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 №M3204

***Белоус Лев Юрьевич***

Проверила

***Титова Анастасия Витальевна***

***САНКТ-ПЕТЕРБУРГ***

***2020***

**Текущая конфигурация операционной системы**

Общий объем оперативной памяти = ***228148 KБ***

Объем раздела подкачки = ***839676 KБ*** *(swapon –show)*

Размер страницы виртуальной памяти = ***4 КБ***

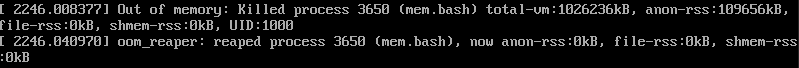
Объем свободной физической памяти в ненагруженной системе = ***3660 КБ***

Объем свободного пространства в разделе подкачки в ненагруженной системе = ***819844 КБ***

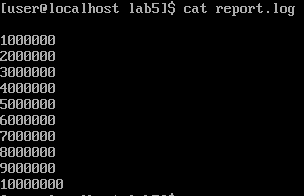
**Эксперимент 1**

**Первый этап**

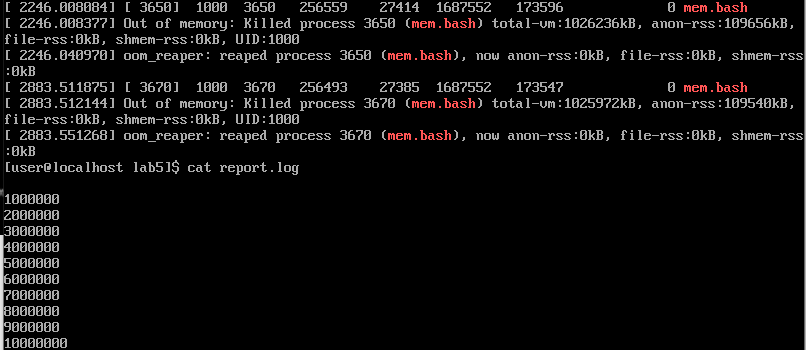
Последняя запись журнала – значенияпараметров, с которыми произошла аварийная остановка процесса.



Значение в последней строке файла report.log – 10000000.



Последние две записи о скрипте в системном журнале. Также значение в последней строке файла report.log - 10000000



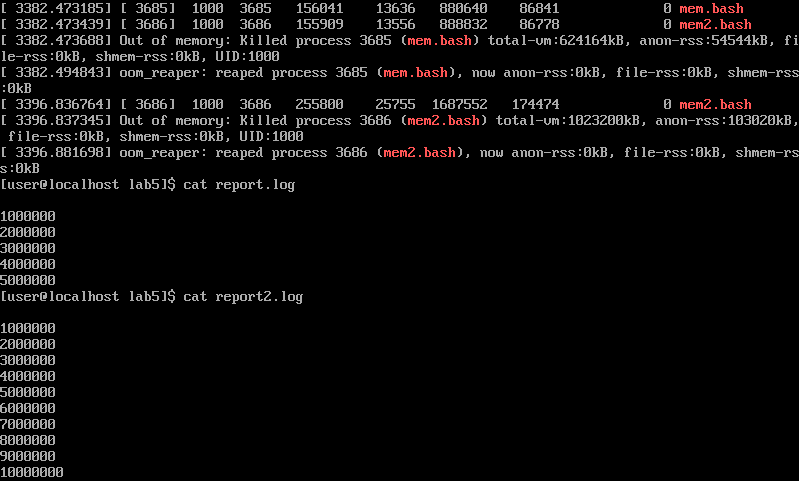
**Графики**

Основные процессы top, kswapd0, tuned, sshd, systend, kworker/0:1H-kblockd, NetworkManager, rcu\_sched

Исходя из графиков, видно, что происходит использование оперативной памяти, потом подключается память из раздела подкачки. Когда заканчивается и она, то процесс убивается.

**Второй этап**

Последние записи о скриптах в системном журнале. Также значения в последних строках файлов report.log и report2.log – 5000000 и 10000000 соответственно.

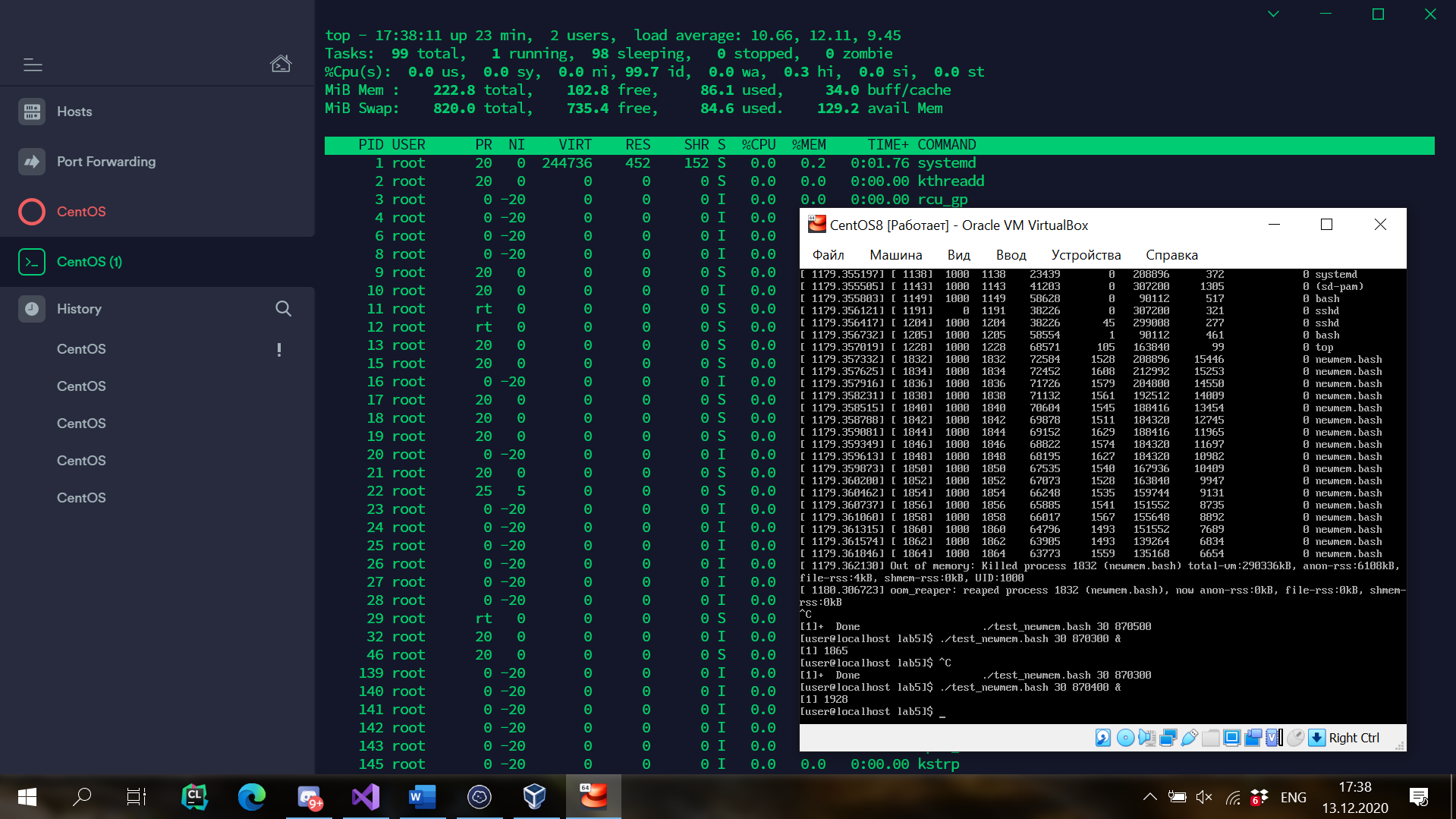


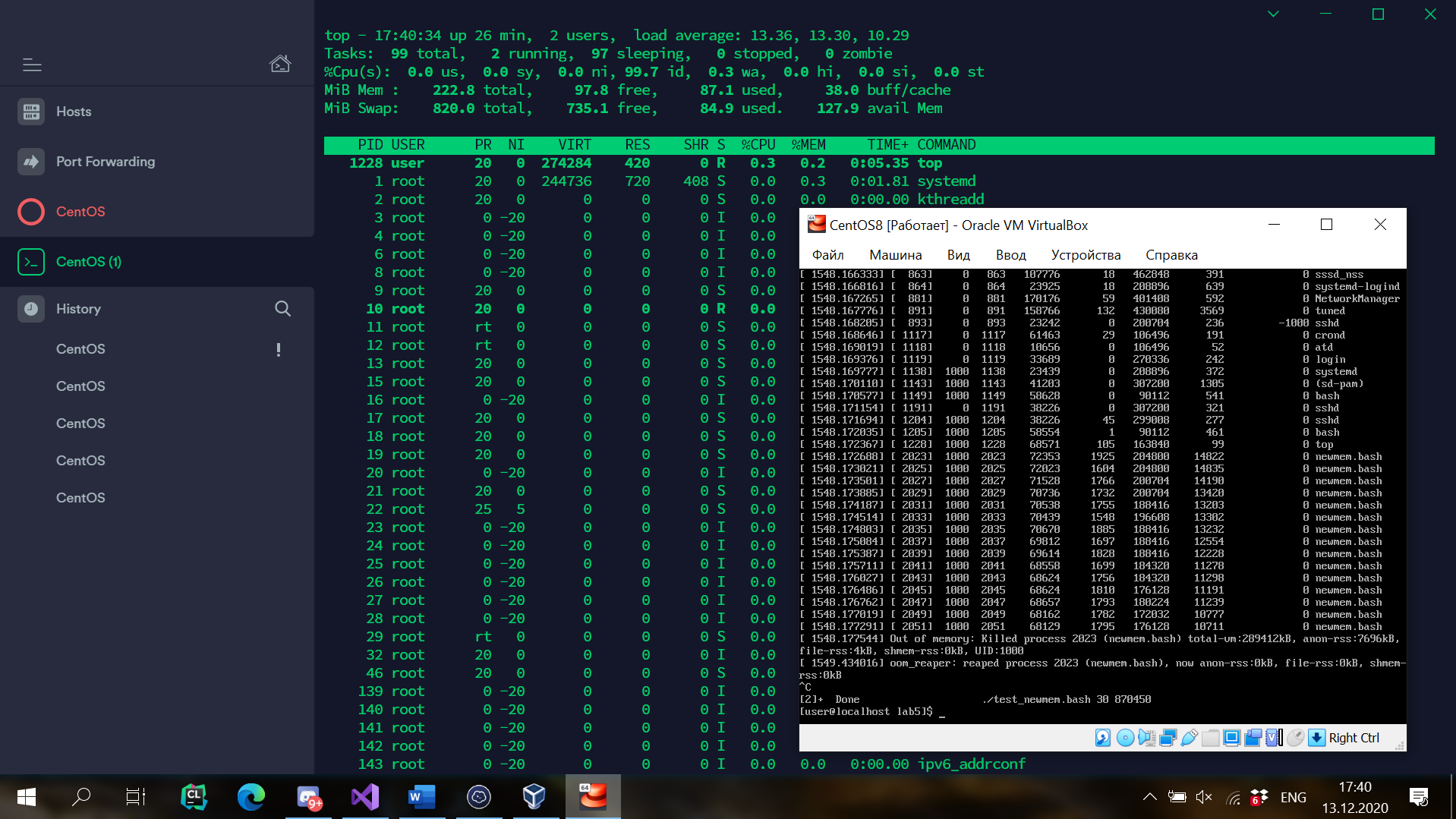
**Графики**

Исходя из графиков, видно, что происходит использование оперативной памяти, потом подключается память из раздела подкачки. Когда заканчивается и она, то процесс убивается.

Дальше память из раздела подчкачки обновляется и один из процессов продолжает свою работу.

**Эксперимент 2**

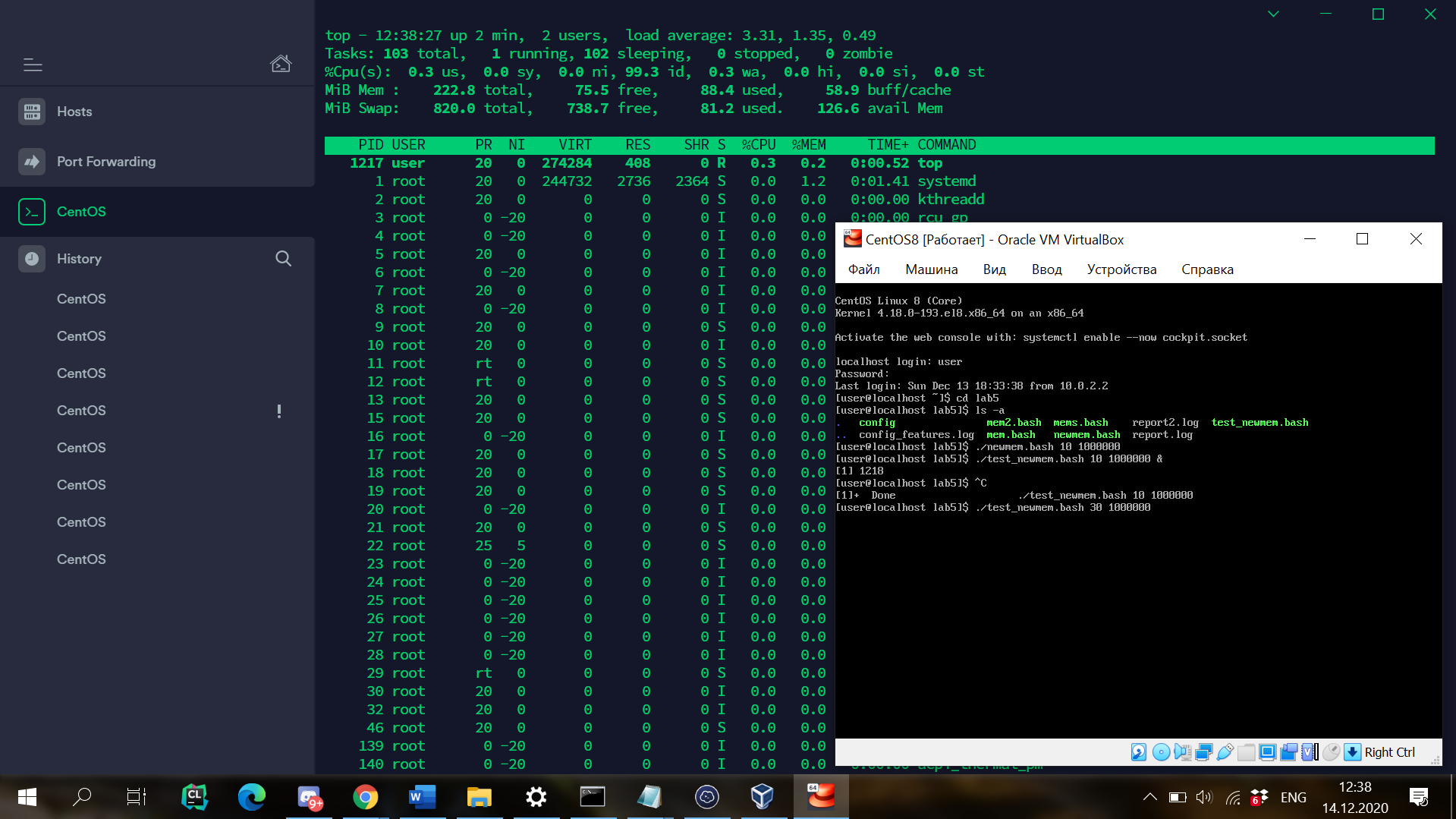




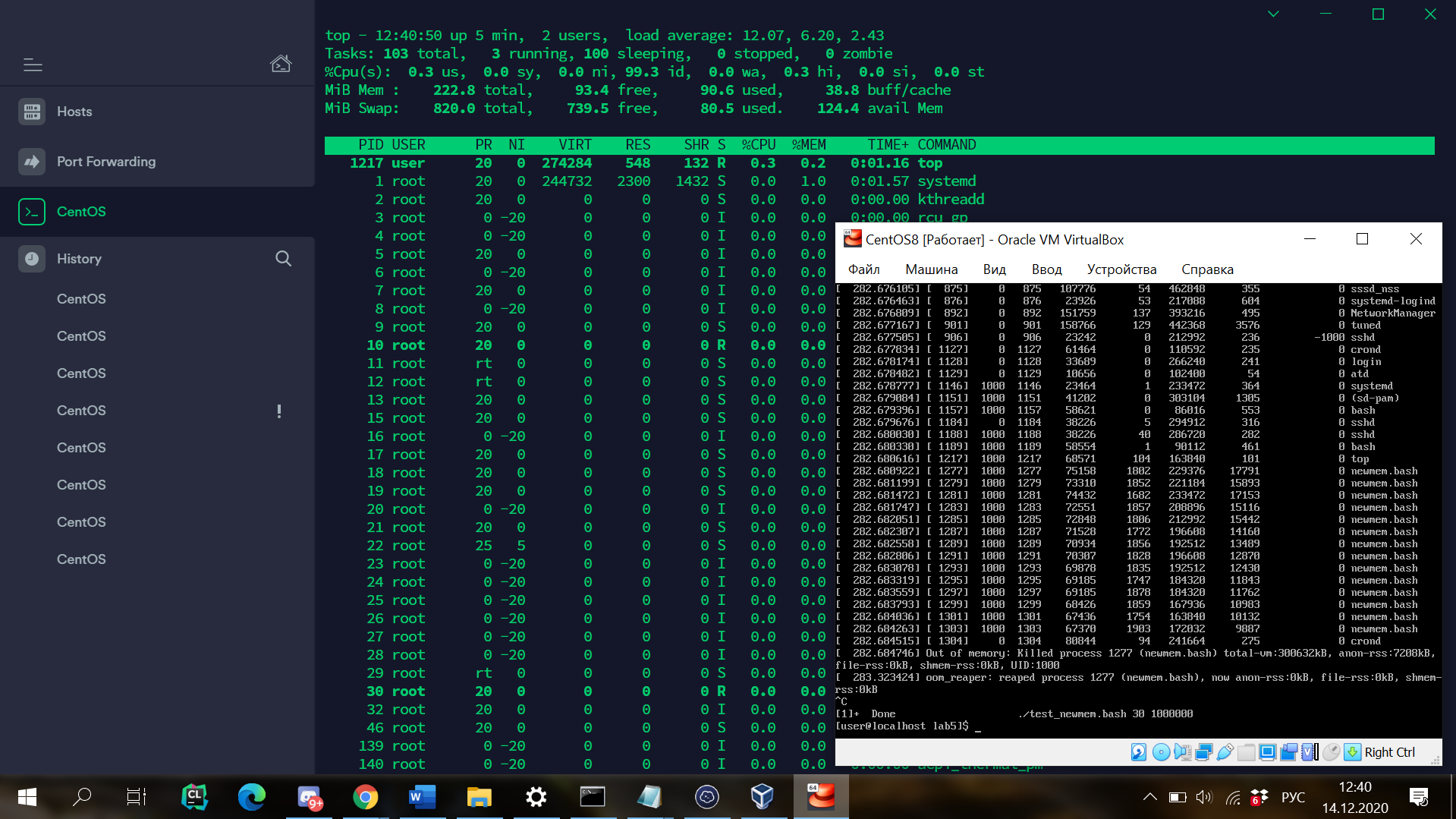
Максимальное значение N при K = 30 – **870400**

**Наблюдения во втором эксперименте:**

При **К = 10** и **N = 1000000** программа закончила свое выполнение без ошибок.



При **К = 30** и **N = 1000000** программа аварийно завершила свое выполнение.



**Вывод:**

Сначала заполняется оперативная память, а потом используется объем из раздела подкачки. Когда вся память заканчивается, то незавершенные процессы убиваются. Если работают одновременно 2 скрипта, то один из них крашится, когда заканчивается память. Дальше память из раздела подкачки обновляется. Скрипт, который не вылетает, продолжает работу, пока память из раздела подкачки опять не закончится. Допустим, у нас заполняется память при массиве из **C** элементов. Если мы берем **C** = N \* K, где N >> K, увеличиваем K в несколько раз, то программа вылетит.