

# Университет ИТМО

## Лабораторная работа №2 по предмету «Архитектура программных систем»

Выполнил: Солодовников Дмитрий Сергеевич

Группа: Р33122

Преподаватель: Покид А. В.

г. Санкт-Петербург

2020/2021 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.

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.

## Снятые показания

### Замеры виртуальной и физической памяти

Утилита top

До аллокации:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10806	tuzserik	20	0	2640	620	544	S	0,0	0,0	0:00.00	lab1

После аллокации:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10806	tuzserik	20	0	160336	620	544	S	0,0	0,0	0:00.00	lab1

После заполнения данными:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10806	tuzserik	20	0	258656	3840	1488	S	29,0	0,1	0:00.87	lab1

После деаллокации:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
10806	tuzserik	20	0	297568	1724	1488	S	0,7	0,0	0:14.59	lab1

### Замеры затраченного процессорного времени

Утилита top

%CPU	%MEM	TIME+
0,7	0,0	0:14.59

### Замеры времени на операции ввода-вывода

Утилита iostat

avg-cpu:	%user	%nice	%system	%iowait	%steal	%idle
	1,20	0,27	10,63	1,23	0,00	86,67

## Адрес начала в адресном пространстве и характеристики

Утилита rmap

```
0000559233ea8000      4      4      0 r--- lab1
0000559233ea9000      4      4      0 r-x-- lab1
0000559233eaa000      4      4      0 r--- lab1
0000559233eab000      4      4      4 r--- lab1
0000559233eac000      4      4      4 rw--- lab1
0000559234578000    132      4      4 rw--- [ anon ]
00007f8ccbe40000    148    144      0 r--- libc-2.31.so
00007f8ccbe65000   1504    804      0 r-x-- libc-2.31.so
00007f8ccbfd000    296     64      0 r--- libc-2.31.so
00007f8ccc027000      4      0      0 ----- libc-2.31.so
00007f8ccc028000     12     12     12 r--- libc-2.31.so
00007f8ccc02b000     12     12     12 rw--- libc-2.31.so
00007f8ccc02e000     16     16     16 rw--- [ anon ]
00007f8ccc038000     28     28      0 r--- libpthread-2.31.so
00007f8ccc03f000     68     68      0 r-x-- libpthread-2.31.so
00007f8ccc050000     20      0      0 r--- libpthread-2.31.so
00007f8ccc055000      4      4      4 r--- libpthread-2.31.so
00007f8ccc056000      4      4      4 rw--- libpthread-2.31.so
00007f8ccc057000     16      4      4 rw--- [ anon ]
00007f8ccc075000     12      8      8 rw--- [ anon ]
00007f8ccc078000      4      4      0 r--- ld-2.31.so
00007f8ccc079000    140    140      0 r-x-- ld-2.31.so
00007f8ccc09c000     32     32      0 r--- ld-2.31.so
00007f8ccc0a5000      4      4      4 r--- ld-2.31.so
00007f8ccc0a6000      4      4      4 rw--- ld-2.31.so
00007f8ccc0a7000     12      8      8 rw--- [ anon ]
00007ffdfefebb000   132     12     12 rw--- [ stack ]
00007ffdfefeb000    12      0      0 r--- [ anon ]
```

## Трассировка системных вызовов

## Утилита strace

```
execve("./lab1", ["/lab1"], 0x7ffea09f4280 /* 52 vars */) = 0
brk(NULL) = 0x55fe5c4de000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffefd4289f0) = -1 EINVAL (Недопустимый аргумент)
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=65903, ...}) = 0
mmap(NULL, 65903, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f1206610000
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"...
, 832) = 832
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\00\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) = 0x7f120665a000
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\00\305\3743\364B\2216\244\224\306@\261\23\327o"...
, 68, 824) = 68
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f12065e8000
mmap(0x7f12065ef000, 69632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f12065ef000
mmap(0x7f1206600000, 20480, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f1206600000
mmap(0x7f1206605000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7f1206605000
mmap(0x7f1206607000, 13432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f1206607000
```

```

NYMOUS, -1, 0) = 0x7f1206607000
close(3) = 0
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\3\0>\0\1\0\0\0\360q\2\0\0\0\0\0"... , 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"... , 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"... , 784, 64) = 784
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) = 32
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\363\377?\332\200\270\27\304d\245n\355Y\377\t\334"... , 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f12063f0000
mprotect(0x7f1206415000, 1847296, PROT_NONE) = 0
mmap(0x7f1206415000, 1540096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f1206415000
mmap(0x7f120658d000, 303104, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f120658d000
mmap(0x7f12065d8000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f12065d8000
mmap(0x7f12065de000, 13528, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f12065de000
close(3) = 0

```

```

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f1206625000
arch_prctl(ARCH_SET_FS, 0x7f1206625740) = 0
mprotect(0x7f12065d8000, 12288, PROT_READ) = 0
mprotect(0x7f1206605000, 4096, PROT_READ) = 0
mprotect(0x55fe5bc7d000, 4096, PROT_READ) = 0
mprotect(0x7f1206655000, 4096, PROT_READ) = 0
munmap(0x7f1206610000, 65903) = 0
set_tid_address(0x7f1206625a10) = 2334
set_robust_list(0x7f1206625a20, 24) = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f12065efbf0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f12065fd3c0}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f12065efc90, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f12065fd3c0}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
brk(NULL) = 0x55fe5c4de000
brk(0x55fe5c4ff000) = 0x55fe5c4ff000
openat(AT_FDCWD, "/dev/urandom", O_RDONLY) = 3
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\224\320\276 \320\260\320\273\320\273\320\276\320\272\320\260\321\206\320\270\320\270", 23До аллокации) = 23
read(0,

```

## Статистика системных вызовов

Утилита stap

```

0      0      0      0
31     1     24     23
8      58     0     0
6      312    0     0
4      238    0     0
8      236    0     0
6      210    0     0
12     105    0     0
2      16729  0     0
18     0      0     0
6      10667  0     0
6      13729  0     0
4      13224  0     0
8      32967  0     0
10     5053   0     0
10     3750   0     0
18     4069   0     0
10     27560  0     0
1      25004   0     0
0      68021  0     0
0      25741  0     0
0      46285  0     0
0      51632  0     0
0      153263 0     0
0      24803  0     0
0      27658  0     0
0      18042  0     0
0      28125  0     0

```

## **Вывод**

В ходе выполнения данной лабораторной работы я познакомился с методами мониторинга выполнения приложений на системе Linux, а также программированию на языке C для этой системы, используя системозависимые команды уровня системных вызовов.