

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики
Факультет программной инженерии и компьютерной техники

Операционные системы
Лабораторная №1

Выполнил:
Беляков Дмитрий
Группа:
Р33122
Преподаватель:
Александр Покид

Санкт-Петербург
2020

Задание

A=248;B=0x936F88A5;C=mmap;D=131;E=170;F=block;G=112;H=random;I=29;J=sum;K=cv

Разработать программу на языке C, которая осуществляет следующие действия

Создает область памяти размером 248 мегабайт, начинающихся с адреса 0x936F88A5 (если возможно) при помощи mmap заполненную случайными числами /dev/urandom в 131 поток. Используя системные средства мониторинга определите адрес начала в адресном пространстве процесса и характеристики выделенных участков памяти. Замеры виртуальной/физической памяти необходимо снять:

До аллокации

После аллокации

После заполнения участка данными

После деаллокации

Записывает область памяти в файлы одинакового размера 170 мегабайт с использованием блочного обращения к диску. Размер блока ввода-вывода 112 байт. Преподаватель выдает в качестве задания последовательность записи/чтения блоков (случайный)

Генерацию данных и запись осуществлять в бесконечном цикле.

В отдельных 29 потоках осуществлять чтение данных из файлов и подсчитывать агрегированные характеристики данных - сумму.

Чтение и запись данных в/из файла должна быть защищена примитивами синхронизации cv.

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

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

#define A 248
#define B 0x936F88A5
```

```

#define C mmap
#define D 131
#define E 170
#define F block
#define G 112
#define H random
#define I 29
#define J sum
#define K cv

typedef struct{
    unsigned char * start;
    size_t count;
    FILE * urandom;
} wrn_args;

typedef struct {
    unsigned int file_number;
    pthread_mutex_t * mutex;
    pthread_cond_t * cv;
} ras_args;

void* write_random_numbers(void * wrn_data){
    wrn_args *data = (wrn_args*) wrn_data;
    for (size_t i = 0; i < data->count; ) {
        i += fread(data->start + i, 1, data->count - i, data-
>urandom);
    }
    return NULL;
}

void generate_and_write(unsigned char *src,
                        unsigned int file_number,
                        pthread_mutex_t * mutex,
                        pthread_cond_t * cv){

    FILE * urandom = fopen("/dev/urandom", "rb");
    size_t wrn_part = A * 1024 * 1024 / D;
    unsigned char * start = src;
    pthread_t wrn_threads[D];
    for (int i = 0; i<D-1; i++){
        wrn_args wrn_data = {start, wrn_part, urandom};
        pthread_create(&(wrn_threads[i]), NULL, write_random_numbers,
&wrn_data);
        start+=wrn_part;
    }

    wrn_args wrn_data = {start, wrn_part + A * 1024 * 1024 % D,
urandom};

```

```

pthread_create(&(wrn_threads[D-1]), NULL, write_random_numbers,
&wrn_data);
for(int i = 0; i < D; i++)
    pthread_join(wrn_threads[i], NULL);

fclose(urandom);
pthread_mutex_lock(mutex);
for (int i = 0; i < file_number; i++){
    char result_name[13];
    snprintf(result_name,13, "lab_os_%d.bin", i);
    FILE * file = fopen(result_name, "wb");
    size_t file_size = E*1024*1024;
    for (int j = 0; j < file_size; ){
        unsigned int block = rand() % (A * 1024 * 1024 / G);
        unsigned int block_size = file_size - j < G ? file_size -
j < G : G;
        j += fwrite( src + block*G, 1, block_size, file);
    }
    fclose(file);
    printf("%d data_generated\n", i);
}
pthread_cond_broadcast(cv);
pthread_mutex_unlock(mutex);
}

int convert_char_buf_2_int(unsigned char buf[]){
    int sum = 0;
    for (int i = 0; i < G / sizeof (int); i+=sizeof (int)){
        int num = 0;
        for (int j = 0; j < sizeof (int); j++){
            num = (num<<8) + buf[i+j];
        }
        sum += num;
    }
    return sum;
}

_Noreturn void* read_and_sum(void * ras_data){
    ras_args * args = (ras_args*) ras_data;
    char filename[13];
    snprintf(filename, 13, "lab_os_%d.bin", args->file_number);
    while(1){
        long long sum = 0;
        pthread_mutex_lock(args->mutex);
        printf("%ld wait on cond\n", pthread_self());
        pthread_cond_wait(args->cv, args->mutex);
        printf("%ld waited\n", pthread_self());
        FILE *file = fopen(filename, "rb");
        unsigned char buf[G];

```

```

        for (unsigned int i = 0; i < 2*E*1024*1024/G; i++){
            unsigned int block = rand() % (2*E*1024*1024/G);
            fseek(file, block*G, 0);
            if (fread(&buf, 1, G, file) != G) continue;
            else sum+=convert_char_buf_2_int(buf);
        }
        printf("%lld\n", sum);
        fclose(file);
        pthread_mutex_unlock(args->mutex);
    }
}

unsigned char* allocate_memory(){
    return C((void *)B, A*1024*1024, PROT_READ | PROT_WRITE,
MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
}

int main() {
    unsigned char *src;
    printf("Before allocation");
    getchar();
    src = allocate_memory();
    if ( src == MAP_FAILED ){
        printf("Error: can't allocate to the address");
        return 1;
    }
    printf("After allocation");
    getchar();

    unsigned int file_number = A / E + (A % E > 0 ? 1 : 0);

    pthread_mutex_t mutex;
    pthread_cond_t cv;

    pthread_mutex_init(&mutex, NULL);
    pthread_cond_init(&cv, NULL);

    generate_and_write(src, file_number, &mutex, &cv);
    printf("After filling");
    getchar();
    munmap(src, A*1024*1024);
    printf("After deallocation");
    getchar();

    src = allocate_memory();
    if ( src == MAP_FAILED ){
        printf("Error: can't allocate to the address");
        return 1;
    }
}

```

```

pthread_t ras_threads[I];

for (int i = 0; i < I; i++){
    int file_count = i / (I / file_number);
    if (file_count == file_number) file_count--;
    ras_args ras_data = {file_count, &mutex, &cv};
    pthread_create(&ras_threads[i], NULL, read_and_sum,
&ras_data);
}

while(1)
    generate_and_write(src, file_number, &mutex, &cv);

for (int i = 0; i < I; i++)
    pthread_join( ras_threads[0], NULL);

return 0;
}

```

Сбор статистики

1. ps -A | grep main — узнаём PID
2. rmap -x PID — смотрим на память

Before allocation

```

49006:    ./main
Address      Kbytes      RSS      Dirty Mode  Mapping
000055d6c9acc000      4        4        0 r---- main
000055d6c9acd000      4        4        0 r-x-- main
000055d6c9ace000      4        4        0 r---- main
000055d6c9acf000      4        4        4 r---- main
000055d6c9ad0000      4        4        4 rw--- main
000055d6ca0a3000     132        4        4 rw--- [ anon ]
00007f99ce15b000      12         8        8 rw--- [ anon ]
00007f99ce15e000     152       152        0 r---- libc-2.32.so
00007f99ce184000    1332       752        0 r-x-- libc-2.32.so
00007f99ce2d1000     304        64        0 r---- libc-2.32.so
00007f99ce31d000      12        12       12 r---- libc-2.32.so
00007f99ce320000      12        12       12 rw--- libc-2.32.so
00007f99ce323000      16         8        8 rw--- [ anon ]
00007f99ce327000      28        28        0 r---- libpthread-2.32.so
00007f99ce32e000      64        64        0 r-x-- libpthread-2.32.so
00007f99ce33e000      20         0        0 r---- libpthread-2.32.so
00007f99ce343000       4         4        4 r---- libpthread-2.32.so
00007f99ce344000       4         4        4 rw--- libpthread-2.32.so
00007f99ce345000      24        12       12 rw--- [ anon ]
00007f99ce382000       8         8        0 r---- ld-2.32.so
00007f99ce384000     132       132        0 r-x-- ld-2.32.so

```

00007f99ce3a5000	36	36	0	r----	ld-2.32.so
00007f99ce3ae000	4	4	4	r----	ld-2.32.so
00007f99ce3af000	8	8	8	rw---	ld-2.32.so
00007ffd86468000	132	12	12	rw---	[stack]
00007ffd86569000	12	0	0	r----	[anon]
00007ffd8656c000	4	4	0	r-x--	[anon]
fffffffffff600000	4	0	0	--x--	[anon]

total kB	2476	1348	96		

After allocation

```
49006: ./main
```

Address	Kbytes	RSS	Dirty	Mode	Mapping
00000000936f8000	253952	0	0	rw---	[anon]
000055d6c9acc000	4	4	0	r----	main
000055d6c9acd000	4	4	0	r-x--	main
000055d6c9ace000	4	4	0	r----	main
000055d6c9acf000	4	4	4	r----	main
000055d6c9ad0000	4	4	4	rw---	main
000055d6ca0a3000	132	4	4	rw---	[anon]
00007f99ce15b000	12	8	8	rw---	[anon]
00007f99ce15e000	152	152	0	r----	libc-2.32.so
00007f99ce184000	1332	752	0	r-x--	libc-2.32.so
00007f99ce2d1000	304	64	0	r----	libc-2.32.so
00007f99ce31d000	12	12	12	r----	libc-2.32.so
00007f99ce320000	12	12	12	rw---	libc-2.32.so
00007f99ce323000	16	8	8	rw---	[anon]
00007f99ce327000	28	28	0	r----	libpthread-2.32.so
00007f99ce32e000	64	64	0	r-x--	libpthread-2.32.so
00007f99ce33e000	20	0	0	r----	libpthread-2.32.so
00007f99ce343000	4	4	4	r----	libpthread-2.32.so
00007f99ce344000	4	4	4	rw---	libpthread-2.32.so
00007f99ce345000	24	12	12	rw---	[anon]
00007f99ce382000	8	8	0	r----	ld-2.32.so
00007f99ce384000	132	132	0	r-x--	ld-2.32.so
00007f99ce3a5000	36	36	0	r----	ld-2.32.so
00007f99ce3ae000	4	4	4	r----	ld-2.32.so
00007f99ce3af000	8	8	8	rw---	ld-2.32.so
00007ffd86468000	132	12	12	rw---	[stack]
00007ffd86569000	12	0	0	r----	[anon]
00007ffd8656c000	4	4	0	r-x--	[anon]
fffffffffff600000	4	0	0	--x--	[anon]

total kB	256428	1348	96		

After filling

```
49006: ./main
```

Address	Kbytes	RSS	Dirty	Mode	Mapping
---------	--------	-----	-------	------	---------

00000000936f8000	253952	151296	151296	rw---	[anon]
000055d6c9acc000	4	4	0	r----	main
000055d6c9acd000	4	4	0	r-x--	main
000055d6c9ace000	4	4	0	r----	main
000055d6c9acf000	4	4	4	r----	main
000055d6c9ad0000	4	4	4	rw---	main
000055d6ca0a3000	132	40	40	rw---	[anon]
00007f9988781000	4	0	0	-----	[anon]
00007f9988782000	8192	8	8	rw---	[anon]
00007f9988f82000	4	0	0	-----	[anon]
00007f9988f83000	8192	8	8	rw---	[anon]
00007f9989783000	4	0	0	-----	[anon]
00007f9989784000	8192	8	8	rw---	[anon]
00007f9989f84000	4	0	0	-----	[anon]
00007f9989f85000	8192	8	8	rw---	[anon]
00007f99c8000000	132	8	8	rw---	[anon]
00007f99c8021000	65404	0	0	-----	[anon]
00007f99ce15b000	12	8	8	rw---	[anon]
00007f99ce15e000	152	104	0	r----	libc-2.32.so
00007f99ce184000	1332	876	0	r-x--	libc-2.32.so
00007f99ce2d1000	304	52	0	r----	libc-2.32.so
00007f99ce31d000	12	12	12	r----	libc-2.32.so
00007f99ce320000	12	12	12	rw---	libc-2.32.so
00007f99ce323000	16	8	8	rw---	[anon]
00007f99ce327000	28	24	0	r----	libpthread-2.32.so
00007f99ce32e000	64	64	0	r-x--	libpthread-2.32.so
00007f99ce33e000	20	20	0	r----	libpthread-2.32.so
00007f99ce343000	4	4	4	r----	libpthread-2.32.so
00007f99ce344000	4	4	4	rw---	libpthread-2.32.so
00007f99ce345000	24	12	12	rw---	[anon]
00007f99ce382000	8	8	0	r----	ld-2.32.so
00007f99ce384000	132	132	0	r-x--	ld-2.32.so
00007f99ce3a5000	36	36	0	r----	ld-2.32.so
00007f99ce3ae000	4	4	4	r----	ld-2.32.so
00007f99ce3af000	8	8	8	rw---	ld-2.32.so
00007ffd86468000	132	12	12	rw---	[stack]
00007ffd86569000	12	0	0	r----	[anon]
00007ffd8656c000	4	4	0	r-x--	[anon]
fffffffffff600000	4	0	0	--x--	[anon]
-----	-----	-----	-----		
total kB	354748	152800	151468		

After deallocation

49006:	./main				
Address	Kbytes	RSS	Dirty	Mode	Mapping

000055d6c9acc000	4	4	0	r----	main
000055d6c9acd000	4	4	0	r-x--	main
000055d6c9ace000	4	4	0	r----	main
000055d6c9acf000	4	4	4	r----	main
000055d6c9ad0000	4	4	4	rw---	main
000055d6ca0a3000	132	40	40	rw---	[anon]
00007f9988781000	4	0	0	-----	[anon]
00007f9988782000	8192	8	8	rw---	[anon]
00007f9988f82000	4	0	0	-----	[anon]
00007f9988f83000	8192	8	8	rw---	[anon]
00007f9989783000	4	0	0	-----	[anon]
00007f9989784000	8192	8	8	rw---	[anon]
00007f9989f84000	4	0	0	-----	[anon]
00007f9989f85000	8192	8	8	rw---	[anon]
00007f99c8000000	132	8	8	rw---	[anon]
00007f99c8021000	65404	0	0	-----	[anon]
00007f99ce15b000	12	8	8	rw---	[anon]
00007f99ce15e000	152	152	0	r----	libc-2.32.so
00007f99ce184000	1332	876	0	r-x--	libc-2.32.so
00007f99ce2d1000	304	52	0	r----	libc-2.32.so
00007f99ce31d000	12	12	12	r----	libc-2.32.so
00007f99ce320000	12	12	12	rw---	libc-2.32.so
00007f99ce323000	16	8	8	rw---	[anon]
00007f99ce327000	28	24	0	r----	libpthread-2.32.so
00007f99ce32e000	64	64	0	r-x--	libpthread-2.32.so
00007f99ce33e000	20	20	0	r----	libpthread-2.32.so
00007f99ce343000	4	4	4	r----	libpthread-2.32.so
00007f99ce344000	4	4	4	rw---	libpthread-2.32.so
00007f99ce345000	24	12	12	rw---	[anon]
00007f99ce382000	8	8	0	r----	ld-2.32.so
00007f99ce384000	132	132	0	r-x--	ld-2.32.so
00007f99ce3a5000	36	36	0	r----	ld-2.32.so
00007f99ce3ae000	4	4	4	r----	ld-2.32.so
00007f99ce3af000	8	8	8	rw---	ld-2.32.so
00007ffd86468000	132	12	12	rw---	[stack]
00007ffd86569000	12	0	0	r----	[anon]
00007ffd8656c000	4	4	0	r-x--	[anon]
ffffffffffff600000	4	0	0	--x--	[anon]
-----	-----	-----	-----	-----	-----
total kB	100796	1552	172		

3. ps -T -p PID -смотрим threads процесса

PID	SPID	TTY	TIME	CMD
49006	49006	pts/1	00:00:02	main
49006	51612	pts/1	00:00:00	main
49006	51613	pts/1	00:00:00	main
49006	51614	pts/1	00:00:00	main
49006	51615	pts/1	00:00:00	main

```

49006 51616 pts/1 00:00:00 main
49006 51617 pts/1 00:00:00 main
49006 51618 pts/1 00:00:00 main
49006 51619 pts/1 00:00:05 main
49006 51620 pts/1 00:00:00 main
49006 51621 pts/1 00:00:00 main
49006 51622 pts/1 00:00:00 main
49006 51623 pts/1 00:00:00 main
49006 51624 pts/1 00:00:00 main
49006 51625 pts/1 00:00:00 main
49006 51626 pts/1 00:00:00 main
49006 51627 pts/1 00:00:00 main
49006 51628 pts/1 00:00:00 main
49006 51629 pts/1 00:00:00 main
49006 51630 pts/1 00:00:00 main
49006 51631 pts/1 00:00:00 main
49006 51632 pts/1 00:00:05 main
49006 51633 pts/1 00:00:00 main
49006 51634 pts/1 00:00:00 main
49006 51635 pts/1 00:00:00 main
49006 51636 pts/1 00:00:00 main
49006 51637 pts/1 00:00:00 main
49006 51638 pts/1 00:00:00 main
49006 51639 pts/1 00:00:00 main
49006 51640 pts/1 00:00:00 main
49006 51813 pts/1 00:00:00 main
49006 51821 pts/1 00:00:00 main
49006 51825 pts/1 00:00:00 main

```

...

или

top -H -p PID

```

top - 23:25:58 up 10:06, 1 user, load average: 0,97, 1,82, 1,61
Threads: 30 total, 0 running, 30 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5,8 us, 2,4 sy, 0,0 ni, 91,1 id, 0,2 wa, 0,4 hi, 0,2 si,
0,0 st
MiB Mem : 7872,5 total, 368,7 free, 5856,4 used, 1647,5 buff/cache
MiB Swap: 0,0 total, 0,0 free, 0,0 used. 541,0 avail Mem

```

	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	
COMMAND												
	49006	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:02.05	main
	51612	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51613	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51614	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51615	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51616	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51617	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51618	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:00.00	main
	51619	kevinche	20	0	1641004	179124	288	S	0,0	2,2	0:21.80	main

```

51620 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:00.00 main
51621 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:00.00 main
51622 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:00.00 main
51623 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:00.00 main
51624 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:00.00 main
51625 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:10.25 main
51626 kevinche 20 0 1641004 179124 288 S 0,0 2,2 0:10.34 main

```

Чтобы поменять приоритет

```
sudo chrt -r -p PR TID
```

Посмотреть загрузку на CPU

```
watch -n 1 pidstat -p PID
```

```

11:36:28      UID      PID    %usr %system  %guest  %wait   %CPU   CPU  Command
11:36:28    1000    82343    0,15    0,22    0,00    0,00    0,37     3   main

```

или

```
top -H -p PID
```

```

top - 11:40:03 up 22:20,  1 user,  load average: 1,54, 1,43, 1,40
Threads:  30 total,   1 running,  29 sleeping,   0 stopped,   0 zombie
%Cpu(s):  6,0 us,  6,8 sy,   0,0 ni, 87,1 id,   0,1 wa,   0,0 hi,   0,0 si,   0,0 st
MiB Mem :  7872,5 total,  1057,6 free,  4084,9 used,  2730,1 buff/cache
MiB Swap:   0,0 total,    0,0 free,    0,0 used.  2045,9 avail Mem

```

Посмотреть на статистику по IO

```
watch -n 1 pidstat -d -p 82343
```

```

11:43:53      UID      PID  kB_rd/s  kB_wr/s kB_ccwr/s iodelay  Command
11:43:53    1000    82343    2,82    17,26    0,00    127   main

```

Посмотреть на системные вызовы

```
sudo strace -p PID
```

```

write(3, "\323U.\231k\373@I\210\246*\\"\303\331=W\340\306\2610'0:\257G\
345\256\356\225\306\f\244"...", 4096) = 4096
write(3, "w\22\371\264\336\260\266\304&\35\7\374E\203kG\207\232o\24+\3123#4t\
324\231]\322\360\317"...", 4096) = 4096
write(3, "\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"...
, 4096) = 4096
write(3, "\17{<\313A\10\235\1\37+\364\275r%-80c\342\221+\3724Y?*F\
213\371\252\311\273"...", 4096) = 4096
write(3, "\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"...
, 4096) = 4096
write(3, "\316\343\367\324B=\262\177\330[\254\344\340\367\275\213\34w\2=\211&\
354\222\240.\257\255\20\373f\241"...", 4096) = 4096
write(3, "\322\340\366\273cuQ3\357Wl\212\222\257\324\340\36\4F\270c\
30\1\372\304\260\265y\371\363\322\260"...", 4096) = 4096

```

Замеры io

```
sudo iotop
```

Total DISK READ :		0.00 B/s		Total DISK WRITE :		312.13 M/s		
Actual DISK READ:		0.00 B/s		Actual DISK WRITE:		372.72 M/s		
TID	PRI	USER		DISK READ	DISK WRITE	SWAPIN	IO>	COMMAND
57667	be/4	kevinche		0.00 B/s	312.13 M/s	0.00 %	4.75 %	./main

Использование systemtap

```
sudo stap -x $(pidof main) system_monitor.stp
```

				read	read		write	write
name	open	read	MB tot	B avg	write	MB tot	B avg	
main	0	207	193	978192	217606	850	4095	

Вывод: в данной лабораторной я преисполнился кодом на C, воспроизвёл основные операции программ, попробовал несколько средств мониторинга для наблюдения за состоянием системы в целом, а также процесса.