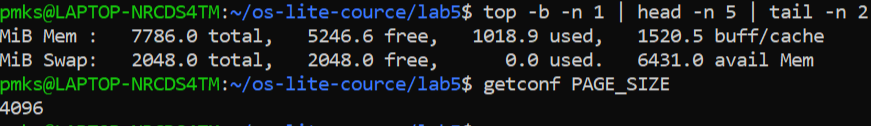
**Отчет по 6 лабораторной.**

1. **Параметры системы.**

****

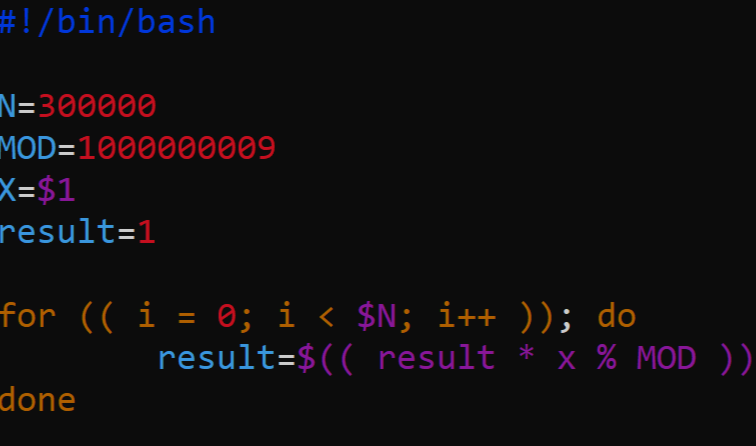
1. Общий объем оперативной памяти: 7786 Мб
2. Объем раздела подкачки: 2048 Мб
3. Размер страницы виртуальной памяти: 4096
4. Объем свободной физической памяти в ненагруженной системе: 5247 Мб
5. Объем свободного пространства в разделе подкачки в ненагруженной системе: 2048 Мб

**Характеристики процессора.**

1. Модель: 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
2. Процессоров: 1
3. Ядер: 4
4. Потоков: 2
5. **Небольшое уточнение.**

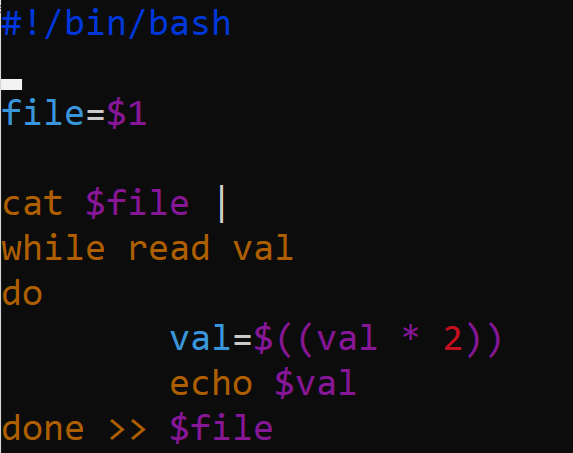
Если посчитать общее время на запуск всех скриптов по условию, получится порядка 13-14 часов, что довольно затруднительно. Поэтому немного уменьшим количество запусков: каждый эксперимент запускаю 5 раз, внутри одного эксперимента для каждого N в диапазоне от 1 до 10 запускаю скрипт по 5 раз, причем 1 запуск 1 скрипта будет занимать 1 секунду.

1. **Скрипты.**



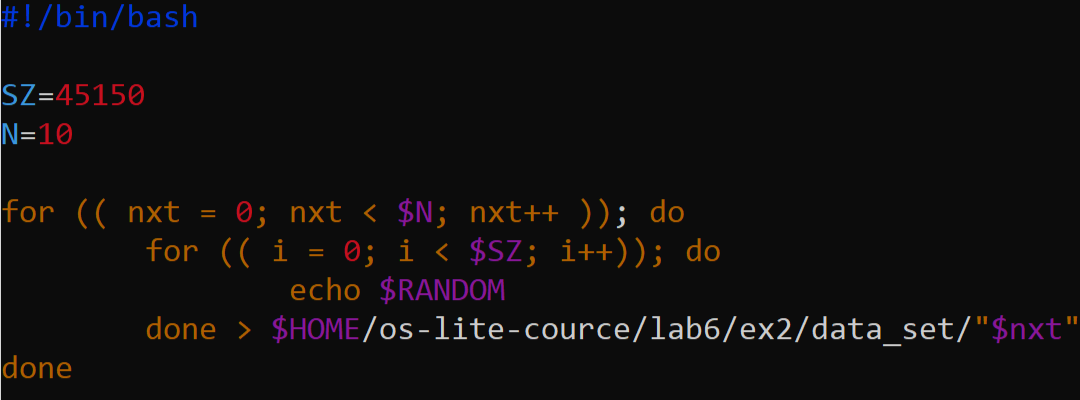
1. Функция для первого эксперимента.

Функция считает (X ^ N) % MOD, где X – входной параметр, N и MOD – специально подобранные константы, чтобы время выполнения было около 1с.



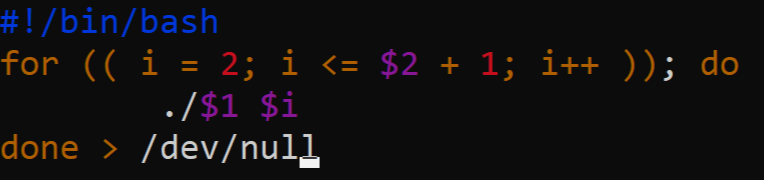
2. Скрипт для второго эксперимента.

Скрипт берёт данный как параметр файл датасета, умножает все числа в нем на 2 и дописывает в конец того же файла.



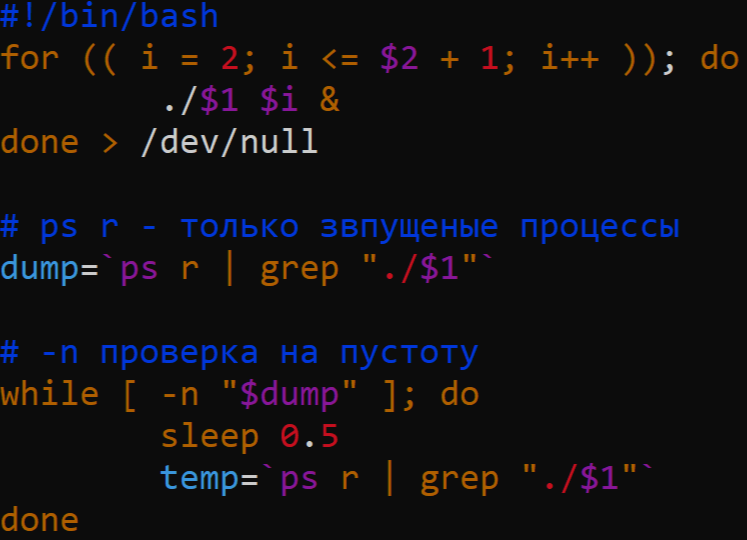
3. Скрипт для генерации датасета.

Данный скрипт, как понятно, генерирует случайные файлы с числами. SZ специально подобран, чтобы размер файла был 250 Кб.



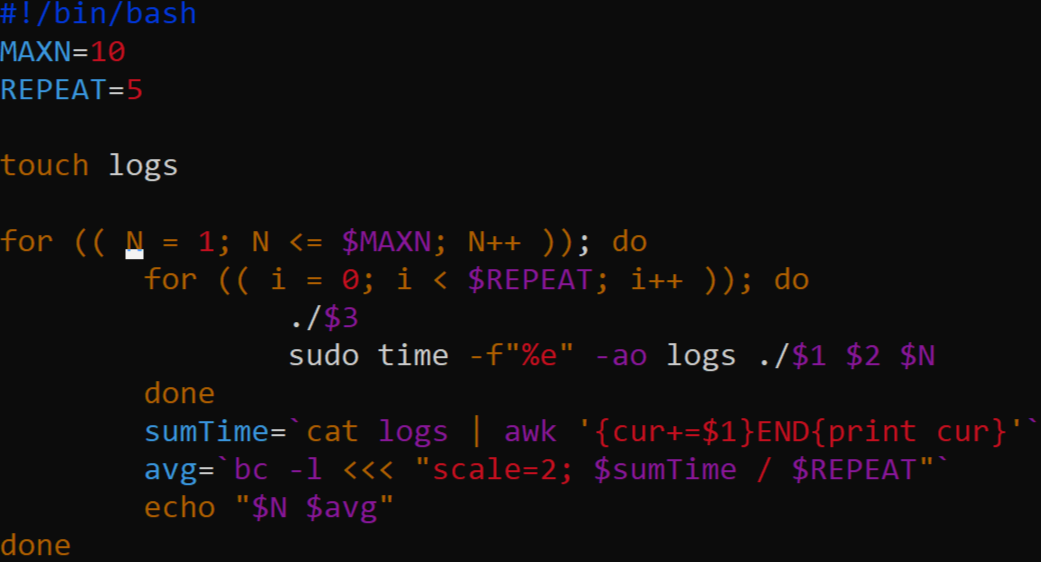
4. Скрипт запуска в последовательном режиме.

Этот скрипт принимает первым параметром запускаемый скрипт, вторым – сколько раз надо исполнить данный скрипт.



5. Скрипт запуска в параллельном режиме.

Принимает то же самое, что и предыдущий, только запускает в фоновом режиме и ждёт пока все запущенные скрипты не завершатся. Отмечу, что sleep вносит небольшие неточности в подсчёт общего времени, но дальше будем им пренебрегать.



6. Скрипт для общего запуска всех скриптов и режимов.

Скрипт принимает 3 параметра: 1 – скрипт модификации запуска (последовательно/параллельно), 2 – скрипт, который будет запускаться внутри мода, 3 – скрипт, который надо сделать до запуска модификации (для 1 эксперимента пустой скрипт, для 2го – генератор датасета).

**Вариации запуска:**

Эксперимент 1:

1 часть: ./start.sh seq\_executor.sh ex1/func.sh ex1/nop.sh

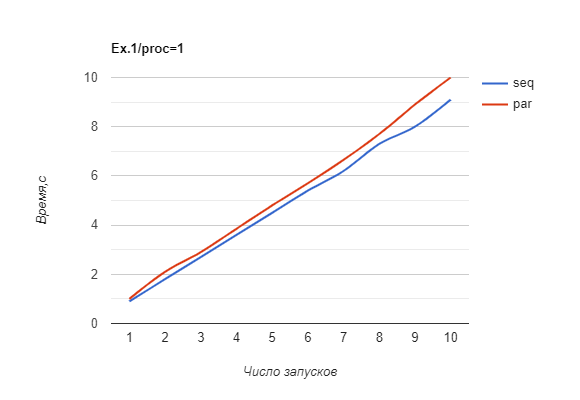
2 часть: ./start.sh parallel\_executor.sh ex1/func.sh ex1/nop.sh

Эксперимент 2:

1 часть: ./start.sh seq\_executor.sh ex2/algo.sh ex2/ data\_generator.sh

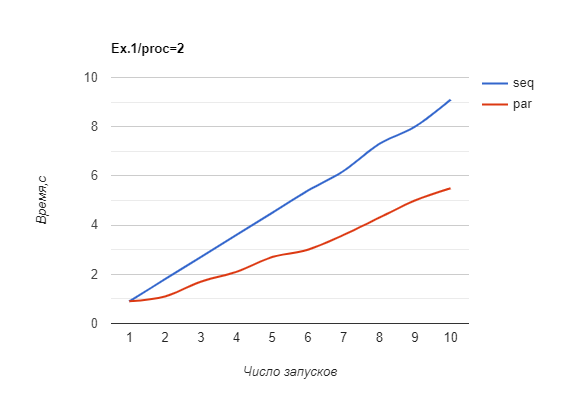
2 часть: ./start.sh parallel\_executor.sh ex2/algo.sh ex2/ data\_generator.sh

1. **Графики.**
2. Первый эксперимент, 1 процессор:



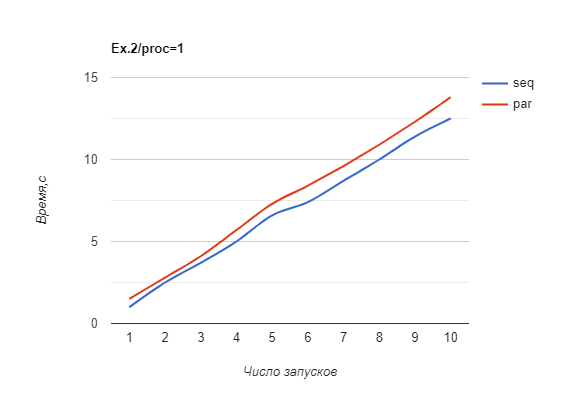
Оба графика экстраполируются прямой линией достаточно хорошо, особенно при последовательном запуске. Видно, что графики одинаково возрастают, небольшая разница обуславливается спецификой запуска скриптов параллельно (проблема в sleep). Результат ожидаемый, так как процессор 1 и нет разницы, как именно запускать скрипты, если исполняться будет только один.

1. Первый эксперимент, 2 процессора:



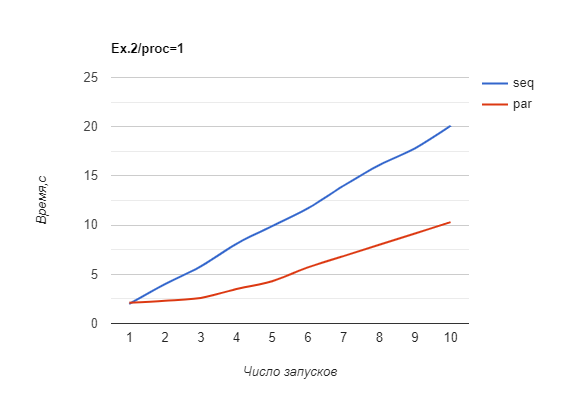
Опять видно, что линейный рост, но у параллельного запуска коэффициент примерно в 2 раза меньше. На отрезке [1;2] почти нет роста у параллельного, поскольку второй процессор начинает обрабатывать второй вызов скрипта. Результат ожидаемый: 2 процессора дают выигрыш при параллельном исполнении.

1. Второй эксперимент, 1 процессор:



Растут линейно, почти одинаково, потому что 1 процессор.

1. Второй эксперимент, 2 процессора:



Рост последовательного линеен, у параллельного для [1;2] почти константа, после 3 тоже линеен. Стоит отметить, что тут сильнее разница в скорости работы при хотя бы 3 запусках. Так происходит из-за того, что во втором эксперименте скрипт не имеет сложных вычислений, проще распараллеливается простыми операциями ввода и вывода.

1. **Выводы.**
2. При использовании 1 процессора никакой выгоды нет от способа запуска скриптов.
3. Больший выигрыш от нескольких процессоров получают скрипты, в которых используются несложные операции, не требующие много процессорного времени.
4. Выигрыш параллельного запуска наибольший, если число запускаемых скриптов равно числу процессоров.