**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

**по лабораторная работа №5. «Управление памятью в ОС Linux»**

**по дисциплине «Операционные системы»**

Автор: Жуков Максим Александрович

Факультет: ИТиП

Группа: M3211

Преподаватель: Дюкарева Вероника Максимовна



Санкт-Петербург 2020

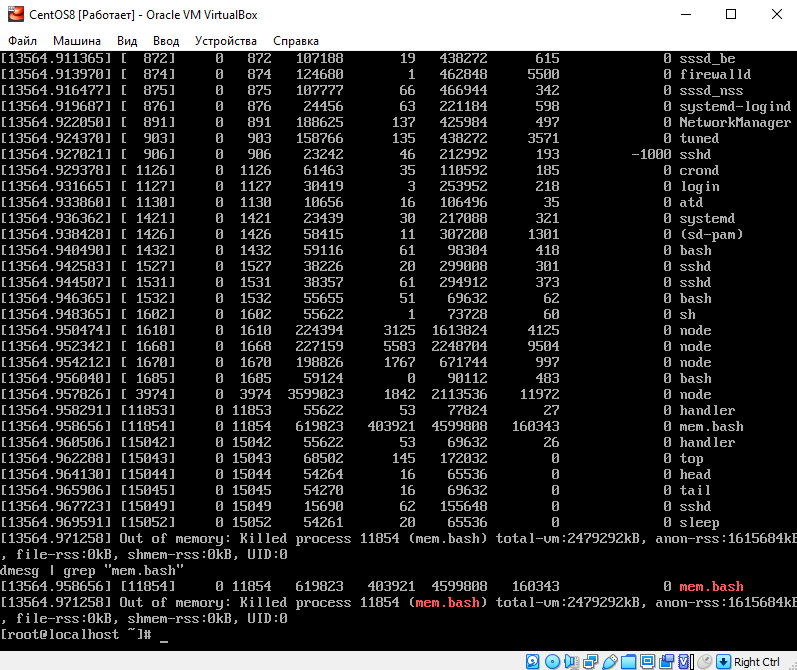
*Данные о текущей конфигурации операционной системы:*

1. Общий объём оперативной памяти: 2048 МБ
2. Объём раздела подкачки: 819 МБ
3. Раздел страницы виртуально памяти: 4 КБ
4. Объем свободной физической памяти в ненагруженной системе: 1262 МБ
5. Объем свободного пространства в разделе подкачки в ненагруженной системе: 819 МБ

**Эксперимент №1**

*Первый этап:*

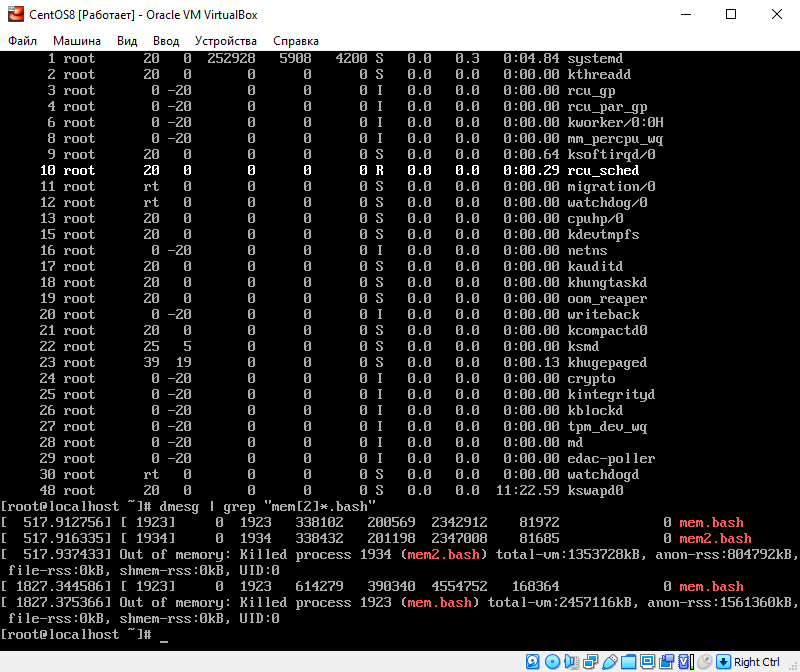
1. Последняя запись журнала – значение параметров, с которыми произошла аварийная остановка процесса:



1. Значение в последней строке файла report.log: 28000000.
2. Данные, полученные во время наблюдения top, находятся в файле data1.
3. График зависимости времени от памяти полученный на основе данных из (3):

*Второй этап:*

1. Последние записи о скриптах в системном журнале:



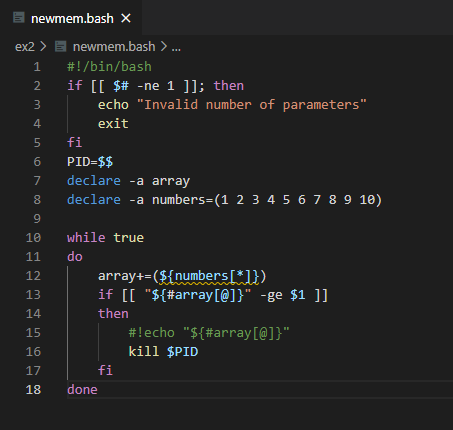
1. Значения в последних строках report2\_1.log и report2\_2.log: 28000000, 14000000.
2. Данные, полученные во время наблюдения top, находятся в data2\_1 и data2\_2.
3. Графики зависимости времени от памяти полученный на основе данных из (3):

*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.

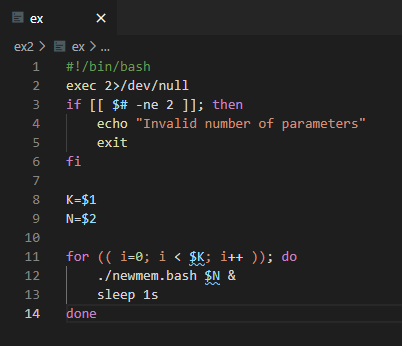
**Эксперимент №2**

*Подготовительный этап:* скрипт newmem.bash:



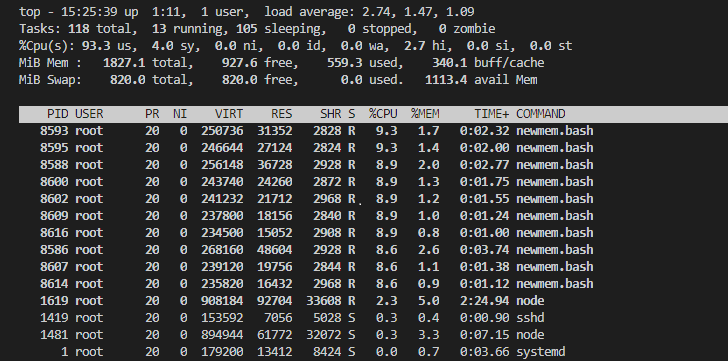
*Основной этап:*

Скрипт, который будет запускать newmem.bash:

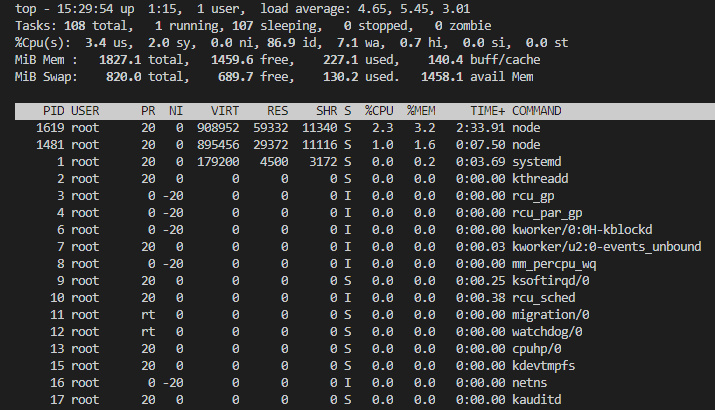


1. N = 2800000, K = 10:

Видим, что newmem.bash был запущен 10 раз:



И каждый процесс был успешно завершён:

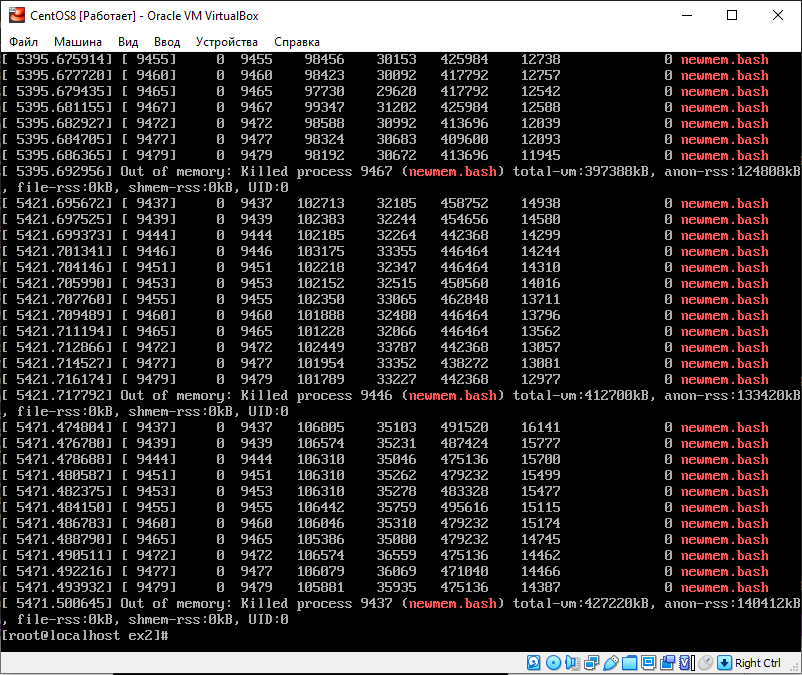


Как видно, системном журнале нет записей об аварийной остановке процесса:



1. N = 2800000, K = 30:

Как видно, множество процессов завершились аварийно:



Это происходит из-за того, что для всех процессов не хватает памяти, что логично. В первом пункте мы использовали N = 28000000(максимальный размер, при котором не происходит аварийной остановки) / 10, поэтому памяти на всех хватило. Но в случае, когда K = 30 получается, что N \* K > 28000000.

1. Для подбора такого максимальное значение N, чтобы при K=30 не происходило аварийных завершений процессов, используем: N \* 30 = 28000000; N = 933333 – с такого и начинаем.

После пары десятков тестов, я выяснил, что он находится в районе 1300000.