**Отчёт по лабораторной работе**

*Фамилия Имя*

Общие параметры системы(Ubuntu 20.4.1 LTS)

* Общий объем оперативной памяти: 3908 MiB
* Объем раздела подкачки: 3921 MiB
* Размер страницы виртуальной памяти: 2048 KB
* Объем свободной физической памяти в ненагруженной системе: 3539 MiB
* Объем свободного пространства в разделе подкачки в ненагруженной системе: 3872 MiB

Эксперимент 1

*Первый этап, снято с интервалом в 10 секунд:*

Всё время жизни процесс mem.bash висел в первой строчке top.

N = 9000000.

*Второй этап:*

График получился такой же для каждого из процессов. Запущенные почти одновременно, они работали почти синхронно, и упали тоже одновременно. Вероятно, наличие двух ядер в распоряжении виртуальной машины позволили планировщику создать условия для такого поведения.

Эксперимент 2, снято с интервалом в 1 секунду

K=10, n = 900000:

Пороговое значение при K=30 n=3250000.

Вывод: операционной системе удаётся управлять памятью 30 маленьких процессов эффективнее, чем памятью одного большого, хотя суммарно в первом случае используется памяти больше, чем во втором. Скорее всего, это происходит из-за того, что массив, используемый в скриптах – цельная структура, которая должна лежать в памяти подряд. Соответственно, набор маленьких массивов легче перемещать между оперативной памятью и подкачкой, чем один большой, вследствие чего программа с одним массивом падает быстрее. Кроме того, вероятно, что когда был запущен последний процесс из 30, несколько первых уже успели завершиться, ведь в начале работы им была предоставлена вся память. Поэтому в каждый момент времени в памяти находилось меньше 30 процессов.