ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет безопасности информационных технологий
Дисциплина:

«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

«Membomb»

Выполнил:

Чу Ван Доан N3247

(подпись)

Проверил:

Савков Сергей Витальевич

(подпись)

Санкт-Петербург

Задание

1. Написать программу выделения памяти и заполнения ее нулями с шагом, равным размеру страницы памяти (mmap, VirtualAlloc)
2. Составить график свободной памяти
3. Ознакомиться с работой демона OOM Killer в Linux
4. Достичь сообщения о невозможности выделить память в Windows

1. Написать программу выделения памяти и заполнения ее нулями с шагом, равным размеру страницы памяти (mmap, VirtualAlloc).

a. Linux (Ubuntu 22.04 LTS)

1. Написать программу выделения памяти и заполнения ее нулями с шагом, равным размеру страницы памяти (mmap, VirtualAlloc)

Программа:

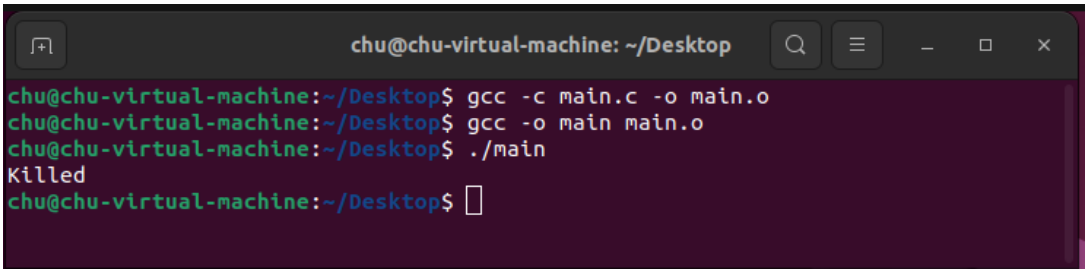
```
#include <stdlib.h>
#include <sys/mman.h>
#include <string.h>
#include <unistd.h>

int main(){
unsigned int size = 50*1024*1024;
    while(1){
        unsigned char *p = mmap(NULL, size, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, 0,0);
        for(int i = 0; i < size; i += 4096)
                p[i] = 0;
        system("free -m >> membomb.txt");
                }
return 0;
}
```

2. Составить график свободной памяти

- До запуска программы:

- После запуска программы:

- OOM



b. Windows

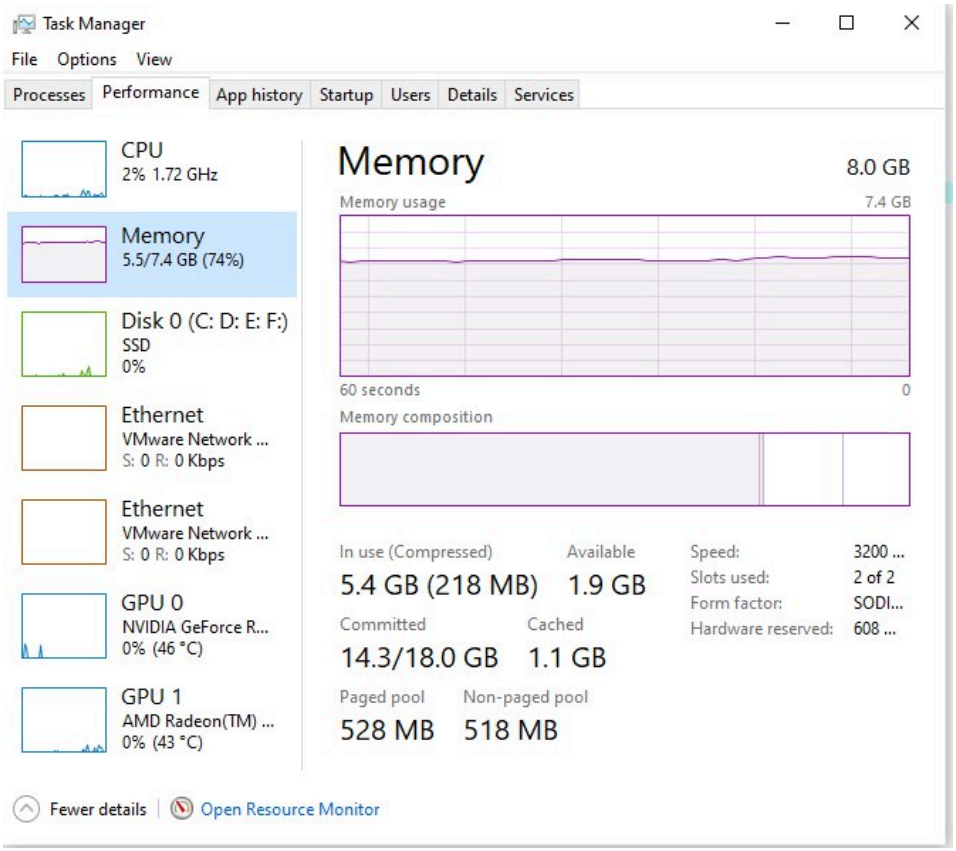
1. Написать программу выделения памяти и заполнения ее нулями с шагом, равным размеру страницы памяти (mmap, VirtualAlloc)

Программа:

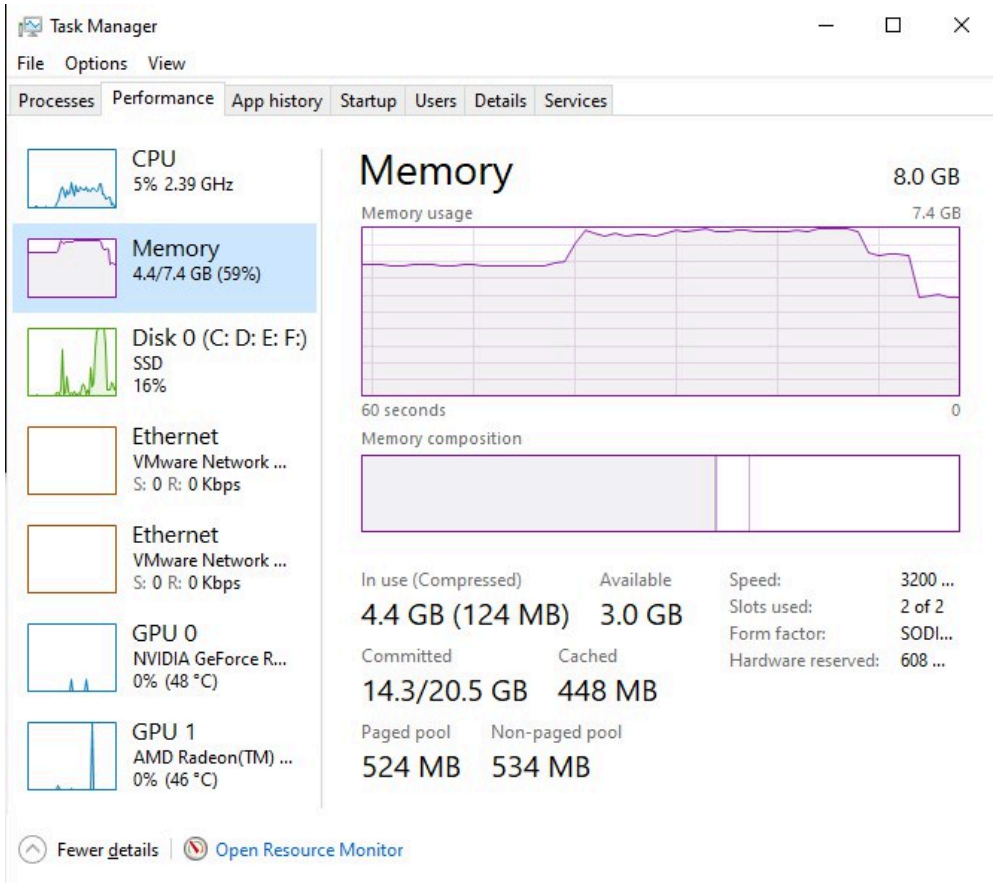
```
#include <stdlib.h>
#include <windows.h>
#include <cstring>
#include <unistd.h>
using namespace std;
int main() {
    SYSTEM_INFO si;
    GetSystemInfo(&si);
    int size = si.dwPageSize;

    void* ptr;
    while (1){
        LPVOID ptr = VirtualAlloc(NULL, size, MEM_RESERVE, PAGE_READWRITE);
        ptr = VirtualAlloc(ptr, size, MEM_COMMIT, PAGE_READWRITE);
        memset(ptr, '0', size);
    }
    return 0;
}
```

2. Составить график свободной памяти

- До запуска программы:



- **После запуска программы:**



2. Ознакомиться с работой демона OOM Killer в Linux

Когда у сервера или процесса заканчивается память, Linux предлагает 2 пути решения: обрушить всю систему или завершить процесс (приложение), который съедает память. Лучше, конечно, завершить процесс и спасти ОС от аварийного завершения. В двух словах, Out-Of-Memory Killer (OOM Killer)— это процесс, который завершает приложение, чтобы спасти ядро от сбоя. Он жертвует приложением, чтобы сохранить работу ОС.

OOM Killer - это компонент ядра Linux, призванный решать проблему недостатка памяти. Известно, что виртуальной памяти может быть бесконечно много (в пределах адресации), а вот физической - вполне конечное число. Ядро выделяет память процессам "с запасом" в сумме превышающую физическую память системы. В основном, всё разруливается нормально (вся выделенная память одновременно редко требуется), но бывает ситуация когда становится нужно памяти больше, чем ее физически есть. И системе тогда нужно завершить какой-то процесс, чтобы продолжить работу. Вот этим и занимается OOM Killer.

Когда заканчивается память, вызывается функция out_of_memory(). В ней есть функция select_bad_process(), которая получает оценку от функции badness(). Под раздачу попадет самый «плохой» процесс. Функция badness() выбирает процесс по определенным правилам.

1. Ядру нужен какой-то минимум памяти для себя.
2. Нужно освободить много памяти.
3. Не нужно завершать процессы, которые используют мало памяти.
4. Нужно завершить минимум процессов.
5. Сложные алгоритмы, которые повышают шансы на завершение для тех процессов, которые пользователь сам хочет завершить.

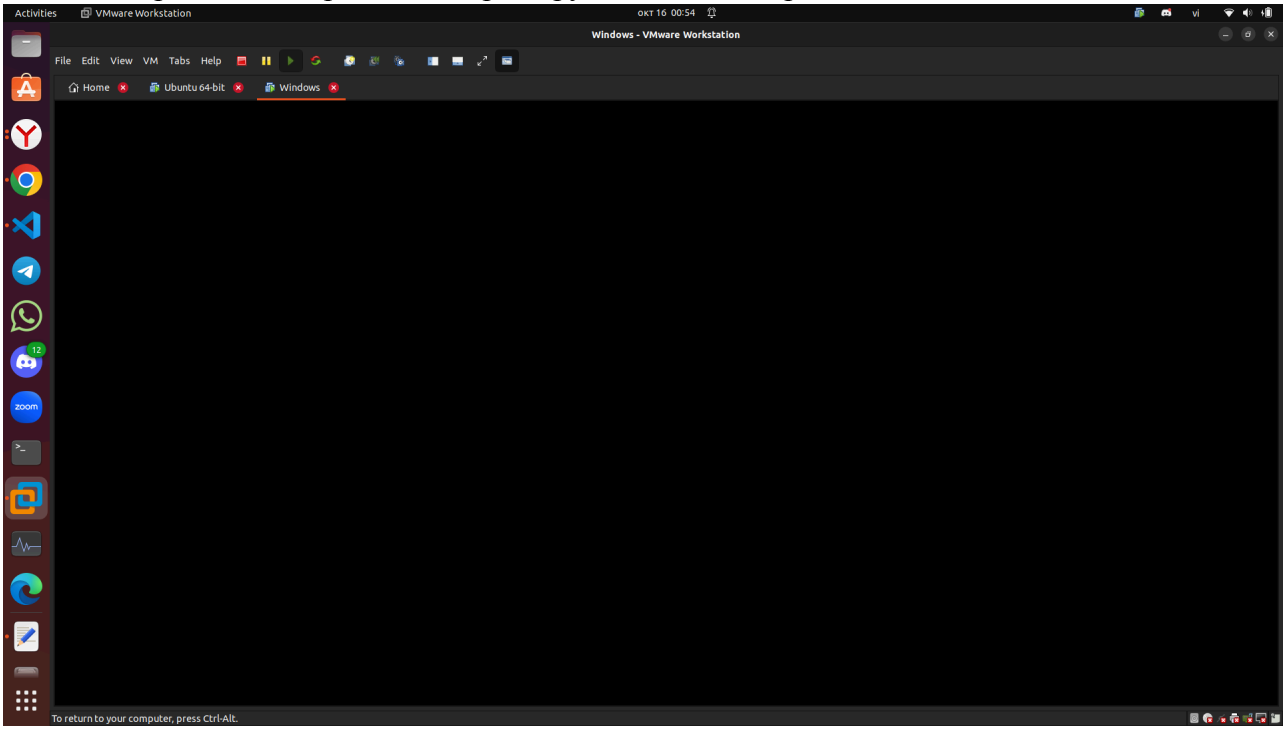
Выполнив все эти проверки, OOM изучает оценку (oom_score). OOM назначает oom_score каждому процессу, а потом умножает это значение на объем памяти. У процессов

с большими значениями больше шансов стать жертвами OOM Killer. Процессы, связанные с привилегированным пользователем, имеют более низкую оценку и меньше шансов на принудительное завершение.

Всякий раз, когда OOM Killer вызывается для уничтожения процесса, он записывает информацию в системный журнал, включая информацию о том, какой процесс был убит и почему. Проверяем следующее: **dmesg | egrep -i “killed process”**

IV. Достичь сообщения о невозможности выделить память в Windows

- После первого запуска программы membomb в Windows, как и на графике, память выделяется и используется очень быстро, и программа израсходовала память, компьютер начинает зависать, и операционная система немедленно реагирует, закрывая программу и возвращая память. И после этого компьютер продолжает нормально работать.
- После перезагрузки компьютера и повторного запуска программы membomb память выделялась и использовалась очень быстро, и программа израсходовала всю память, компьютер начал зависать и сразу экран становился черным . Я ничего не мог сделать с компьютером, и мне пришлось перезагрузить компьютер.



Вывод: При выделении памяти в windows с помощью функции VirtualAlloc(). Свободная память резко упадет. При выделении памяти в Linux (функция mmap())свободная память будет медленно уменьшаться, потому что в ядре Linux есть OOM Killer, поэтому, когда памяти выделяется слишком много, вызывая переполнение физической памяти системы, OOM отключит этот процесс.