

ПРОВЕДЕНИЕ ЭКСПЕРЕМЕНТОВ:

- Общий объем оперативной памяти – 3.8 (Gb);
- Объем раздела подкачки – 3.8 (Gb);
- Размер страницы виртуальной памяти – 4096 (4 КБ);
- Объем свободной физической памяти в ненагруженной системе – 3.2 (Gb);
- Объем свободного пространства в разделе подкачки в ненагруженной системе – 3.3 (Gb).

Записи скрипт mem.bash. Вывод последней строки в файл report.log – “ Array size: 89000000”. То есть скрипт завершил аварийно работу, когда размер массива был между 89000000 и 90000000.

Когда программа завершила работу в консоль вывелось слово «Killed», что означает, что программа завершилась.

Первый Этап:

После завершения скрипта и использования команды для просмотра журнала была такая информация:

```
sudo dmesg | grep "mem.bash"
```

```
[ 623.470528] [ 4239] 1000 4239 1767490 880960 880960 0 0 14209024
884096 200 mem.bash
```

```
[ 623.470562] oom-
```

```
kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom
,task_memcg=/user.slice/user-1000.slice/user@1000.service/app.slice/app-
org.gnome.Terminal.slice/vte-spawn-e736e3ab-4a6a-48c8-8787-
59652be0b119.scope,task=mem.bash,pid=4239,uid=1000
```

```
[ 623.470616] Out of memory: Killed process 4239 (mem.bash) total-vm:7069960kB, anon-
rss:3523840kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:13876kB oom_score_adj:200
```

```
[ 626.824690] oom_reaper: reaped process 4239 (mem.bash), now anon-rss:52kB, file-rss:0kB,
shmem-rss:0kB
```

-
- [623.470528]:
 - Это метка времени, которая показывает, через сколько секунд после загрузки системы произошло событие (в секундах).
 - [4239]:
 - Идентификатор процесса (PID).
 - 1000:
 - Идентификатор пользователя (UID), от имени которого запущен процесс.
 - 4239:
 - Снова PID процесса.
 - 1767490:
 - Общий объем виртуальной памяти, используемой процессом, в страницах. Если размер страницы равен 4 КБ, то общий объем виртуальной памяти:

$$1767490 * 4 \text{ КБ} = 7069960 \text{ КБ} = 7069.96 \text{ МБ} = \sim 7 \text{ ГБ}$$

- **880960:**
 - Объем анонимной памяти (anon-rss), используемой процессом, в страницах. Это память, которая не связана с файлами.
$$880960 * 4 \text{ КБ} = 3523840 \text{ КБ} = 3523.84 \text{ МБ} = \sim 3.5 \text{ ГБ}$$
 - **880960:**
 - Объем памяти, связанной с файлами (file-rss), в страницах. В данном случае это 0 КБ, так как процесс использует только анонимную память.
 - **0:**
 - Объем разделяемой памяти (shmem-rss), в страницах
 - **14209024:**
 - Общий объем памяти, выделенной процессу, в страницах.
 - **884096:**
 - Объем памяти, используемой процессом, в страницах.
 - **200:**
 - Значение `oom_score_adj`, которое влияет на приоритет процесса при выборе жертвы для завершения из-за нехватки памяти. Значение 200 означает, что процесс имеет высокий приоритет для завершения.
 - **mem.bash:**
 - Имя процесса, который был завершен.
-

2. Вторая строка:

```
[ 623.470562] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-1000.slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-spawn-e736e3ab-4a6a-48c8-8787-59652be0b119.scope,task=mem.bash,pid=4239,uid=1000
```

- **oom-kill:**
 - Указывает, что произошло завершение процесса из-за нехватки памяти (Out of Memory Killer).
- **constraint=CONSTRAINT_NONE:**
 - Указывает, что процесс не ограничен никакими конкретными ограничениями памяти.
- **nodemask=(null):**
 - Указывает, что процесс не ограничен конкретными узлами в многоузловой системе.
- **cpuset=/:**
 - Указывает, что процесс не ограничен конкретными процессорами.
- **mems_allowed=0:**
 - Указывает, что процесс не ограничен конкретными узлами памяти.
- **global_oom:**
 - Указывает, что процесс был завершен из-за глобальной нехватки памяти в системе.
- **task_memcg=/...:**
 - Указывает группу управления памятью (Memory Control Group), к которой принадлежит процесс. В данном случае это группа, связанная с терминалом GNOME.
- **task=mem.bash:**

- Имя процесса, который был завершен.
 - **pid=4239:**
 - Идентификатор процесса, который был завершен.
 - **uid=1000:**
 - Идентификатор пользователя, от имени которого запущен процесс.
-

3. Третья строка:

```
[ 623.470616] Out of memory: Killed process 4239 (mem.bash) total-vm:7069960kB, anon-rss:3523840kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:13876kB oom_score_adj:200
```

- **Out of memory:**
 - Указывает, что процесс был завершен из-за нехватки памяти.
 - **Killed process 4239 (mem.bash):**
 - Идентификатор процесса и его имя.
 - **total-vm:7069960kB:**
 - Общий объем виртуальной памяти, используемой процессом, в килобайтах.
 - **anon-rss:3523840kB:**
 - Объем анонимной памяти, используемой процессом, в килобайтах.
 - **file-rss:0kB:**
 - Объем памяти, связанной с файлами, в килобайтах.
 - **shmem-rss:0kB:**
 - Объем разделяемой памяти, в килобайтах.
 - **UID:1000:**
 - Идентификатор пользователя, от имени которого запущен процесс.
 - **pgtables:13876kB:**
 - Объем памяти, используемой для таблиц страниц процесса, в килобайтах.
 - **oom_score_adj:200:**
 - Значение `oom_score_adj`, которое влияет на приоритет процесса при выборе жертвы для завершения.
-

4. Четвертая строка:

```
[ 626.824690] oom_reaper: reaped process 4239 (mem.bash), now anon-rss:52kB, file-rss:0kB, shmem-rss:0kB
```

- **oom_reaper:**
 - Указывает, что процесс был "почищен" (reaped) после завершения. Это означает, что система освободила ресурсы, связанные с процессом.
- **reaped process 4239 (mem.bash):**
 - Идентификатор процесса и его имя.
- **anon-rss:52kB:**
 - Объем анонимной памяти, оставшейся после завершения процесса.
- **file-rss:0kB:**
 - Объем памяти, связанной с файлами, оставшейся после завершения процесса.
- **shmem-rss:0kB:**

- Объем разделяемой памяти, оставшейся после завершения процесса.

```
[ 623.470616] Out of memory: Killed process 4239 (mem.bash) total-vm:7069960kB, anon-  
rss:3523840kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:13876kB oom_score_adj:200
```

**Длина массива на момент остановки скрипта (из файла report.log): Array size:
90000000.**

О варьировании 5-ти верхних процессов в команде top можно посмотреть в файле top_data.log.

Во время выполнения скриптов mem.bash, mem2.bash они не всегда находились на самых верхних строках при просмотре команды top. Это значит, что они не все время задействовали CPU больше чем остальные процессы. Основная работа этих процессов — это выделение памяти и добавление элементов в массив, что не является CPU-интенсивной операцией.

Тенденцию задействования ресурсов можно увидеть на графиках (mem_first_grafics, ram_swap_first_grafics). Объяснение такое же, как и у двух процессов. Об этом ниже.

Этап 2

Запустив два скрипта, я получил вот такие результаты:

```
sudo dmesg | grep "mem.bash"
```

```
[ 1572.268772] [ 4380] 1000 4380 872398 431520 431488 32 0 7036928 438464  
200 mem.bash
```

```
[ 1795.188667] [ 4380] 1000 4380 1757425 871424 871424 0 0 14131200 883552  
200 mem.bash
```

```
[ 1795.188678] oom-  
kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_me  
mcg=/user.slice/user-1000.slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-spawn-  
a933b1a8-cbbf-4ee1-bd3b-9c2999698e9e.scope,task=mem.bash,pid=4380,uid=1000
```

```
[ 1795.188727] Out of memory: Killed process 4380 (mem.bash) total-vm:7029700kB, anon-  
rss:3485696kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:13800kB oom_score_adj:200
```

```
sudo dmesg | grep "mem2.bash"
```

```
[ 1572.268775] [ 4381] 1000 4381 889129 444384 444352 32 0 7168000 442400  
200 mem2.bash
```

```
[ 1572.268860] oom-  
kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_me
```

mcg=/user.slice/user-1000.slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-spawn-a933b1a8-cbbf-4ee1-bd3b-9c2999698e9e.scope,task=mem2.bash,pid=4381,uid=1000

[1572.269099] Out of memory: Killed process 4381 (mem2.bash) total-vm:3556516kB, anon-rss:1777408kB, file-rss:128kB, shmem-rss:0kB, UID:1000 pgtables:7000kB oom_score_adj:200

[1574.797446] oom_reaper: reaped process 4381 (mem2.bash), now anon-rss:148kB, file-rss:128kB, shmem-rss:0kB

Выводы:

Общий объем виртуальной памяти (`total-vm`) для `mem.bash` был примерно в два раза больше, чем для `mem2.bash` (7 ГБ против 3.5 ГБ).

• Механизм OOM Killer:

- Процессы были завершены из-за нехватки памяти.
- Оба процесса имели высокий приоритет для завершения (`oom_score_adj:200`), что указывает на их высокую вероятность быть завершенными.

• Раздел подкачки:

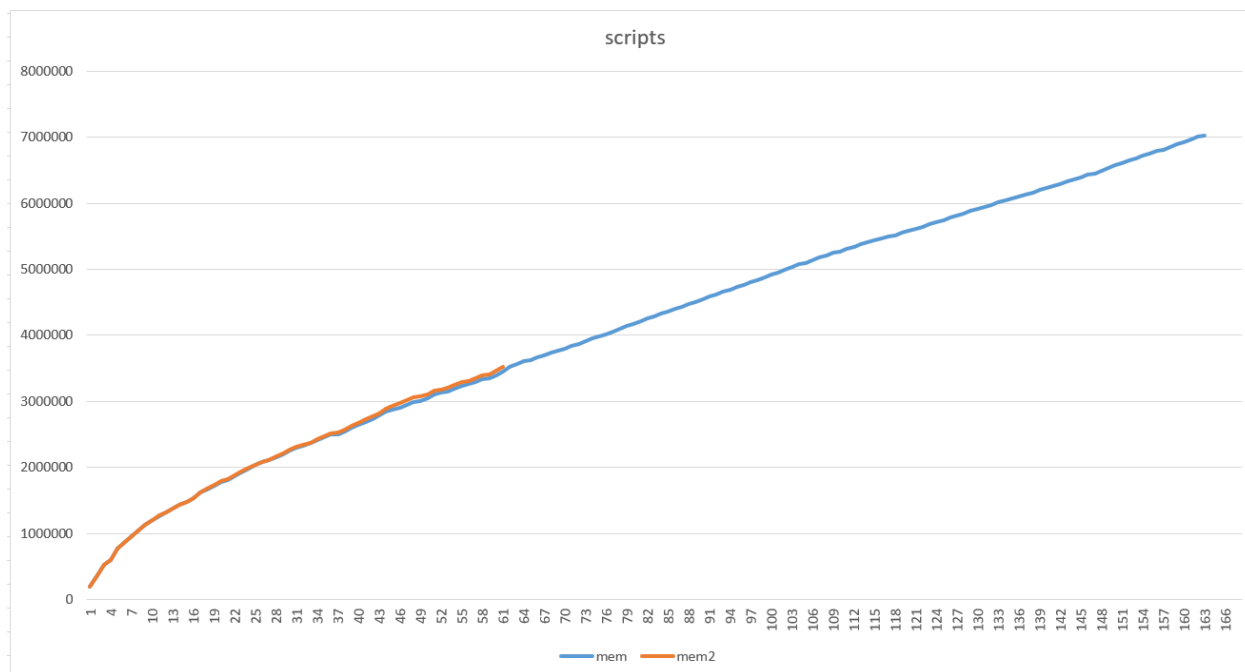
- В данном случае раздел подкачки не использовался, так как объем анонимной памяти (`anon-rss`) был значительным, и система не смогла вытеснить страницы в `swap`.

• Пороговые значения:

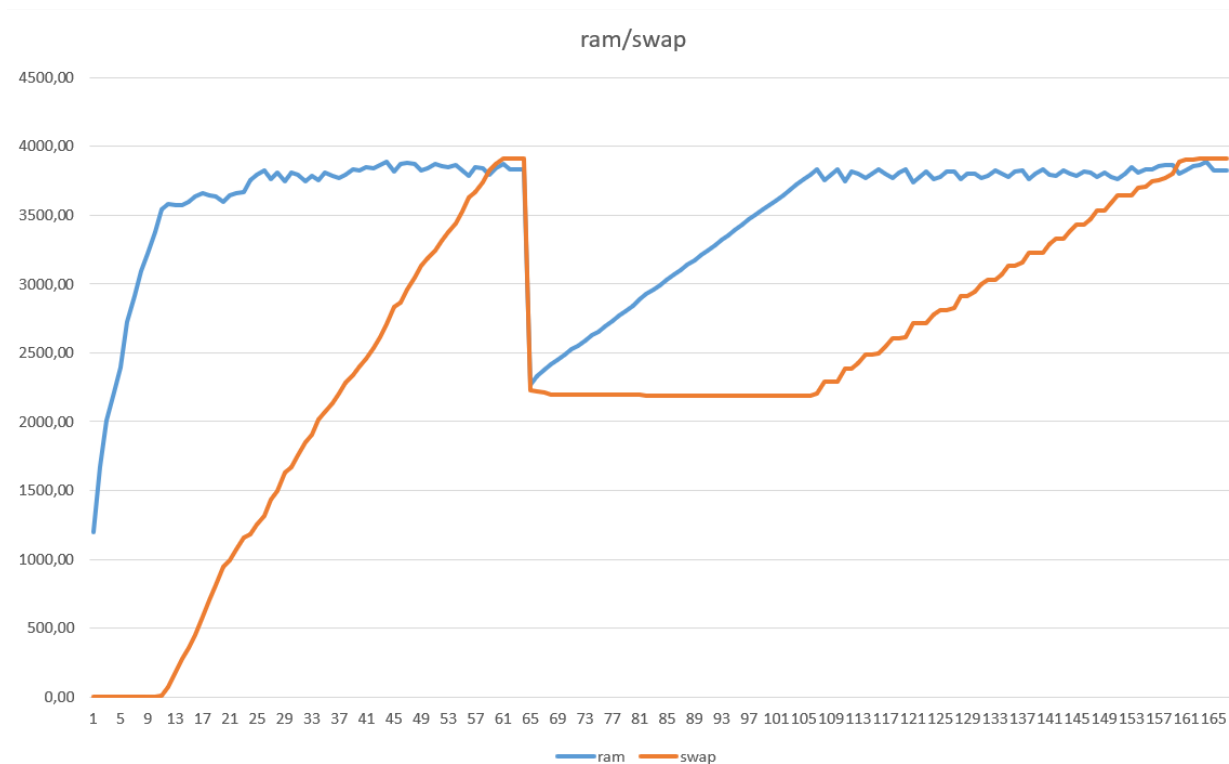
- Процесс `mem2.bash` был завершен при использовании ~3.5 ГБ виртуальной памяти и ~1.7 ГБ анонимной памяти.
- Процесс `mem.bash` был завершен при использовании ~7 ГБ виртуальной памяти и ~3.5 ГБ анонимной памяти.

Оба процесса были запущены одновременно и стали занимать какую-то память. Когда они в сумме стали занимать практически всё (7 Гб на двоих (по 3.5 на каждого)), то один из процессов аварийно завершился. И память, которую он занимал, освободилась. И второй процесс смог работать дальше. И когда уже он занял практически всё, то и его отключили. Но в тот момент не было кандидата, который тоже занимал часть из этих 7 Гб. Поэтому последний процесс смог занять все 7 Гб.

На графиках можно проследить тенденцию изменения величин. На графике `scripts` видно, что оба процесса одинаково забирают память. Пока они оба не стали занимать практически всё. Один из них завершился и второй продолжил занимать память, которая освободилась.



На графике ram/swap можно проследить сколько использовалось каждой величины. Пока мог использоваться ram, он и использовался. Когда уже его стало не хватать (использовался больше 3.500) стал использоваться swap. Он активно работал и когда они уже оба были на пределе, то произошло завершение одного из скриптов и значение ram и swap уменьшилось. Потом ram снова рос. Когда он снова достигал пиковой точки, то подключился в работу swap (стал занимать больше памяти). И когда они оба уже были на пределе, то скрипт аварийно завершился.



Эксперимент 2

- При большом количестве запусков ($K=30$) и относительно большом значении N каждый запуск `newmem.bash` выделяет значительный объем памяти.
- Если общий объем выделенной памяти превышает доступную оперативную память и раздел подкачки, OOM Killer начинает завершать процессы.

Написав скрипт и экспериментально вычисляя, было подобрано наибольшее значение N , при котором все 30 запусков скрипта `newmem.bash` проходят успешно. Это при $N=3600000$. Такой вывод можно сделать, потому что часть памяти, занятая одним из процессов, успела освободиться до того, как очередному процессу понадобилась она.