Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)
Факультет Программной инженерии и компьютерной техники

Лабораторная работа № 1

По дисциплине «Операционные системы»

Вариант:

A=190;B=0x7B10F87C;C=malloc;D=29;E=91;F=block;G=33;H=seq;I=17;J=min;K=sem

Выполнили:

Самойлова Анна,

Пилацис Дамир

P33211

Санкт - Петербург

2020г.

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.

Выполнение и анализ

Замеры виртуальной и физической памяти (top):

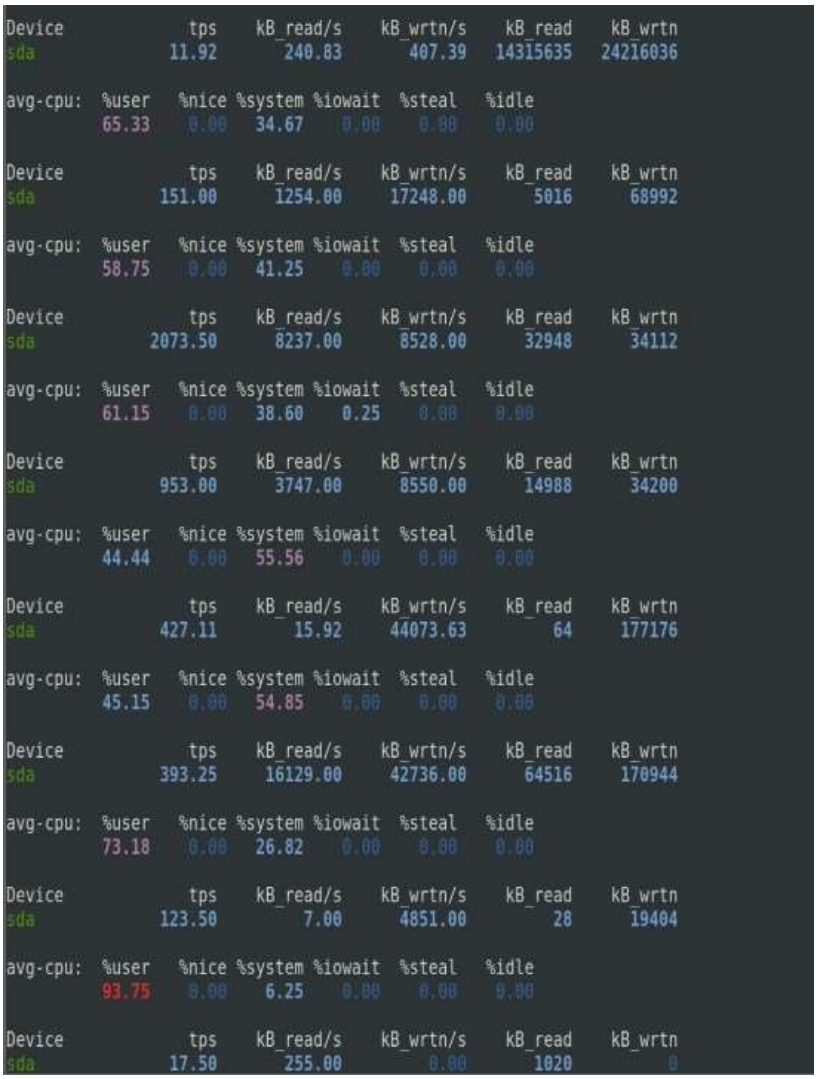
	VIRT	RES
До аллокации	2288	872
После аллокации	187968	872
После заполнения участка данными	220752	187232
После деаллокации	297348	24393

Потребление процессорного времени (top):

max %CPU = 94

min %CPU = 33

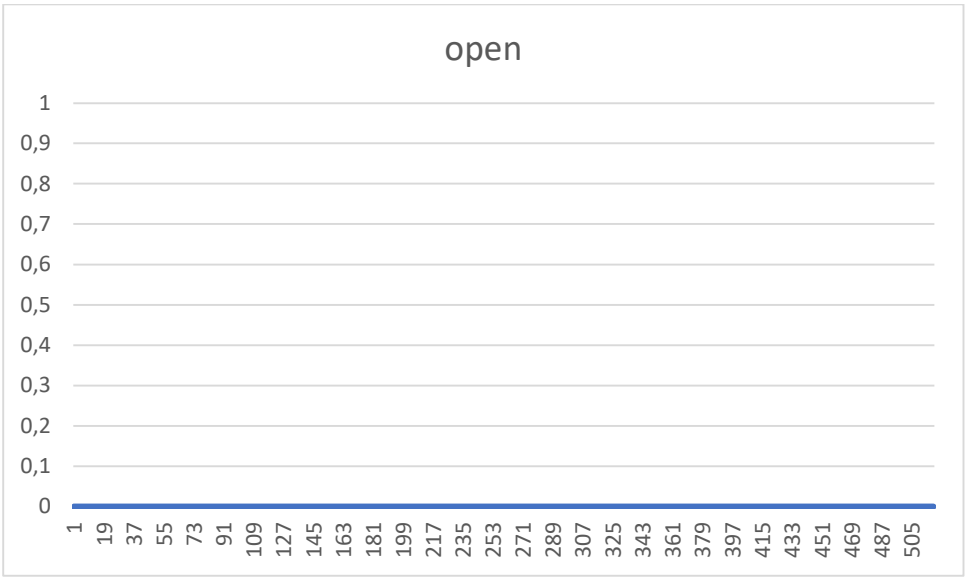
на чтение и запись (iostat):

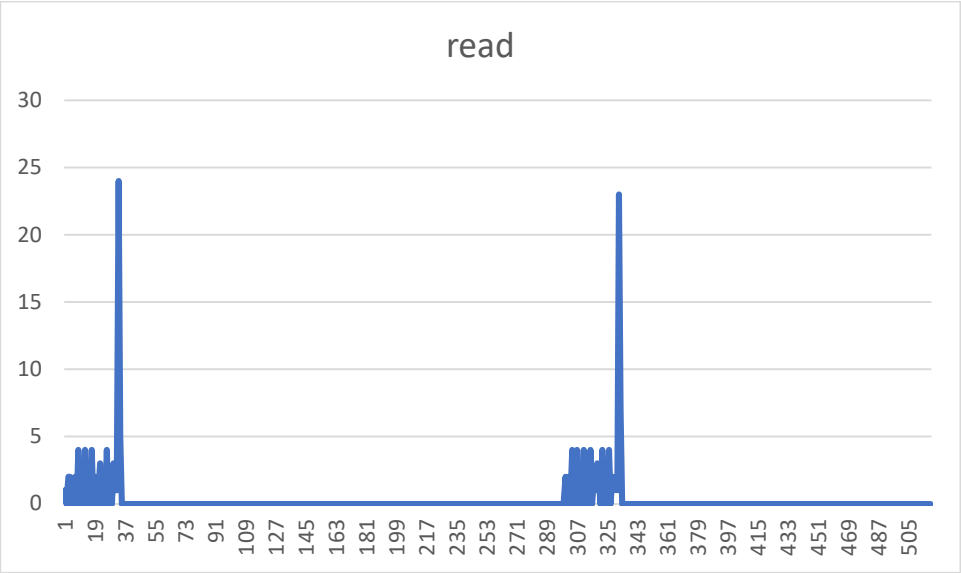


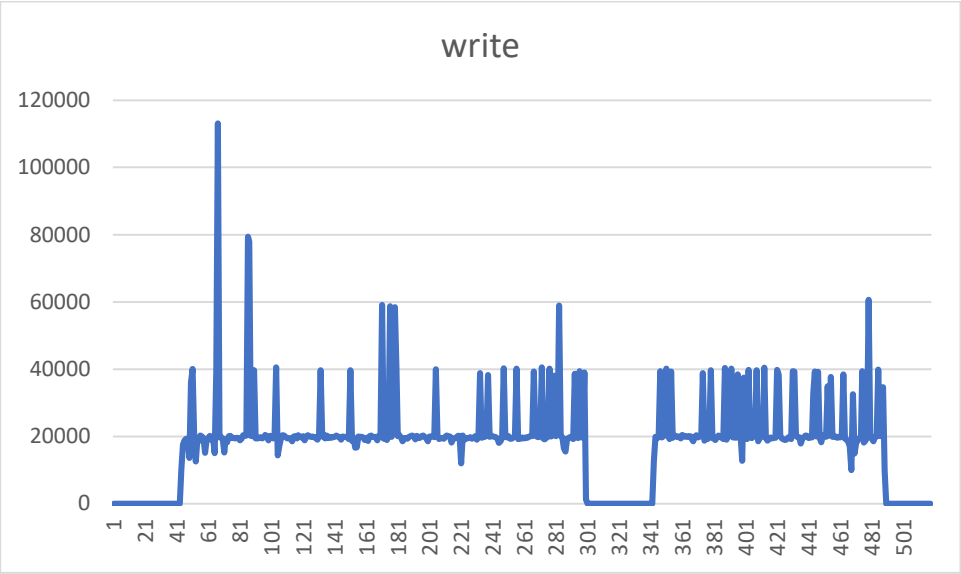
Трасса системных вызовов (strace):

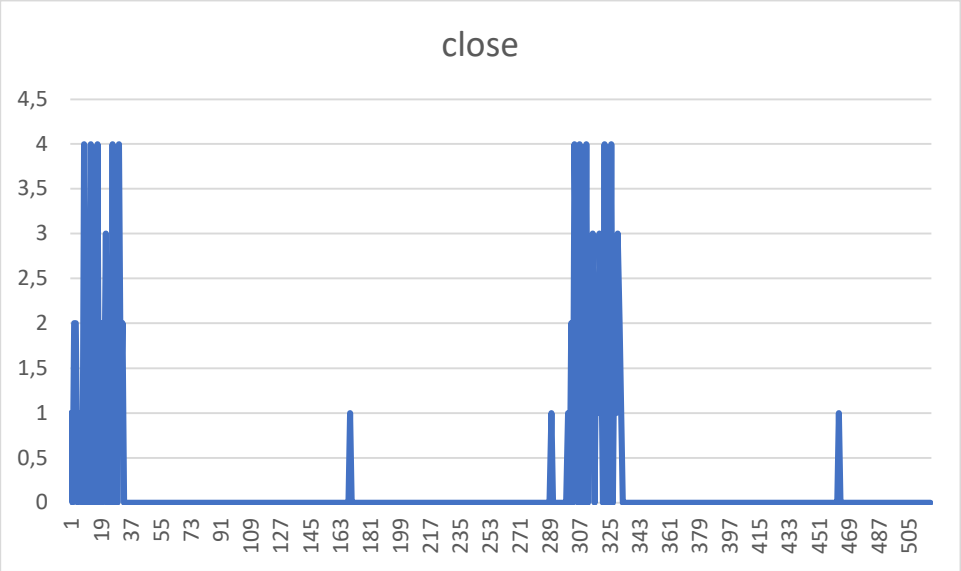
```
execve("./try1", ["./try1"], 0x7fff298eddc0 /* 44 vars */) = 0
brk(NULL)                               = 0x563bfff90000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=100288, ...}) = 0
mmap(NULL, 100288, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7efdbb82a000
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\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0@l\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=146968, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7efdbb828000
mmap(NULL, 132288, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7efdbb807000
mmap(0x7efdbb80d000, 61440, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7efdbb80d000
mmap(0x7efdbb81c000, 24576, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x15000) = 0x7efdbb81c000
mmap(0x7efdbb822000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x7efdbb822000
mmap(0x7efdbb824000, 13504, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7efdbb824000
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\0\3\0>\0\1\0\0\0\260A\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1824496, ...}) = 0
mmap(NULL, 1837056, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7efdbb646000
mprotect(0x7efdbb660000, 1658880, PROT_NONE) = 0
mmap(0x7efdbb660000, 1343488, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x22000) = 0x7efdbb660000
mmap(0x7efdbb7b0000, 311296, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x16a000) = 0x7efdbb7b0000
mmap(0x7efdbb7fd000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b6000) = 0x7efdbb7fd000
mmap(0x7efdbb883000, 14336, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7efdbb883000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7efdbb643000
arch_prctl(ARCH_SET_FS, 0x7efdbb643740) = 0
mprotect(0x7efdbb7fd000, 16384, PROT_READ) = 0
mprotect(0x7efdbb822000, 4096, PROT_READ) = 0
mprotect(0x563bff69f000, 4096, PROT_READ) = 0
mprotect(0x7efdbb86a000, 4096, PROT_READ) = 0
munmap(0x7efdbb82a000, 100288)          = 0
set_tid_address(0x7efdbb643a10)         = 46991
set_robust_list(0x7efdbb643a20, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7efdbb80d6b0, sa_mask=[], sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7efdbb819730}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7efdbb80d740, sa_mask=[], sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7efdbb819730}, 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
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...}) = 0
brk(NULL)                               = 0x563bfff90000
brk(0x563bfffb1000)                     = 0x563bfffb1000
write(1, "Start program\n", 14)         = 14
mmap(NULL, 190001152, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7efdb0110000
write(1, "Launch threads, which reads dev/"..., 39) = 39
write(1, "start reading random threads(): "..., 60) = 60
```

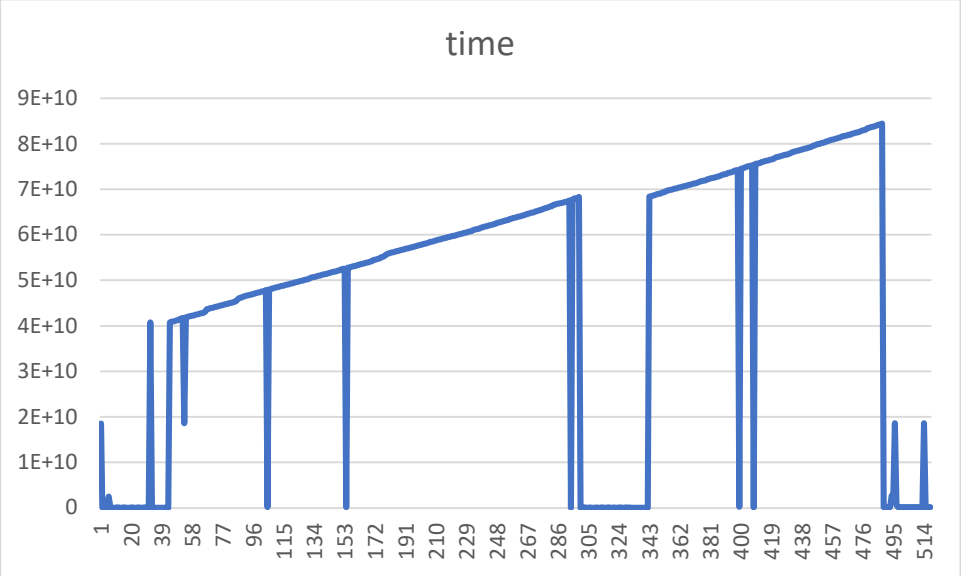
Графики системных характеристик:

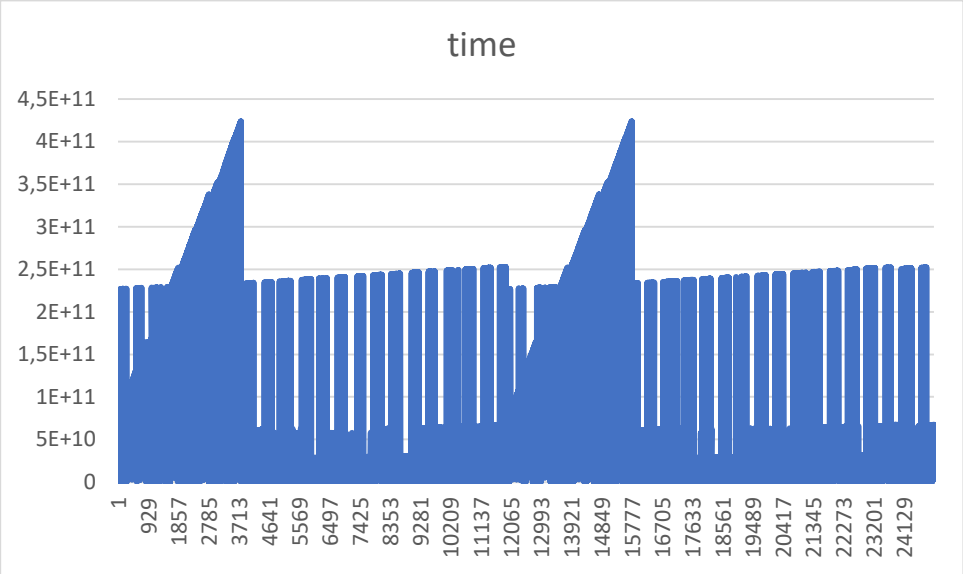



read
30
25
20
15
10
5
0
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307 325 343 361 379 397 415 433 451 469 487 505


write
120000
100000
80000
60000
40000
20000
0
1 21 41 61 81 101 121 141 161 181 201 221 241 261 281 301 321 341 361 381 401 421 441 461 481 501


close
4,5
4
3,5
3
2,5
2
1,5
1
0,5
0
1 19 37 55 73 91 109 127 145 163 181 199 217 235 253 271 289 307 325 343 361 379 397 415 433 451 469 487 505


time
9E+10
8E+10
7E+10
6E+10
5E+10
4E+10
3E+10
2E+10
1E+10
0
1
20
39
58
77
96
115
134
153
172
191
210
229
248
267
286
305
324
343
362
381
400
419
438
457
476
495
514


time
4,5E+11
4E+11
3,5E+11
3E+11
2,5E+11
2E+11
1,5E+11
1E+11
5E+10
0
1
929
1857
2785
3713
4641
5569
6497
7425
8353
9281
10209
11137
12065
12993
13921
14849
15777
16705
17633
18561
19489
20417
21345
22273
23201
24129

Вывод

В процессе работы мы изучили основы программирования на языке C, в том числе многопоточное программирование. Узнали про примитив синхронизации semaphore. Поставили виртуальную машину на windows и узнали основные команды в Linux. Изучили системные утилиты, которые позволяют измерять характеристики программ/процессов. Написали скрипт для systemtap, с помощью которого построили графики системных характеристик.