## 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 № **M3209**

##### Герасимов Григорий Кириллович

Преподаватель:  
**Осипов Святослав Владимирович**

***САНКТ-ПЕТЕРБУРГ 2020***

*Данные о текущей конфигурации ОС:*

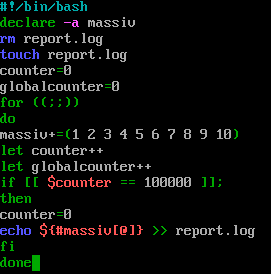
**Отчет**

1. Общий объем оперативной памяти: 512 Мб
2. Объем раздела подкачки: 820 Мб
3. Размер страницы виртуальной памяти: 4 Кб
4. Объем свободной физической памяти в ненагруженной системе: ~ 102 Мб
5. Объем свободного пространства в разделе подкачки в ненагруженной системе: ~ 820 Мб

# Эксперимент I

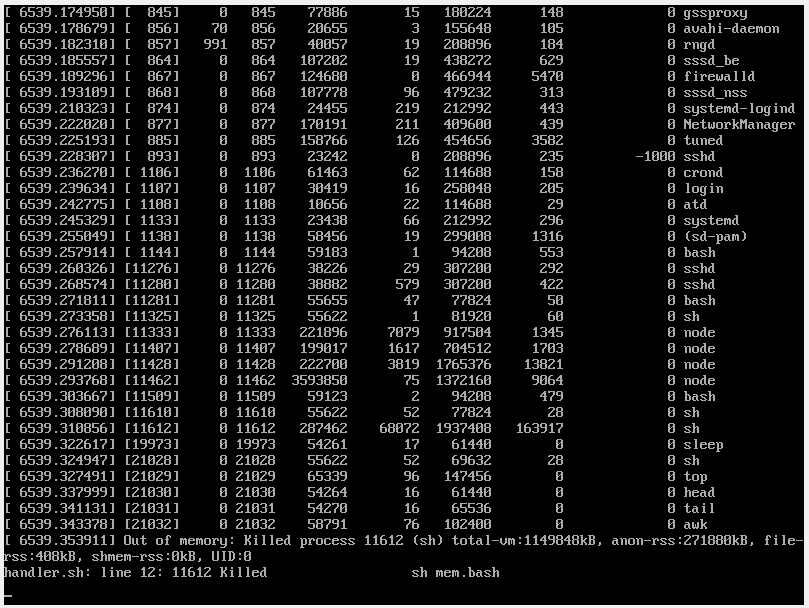
### Подготовительный этап:

1. Уменьшил объем оперативной памяти с 4096 Мб на 512 Мб
2. Был написан скрипт mem.bash, который на каждом шаге цикла в конец массива добавлял последовательность из 10 элементов. Далее, каждую 10000-ый шаг добавлял в файл report.log строка с текущим значением размера массива

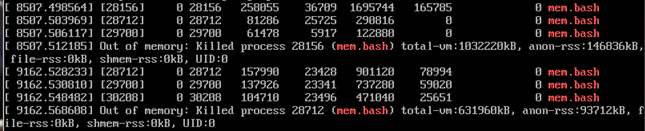


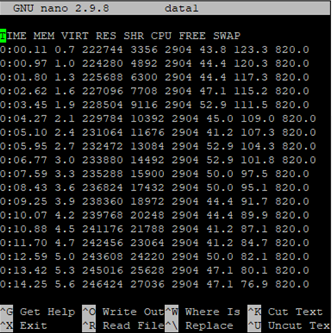
### Первый этап:

1. Последняя запись журнала – значения параметров, с которыми произошла аварийная остановка процесса.



1. Две записи о скрипте в системном журнале

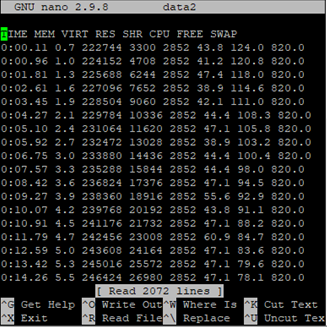


1. Значение в последней строке файла report.log: 49100000
2. Данные который получены во время наблюдения (top): 
3. График зависимости времени от памяти полученный на основе данных из (3)

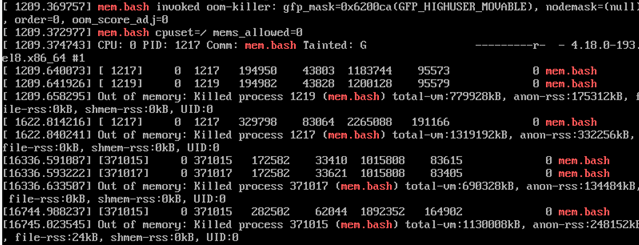
### 

### Второй этап:

1. Производим аналогичные выше действия только с еще одним скриптом
2. Данные который получены во время наблюдения (top):

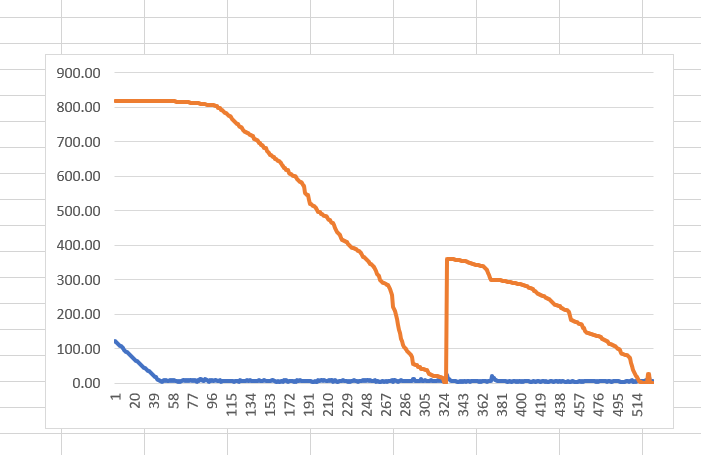


1. Значение в последней строке файла report.log: 23200000 и 44600000
2. Две записи о скрипте в системном журнале



1. График зависимости времени от памяти полученный на основе данных из (3)

MEMORY



### Наблюдения:

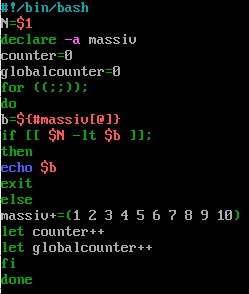
При низкой величине физической памяти происходит переход на файлы подкачки. Как только их значение станет критично, то произойдет аварийное завершение программы.

Если запусти 2 скрипта одновременно, то из графика можно заметить, что один из них продолжил работу, это обусловлено тем, что во время

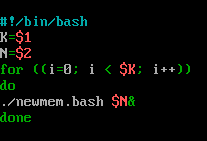
аварийного завершения первого скрипта у нас восстанавливается физическая память, далее идет сценарий как из первого этапа.

**Эксперимент II**

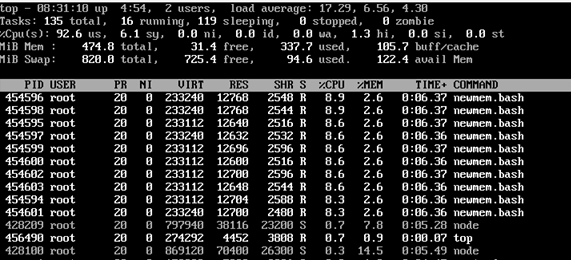
1. Подкорректирован newmem.bash – добавлено условие на прекращение скрипта



1. Был написан обработчик, через который будет производиться запуск

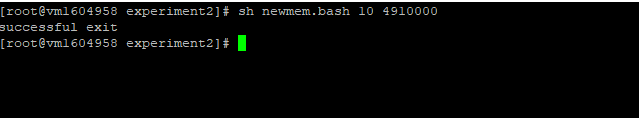


1. Убеждаемся, что было запущено 10 скриптов

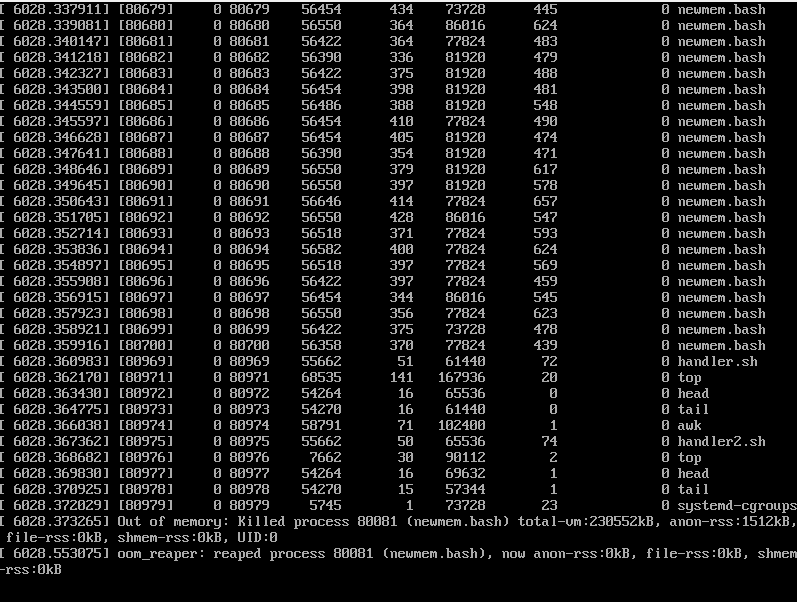


### Наблюдения:

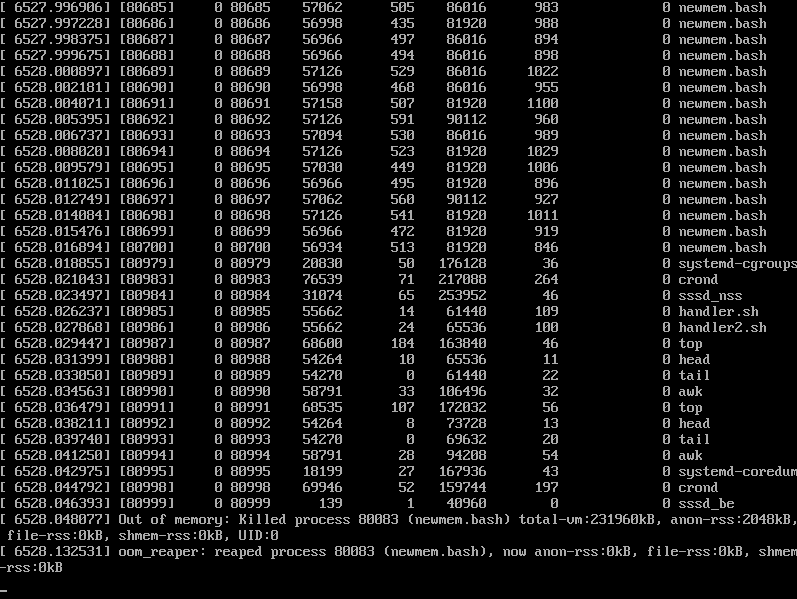
При К = 10 и N = 4910000 программа успешно завершила свое выполнение.



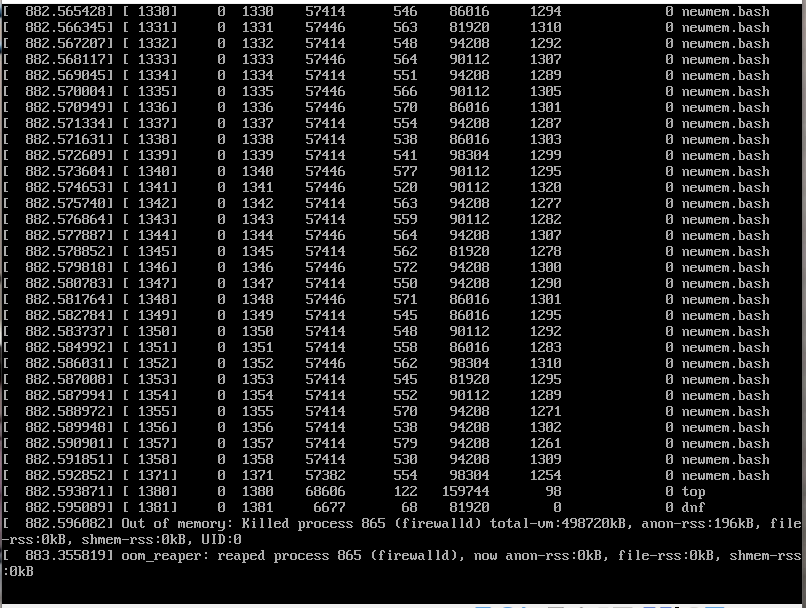
При К = 30 и N = 4910000 программа аварийно завершила свое выполнение.



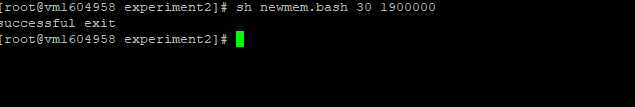
При К = 30 и N = 3900000 программа аварийно завершила свое выполнение.



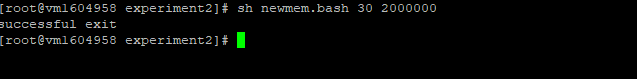
При К = 30 и N = 2900000 программа аварийно завершила свое выполнение.



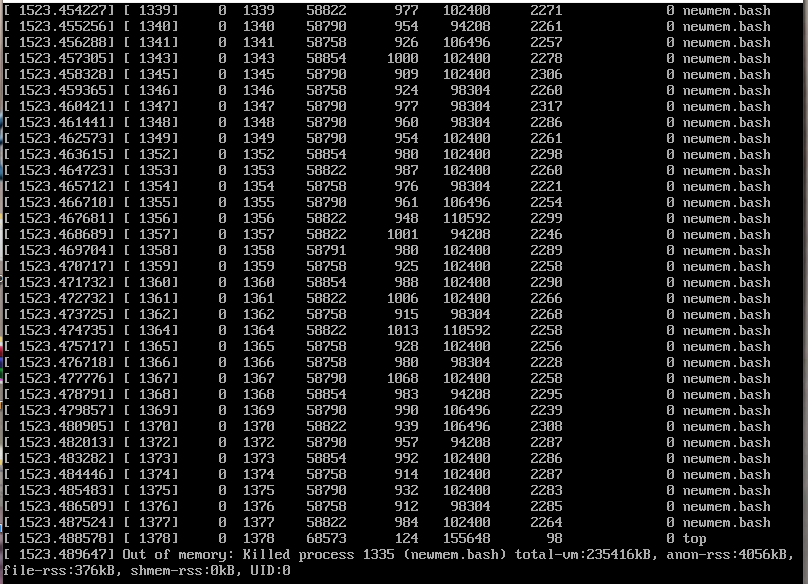
При К = 30 и N = 1900000 программа успешно завершила свое выполнение.



#### При К = 30 и N = 2000000 программа успешно завершила свое выполнение.

****

При К = 30 и N = 2100000 программа аварийно завершила свое выполнение.



**Вывод**

На практике убедился, что во время работы с оперативной памятью данные заполняют физическую память, пока не дойдет до критического значения. Далее ОС начинает заполнять файлы подкачки. В случае, когда фалов подкачки будет не хватать

– произойдет аварийная остановка процесса