

Лабораторная работа №5

Общий объем оперативной памяти: 1 860 932

Объем раздела подкачки: 839 676

Размер страницы виртуальной памяти: 4096

Объем свободной физической памяти в ненагруженной системе: 1 044 664

Объем свободного пространства в разделе подкачки в ненагруженной системе: 839676

Первый эксперимент

→ Первый этап

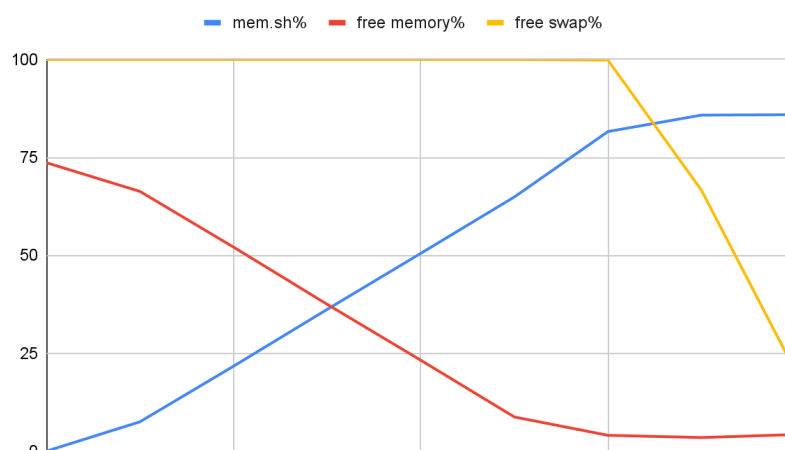
```
GNU nano 2.9.8          mem.sh
#!/bin/bash
echo "" > report.log
arr=()
cnt=0
while true
do
    arr+=(1 2 3 4 5 6 7 8 9 10)
    ((cnt++))
    if [[ $cnt -eq 100000 ]]
    then
        cnt=0
        echo "${#arr[@]}" >> report.log
    fi
done
```

◆ Значение в report.log: 31 000 000

◆ Последние две записи о скрипте в системном журнале:

```
[14754.615516] Out of memory: Killed process 1984 (bash) total-vm:2659632kB, anon-rss:1645484kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:4836kB oom_score_adj:0
[14754.695789] oom_reaper: reaped process 1984 (bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
[root@localhost lab5]#
```

◆ Диаграмма изменения состояния памяти:



Система сначала использовала оперативную память. Когда память закончилась, система начала использовать файлы подкачки. Когда и они заканчиваются, система вынуждена убивать процесс.

→ Второй этап

```
GNU nano 2.9.8 mem2.sh
#!/bin/bash

echo "" > report2.log
arr=()
cnt=0
while true
do
    arr+=(1 2 3 4 5 6 7 8 9 10)
    ((cnt++))
    if [[ $cnt -eq 100000 ]]
    then
        cnt=0
        echo "${arr[0]}" >> report2.log
    fi
done

GNU nano 2.9.8 launch.sh
#!/bin/bash

./mem.sh &
./mem2.sh
```

- ◆ Значение в report.log: 15 000 000
Значение в report2.log: 31 000 000

- ◆ Последние две записи о скриптах в системном журнале

```
[27886.839420] Out of memory: Killed process 2322 (mem.sh) total-vm:1442460kB, anon-rss:831400kB, file-rss:4kB, shmem-rss:0kB, UID:0 pgtables:2464kB oom_score_adj:0
[27886.888925] oom_reaper: reaped process 2322 (mem.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB

[27901.687697] Out of memory: Killed process 2323 (mem2.sh) total-vm:2652768kB, anon-rss:1665032kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:4840kB oom_score_adj:0
[27901.772514] oom_reaper: reaped process 2323 (mem2.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```

- ◆ Диаграмма изменения состояния памяти:



Как и в прошлом этапе система сначала использует оперативную память, а потом файлы подкачки. Когда закончилась свободная память, система убила первый процесс(mem.sh), тем самым освободив память для второго. Когда память снова закончилась, системе пришлось убить и второй процесс (mem2.sh).

Можно заметить что размер массива первой программы составляет примерно половину от размера второй, что подтверждает догадку.

В момент, когда закончилась оперативная память, в списке процессов можно было заметить процесс kswarp0 работающий с виртуальной памятью.

Эксперимент 2

```
GNU nano 2.9.8 newmem.sh

#!/bin/bash

N=$1
arr=()

while true
do
    arr+=(1 2 3 4 5 6 7 8 9 10)
    if [[ ${#arr[@]} -gt $N ]]
    then
        break
    fi
done

GNU nano 2.9.8 launchmk.sh

#!/bin/bash

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

- N=3 100 000, K=10 - завершилось без ошибок
- N=3 100 000, K=30 - крашнулось 10 из 30 программ

```
[root@localhost lab51# dmesg | grep "newmem.sh,pid"
[36839.977039] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2610,uid=0
[36842.489620] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2612,uid=0
[36844.722609] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2614,uid=0
[36847.179645] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2616,uid=0
[36849.431527] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2618,uid=0
[36851.571540] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2620,uid=0
[36853.668540] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2622,uid=0
[36858.757055] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2624,uid=0
[36861.148511] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2626,uid=0
[36863.560749] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-0.slice/session-3.scope,task=newmem.sh,pid=2628,uid=0
```

- N=2 900 000, K=30 - завершилось без ошибок

На удивление, при N всего в 11 раз меньше того, что было в первом эксперименте, программа работает без каких-либо проблем. Благодаря тому, что процессы запускаются с промежутком в секунду, некоторые из них успевают завершиться, освобождая место для новых.