**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе «Управление памятью в ОС Linux»

по дисциплине «**Операционные системы**»

Автор: Чесноков Александр Сергеевич

Факультет: Информационных технологий   
и программирования

Группа: М3203

Преподаватель: Титова Анастасия Витальевна



Санкт-Петербург 2020**Задачи работы:**

Проведите два виртуальных эксперимента в соответствии с требованиями и проанализируйте их результаты. В указаниях ниже описано, какие данные необходимо фиксировать в процессе проведения экспериментов. Рекомендуется написать «следящие» скрипты и собирать данные, например, из вывода утилиты top автоматически с заданной периодичностью, например, 1 раз в секунду. Можно проводить эксперименты и фиксировать требуемые параметры и в ручном режиме, но в этом случае рекомендуется замедлить эксперимент, например, уменьшив размер добавляемой к массиву последовательности с 10 до 5 элементов.

**Ход работы:**

Общий объем оперативной памяти MemTotal: 1870900 kB

Объем раздела подкачки SwapTotal: 839676 kB;

Размер страницы виртуальной памяти PAGE\_SIZE: 4 kB

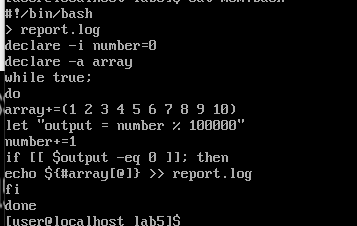
Объем свободной физической памяти в ненагруженной системе MemFree: 1470132 kB

Объем свободного пространства в разделе подкачки в ненагруженной системе SwapFree: 839676 kB

**Эксперимент №1**

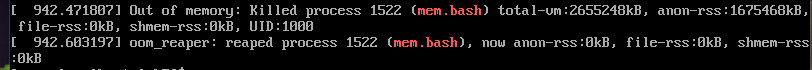
*Подготовительный этап:*

Скрипт mem.bash:



*Первый этап:*

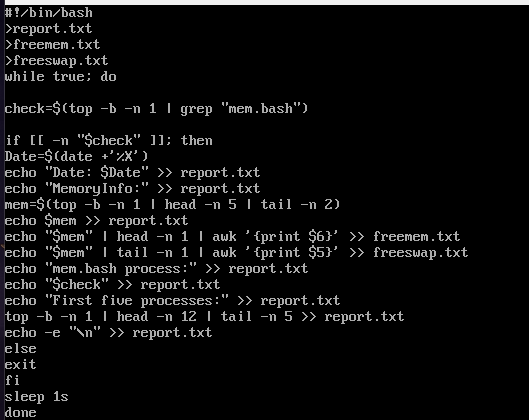
Записи системного журнала:

**

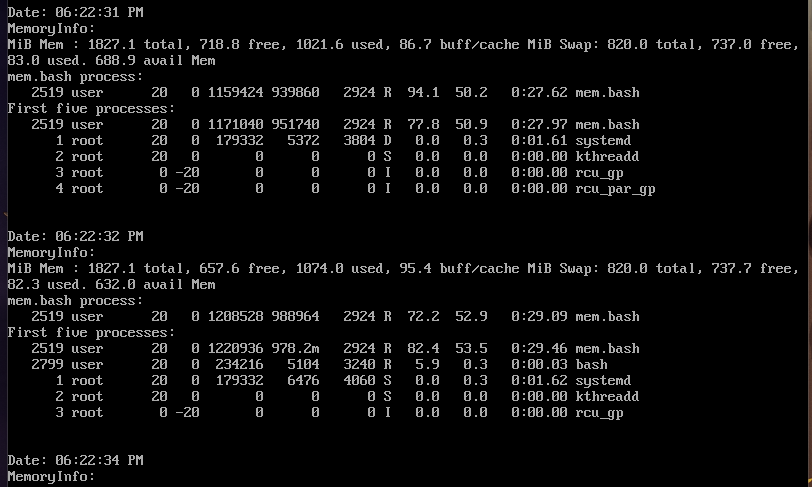
Последняя строка report.log:

**

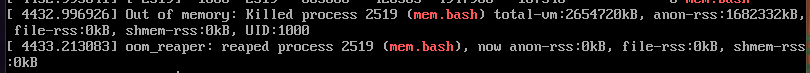
Скрипт-шпион за памятью и процессами:

**

Фрагмент отчета работы шпионa report.txt:

**

Записи системного журнала:

**

Последняя строка report.log:

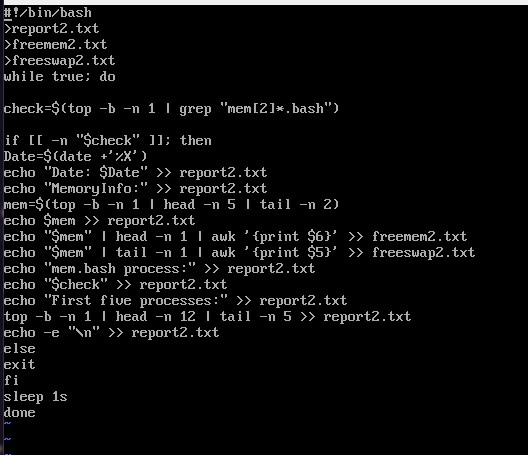
**

*Второй этап:*

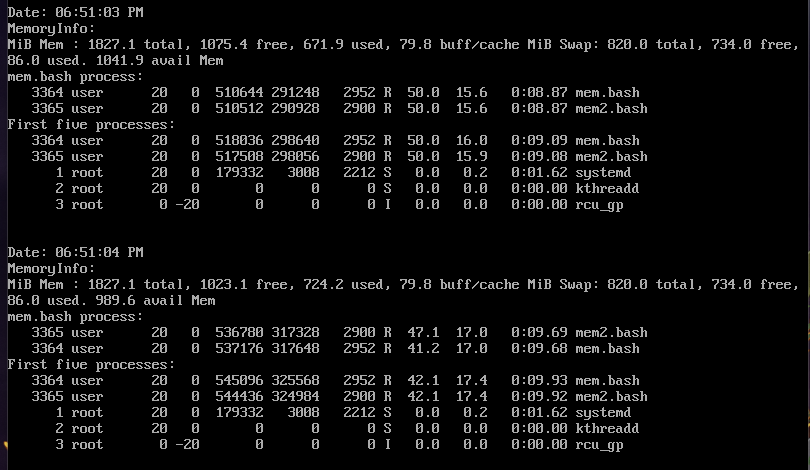
Скрипт mems.sh



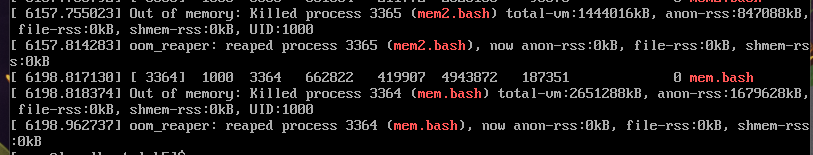
Скрипт-шпион за памятью и процессами:



Результат работы шпиона: фрагмент report2.txt



Записи системного журнала:



Последняя строка report.log:



Последняя строка report2.log:

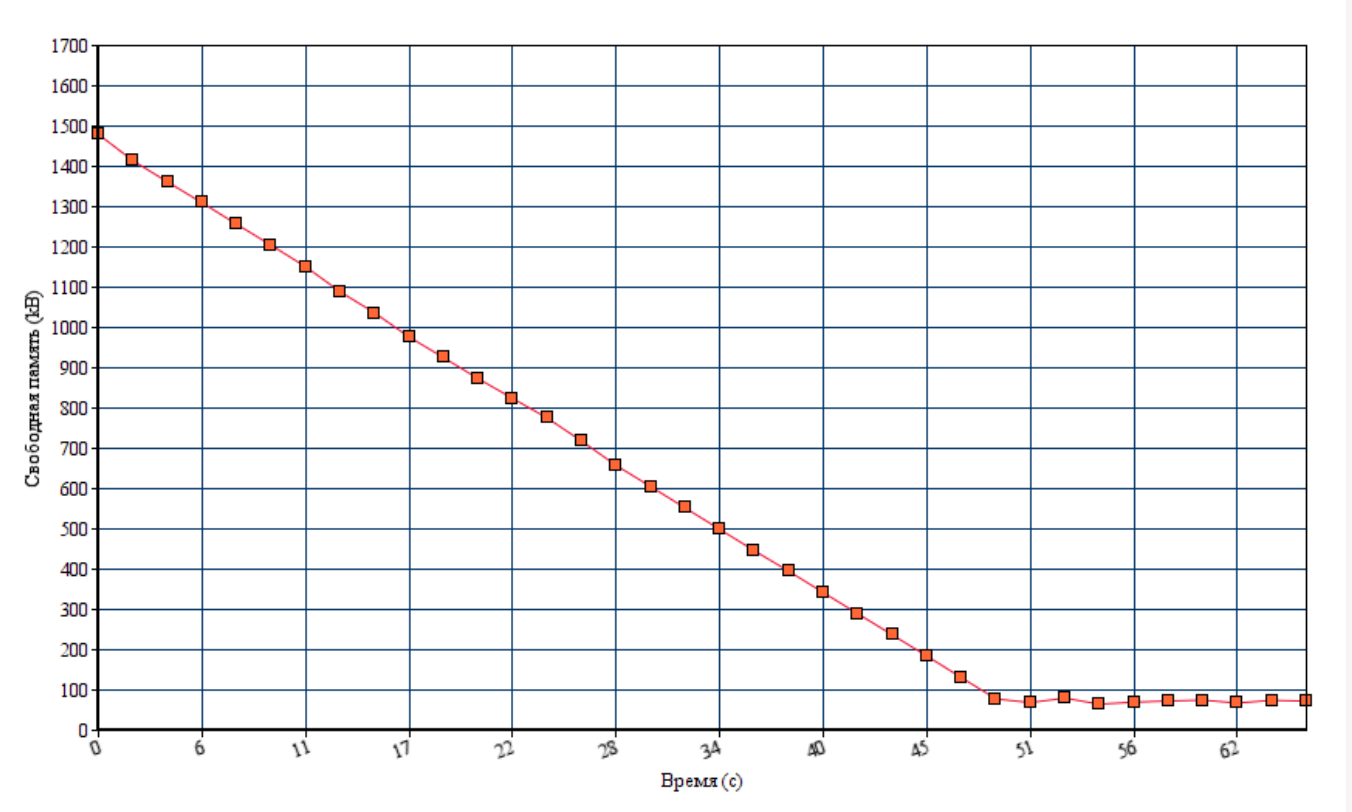


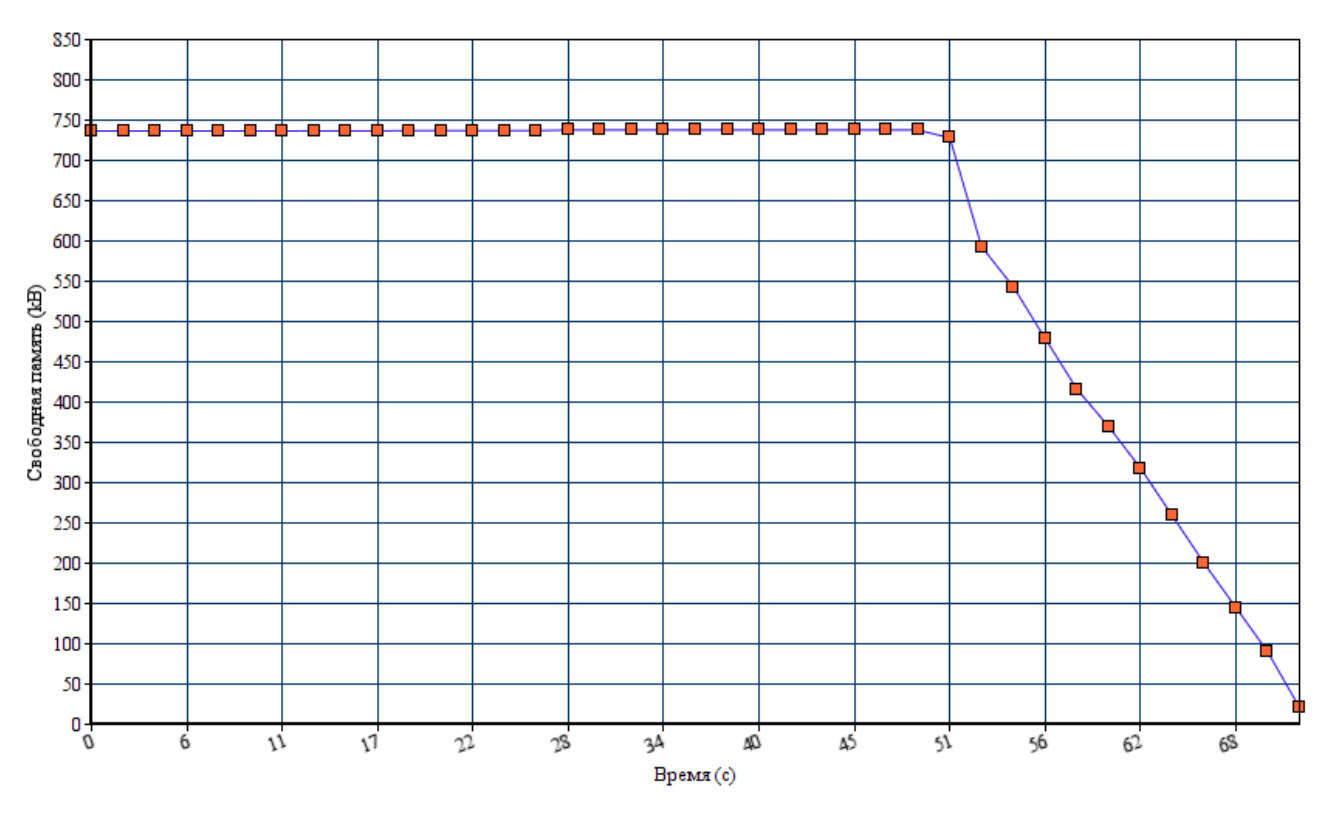
*Обработка:*

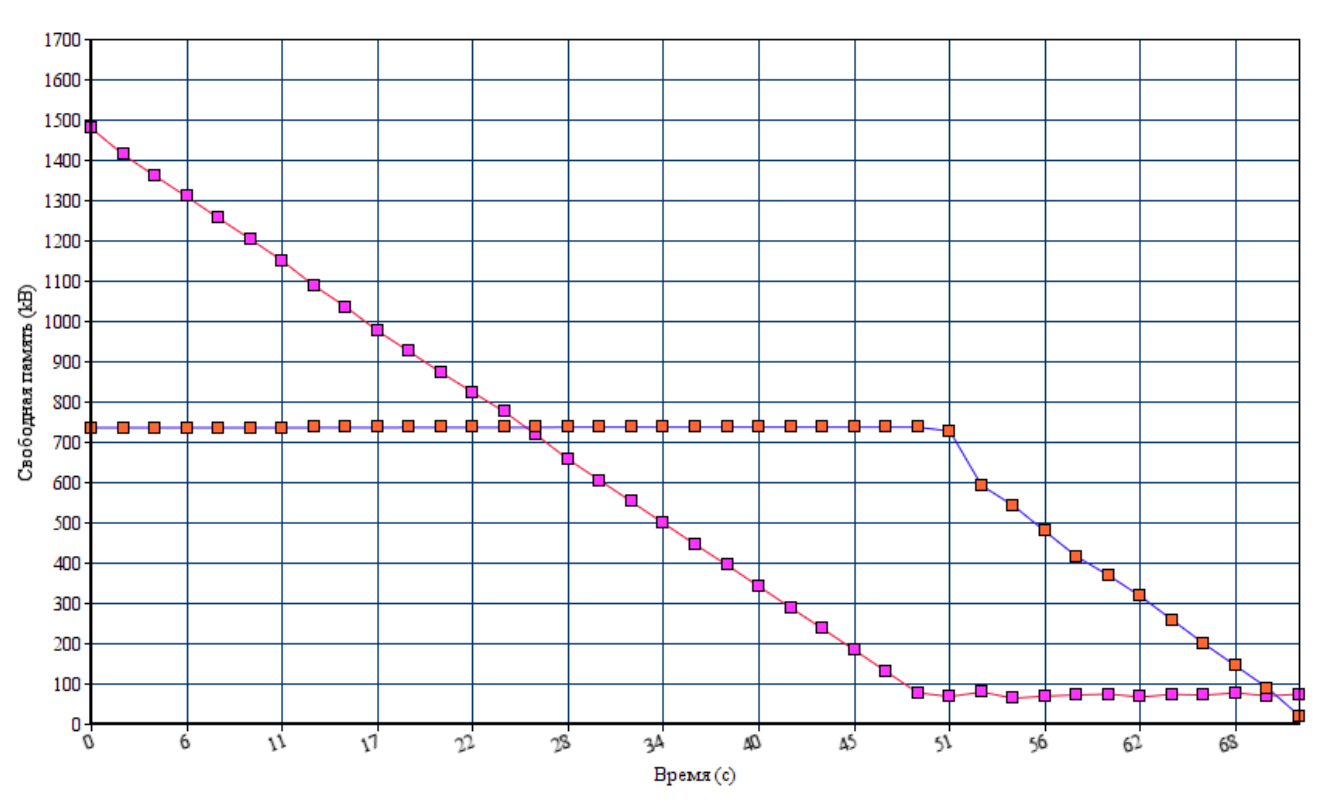
*Первый этап:*

*Красный – Memfree*

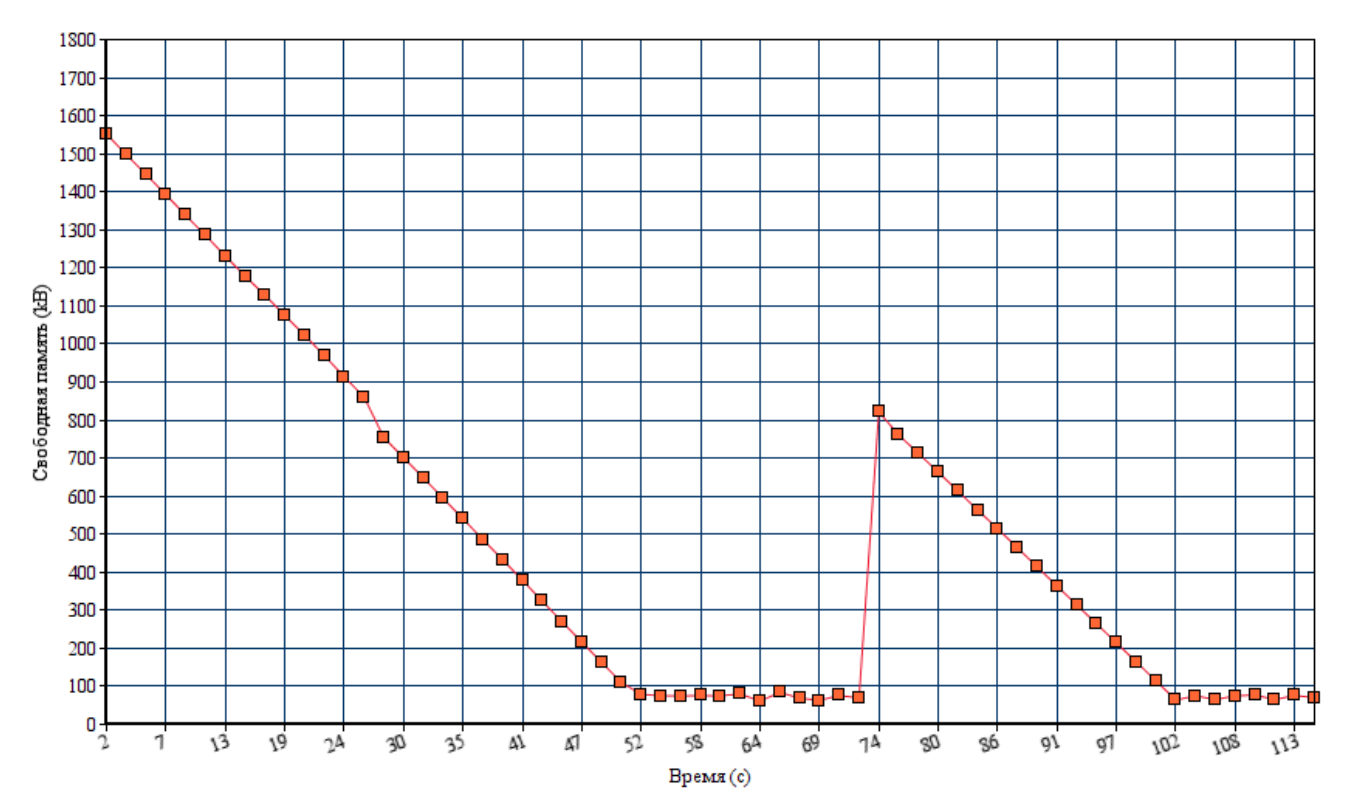
*Синий - Swapfree*

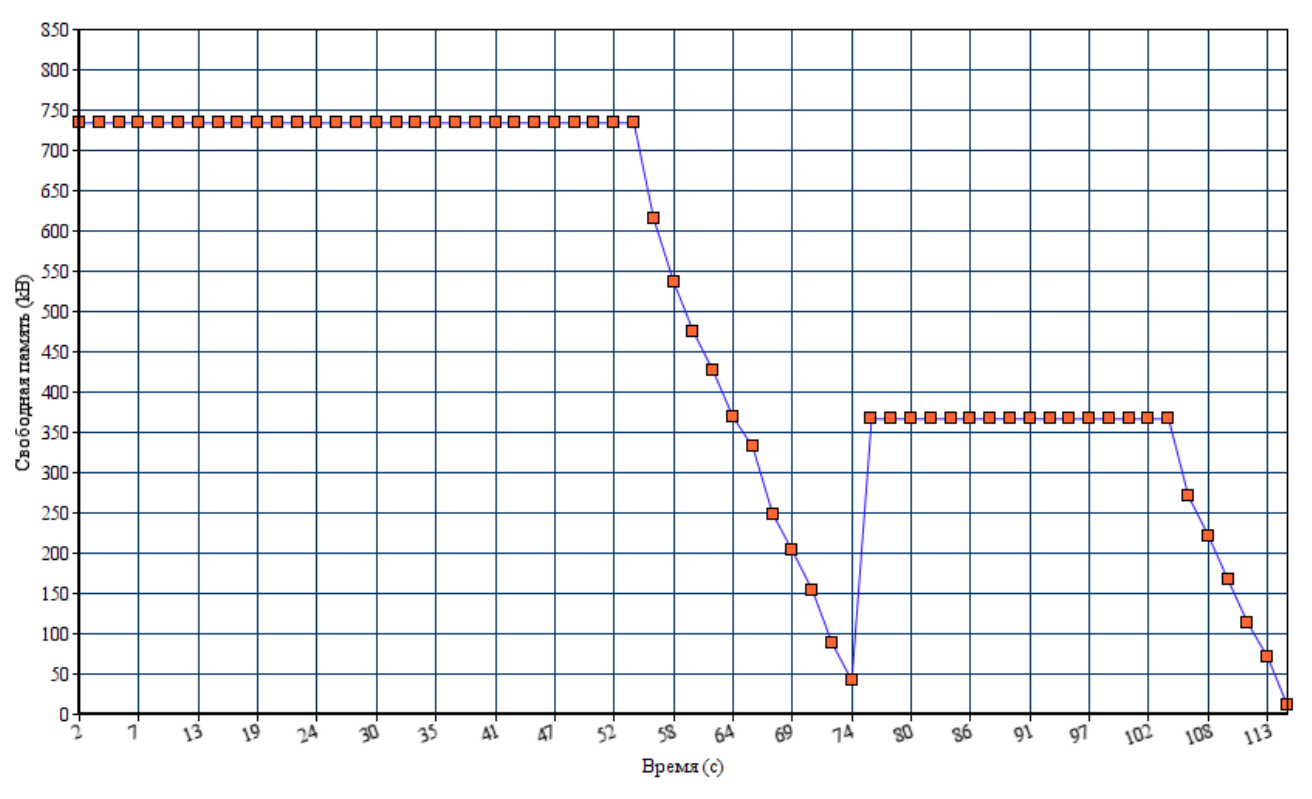
**

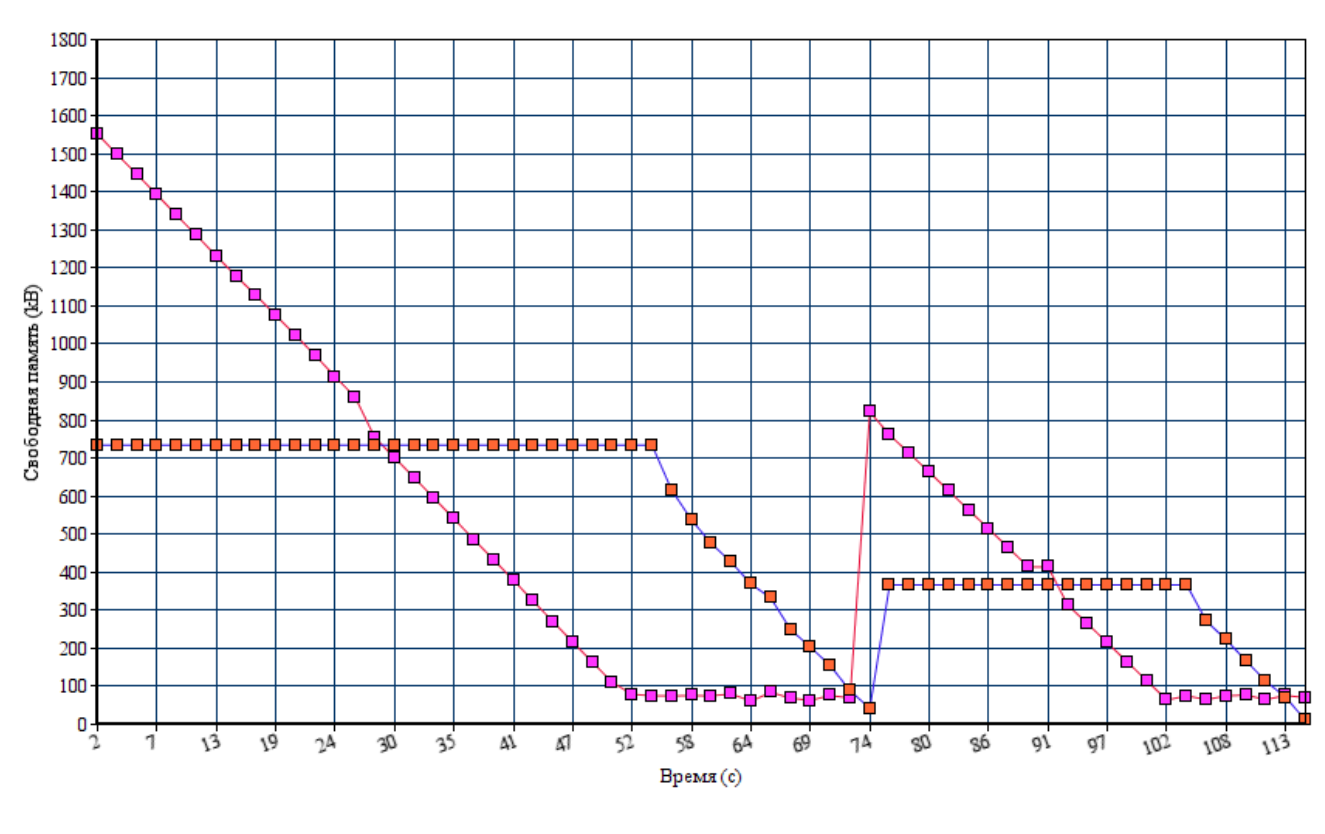
**

**

*Второй этап:*

**

**

**

**Вывод:**

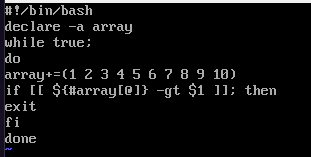
В начале каждого этапа используется лишь физическая память Mem, как только она рискует закончиться начинает использоваться память подкачки Swap (в 1 этапе на 51 секунде, во втором на 55 и 105). Как только память подкачки заканчивается процесс аварийно завершается.

Во втором этапе память для процессов делится примерно поровну, как только память подкачки закончилась, один процесс завершился аварийно, и тогда освободилась половина памяти, которую начал использовать оставшийся процесс.

**Эксперимент №2**

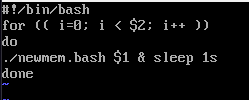
*Подготовительный этап:*

Скрипт newmem.bash, ограничивающийся параметром N($1):

****

*Основной этап:*

Скрипт, запуска go.sh запускающий K ($2) newmem.bash с параметром N($1):



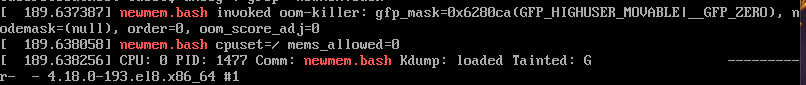
Аварийный размер массива: 31000010

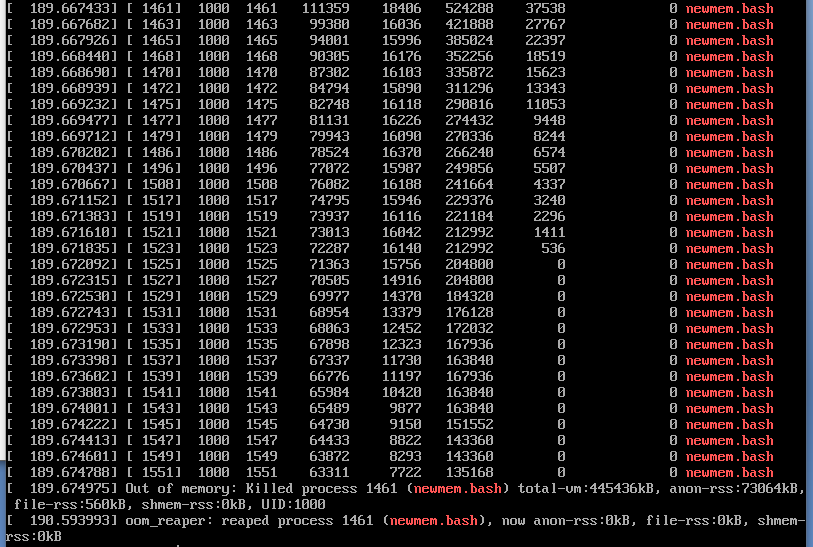
Проверим при N= 3100001 и K=10



Аварийной остановки не было:

Проверим при N= 3100001 и K=30





Аварийно завершены 17 процессов.

Найдем максимальную N при K=30

При N = 1900000 скрипт успешно работает, при N= 2000000 некоторые процессы аварийно завершаются.

Nmax **≈ 1900000.**

**Вывод:**

Процессы полностью заполняют физическую память и начинают использовать файлы подкачки. Когда они заканчиваются ОС аварийно завершает часть процессов, чтобы освободить память для других. Если памяти для них хватает, они успешно завершают процесс.