

Данные о текущей конфигурации операционной системы в аспекте управления памятью:

- Общий объем оперативной памяти.
- Объем раздела подкачки.
- Объем свободной физической памяти в ненагруженной системе.
- Объем свободного пространства в разделе подкачки в ненагруженной системе.
- Размер страницы виртуальной памяти.

	total	used	free	shared	buff/cache	available
Mem:	1870900	172372	1430720	8948	267808	1542704
Swap:	839676	0	839676			

```
[root@localhost lab5]# getconf PAGESIZE
4096
```

Эксперимент №1

Первый этап:

Запущенный скрипт:

```
#!/bin/bash

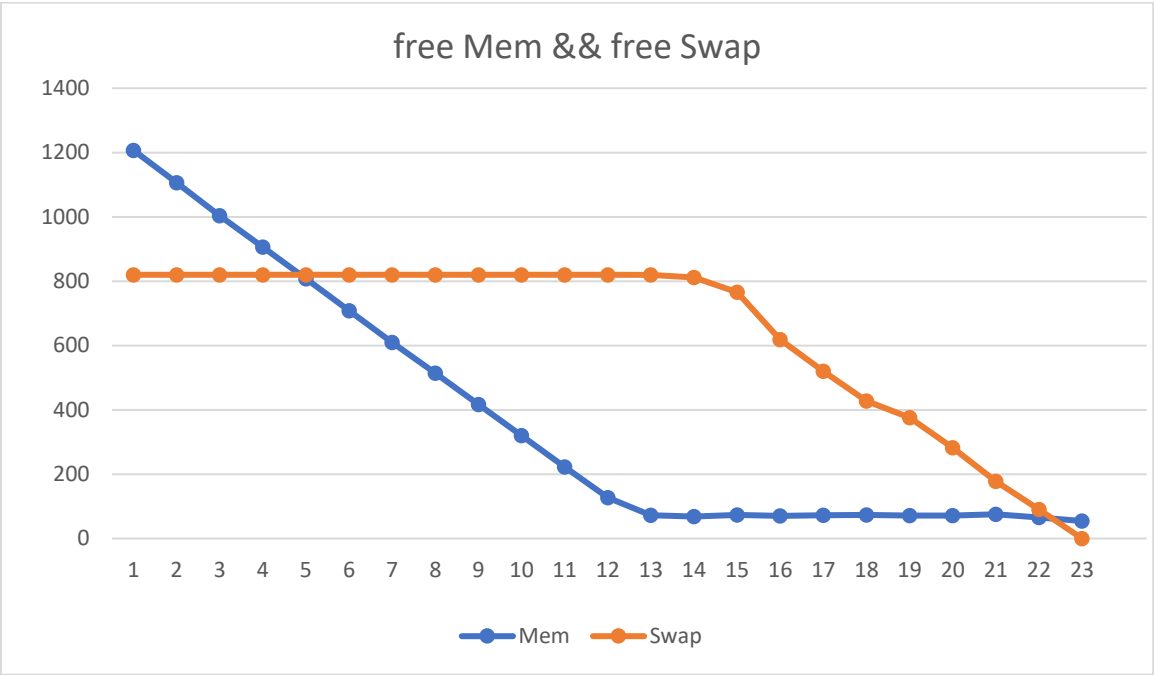
cnt=0
arr=()
while true
do
    arr+=(1 2 3 4 5 6 7 8 9 10)
    cnt=$((cnt + 1))
    if [[ $cnt -eq 100000 ]]
    then
        echo ${#arr[@]} > report.log
        cnt=0
    fi
done
```

Последняя запись в файле report.log (размер массива):

```
[root@localhost lab5]# cat report.log
28000000
```

Последние две записи о скрипте в системном журнале:

```
[ 339.336619] Out of memory: Killed process 1503 (mem.sh) total-vm:2435336kB, anon-rss:1595168kB, f
ile-rss:0kB, shmem-rss:0kB, UID:0
[ 339.441300] oom_reaper: reaped process 1503 (mem.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0
kB
```



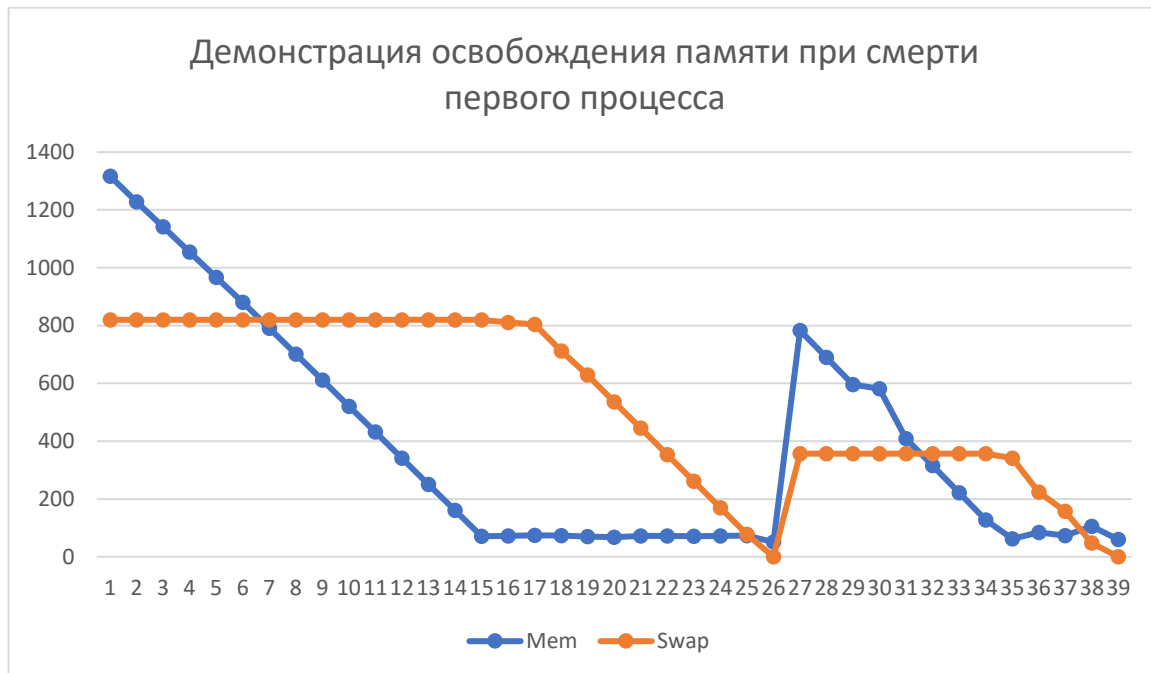
Второй этап:

Последняя запись в файлах report.log report2.log (размеры массивов):

```
[root@localhost lab5]# cat report.log
15000000
[root@localhost lab5]# cat report2.log
31000000
```

Последние записи о скриптах в системном журнале:

```
[root@localhost lab5]# dmesg | grep "mem[2]*.sh"
[ 226.411230] mem.sh invoked oom-killer: gfp_mask=0x6200ca(GFP_HIGHUSER_MOVABLE), nodemask=(null),
order=0, oom_score_adj=0
[ 226.412031] mem.sh cpuset=/ mems_allowed=0
[ 226.412266] CPU: 0 PID: 1496 Comm: mem.sh Kdump: loaded Tainted: G
4.18.0-193.el8.x86_64 #1
[ 226.454219] [ 1496] 0 1496 361763 211485 2527232 94727 0 mem.sh
[ 226.454526] [ 1497] 0 1497 359882 210117 2510848 94207 0 mem2.sh
[ 226.454835] Out of memory: Killed process 1496 (mem.sh) total-vm:1447052kB, anon-rss:845940kB, fi
le-rss:0kB, shmem-rss:0kB, UID:0
[ 226.615663] oom_reaper: reaped process 1496 (mem.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0
kB
[ 269.025459] mem2.sh invoked oom-killer: gfp_mask=0x6200ca(GFP_HIGHUSER_MOVABLE), nodemask=(null),
order=0, oom_score_adj=0
[ 269.026301] mem2.sh cpuset=/ mems_allowed=0
[ 269.026585] CPU: 0 PID: 1497 Comm: mem2.sh Kdump: loaded Tainted: G
4.18.0-193.el8.x86_64 #1
[ 269.072398] [ 1497] 0 1497 665165 421674 4956160 187935 0 mem2.sh
[ 269.072991] Out of memory: Killed process 1497 (mem2.sh) total-vm:2660660kB, anon-rss:1686696kB,
file-rss:0kB, shmem-rss:0kB, UID:0
[ 269.250794] oom_reaper: reaped process 1497 (mem2.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:
0kB
```



Выводы:

В какой-то момент работы на первом этапе эксперимента запущенный процесс заполнил физическую память, что вызывало страничный обмен с разделом подкачки и дальнейшее его заполнение. Заполнив виртуальную память, процесс аварийно завершался, так как выделение новых страниц ни в физической, ни в виртуальной памяти было невозможно.

Процессы во втором этапе были убиты не сразу: сначала был убит mem.sh, поскольку на момент исчерпания ресурса виртуальной памяти он занимал ее больше, чем mem2.sh.

После этого mem2.sh продолжил работать, занимая освобожденный первым процессом ресурс, и в дальнейшем был убит.

Эксперимент №2

Скрипт newmem.sh, создающий массив размера N:

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

Вспомогательный скрипт, запускающий newmem.sh k раз с интервалом в 1 секунду:

```
#!/bin/bash

n=1900000
k=30
sudo chmod 755 ./newmem.sh
for ((i = 0; i < k; i++))
do
    sleep 1
    ./newmem.sh $n $i &
done
```

В первый раз запустим вспомогательный скрипт с параметрами $n = 2800000$ (размер массива в 10 раз меньший, чем значение, при котором проходила аварийная остановка в первом эксперименте), $k = 10$. Все k запусков успешно завершились, в системном журнале нет записей об аварийной остановке процесса newmem.sh

Далее запустим скрипт с параметрами $n = 2800000$, $k = 30$.

Некоторые процессы завершились без ошибок, а некоторые аварийно из-за того, что свободная память закончилась раньше, чем все процессы успели выполниться.

Путем подбора с помощью бинарного поиска узнаем, что для того, чтобы было возможно запустить и завершить без ошибок 30 процессов, необходимо, чтобы размер массива не превышал $n = 1850000$.