

Отчёт по лабораторной работе №5

Михайлов Максим, М3237

25 ноября 2020 г.

Текущая конфигурация системы:

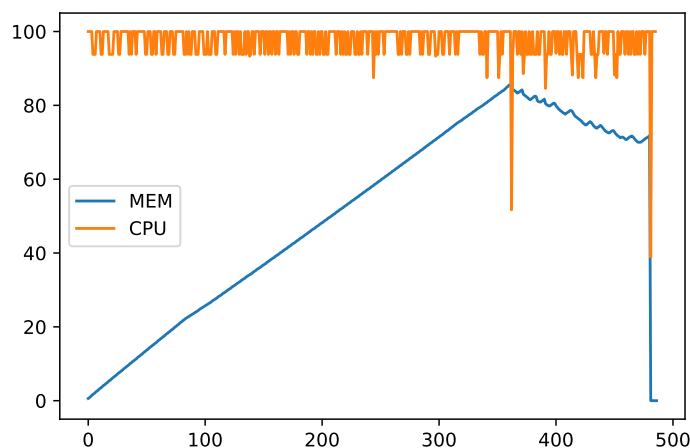
Общий объем оперативной памяти	16G
Объем раздела подкачки	9G
Размер страницы виртуальной памяти	4096B
Объем свободной физической памяти в ненагруженной системе	15141M
Объем свободного пространства в разделе подкачки в ненагруженной системе	9G

Эксперимент №1

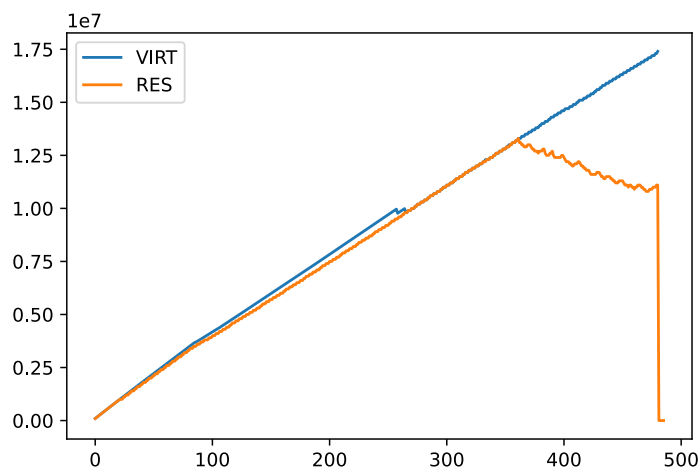
Первый этап

Значения параметров работающего скрипта

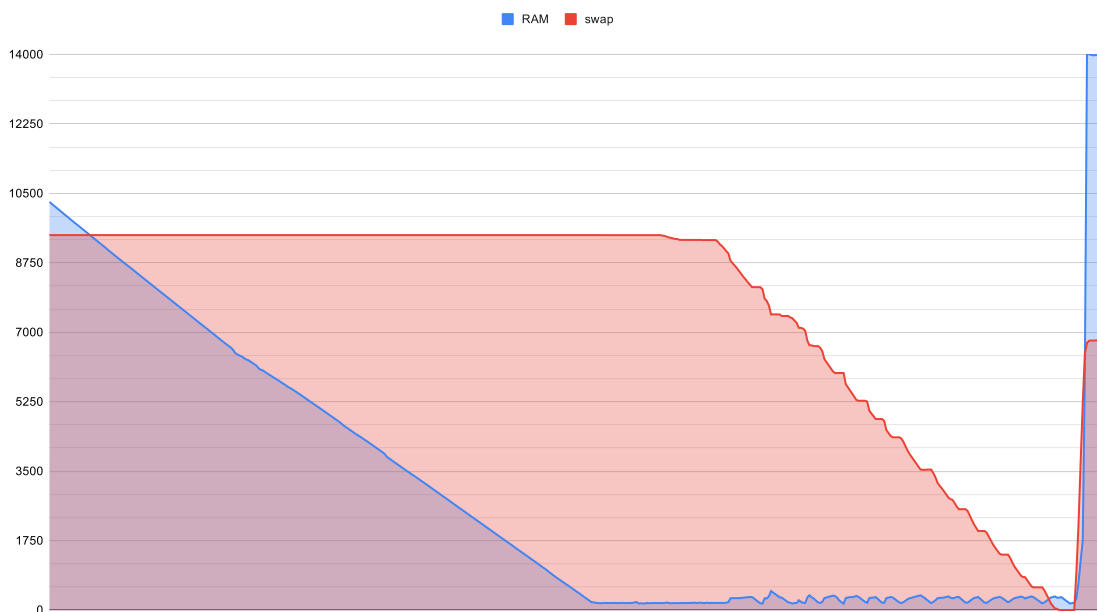
- PID, USER, PR, NI, SHR, S и COMMAND очевидно не изменялись
- TIME+ очевидно росло равномерно
- %MEM, %CPU:



- VIRT, RES:



Свободная память



Изменения в верхних пяти процессах

Изначально было 4 процесса code и 1 процесс brave, на пятой секунде процесс bash занял первое место и вытеснил brave. До остановки скрипта изменений не было, после остановки brave вернулся на пятую позицию.

Команда `dmesg | grep "mem.bash"` дала следующий вывод (красные стрелки обозначают перенос строки):

```
[ 8953.787883] [ 35459] 1000 35459 4016742 2881206 32239616 1133840 0
→ mem.bash
[ 8953.787893] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
→ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem.
→ bash,pid=35459,uid=1000
[ 8953.787902] Out of memory: Killed process 35459 (mem.bash) total-vm:16066968kB, anon-
→ rss:11524824kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:31484kB
```

```

    ↪ oom_score_adj:0
[ 8956.727222] oom_reaper: reaped process 35459 (mem.bash), now anon-rss:0kB, file-rss:0
    ↪ kB, shmem-rss:0kB

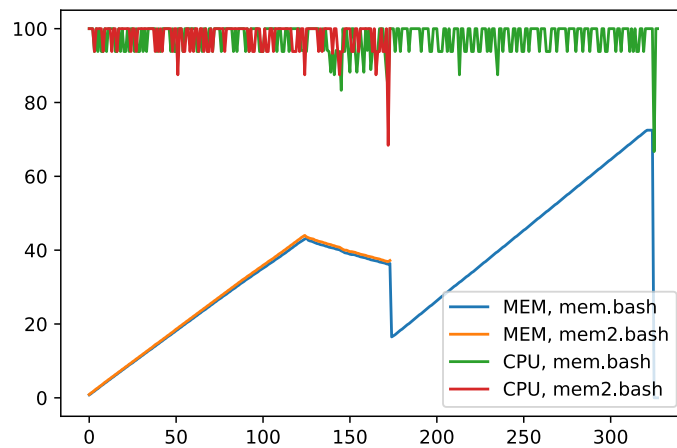
```

Значение последней строки в файле `report.log` — 233000000.

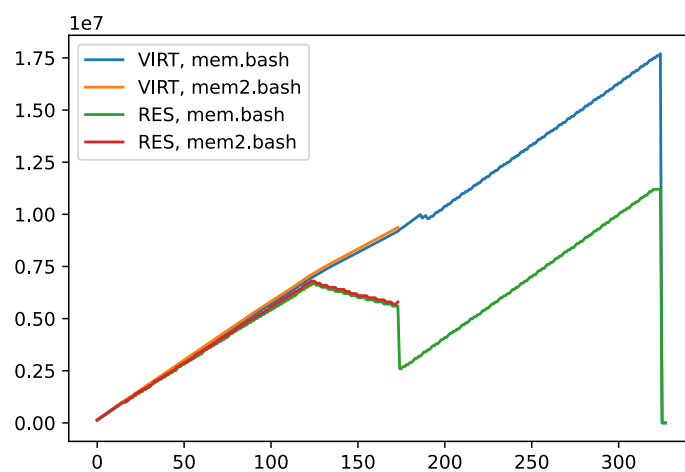
Второй этап

Значения параметров работающих скриптов

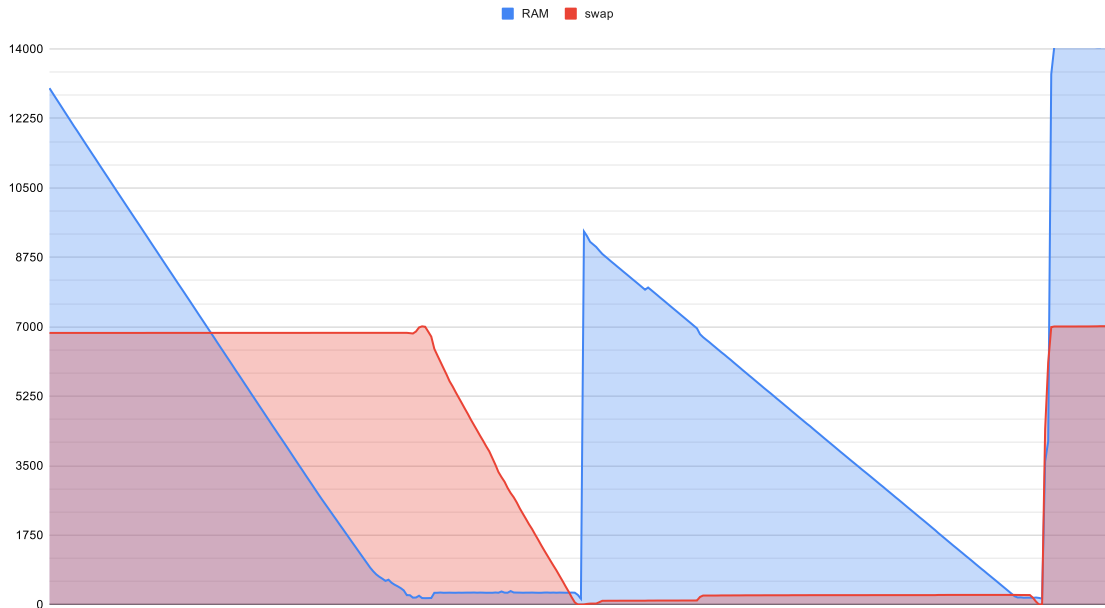
- PID, USER, PR, NI, SHR, S и COMMAND очевидно не изменялись
- TIME+ очевидно росло равномерно
- %MEM, %CPU:



- VIRT, RES:



Свободная память



Изменения в верхних пяти процессах

Изначально было 3 процесса code на первых трёх местах и mem2.bash, mem.bash на четвёртом и пятом местах соответственно. На четвёртой секунде mem2 и mem поднялись на первое и второе место соответственно. Изменений не было до остановки mem2, после чего mem поднялся на первое место, а пятое место занимали попеременно различные процессы. После аварийной остановки mem также вышел из первых пяти процессов.

Команда `dmesg | grep "mem[2]*.bash"` дала следующий вывод:

```
[ 8953.787883] [ 35459] 1000 35459 4016742 2881206 32239616 1133840 0
↳ mem.bash
[ 8953.787893] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
↳ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem.
↳ bash,pid=35459,uid=1000
[ 8953.787902] Out of memory: Killed process 35459 (mem.bash) total-vm:16066968kB, anon-
↳ rss:11524824kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:31484kB
↳ oom_score_adj:0
[ 8956.727222] oom_reaper: reaped process 35459 (mem.bash), now anon-rss:0kB, file-rss:0
↳ kB, shmem-rss:0kB
[11525.860040] [ 40777] 1000 40777 2094921 1431770 16834560 661470 0
↳ mem.bash
[11525.860042] [ 40778] 1000 40778 2080830 1418976 16719872 660169 0
↳ mem2.bash
[11525.860049] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
↳ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem.
↳ bash,pid=40777,uid=1000
[11525.860057] Out of memory: Killed process 40777 (mem.bash) total-vm:8379684kB, anon-
↳ rss:5727080kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:16440kB
↳ oom_score_adj:0
[11527.623645] oom_reaper: reaped process 40777 (mem.bash), now anon-rss:0kB, file-rss:0
↳ kB, shmem-rss:0kB
[11753.389686] [ 40778] 1000 40778 4215171 2756128 33828864 1457336 0
↳ mem2.bash
[11753.389699] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
↳ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem2.
↳ bash,pid=40778,uid=1000
```

```
[11753.389708] Out of memory: Killed process 40778 (mem2.bash) total-vm:16860684kB, anon-
↳ -rss:11024512kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:33036kB
↳ oom_score_adj:0
[11758.184627] oom_reaper: reaped process 40778 (mem2.bash), now anon-rss:0kB, file-rss
↳ :0kB, shmem-rss:0kB
[17130.429021] [ 48359] 1000 48359 2312160 1490357 18563072 820101 0
↳ mem.bash
[17130.429023] [ 48360] 1000 48360 2354136 1520009 18915328 832449 0
↳ mem2.bash
[17130.429033] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
↳ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem2.
↳ bash,pid=48360,uid=1000
[17130.429041] Out of memory: Killed process 48360 (mem2.bash) total-vm:9416544kB, anon-
↳ rss:6080036kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:18472kB
↳ oom_score_adj:0
[17133.115172] oom_reaper: reaped process 48360 (mem2.bash), now anon-rss:0kB, file-rss
↳ :0kB, shmem-rss:0kB
[18483.704393] [ 48359] 1000 48359 4655886 2955116 37351424 1699084 0
↳ mem.bash
[18483.704415] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed
↳ =0,global_oom,task_memcg=/user.slice/user-1000.slice/session-2.scope,task=mem.
↳ bash,pid=48359,uid=1000
[18483.704423] Out of memory: Killed process 48359 (mem.bash) total-vm:18623544kB, anon-
↳ rss:11820464kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:36476kB
↳ oom_score_adj:0
[18487.884211] oom_reaper: reaped process 48359 (mem.bash), now anon-rss:0kB, file-rss:0
↳ kB, shmem-rss:0kB
```

Значение последней строки в файле report.log — 238000000, в report2.log — 120000000.

Объяснение результатов

- %CPU процессов высоко, т.к. процесс ожидает только выделения дополнительной памяти. Это же объясняет неполную загрузку процессора процессами и падение загрузки перед аварийной остановкой.
- %MEM обозначает используемый объём *физической* памяти, поэтому он сначала растёт. При использовании файла подкачки %MEM падает, т.к. часть страниц памяти, используемых процессом, переносится в swap. Это происходит скачкообразно, т.к. страницы переносятся не по одной. Аналогичное верно про RES.
- В эксперименте с двумя процессами %MEM и RES первого (*долгоживущего*) процесса не уменьшаются перед остановкой процесса, т.к. почти весь swap уже занят.
- В случае запуска одного скрипта количество элементов массива при аварийной остановке $\approx 2.38 \cdot 10^8$, т.к. $1.75 \cdot 10^9$ (приблизительно объём используемой памяти в байтах) / $2.38 \cdot 10^8 \approx 8$ байт. bash использует числовые переменные размером 64 бит.
- При запуске двух скриптов второй скрипт останавливается при использовании примерно половины от $2.38 \cdot 10^8$ элементов, т.к. первый скрипт использует столько же и в сумме используется вся доступная память. После остановки второго скрипта первый останавливается при $2.38 \cdot 10^8$ элементов, т.к. теперь память использует только он и фоновые процессы.

Выводы: было проанализировано поведение механизмов управления памятью OS Linux при растущем использовании памяти. Теоретические сведения были подтверждены двумя экспериментами.

Эксперимент №2

При K=30, N=238000000 ряд процессов закончился аварийно, т.к. процессы требуют суммарно приблизительно в 3 раза больше оперативной памяти, чем доступно.

Максимальное значение N , такое что при $K=30$ не происходило аварийных остановок процессов — $9 \cdot 10^7$. Оно отличается от ожидаемого значения — $2.38 \cdot 10^9/30 \approx 8 \cdot 10^7$, т.к. процессы не работают синхронно и некоторые достигают штатного завершения раньше других, пока оперативной памяти хватает.