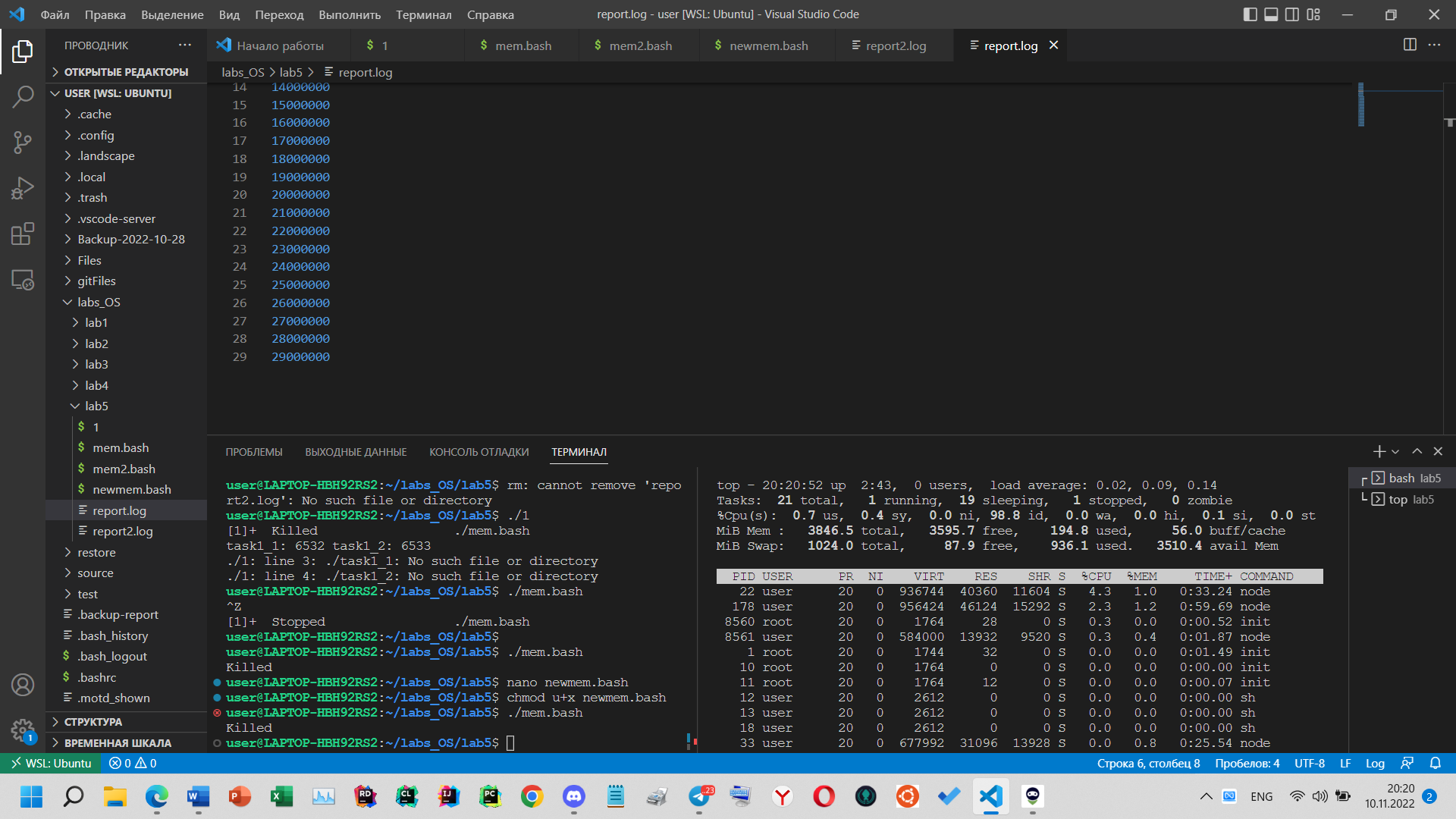
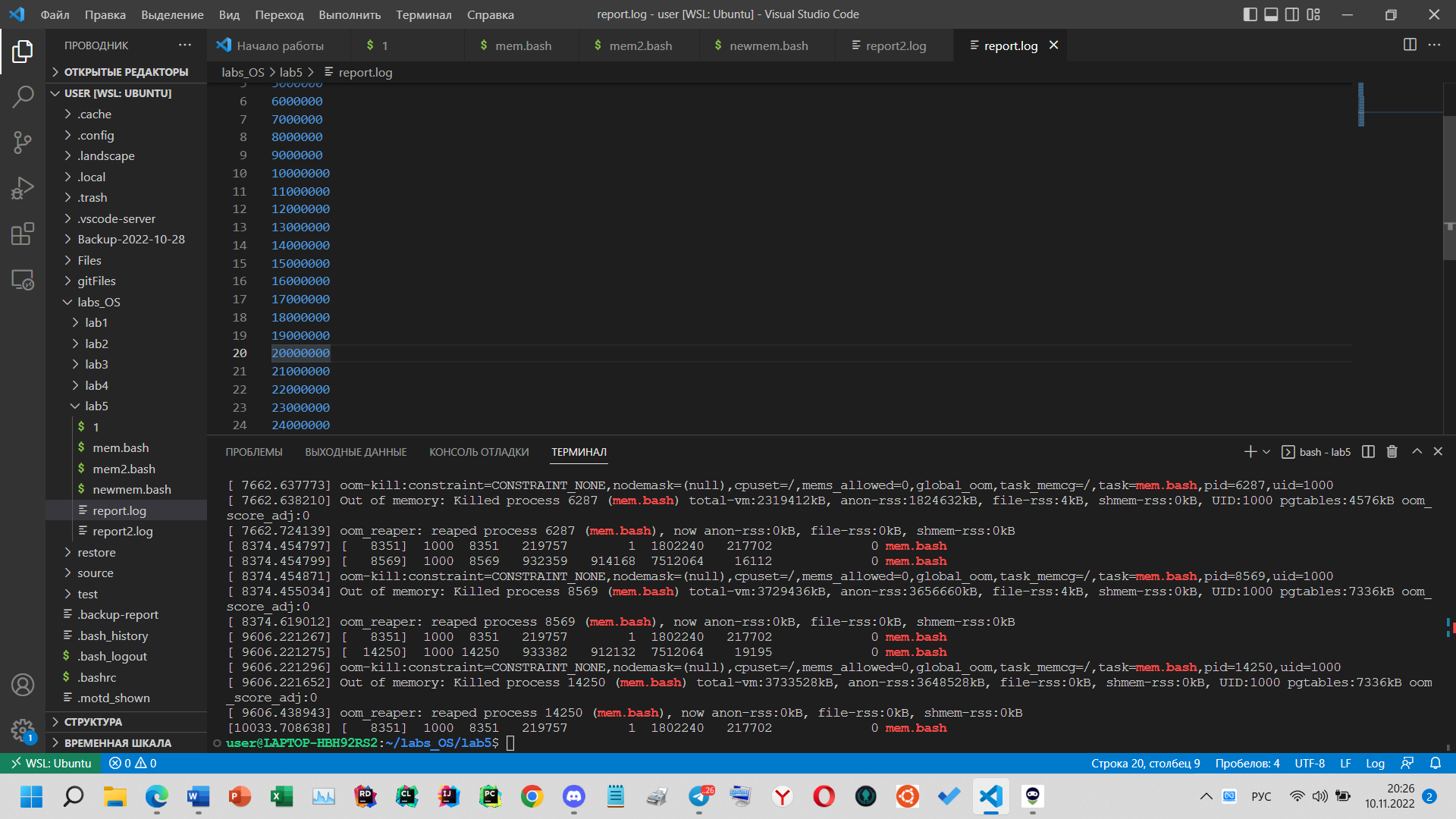
1 эксперимент

1 Этап:

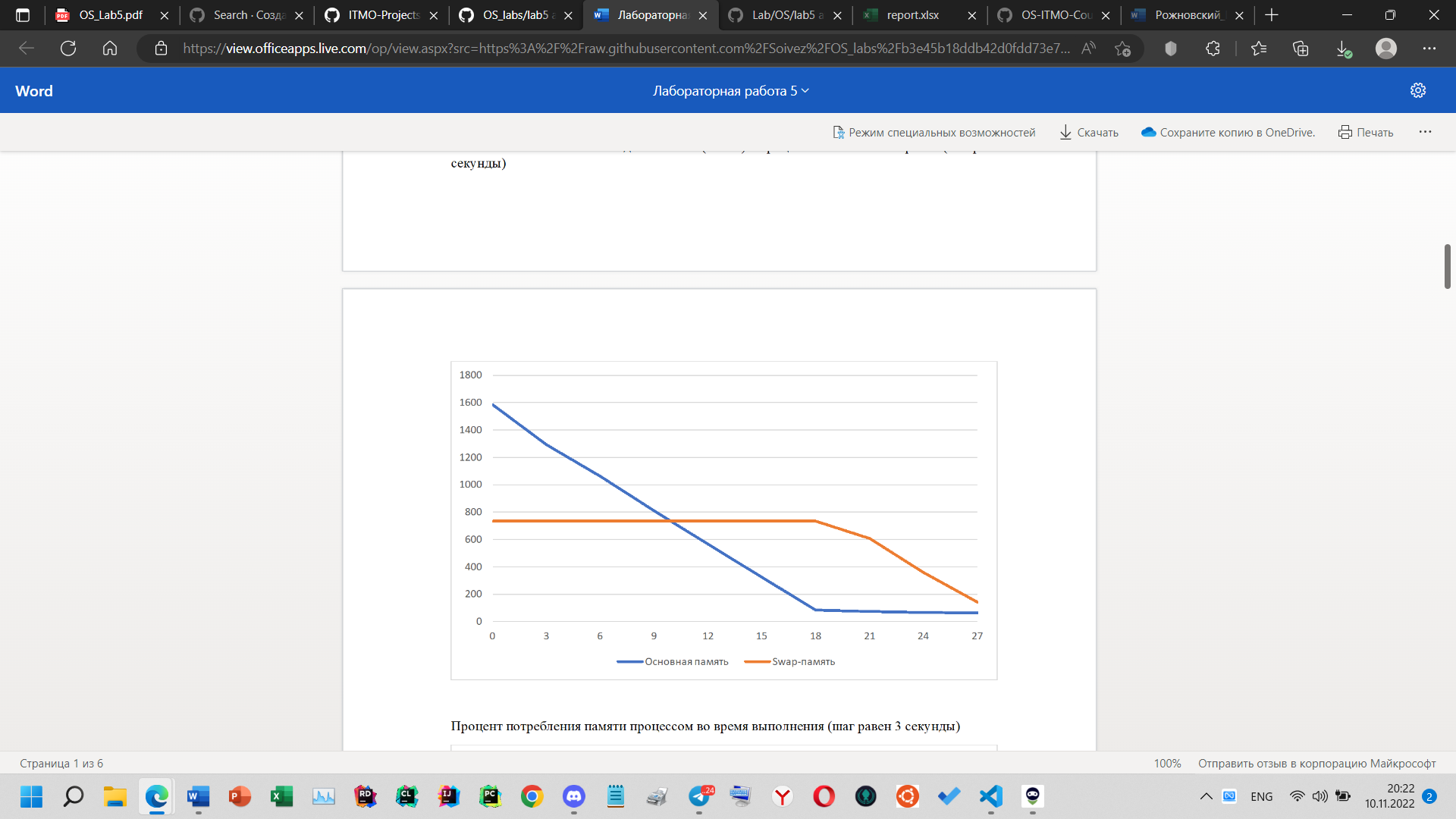
Записи в report.log



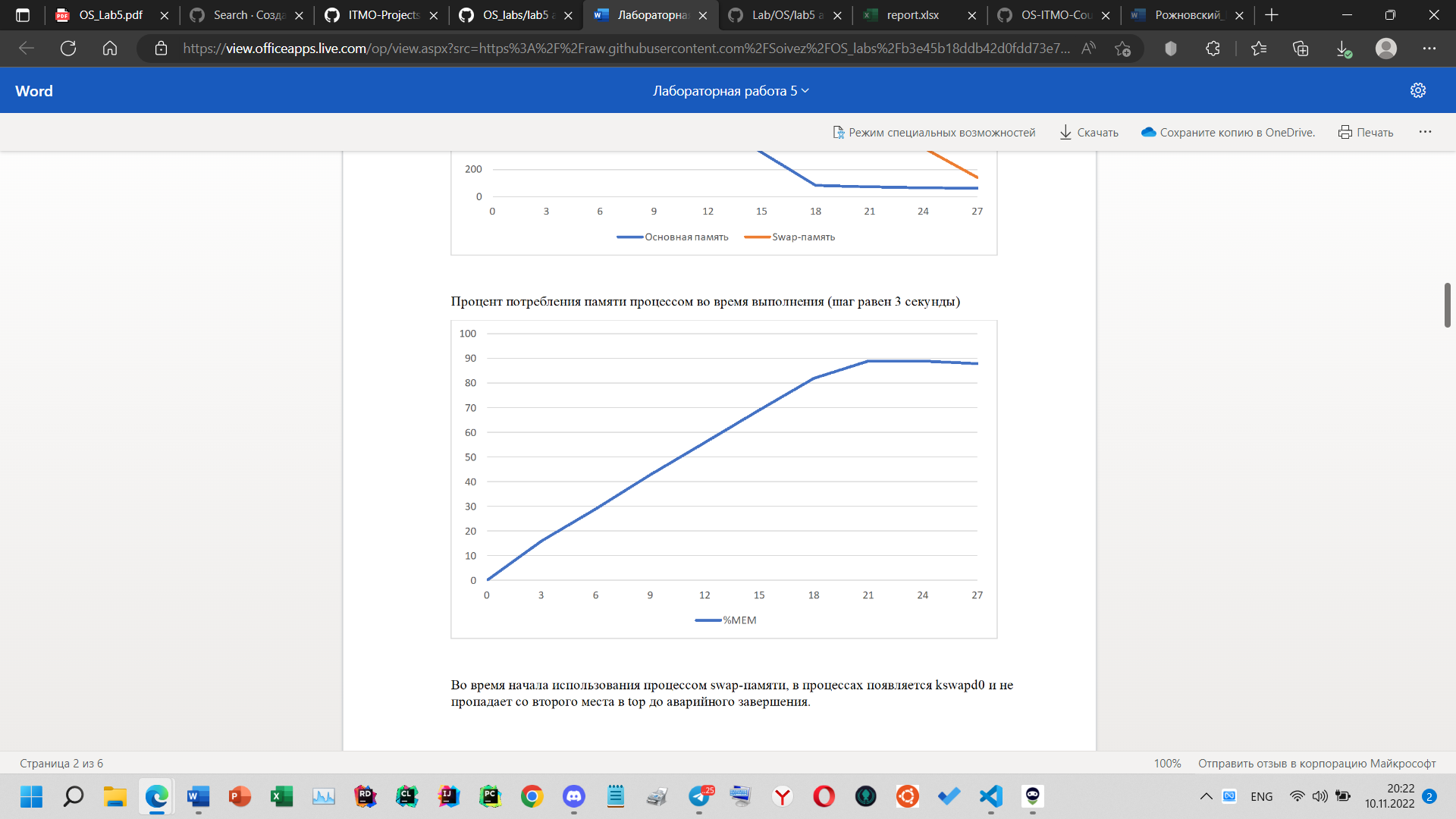
В логах ошибки командой **dmesg | grep "mem.bash"** было получено сообщение:



Изменение количества свободной памяти (в MiB) в процессе выполнения скрипта (шаг равен 3 секунды)

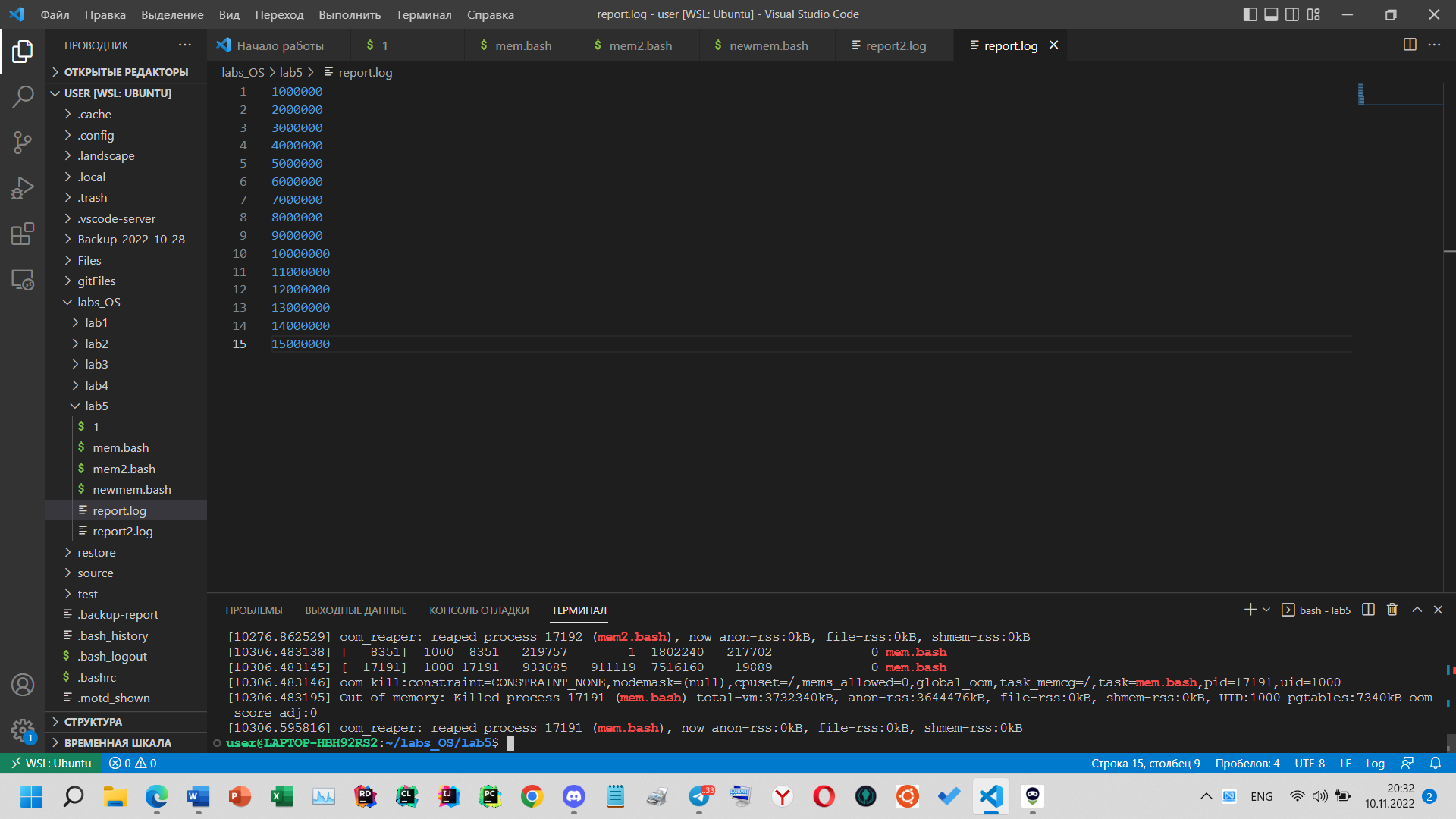


Процент потребления памяти процессом во время выполнения (шаг равен 3 секунды)

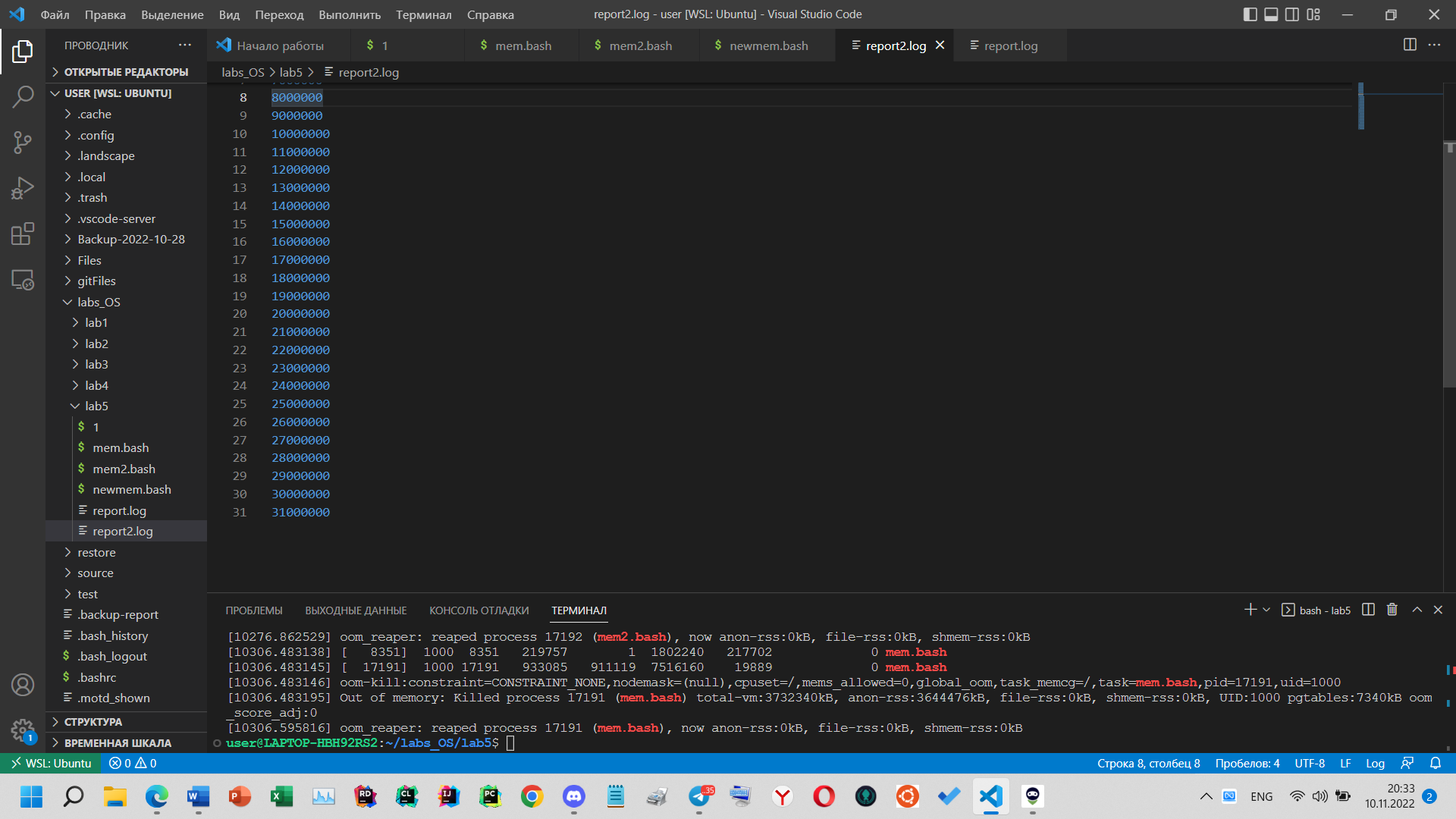


2 этап:

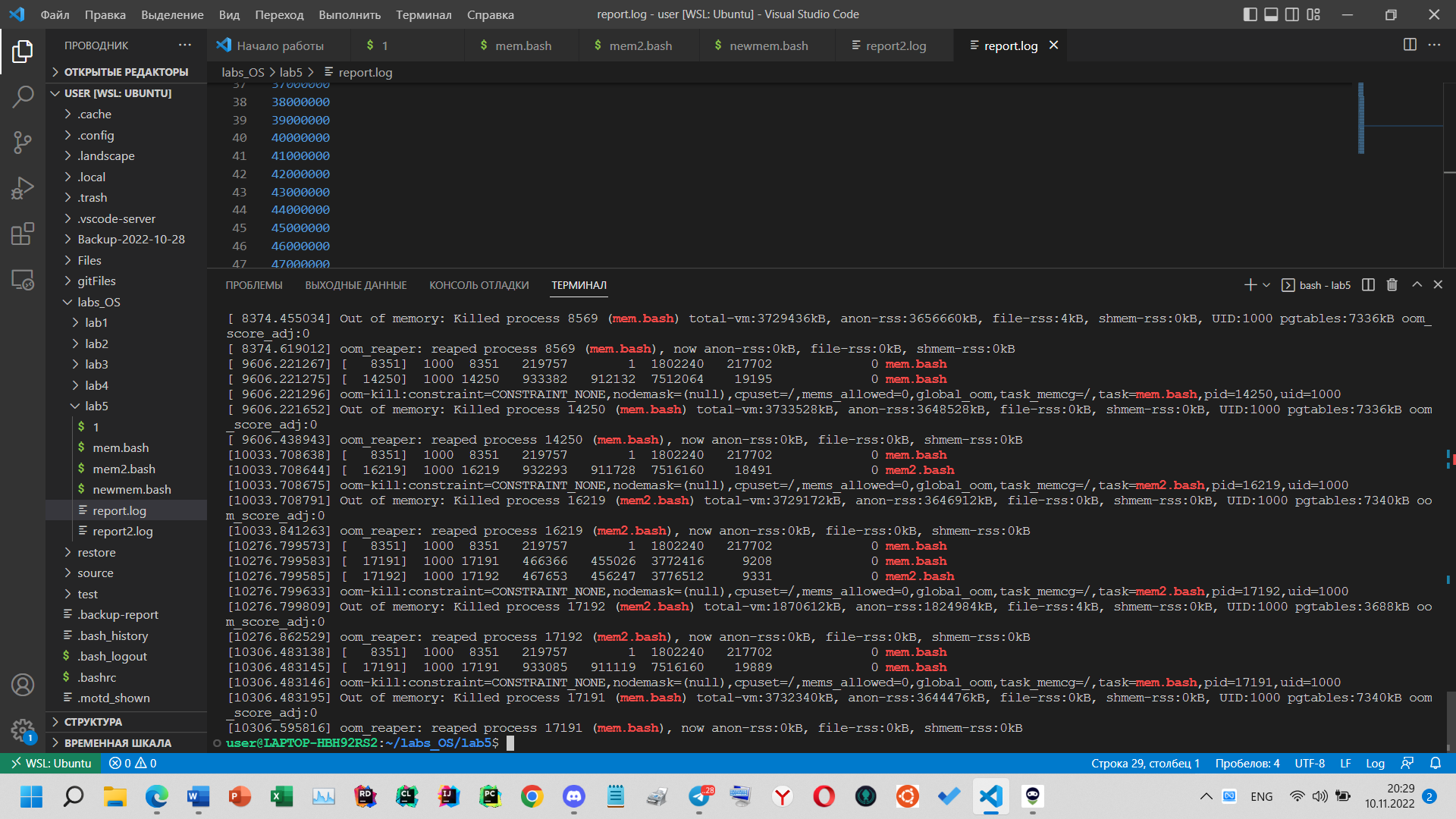
Записи в report.log



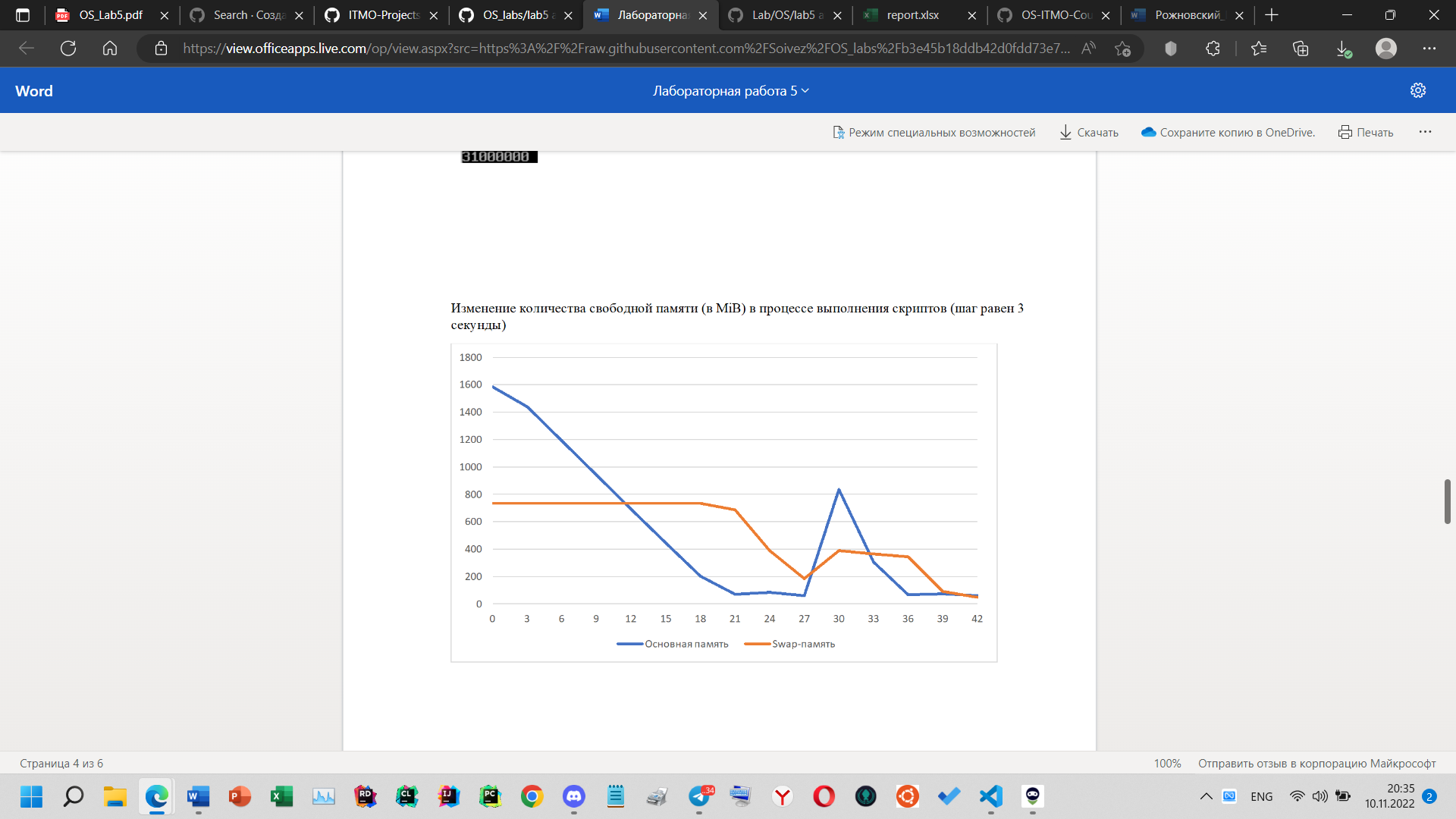
Записи в report2.log



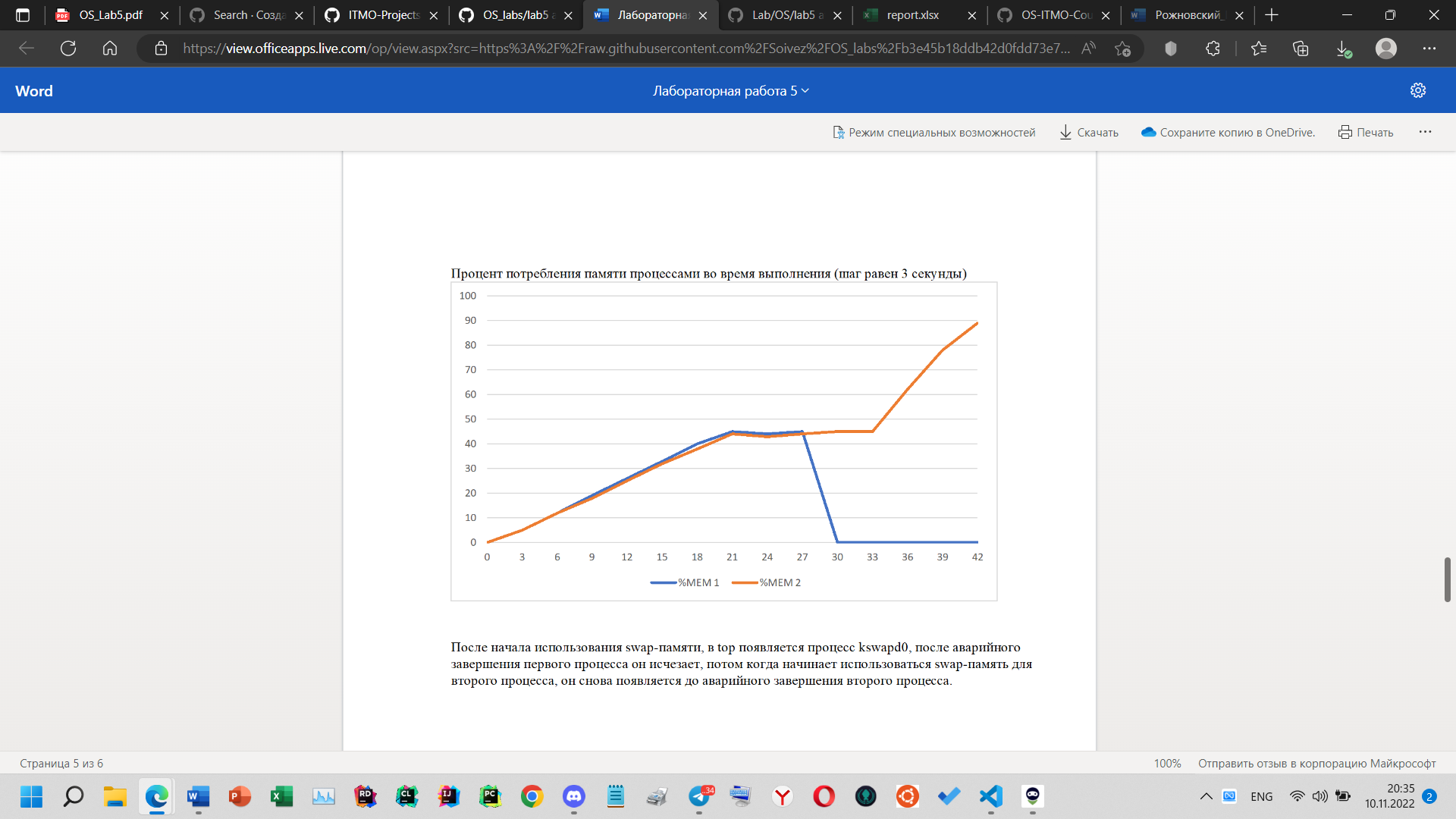
После переполнения памяти одновременно запущенными скриптами **mem.bash** и **mem2.bash** в логах  
с помощью команды **dmesg | grep "mem[2]\*.bash"** были получены записи:



Изменение количества свободной памяти (в MiB) в процессе выполнения скриптов (шаг равен 3 секунды)



Процент потребления памяти процессами во время выполнения (шаг равен 3 секунды)



2 эксперимент

Скрипт gen.bash получает аргументами K и N ($1 = K, $2 = N) и вызывает K раз newmem.bash с аргументом N.

При K = 10, N = 3100000 все запущенные скрипты успешно завершаются.

При K = 30, N = 3100000, сначала уменьшается свободная память, в момент, когда она начинает заканчиваться, начинают использоваться swap-память. В момент, когда количество свободного пространства подкачки опускается до примерно 100 MiB, скрипты начинают завершаться аварийно.

Подберём максимальное N:

При N = 1000000 все скрипты отрабатывают так быстро, что даже не успевают появиться в top с частотой обновления 1 сек.

При N = 3100000 аварийная остановка произошла примерно спустя 26 секунд после запуска gen.bash.

При N = 2000000 все скрипты отработали без аварийных остановок, причем swap-память не была использована.

При N = 2600000 не было аварийных остановок, но на пике оставалось всего 200 MiB swap-памяти.

При N = 2700000 не было аварийных остановок, свободная swap-память снизилась до 80MiB и произошла аварийная остановка

Получается, что максимальное N, при котором все скрипты отрабатывают без аварийных остановок – примерно 2600000. За счет распределения потребления памяти на большое количество процессов (запускаемых с небольшим интервалом) мы можем использовать значительно большее количество памяти, за счет оптимального распределения системой swap-памяти.