

Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

Факультет ПИиКТ

Лабораторная работа №1
по дисциплине
«Операционные системы»

Выполнил: студент группы Р33113
Лукьяненко Никита Игоревич

Преподаватель:
Покид Александр Владимирович

г. Санкт-Петербург
2020 г.

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.

Выполнение

Адрес начала в адресном пространстве процесса и характеристики выделенных участков памяти:

mmap -x [pid]

10436: ./main					
Address	Kbytes	RSS	Dirty	Mode	Mapping
000055cf55bd2000	4	4	0	r----	main
000055cf55bd3000	4	4	0	r-x--	main
000055cf55bd4000	4	4	0	r----	main
000055cf55bd5000	4	4	4	r----	main
000055cf55bd6000	4	4	4	rw---	main
000055cf56bfb000	132	12	12	rw---	[anon]
00007f40a9032000	4	0	0	-----	[anon]
00007f40a9033000	8192	8	8	rw---	[anon]
00007f40a9833000	4	0	0	-----	[anon]
00007f40a9834000	8192	8	8	rw---	[anon]
00007f40aa034000	4	0	0	-----	[anon]
00007f40aa035000	8192	8	8	rw---	[anon]
00007f40aa835000	4	0	0	-----	[anon]
00007f40aa836000	8192	8	8	rw---	[anon]
00007f40b604c000	260112	260108	260108	rw---	[anon]
00007f40c5e50000	148	144	0	r----	libc-2.31.so
00007f40c5e75000	1504	1060	0	r-x--	libc-2.31.so
00007f40c5fed000	296	64	0	r----	libc-2.31.so
00007f40c6037000	4	0	0	-----	libc-2.31.so
00007f40c6038000	12	12	12	r----	libc-2.31.so
00007f40c603b000	12	12	12	rw---	libc-2.31.so
00007f40c603e000	16	16	16	rw---	[anon]
00007f40c6042000	28	28	0	r----	libpthread-2.31.so

00007f40c6049000	68	68	0	r-x--	libpthread-2.31.so
00007f40c605a000	20	0	0	r----	libpthread-2.31.so
00007f40c605f000	4	4	4	r----	libpthread-2.31.so
00007f40c6060000	4	4	4	rw---	libpthread-2.31.so
00007f40c6061000	24	12	12	rw---	[anon]
00007f40c6071000	4	4	0	r----	ld-2.31.so
00007f40c6072000	140	140	0	r-x--	ld-2.31.so
00007f40c6095000	32	32	0	r----	ld-2.31.so
00007f40c609e000	4	4	4	r----	ld-2.31.so
00007f40c609f000	4	4	4	rw---	ld-2.31.so
00007f40c60a0000	4	4	4	rw---	[anon]
00007ffcdade5000	132	12	12	rw---	[stack]
00007ffcdabc1000	12	0	0	r----	[anon]
00007ffcdabc4000	8	4	0	r-x--	[anon]

total kB	295528	261800	260244		

Замеры виртуальной/физической памяти:

ps -eo pid,vsz,rss,comm | grep main

1. До аллокации

```
PID    VSZ    RSS    COM
10326  2644   616    main
```

2. После аллокации

```
10326 262744   616    main
```

3. После заполнения участка данными

```
10326 295528 261740    main
```

4. После деаллокации

```
10326  35428   1640    main
```

Значения затраченного процессорного времени на выполнение программы:

time ./main

```
real    1m11.253s
user    0m37.585s
sys     0m28.746s
```

Значения затраченного процессорного времени на операции ввода-вывода:

sudo strace -c -fp [pid]

% time	seconds	usecs/call	calls	errors	syscall

89.65	216.758376	54	3945775		write
9.62	23.268201	32957	706	1	futex
0.66	1.590793	5098	312		read
0.02	0.053892	160	335		mmap
0.01	0.033838	352	96		clone
0.01	0.023869	76	311		openat
0.01	0.015083	157	96		set_robust_list
0.00	0.011999	77	154		mprotect
0.00	0.006239	64	96		madvise
0.00	0.004230	13	319		munmap
0.00	0.003065	9	311		close
0.00	0.000158	0	561	1	lseek
0.00	0.000000	0	1		brk

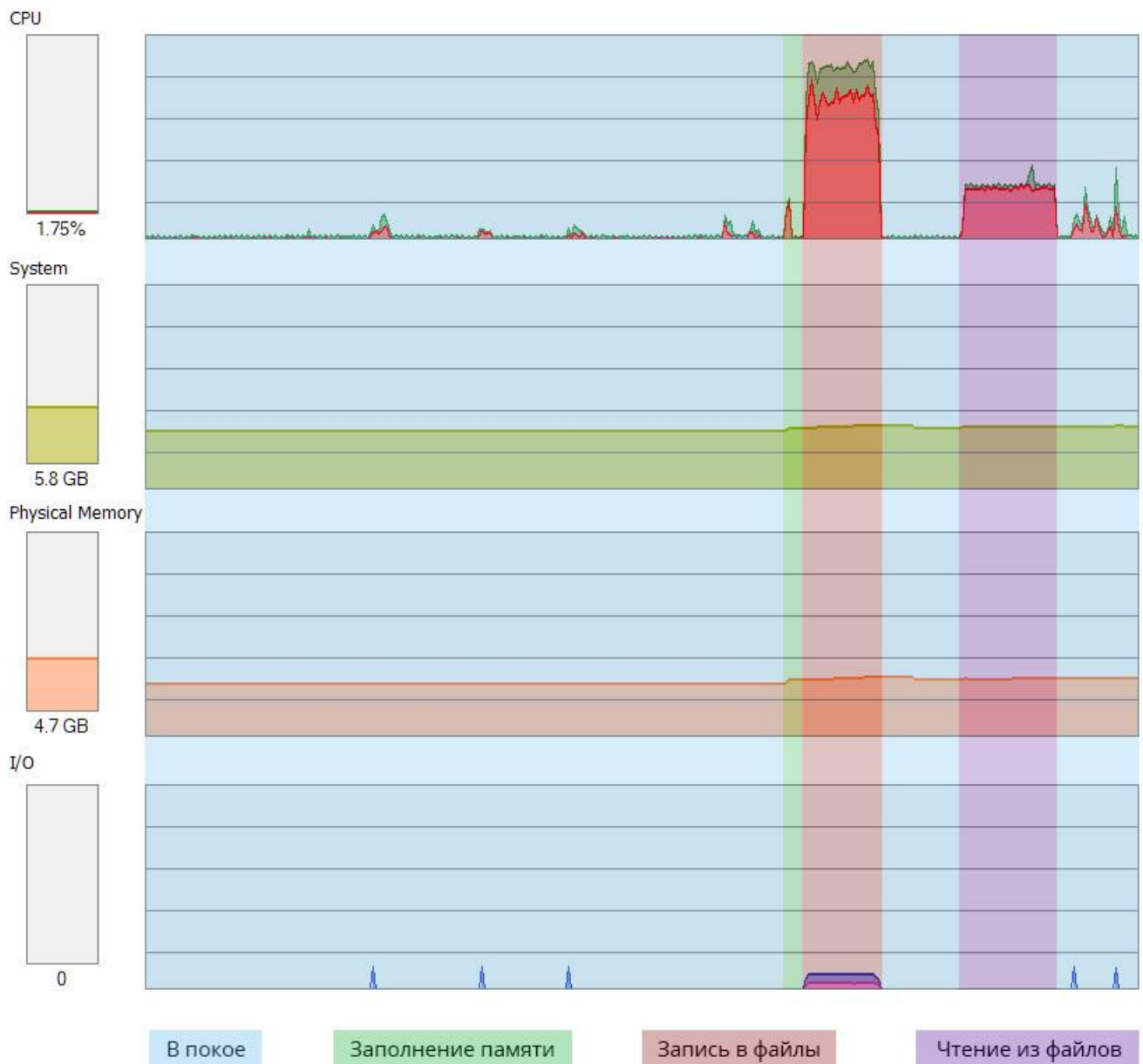
100.00	241.769743		3949073	2	total

Трасса системных вызовов:

sudo strace -fp [pid]

```
strace: Process 10508 attached
read(0, "\n", 1024) = 1
mmap(NULL, 266342400, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fe4cf26f000
write(1, "After allocation [PRESS ENTER]\n", 31) = 31
read(0, "\n", 1024) = 1
write(1, "Filling memory with random numbe...", 52) = 52
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe4cea6e000
mprotect(0x7fe4cea6f000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7fe4cf26dfb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[10519], tls=0x7fe4cf26e700, child_tidptr=0x7fe4cf26e9d0) = 10519
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe4ce26d000
mprotect(0x7fe4ce26e000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7fe4cea6cfb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[10520], tls=0x7fe4cea6d700, child_tidptr=0x7fe4cea6d9d0) = 10520
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe4cda6c000
mprotect(0x7fe4cda6d000, 8388608, PROT_READ|PROT_WRITE) = 0
clone(child_stack=0x7fe4ce26bfb0, flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, parent_tid=[10521], tls=0x7fe4ce26c700, child_tidptr=0x7fe4ce26c9d0) = 10521
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7fe4cd26b000
mprotect(0x7fe4cd26c000, 8388608, PROT_READ|PROT_WRITE) = 0
```

Графики системных характеристик



Вывод

Выполняя данную лабораторную работу, я научился использовать системные средства мониторинга семейства ОС Linux для обнаружения фактических причин проблем с производительностью системы и исполняемых программ. Также ознакомился с особенностями программирования на C: потоками, блокировками, работой с файлами и памятью; и взаимодействия с операционной системой с помощью системных вызовов.