

Университет ИТМО

Лабораторная работа №1

По предмету «Операционные системы»

Вариант

A=119; B=0x122C7834; C=malloc; D=70; E=132; F=nocache; G=16;
H=seq; I=127; J=min; K=futex

Выполнил:

Студент группы Р33111

Мосягин Иван Денисович

Преподаватели:

Осипов С. Б.

Санкт-Петербург

2020

Задание

Разработать программу на языке C, которая осуществляет следующие действия

- Создает область памяти размером A (**119**) мегабайт, начинающихся с адреса B (**0x122C7834**) (если возможно) при помощи C=(**malloc**, **mmap**) заполненную случайными числами /dev/urandom в D (**70**) потоков. 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 E (**132**) мегабайт с использованием F=(блочного, **некешируемого**) обращения к диску. Размер блока ввода-вывода G (**16**) байт. Преподаватель выдает в качестве задания последовательность записи/чтения блоков N=(**последовательный**, заданный или случайный)
- Генерацию данных и запись осуществлять в бесконечном цикле.
- В отдельных I (**127**) 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.

Программа

Гитхаб: <https://github.com/MIDmos/os-lab>

Пример работы

```
pid - 13053
Before the allocation 121856 Kib of memory at the address 0x122c7834. Pause
Data is allocated at the address 0x7f013f66d010. Pause
Write data to the memory. Wait
The memory is filled. Reading...
New min value! (2147483647 -> -2145446593)
Press enter to start deallocation
After deallocation. Pause
Start a loop? (1 or not)
n
```

Данные для анализа

```
(base) До аллокации $ ps -eo pid,vsz,rss,comm | grep 12725
12725 6696 836 main
(base) После аллокации $ ps -eo pid,vsz,rss,comm | grep 12725
12725 128556 836 main
(base) Заполнение данными $ ps -eo pid,vsz,rss,comm | grep 12725
12725 423484 122080 main
(base) После деаллокации $ ps -eo pid,vsz,rss,comm | grep 12725
12725 242356 2958 main
```

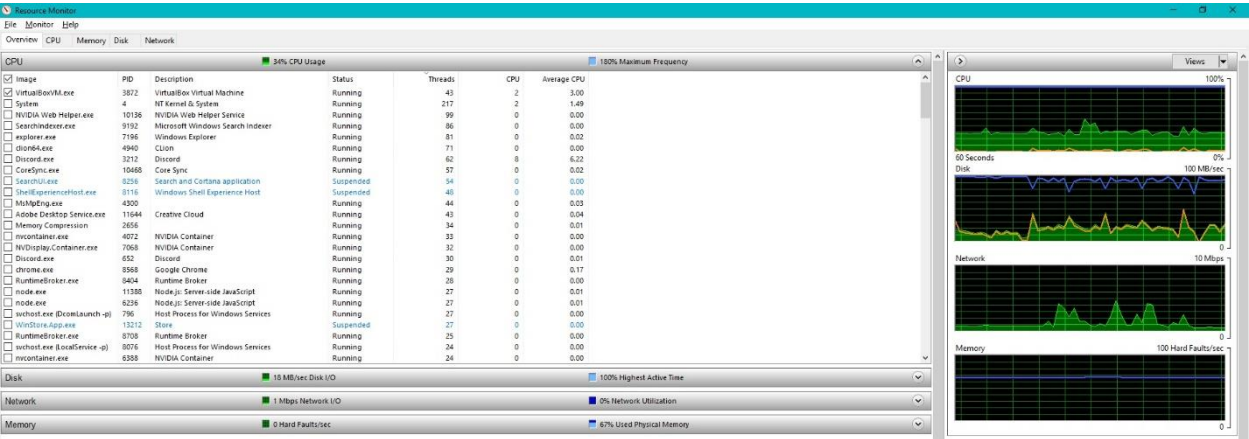
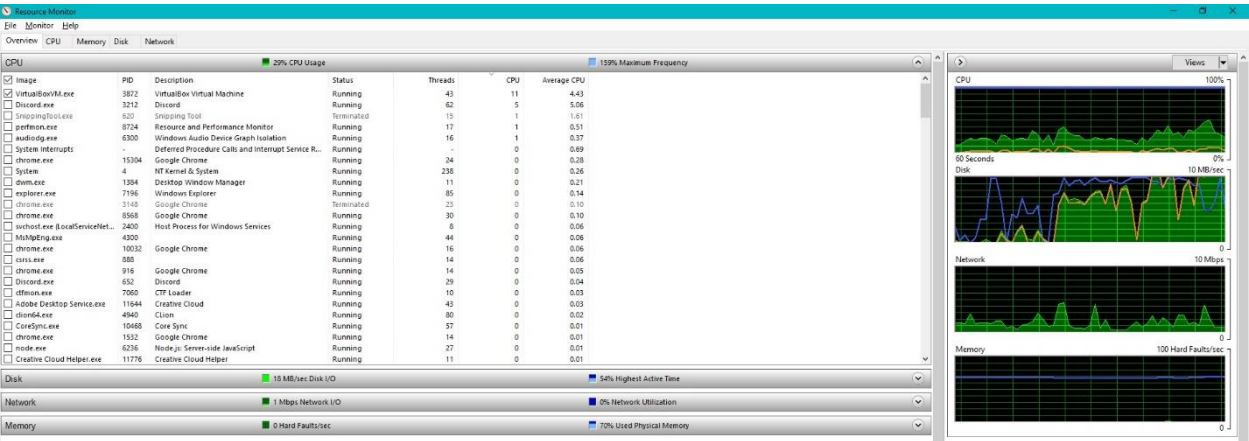
sudo statisc -ps -f [PID]

% time	seconds	usecs/call	calls	errors	syscall
91.38	236.686958	3881	60990	9	futex
7.51	19.441945	3	6455545		pread64
0.70	1.812552	12855	141		read
0.19	0.485247	16	30464		pwrite64
0.12	0.319571	10	30657		openat
0.05	0.140045	5	30534		stat
0.05	0.118796	4	30536		close
0.00	0.001643	14	120		clone
0.00	0.001409	8	182		mmap
0.00	0.001269	7	180		mprotect
0.00	0.001015	10	101		munmap
0.00	0.000901	8	120		set_robust_list
0.00	0.000160	2	70		madvise
0.00	0.000021	7	3		write
0.00	0.000020	20	1	1	ioctl
0.00	0.000006	6	1		fstat
100.00	259.011558		6639645	10	total

sudo strace -fp 13899

```
(base) arthur@arthur-System-Product-Name:~/itmo/lab1$ sudo strace -fp 13899
strace: Process 13899 attached
read(0, "\n", 1024) = 1
mmap(NULL, 124784640, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f58845f2000
write(1, "Data is allocated at the address...", 54) = 54
read(0, "\n", 1024) = 1
close(3) = 0
munmap(0x7f72023ad010, 124780544) = -1 EINVAL (Invalid argument)
write(1, "After deallocation. Pause", 25) = 25
read(0, "\n", 1024) = 1
write(1, "Start a loop? (1 or not)\n", 25) = 25
read(0, "b\n", 1024) = 2
lseek(0, -2, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++
```

Resource Monitor



Вывод

В процессе выполнения лабораторной работы я получил базовые навыки программирования на C, научился вызывать системные функции Linux в своих программах, а также наблюдать за нагрузкой на компьютер средствами операционной системы.