264\240ozWw\303\314\24#Z*\373RX@\222\\\231`\307\320\254\241
\271\327"..., 126) = 126
[pid  1665] read(3, "\307w\350\334\2\252!\5\244\223\342_\7\24JI&\211\275\275\361\310\1H\202
\260 \27\371z\313\303"..., 126) = 126
[pid  1665] read(3, "\353\324f\325\32?\352\243\377\312\372\243^\221n\206\224\300>\266\343\2
63\332\371/Y\202\210\357\347b\325"..., 126) = 126
[pid  1665] read(3, "\272\330\372f\326\365\232\330\355\275\334\247\353Kr\372d\210\376\275\3
60\346\345b\364K\23si\263\213\310"..., 126) = 126
[pid  1665] read(3, "\336mI\v\3559\347\333q\10\222o \257\321\30\34>\n8\302\241\350Y\2$IX\36
3p\221\303"..., 126) = 126
[pid  1665] read(3, "\330\334W\271\n\371\3635`\24\27|\354\213\0278\272\36\277\310\2433\342o
\254\242\270p\345\313\216a"..., 126) = 126
[pid  1665] read(3, "\2665\27M\374\262u\20\312\17\372>1\2\16\310CM\353\243K\2H\31n\\\237\35
3\374r4n"..., 126) = 126
[pid  1665] read(3, "i'\f\207\304\321!\351_\226wJ\260\266\354Ni\260\224\273v\224\213\f\10\3
73\333\343\325\351\320\204"..., 126) = 126
[pid  1665] read(3, "\230\235B`\314fA)R\246f0!\356sy\312\231\276!\225\341\205\312\315\374\2
46C{\26^Y"..., 126) = 126
[pid  1665] read(3, "\26o\276\346\273\213\305\223>{\21\17\262\332\307\221<B\25\370\235\2c\2
32r\270\360\27\344\2612M"..., 126) = 126
[pid  1665] read(3, "\355P\177\330\226 <n9T\vJ\364\227U\203{\204\217\235\2104\353\2665&]\32
1\221\36\244\202"..., 126) = 126
[pid  1665] read(3, "\307\223Z\26?\214BT\341\204\273\247\207\10\343zX\221\330\332\265\373*z
```

```
., 126) = 126
[pid  1676] read(47, "\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"...,
 126) = 72
[pid  1676] read(47, "", 126)           = 0
[pid  1676] futex(0x55baac8e2e84, FUTEX_WAKE, 1) = 0
[pid  1676] madvise(0x7f1b76b1d000, 8368128, MADV_DONTNEED) = 0
[pid  1676] exit(0)                     = ?
[pid  1640] <... futex resumed> )       = 0
[pid  1640] munmap(0x7f1b78b21000, 8392704) = 0
[pid  1640] munmap(0x7f1b78320000, 8392704) = 0
[pid  1640] munmap(0x7f1b77b1f000, 8392704) = 0
[pid  1640] munmap(0x7f1b7731e000, 8392704) = 0
[pid  1640] munmap(0x7f1b76b1d000, 8392704) = 0
[pid  1640] munmap(0x7f1b7631c000, 8392704) = 0
[pid  1640] munmap(0x7f1b75b1b000, 8392704) = 0
[pid  1640] munmap(0x7f1b7531a000, 8392704) = 0
[pid  1640] munmap(0x7f1b74b19000, 8392704) = 0
[pid  1640] munmap(0x7f1b74318000, 8392704) = 0
[pid  1640] munmap(0x7f1b73b17000, 8392704) = 0
[pid  1640] munmap(0x7f1b73316000, 8392704) = 0
```

Графики характеристики





Вывод

В процессе выполнения лабораторной работы я провёл много часов за чтением мануалов по различным командам мониторинга, изучал как работают примитивы синхронизации, в особенности познакомился с futex и использовал его в своей лабораторной, блокируя доступ к файлу пока он полностью не заполнится данными либо полностью не прочитается функцией аггрегатором. Столкнулся с проблемой, что потоки чтения для аггрегатных функций запускались раньше, чем потоки записи файлов, из-за чего функция пыталась прочитать файлы, которые ещё не были заполнены. Провёл часы ожидания вывода strace, оказалось что вывод команды настолько большой, что заполнил всё размапленное место на виртуалке. В итоге немного научился работать с lvm и расширил себе логический раздел, который на моё удивление оказался рутом. В итоге вывод strace занял целых 5гб. Также я первый раз писал многопоточное приложение на си для linux и это был интересный опыт для меня.