

Подготовка:

Перезапустим виртуальную машину, и посмотрим нужные данные:

- Общий объем оперативной памяти - **1870900 Kб**
- Объем раздела подкачки - **839676 Kб**
- Размер страницы виртуальной памяти - **4096 Kб**
- Объем свободной физической памяти в ненагруженной системе - **1466656 Kб**
- Объем свободного пространства в разделе подкачки в ненагруженной системе - **839676 Kб**

MemTotal:	1870900 kB	SwapCached:	0 kB
MemFree:	1423232 kB	SwapTotal:	839676 kB
MemAvailable:	1543676 kB	SwapFree:	839676 kB

Эксперимент №1

Подготовительный этап:

Скрипт mem.sh:

```
#!/bin/bash
reportFile="/home/user/lab5/report.log"

if [[ -z reportFile ]]
then
    touch reportFile
fi

: > $reportFile

array=()
sequence=(1 2 3 4 5 6 7 8 9 10)
step=100000
for (( i = 1; i > 0; i++ ))
do
    array+=($sequence)
    if [[ "$i" == "$step" ]]
    then
        i=0
        echo "${#array[@]} " >> $reportFile
    fi
done
```

```
#!/bin/bash
reportFile="/home/user/lab5/report2.log"

if [[ -z reportFile ]]
then
    touch reportFile
fi

: > $reportFile

array=()
sequence=(1 2 3 4 5 6 7 8 9 10)
step=100000
for (( i = 1; i > 0; i++ ))
do
    array+=($sequence)
    if [[ "$i" == "$step" ]]
    then
        i=0
        echo "${#array[@]} " >> $reportFile
    fi
done
```

Состояние каждого из отслаживаемых параметров было решено фиксировать раз в 10 секунд с помощью видео.

Первый этап:

Запустим скрипт и будем наблюдать за изменением:

- свободной оперативки (Free Mem %)
- свободного файла подкачки (Free Swap %)
- используемой скриптом памяти (Script mem %)

График изменения данных изображен на рис. 1.

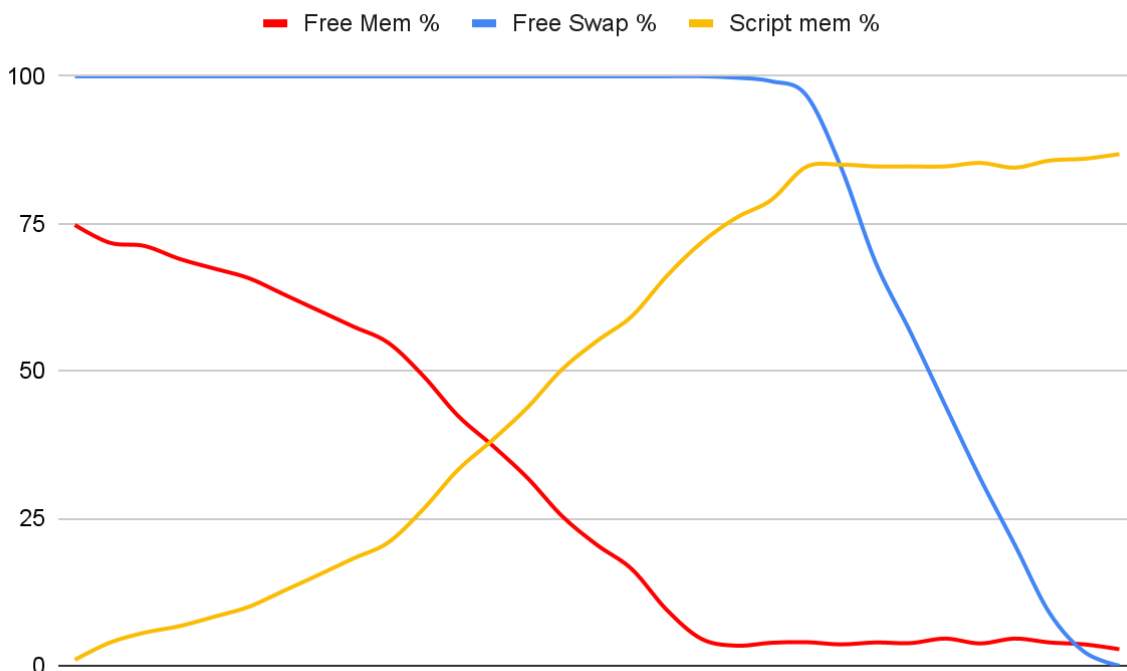


Рис. 1 - Изменения в процессе выполнения 1ого этапа

Анализ графика:

1. Пока хватает оперативки - используется только оперативка.
2. Стало не хватать оперативки - подключается файл подкачки.
3. Стало не хватать и оперативки, и файла подкачки - убиваем процесс.

Последнее значение в report.log:

```
[root@localhost lab5]# cat report.log | tail -1
308000000
```

Записи в журнале:

```
[ 2309.183191] 0 1175 036318 108113 1030818 130323 0 mem
[ 2309.186609] Out of memory: Killed process 1473 (mem) total-vm:2633360kB, anon-rss:1625772kB, file-rss:0kB, shmem-rss:0kB, UID:0
[ 2309.647228] oom_reaper: reaped process 1473 (mem), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
[root@localhost lab5]#
```

Второй этап:

Запустим скрипты mem и mem2 и будем наблюдать за изменением:

- свободной оперативки (Free Mem %).
- свободного файла подкачки (Free Swap %).
- используемой скриптом mem памяти (Script mem %).
- используемой скриптом mem2 памяти (Script mem2 %).

График изменения данных изображен на рис. 2.

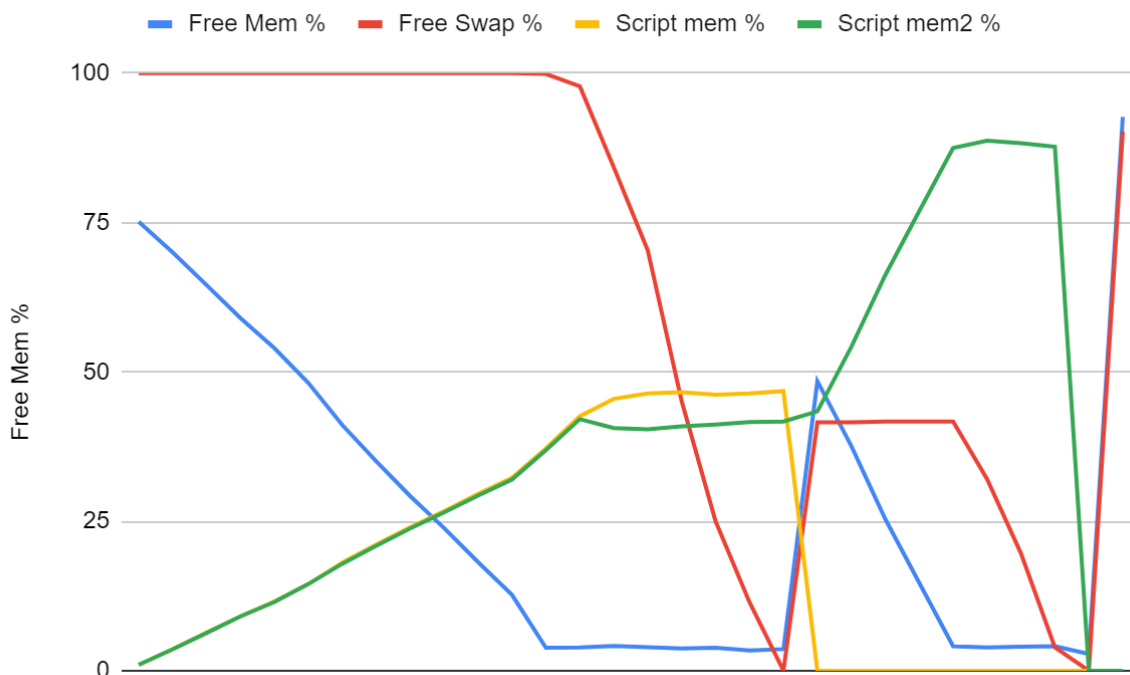


Рис. 2 - Изменения в процессе выполнения 2ого этапа

Анализ графика:

1. Пока хватает оперативки - используется только оперативка (сразу на два скрипта).
2. Стало не хватать оперативки - начинаем использовать файл подкачки (причем приоритет отдается 1ому скрипту).
3. Стало не хватать и оперативки, и файла подкачки - убиваем процесс mem.
4. Память освобождается, используем ее для процесса mem2.
5. Когда начинает не хватать памяти и файла подкачки на процесс mem2 - убиваем и его.

Последние значения в report.log и report2.log :

```
[root@localhost lab5]# cat report.log | tail -1  
15600000
```

```
[root@localhost lab5]# cat report2.log | tail -1  
31000000
```

Размер первого массива в конце работы скрипта составил примерно ½ от второго.

Записи в журнале:

```
[ 5357.332330] Out of memory: Killed process 1484 (mem) total-vm:1441664kB, anon-rss:869740kB, file-  
rss:16kB, shmem-rss:0kB, UID:0  
[ 5358.704183] oom_reaper: reaped process 1484 (mem), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB  
[ 5594.587196] Out of memory: Killed process 1485 (mem2) total-vm:2645372kB, anon-rss:1665116kB, fil  
e-rss:0kB, shmem-rss:0kB, UID:0  
[ 5594.769157] oom_reaper: reaped process 1485 (mem2), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```

Эксперимент №2

Подготовительный этап:

```
#!/bin/bash  
  
array=()  
sequence=(1 2 3 4 5 6 7 8 9 10)  
step=$1  
  
while [[ true ]]  
do  
    array+=($sequence)  
    if [[ ${#array[@]} -gt $step ]]  
    then  
        echo "Success: array size is $1"  
        exit  
    fi  
done
```

```
#!/bin/bash  
  
n=$1  
k=$2  
  
for (( i = 0; i < $k; i++ ))  
do  
    ./newmem $n &  
    sleep 1  
done
```

Основной этап:

Для N = 3100000, K = 10

- Ничего не ломается, т.к. общий размер всех массивов не превысит “порогового” значения.

Для N = 3100000, K = 30

- Уже некоторые процессы ломаются

- Это происходит потому что в какой-то момент времени общий размер массивов превышает “пороговое значение”, и системе приходится убивать некоторые процессы.

Очевидно, что нужно подобрать такое N , что в один момент времени общий размер всех массивов не превышал пороговое значение.

В результате упрощенного подбора оптимального значения с помощью бин. поиска, получаем примерное наибольшее значение $N = 1400000$.

Благодаря интервалу в 1 секунду ранее запущенные процессы успевают закончиться и освободить место под следующие, из-за чего системе не приходится убивать процессы.