**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: Коверин Анатолий Денисович

Факультет: ИТиП

Группа: M3207

Преподаватель: Осипов Святослав Владимирович



Санкт-Петербург 2020

# Эксперимент №1

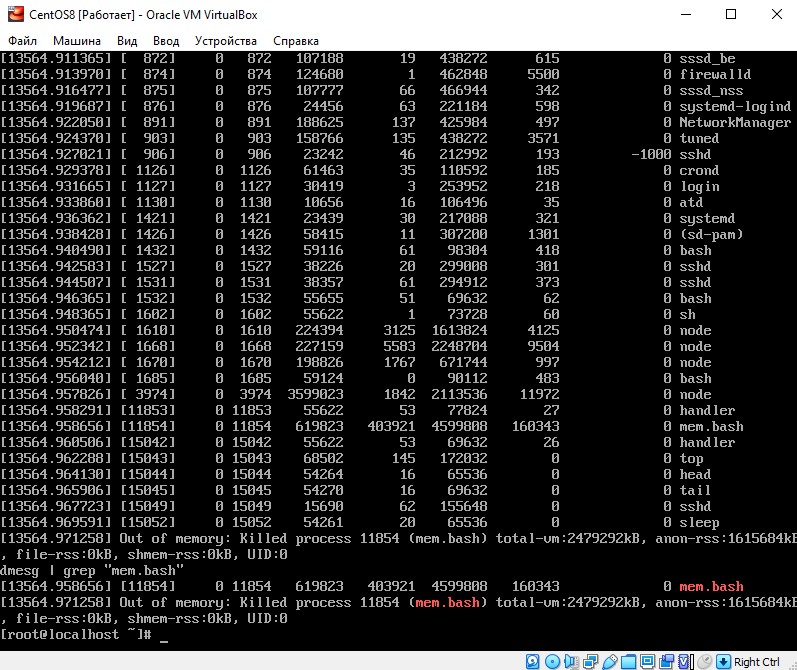
Первый этап*:*

**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

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



1. Значение в последней строке файла: 28000000.
2. График зависимости времени от памяти

0

200

400

600

800

1000

1200

0:00

0:08

0:17

0:26

0:35

0:43

0:52

1:01

1:10

1:17

1:21

1:26

1:34

1:43

1:51

2:00

2:09

2:20

2:29

2:38

2:47

2:56

3:05

3:13

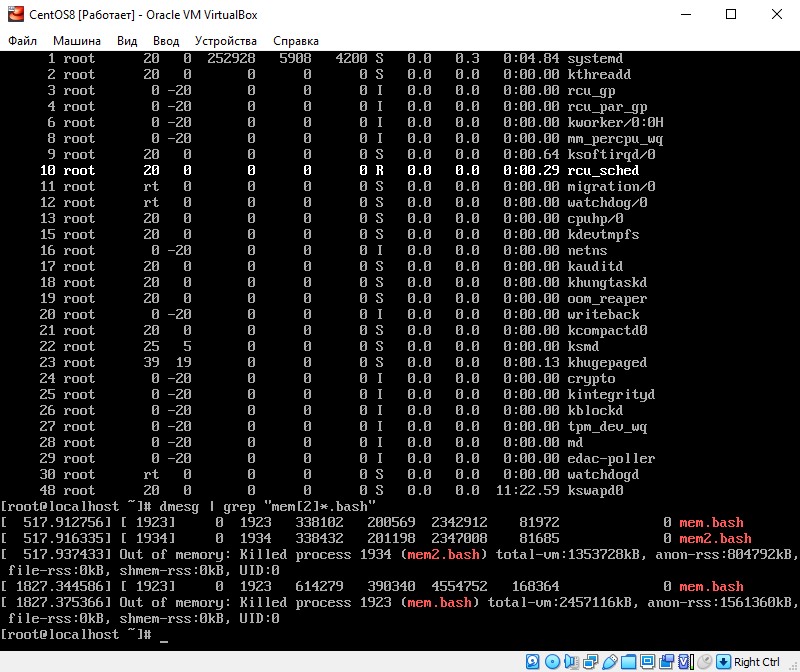
3:23

FREE

SWAP

*Второй этап:*

1. Последние записи о скриптах в системном журнале:



1. Значения в последних строках: 28000000, 14000000.
2. Графики зависимости времени от памяти

00

,

0

200,00

400,00

600,00

800,00

1000,00

1200,00

1400,00

0:00

0:05

0:10

0:16

0:21

0:26

0:32

0:37

0:42

0:48

0:53

0:59

1:06

1:11

1:16

1:22

1:28

1:33

1:39

1:52

2:01

2:11

2:21

2:31

2:40

2:50

3:06

mem.bash

FREE

SWAP

0

,

00

200,00

400,00

600,00

800,00

1000,00

1200,00

1400,00

0:00

0:03

0:07

0:11

0:15

0:18

0:22

0:26

0:30

0:34

0:37

0:41

0:45

0:49

0:53

0:57

1:01

1:06

1:10

1:14

1:18

1:22

1:26

1:30

1:34

1:38

mem2.bash

FREE

SWAP

**Наблюдения***:*

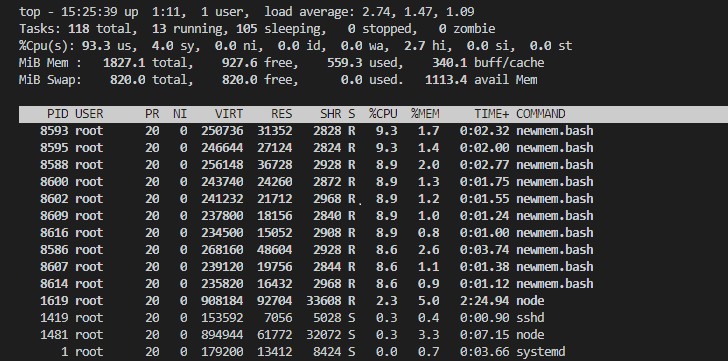
Если физическая память маленькая, переходим на файлы подкачки. Аварийное заверщение программы происходит при критическом значении. При запуске 2х скриптов, один из них продолжил работу, так как во время аварийного завершения первого скрипта физическая память восстанавливается. Далее идет сценарий из первого этапа.

**Эксперимент №2**

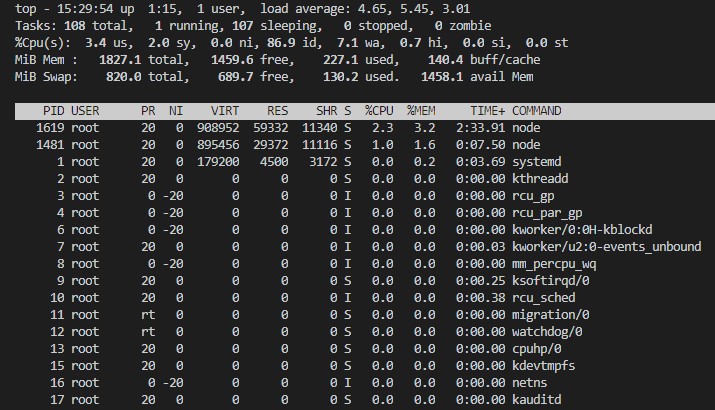
1. N = 2800000, K = 10:

Видим, что newmem.bash был запущен 10 раз:

В системном журнале нет записей об аварийной остановке процесса:

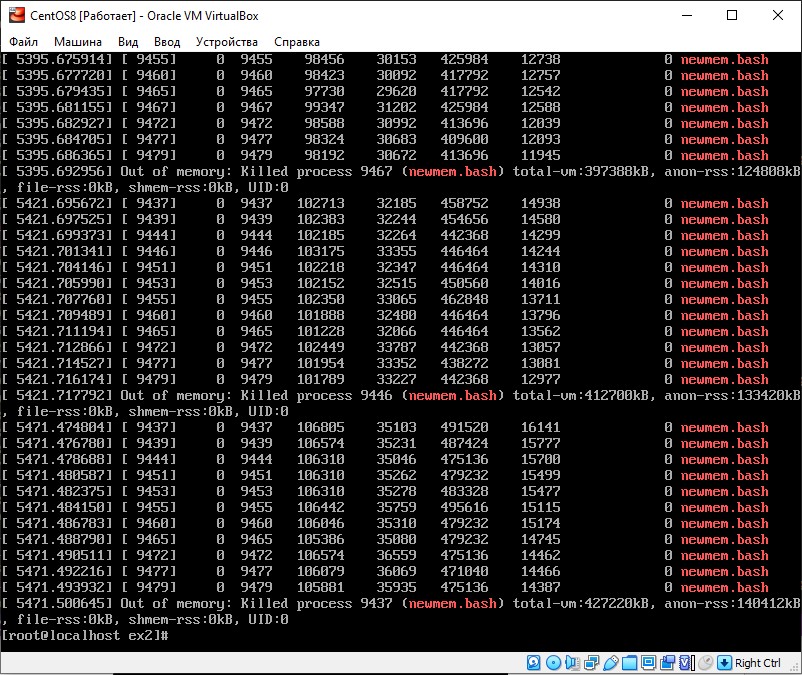


каждый процесс был успешно завершён:



1. N = 2800000, K = 30:

Множество процессов завершились аварийно:



Это происходит из-за того, что для всех процессов не хватает памяти. 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.