



УНИВЕРСИТЕТ ИТМО

Факультет ПИиКТ

Дисциплина: операционные системы

Лабораторная работа №1

Выполнил: Иванов Илья

Преподаватель: Покид А.В.

Группа: Р33122

Санкт-Петербург, 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, кешируемого) 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=(mutex, 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.

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.

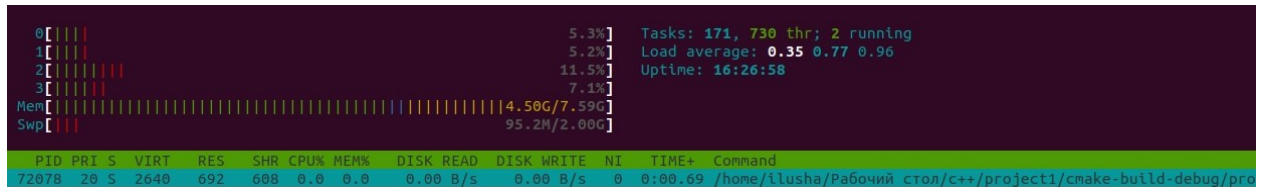
Вариант

A=213;B=0xD9B51893;C=mmap;D=62;E=177;F=block;G=21;H=seq;I=40;J=max;K=cv

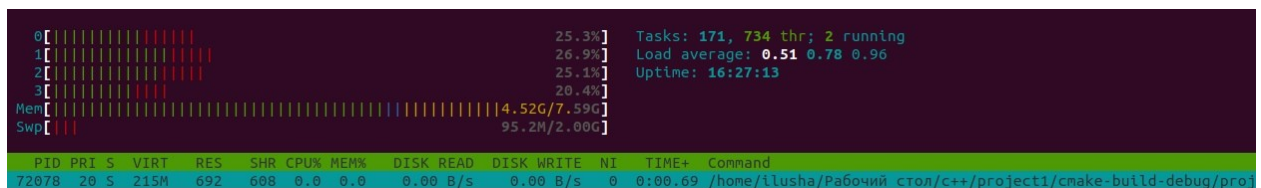
Выполнение

1) Замеры виртуальной/физической памяти:

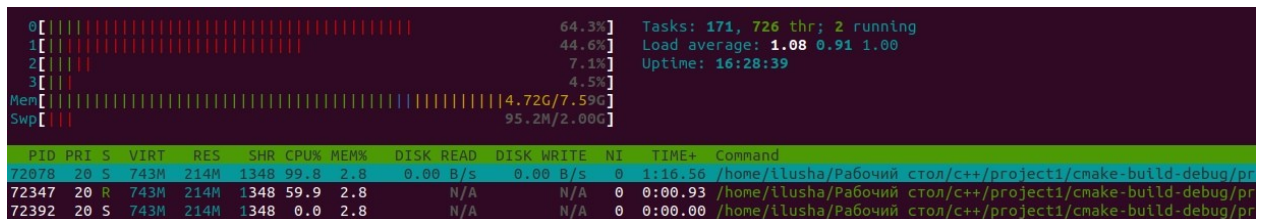
До аллокации:



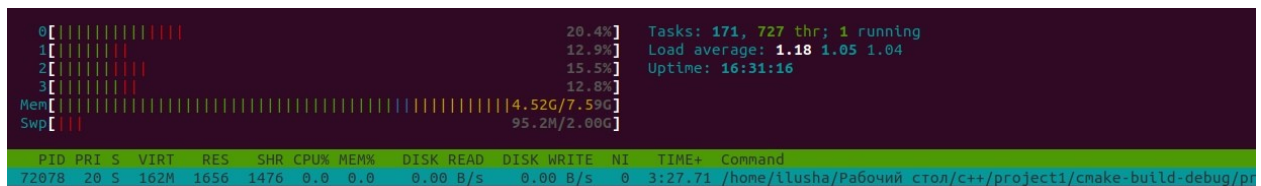
После аллокации:



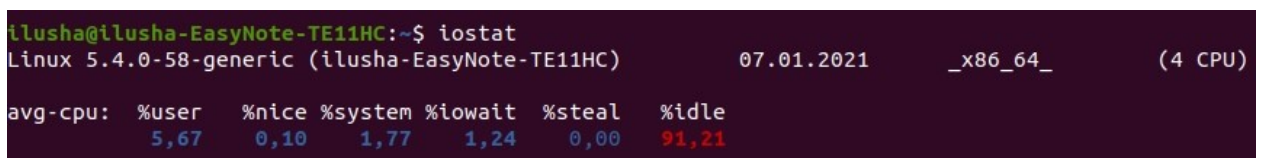
После заполнения участка данными (под конец):



После деаллокации:



2) Ввод-вывод:



3) Трасса системных вызовов

```
[pid 5316] read(0, "Content-Length: 79\r\n\r\n{\"jsonrpc\":\"...\", 4096) = 101
[pid 5316] clock_gettime(CLOCK_MONOTONIC, {tv_sec=41693, tv_nsec=665369278}) = 0
[pid 5316] clock_gettime(CLOCK_REALTIME, {tv_sec=1610353466, tv_nsec=439832624}) = 0
[pid 5316] write(2, "I[11:24:26.439] <-- clionDebugDu"... , 53) = 53
[pid 5316] stat("/tmp/preamble-8055b4.pch", {st_mode=S_IFREG|0664, st_size=850920, ...}) = 0
[pid 5316] stat("/tmp/preamble-c7fa4e.pch", {st_mode=S_IFREG|0664, st_size=932304, ...}) = 0
[pid 5316] openat(AT_FDCWD, "/proc/self/statm", O_RDONLY) = 3
[pid 5316] fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
[pid 5316] read(3, "88817 11609 5698 7857 0 21249 0\n", 1024) = 32
[pid 5316] close(3) = 0
```

4) Вывод stat

open	read	write	current_thread
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
1	2	3	0
1	0	0	0
1	0	0	0
1	0	0	0
1	0	0	0
2	2	3	0
2	0	0	0
2	0	0	0
2	0	0	0
2	0	0	0
3	2	3	0
3	0	0	0
3	0	0	0
3	0	0	0
3	0	0	0
4	2	3	0
4	0	0	0
4	0	0	0
4	0	0	0
4	0	0	0
5	2	3	2046
5	0	0	0
5	0	0	0
232	9	42	0
232	0	0	4716
233	2	3	0
233	0	0	0
233	0	0	0
233	0	0	0
233	0	0	0

234	0	0	4412
234	0	0	0
234	0	0	0
235	2	3	4434
235	0	0	0
235	0	0	0
235	0	0	0
235	0	0	0
236	2	3	0
236	0	0	0
236	0	0	0
236	0	0	0
236	0	0	0
237	2	3	0
237	0	0	0
237	0	0	0
237	0	0	0
237	0	0	0
238	2	3	0

Вывод: В процессе выполнения данной лабораторной работы я рассмотрел системные характеристики и понял как они меняются во время выполнения программы. В процессе использовал такие утилиты как `stap`, `strace`, `atop`, `htop`.