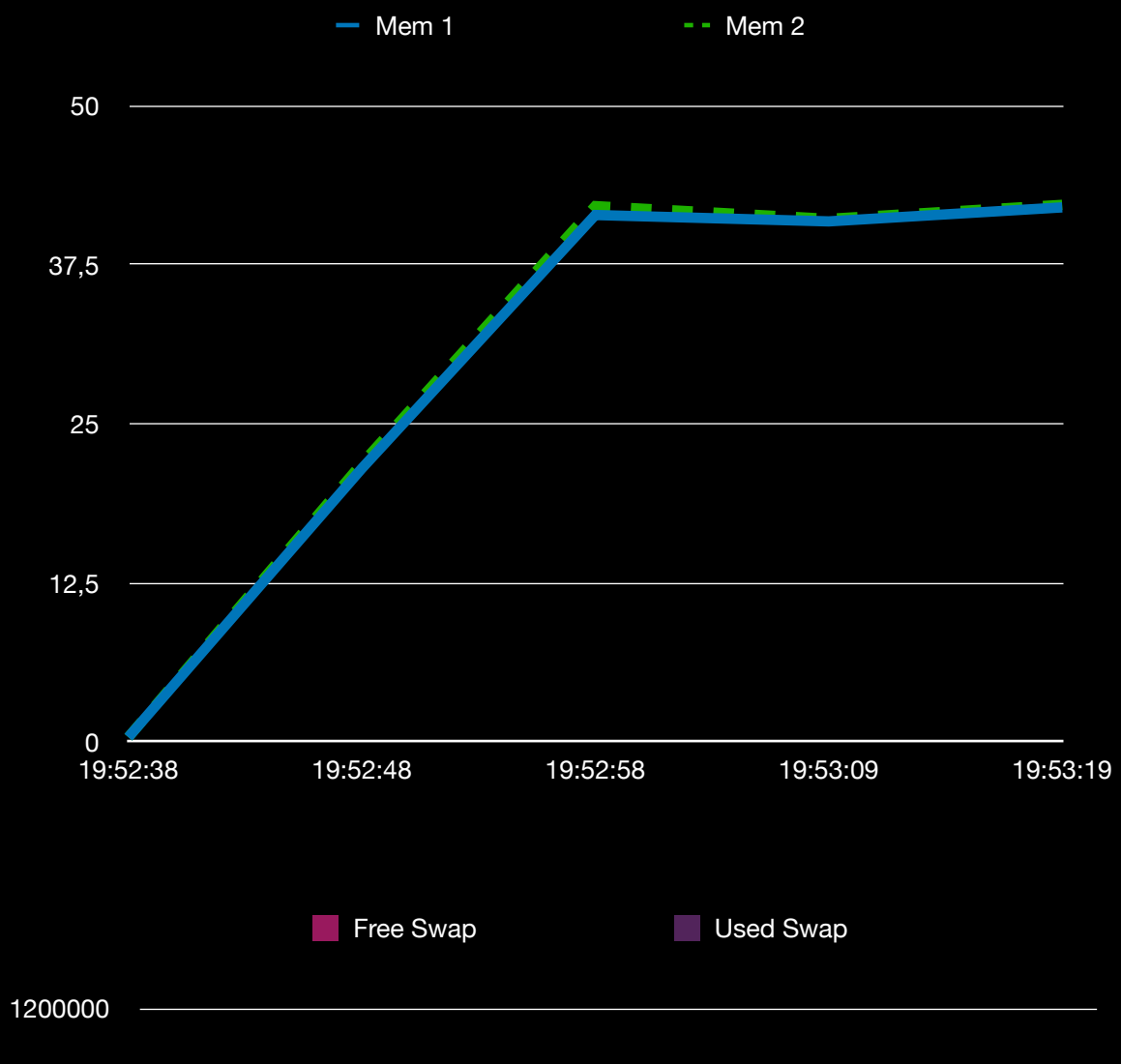
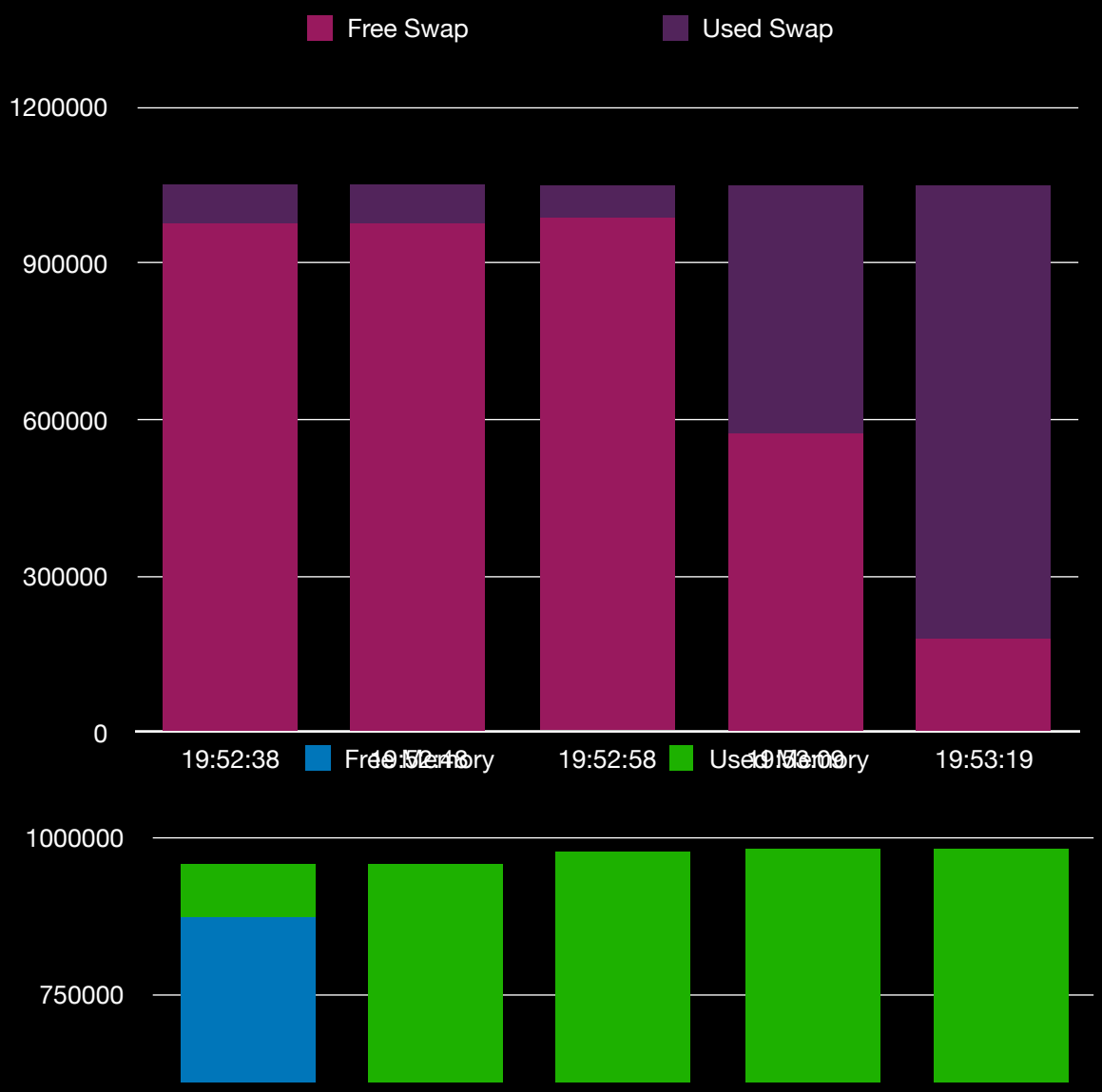
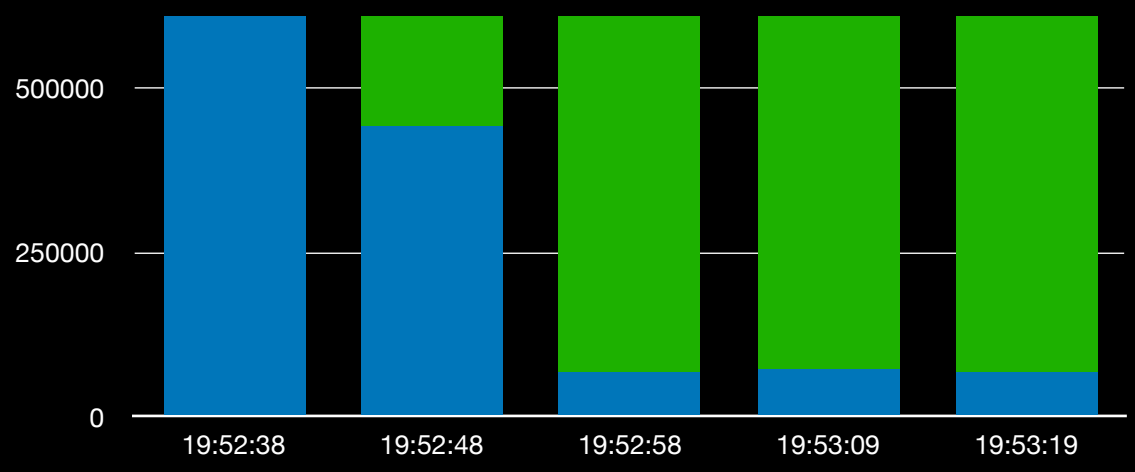
Data

Time/Memory	19:52:38	19:52:48	19:52:58	19:53:09	19:53:19
Mem 1	0,4	21,5	41,4	40,9	42,0
Mem 2	0,4	21,9	42,1	41,1	42,2
Total Memory	1006944	1006944	1006944	1006944	1006944
Total Swap	1048572	1048572	1048572	1048572	1048572
Free Memory	870896	441188	67612	73804	69668
Free Swap	978360	978360	987832	573880	178616
Used Memory	86700	516092	909172	909052	913224
Used Swap	70212	70212	60740	474692	869956







Общий объем оперативно памяти: 1006944 kB
Объем раздела подкачки: 1048572 kB
Размер страницы виртуальной памяти: 4096
Объем свободной физической памяти в ненагруженной системе: 238352
Объем свободного пространства в разделе подкачки в ненагруженной системе: 1048572 kB

report.log ---- 10000010

Достаточно ~40 секунд, чтобы процесс съел 80% памяти и погиб смертью храбрых.

При запуске двух одновременных скриптов из-за того, что мой bash использует xmalloc они убивают сами себя.

report.log ---- 5000010
report.log2 ---- 5000010

Использование памяти, как можно заметить в сумме ~ если работает один процесс.

При N = 1000001 и K = 10 процессы также запускаются, но процесс убивает сам себя когда достигается ~80% загрузки ОЗУ и он пытается запросить еще память.

При K = 30 ситуация аналогичная.

При K = 10 maxN ~ 1000001. Получено при помощи бинпоиска. При таком N процессы не успевают в один момент занять 80% памяти и умирают.

Вывод: в ситуации, когда ваша система использует xmalloc вы можете полностью на него положиться. Процессы будут уничтожаться сами по себе следуя принципу: “succeed or die”. Однако, никакой информации о том в каком состоянии процесс в логах вы не сможете увидеть, что безусловно является минусом. На моей машине поедание памяти процессами росло примерно одинаково, однако когда памяти не хватало, то был задействован Swap.

Из полезного: научился ручками выделять своп + перебилдил баш в попытках убрать xmalloc (не убрал)