

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет систем управления и робототехники

Операционные системы

Лабораторная работа № 1

Выполнил студент

Ефаринов Павел Андреевич

Группа № Р33122

Преподаватели: Клименков Сергей Викторович

Покид Александр Владимирович

г. Санкт-Петербург

2020

Вариант: A=186;B=0xA6EF4BD5;C=mmap;D=32;E=112;F=block;G=37;H=seq;I=27;J=sum;K=sem

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 star построить графики системных характеристик.

Отчет:

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

```

const size_t A = 186 * 1024 * 1024;
void* B = (void*) 0xA6EF4BD5;
void* page_aligned_B = (void*) 0xA6EF4000;
const int D = 32;
const int E = 112 * 1024 * 1024;
const size_t G = 37;
const int I = 27;
const int file_number = 20;
_Atomic unsigned long long processed_files_count = 0;
sem_t files_semaphore;

struct memory_filler_data {
    size_t part_length;
    void* address;
    FILE* random_file;
};

void* memory_filler_thread(void* in_thread_data) {
    struct memory_filler_data thread_data = *(struct memory_filler_data*)
in_thread_data;
    fread(thread_data.address, thread_data.part_length, 1,
thread_data.random_file);
    free(in_thread_data);
    pthread_exit(NULL);
}

_Noreturn void* file_aggregator_thread() {
    while (1) {
        sem_wait(&files_semaphore);
        printf("Aggregator thread awakened!\n");
        unsigned long long current_file = processed_files_count++;
        char filename[20];
        sprintf(filename, "%llu", current_file);

        long long sum = 0;
        int num;

        FILE* output_file = fopen(filename, "r");

        while (fread(&num, sizeof(int), 1, output_file) == 1) {
            sum += num;
        }

        fclose(output_file);
        printf("My current file is: %llu It's sum is: %lld\n", current_file,
sum);

        remove(filename);
    }
    pthread_exit(NULL);
}

```

```

void fill_memory(void* memory_address, size_t memory_size) {

    //before allocation

    mmap(memory_address, memory_size, PROT_READ | PROT_WRITE, MAP_ANONYMOUS |
MAP_PRIVATE, -1, 0);

    //after allocation

    FILE* random_file = fopen("/dev/urandom", "r");

    pthread_t filler_threads[D];
    size_t part_length = memory_size / D;
    for (int i = 0; i < D; ++i) {
        struct memory_filler_data* thread_data = malloc(sizeof(struct
memory_filler_data));

        thread_data->part_length = part_length;
        thread_data->address = memory_address + part_length * i;
        thread_data->random_file = random_file;

        pthread_create(&filler_threads[i], NULL, memory_filler_thread, (void*)
thread_data);
    }

    for (int i = 0; i < D; ++i) {
        pthread_join(filler_threads[i], NULL);
    }

    fclose(random_file);

    //after memory fill
}

void fill_file_from_memory(void* memory_address, size_t data_length, char*
filename) {
    FILE* output_file = fopen(filename, "w");
    fwrite(memory_address, G, data_length / G + 1, output_file);
    fclose(output_file);
}

void* fill_memory_write_file() {
    sem_init(&files_semaphore, 0, 0);
    unsigned long long created_files_count = 0;
    while (1) {
        // for (int i = 0; i < file_number; ++i) {
        fill_memory(B, A);
        char* filename = malloc(20 * sizeof(char));
        sprintf(filename, "%llu", created_files_count);
        printf("Current file: %s\n", filename);
        fill_file_from_memory(B, E, filename);
    }
}

```

```

        created_files_count++;
        sem_post(&files_semaphore);

        munmap(page_aligned_B, A);
        free(filename);

        // after deallocating
    }
    sem_destroy(&files_semaphore);
    pthread_exit(NULL);
}

void* aggregate_files() {
    sleep(25);
    pthread_t aggregation_threads[I];
    for (int i = 0; i < D; ++i) {
        pthread_create(&aggregation_threads[i], NULL, file_aggregator_thread,
NULL);
    }
    pthread_exit(NULL);
}

int main() {
    pthread_t* memory = malloc(sizeof(pthread_t));
    pthread_t* files = malloc(sizeof(pthread_t));

    pthread_create(memory, NULL, fill_memory_write_file, NULL);
    pthread_create(files, NULL, aggregate_files, NULL);

    pthread_exit(NULL);
}

```

Ход выполнения.

Мониторинг и профилирование выполнялись в Ubuntu 18.04. Средствами мониторинга являлись утилиты gdb, htop, ps, rmap, iostat, spstat (top не хотел показывать ресурсы потоков).

Результаты замеров потребляемой памяти.

Точка замера	Результат замера
До аллокации	<pre> 1 [] 11.7% Tasks: 150, 447 thr: 1 running 2 [] 7.3% Load average: 0.43 0.40 0.39 3 [] 14.5% Uptime: 23:27:21 4 [] 4.1% Mem[] 1.97G/5.70G Swp[] 157M/976M PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command 11287 l1zzka239 20 0 99048 820 688 t 0.0 0.0 0:00.00 /home/l1zzka239/CLionProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre 11288 l1zzka239 20 0 99048 820 688 t 0.0 0.0 0:00.00 /home/l1zzka239/CLionProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre 11289 l1zzka239 20 0 99048 820 688 t 0.0 0.0 0:00.00 /home/l1zzka239/CLionProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre 11283 l1zzka239 20 0 99048 820 688 t 0.0 0.0 0:00.00 /home/l1zzka239/CLionProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre 11273 l1zzka239 20 0 99048 820 688 t 0.0 0.0 0:00.00 /home/l1zzka239/CLionProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre </pre>

Загрузка CPU:

```

File Edit View Search Terminal Help

1 [||||||||||||||||||||||||||||||||||||||||| 55.9%] Tasks: 154, 674 thr; 2 running
2 ||||||||||||||||||||||||||||||||||||||| 51.9%] Load average: 2.24 1.00 0.61
3 ||||||||||||||||||||||||||||||||||||||| 76.6%] Uptime: 23:33:15
4 [||||||||||||||||||||||||||||||||||||| 26.5%]

Mem[||||||||||||||||||||||||||||||||||||| 2.57G/5.70G]
Swp[||||||||||||||||| 157M/976M]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
11796 l1zka239 20 0 1754M 99M 1840 s 81.6 1.7 0:01.36 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
12351 l1zka239 20 0 1754M 99M 1840 R 3.2 1.7 0:00.05 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11778 l1zka239 20 0 1754M 99M 1840 s 0.6 1.7 0:01.54 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11788 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.43 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11789 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.16 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11790 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.30 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11780 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.53 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11782 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.26 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11784 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.45 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11786 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.23 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11790 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.24 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11806 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.20 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11791 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.17 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11805 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.19 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11807 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11808 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11803 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:01.29 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11781 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11783 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11705 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11792 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11787 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11793 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11794 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11797 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11798 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre
11799 l1zka239 20 0 1754M 99M 1840 s 0.0 1.7 0:00.00 /home/l1zka239/CLlonProjects/osThreadMassacre/cnake-build-debug/osThreadMassacre

File Edit View Search Terminal Help
F1Help F2Setup F3Search F4Filter F5Free F6Sortby F7Lic - F8Lic + F9All F10Quit

```

Статистика IO:

Среднее за 10с (при замере 1 раз в секунду загрузка доходила до 100МБ/с):

```
Total DISK READ : 0.00 B/s | Total DISK WRITE : 44.69 M/s
Actual DISK READ: 0.00 B/s | Actual DISK WRITE: 14.72 K/s

TID PRIO< USER      DISK READ  DISK WRITE  SWAPIN     IO    COMMAND
253 be/3 root        0.00 B/s   9.55 K/s   0.00 %   0.06 % [jbd2/dm-0-8]
295 be/3 root        0.00 B/s  136.03 K/s  0.00 %   0.00 % systemd-journald
869 be/4 syslog      0.00 B/s 2036.51 B/s  0.00 %   0.00 % rsyslogd -n [rs:main Q:Reg]
11778 be/4 lizzka23  0.00 B/s  44.55 M/s   0.00 %   0.00 % osThreadMassacre
11780 be/4 lizzka23  0.00 B/s   0.00 B/s   0.00 %   0.00 % osThreadMassacre
11789 be/4 lizzka23  0.00 B/s   0.00 B/s   0.00 %   0.00 % osThreadMassacre
11795 be/4 lizzka23  0.00 B/s   0.00 B/s   0.00 %   0.00 % osThreadMassacre
19748 be/4 root        0.00 B/s   0.00 B/s   0.00 %   0.08 % [kworker/u16:1-phy0]
27012 be/4 root        0.00 B/s   0.00 B/s   0.00 %   0.04 % [kworker/u16:0-events_unbound]

(base) lizzka239@lizzka239-Lenovo-Z50-70:~/CLionProjects/osThreadMassacre/cmake-build-debug$ iostat
Linux 5.4.0-52-generic (lizzka239-Lenovo-Z50-70)      01.11.2020      _x86_64_      (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           9,66    0,04    8,64    0,52    0,00   81,14

Device            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
loop0              0,00         0,03         0,00       2310         0
loop1              0,08         0,10         0,00       8420         0
loop2              0,01         0,02         0,00       1756         0
loop3              0,00         0,01         0,00        658         0
loop4              3,30         3,33         0,00     293347         0
loop5              0,11         0,14         0,00     12384         0
loop6              0,00         0,00         0,00        239         0
loop7              2,10         2,12         0,00    186851         0
sda                5,51        85,25       219,91   7506006   19361881
dm-0               8,79        84,42       218,86   7432389   19269156
dm-1               0,46         0,06         1,83     5128     160912
loop8              0,00         0,00         0,00        226         0
loop9              4,21         4,23         0,00    372246         0
loop10             0,00         0,00         0,00         87         0
loop11             0,00         0,01         0,00       1098         0
loop12             0,03         0,05         0,00       4592         0
loop13             0,00         0,02         0,00       1551         0
loop14             0,02         0,03         0,00       2228         0
loop15             0,01         0,03         0,00       2497         0
loop16             0,01         0,02         0,00       1361         0
loop17             0,00         0,03         0,00       2278         0
loop18             0,00         0,01         0,00        892         0
loop19             0,03         0,04         0,00       3189         0
loop20             0,00         0,00         0,00         51         0
loop21             0,00         0,01         0,00        597         0
loop22             0,00         0,00         0,00         52         0
loop23             0,00         0,00         0,00        140         0
loop24             0,00         0,00         0,00        124         0
loop25             0,00         0,01         0,00       1285         0
```

Вывод:

В результате выполнения лабораторной работы я понял, как можно осуществлять профилирование и мониторинг приложений в OS *nix. Изучил основы многопоточного программирования на языке C.