МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа №5

По дисциплине «Операционные системы»

Управление памятью в ОС Linux

Выполнил студент группы M3205 Решетников Роман Олегович

Проверил *Ефимов Тимур Андреевич*

*САНКТ-ПЕТЕРБУРГ*

*2020*

**Отчет по Лабораторной работе №5**

*Конфигурация ОС:*

 Общий объем оперативной памяти – 512 МБ

 Объем раздела подкачки – 820 МБ

 Размер страницы виртуальной памяти – 4 КБ

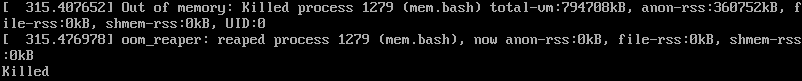
 Объем свободной физической памяти в ненагруженной системе – 120 МБ

 Объем свободного пространства в разделе подкачки в ненагруженной системе – 820 МБ

Эксперимент №1

Первый этап.

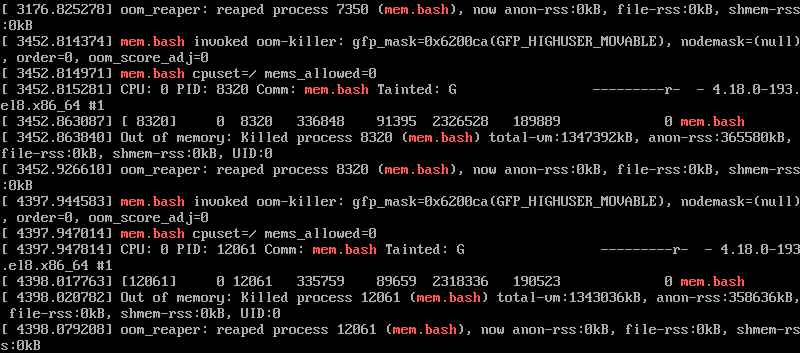
1) Создадим скрипт mem.sh, добавляющий в конец массива 100 элементов и выводящий каждый 10000-й шаг размер массива в report.log

Зафиксируем последнюю строку файла report.log – 8880000; 

2)

Создадим скрипт handler.sh, который запускает скрипт и считывает показания команды top каждую секунду.

Зафиксируем последнюю строку файла report1.log – 8850000

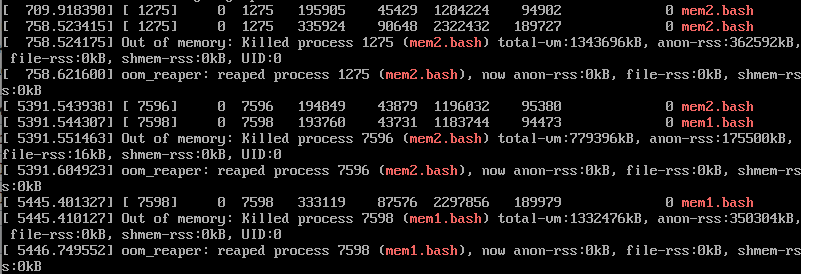
Зафиксируем последнюю строку системного отчета:

Построим график изменения размера оперативной памяти и размера файла подкачки от времени:

Второй этап.

Запустим handler1.sh и подождем, пока система не убьет процессы.

Зафиксируем последние строки файлов report1.log - 4470000 и report2.log - 8910000

Зафиксируем строки системного отчета о завершении процессов: 

Построим график размера оперативной памяти и размера файла подкачки от времени для первого mem1.bash

и второго mem2.bash

*Наблюдения:*

И в первом и во втором случаях размер массива ограничивается примерно 8900000 элементами. Когда количество используемой оперативной памяти достигает 400 Мб, то данные начинают записываться в файл подкачки.

Скачок графика на 51 секунде можно объяснить тем, что в это время завершился процесс mem1.sh, освободив память. Также можно заметить, что во время остановки mem1.sh оперативной памяти освободилась треть, а память файла подкачки освободилось наполовину. Это означает, что элементы из mem1.sh чаще записывались в оперативную память.

Эксперимент №2

Макс N = 8880000 (из эксперимента 1.1)

При запуске N = 888000 и K = 10 все процессы завершились успешно без аварийной остановки.

При запуске же N = 1000000 и K = 30, часть процессов завершилась аварийно. Процессор стал тратить больше времени на межстраничный обмен, чем на непосредственное выполнение программ.

Дальнейшие запуски для K = 30. Найденное: - – есть авария, + – без аварий:

N = 256000 - +

N = 800000 – -

N = 700000 – -

N = 600000 – +

N = 690000 - +

Итог N’ = 690000

Наше N получилось не в 30 раз меньше потому, что процессы успевают отработать и освободить память.

*Вывод: мы убедились на практике, что сначала заполняется оперативная память, при заполнении оперативной памяти начинает заполняться файл подкачки, а после заполнения файла подкачки, процесс аварийно завершается.*