

Лабораторная работа No5. Управление памятью в ОС Linux

---Ли Яньюй (M3209)

Эксперимент No1

Подготовительный этап:

- 1) Создайте скрипт **mem.**

```
#!/bin/bash

step=0
my_array=()
size_array=0

touch report.log

while true
do
    step=$(( $step + 1 ))
    my_array+=(1 2 3 4 5 6 7 8 9 10)
    size_array=$(( $size_array + 10 ))

    if [[ $step == 100000 ]]; then
        echo $size_array >> report.log
        step=0
    fi
done
```

- 2) Написать «следящие» скрипты и собирать данные, ---- **top_tracker**

```
#!/bin/bash

while true
do
    top -b |grep 'top - ' -A 11 >> top_history.txt
    sleep 1
done
```

Первый этап:

Последняя запись журнала – значения параметров, с которыми произошла аварийная остановка процесса.

```
top - 23:32:12 up 3 min, 1 user, load average: 2.17, 1.21, 0.50
Tasks: 223 total, 3 running, 220 sleeping, 0 stopped, 0 zombie
%Cpu(s): 63.2 us, 32.0 sy, 0.0 ni, 0.0 id, 1.2 wa, 3.2 hi, 0.5 si, 0.0 st
MiB Mem : 2025.1 total, 57.1 free, 1916.3 used, 51.6 buff/cache
MiB Swap: 820.0 total, 0.0 free, 820.0 used, 4.7 avail Mem

    PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
  2777 yanyu    20   0 1639596 1.2g 1428 R 85.1  59.6   0:55.43 bash
    47 root      20   0 0        0      0 S  8.7   0.0   0:01.62 kswapd0
 2032 yanyu    20   0 2920456 152780 5524 R 1.3   7.4   0:12.17 gnome-s+
 2774 yanyu    20   0 64508    1708 1240 R 0.6   0.1   0:00.16 top
    469 root      0 -20 0         0      0 I  0.3   0.0   0:00.04 kworker+

--
top - 23:32:17 up 3 min, 1 user, load average: 2.17, 1.21, 0.50
Tasks: 224 total, 7 running, 217 sleeping, 0 stopped, 0 zombie
%Cpu(s): 11.5 us, 77.3 sy, 0.0 ni, 0.0 id, 3.3 wa, 6.6 hi, 1.3 si, 0.0 st
MiB Mem : 2025.1 total, 51.6 free, 1924.6 used, 48.9 buff/cache
MiB Swap: 820.0 total, 0.0 free, 820.0 used, 16.4 avail Mem

    PID USER      PR  NI  VIRT  RES  SHR S %CPU  %MEM    TIME+  COMMAND
    47 root      20   0 0         0      0 R 61.5   0.0   0:04.16 kswapd0
```

И значение в последней строке файла report.log.

```
-----  
14000000  
15000000  
16000000  
17000000  
18000000  
19000000  
20000000  
21000000
```

Наблюдайте за следующими значениями (и фиксируйте их изменения во времени в отчете):

- 1) значения параметров памяти системы (верхние две строки над основной таблицей);
- 2) значения параметров в строке таблицы, соответствующей работающему скрипту;
- 3) изменения в верхних пяти процессах (как меняется состав и позиции этих процессов).

Полные данные находятся в файле **top_history.txt**

```
MiB Mem : 2025.1 total, 192.4 free, 1188.6 used, 644.1 buff/cache  
MiB Swap: 820.0 total, 820.0 free, 0.0 used. 664.5 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND  
 2777 yanyu     20   0   37380  27796  2836 R   28.5   1.3   0:00.86  bash  
 2032 yanyu     20   0 2920456 275068 112540 S    4.6  13.3   0:12.09  gnome-s+  
 2656 yanyu     20   0  521736  43244  30012 S    1.0   2.1   0:00.99  gnome-t+  
 2092 yanyu     20   0  387324  10340   8556 S    0.7   0.5   0:00.21  ibus-da+  
 2334 yanyu     20   0  228900   9056   8016 S    0.7   0.4   0:00.07  ibus-en+
```

Через почти 1 сек после старта, процесс становится топ1 по требуемой ему оперативной памяти и занимает эту позицию до завершения процесса.

```
MiB Mem : 2025.1 total, 109.8 free, 1271.2 used, 644.1 buff/cache  
MiB Swap: 820.0 total, 820.0 free, 0.0 used. 581.9 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND  
 2777 yanyu     20   0  121992 112276  2836 R   97.0   5.4   0:03.79  bash  
 2032 yanyu     20   0 2920456 274952 112540 S    0.7  13.3   0:12.11  gnome-s+  
 2180 root       20   0  204924  32272  10348 S    0.3   1.6   0:00.25  sssd_kcm  
 2774 yanyu     20   0   64508   4848   4020 R    0.3   0.2   0:00.03  top  
    1 root       20   0  245392  14188   9272 S    0.0   0.7   0:01.83  systemd
```

```
MiB Mem : 2025.1 total, 74.2 free, 1355.3 used, 595.6 buff/cache  
MiB Swap: 820.0 total, 819.2 free, 0.8 used. 498.5 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND  
 2777 yanyu     20   0  208320 198604  2836 R   97.0   9.6   0:06.73  bash  
    47 root       20   0         0         0         0 S    1.3   0.0   0:00.04  kswapd0  
 2774 yanyu     20   0   64508   4804   3976 R    0.3   0.2   0:00.04  top  
    1 root       20   0  245392  14176   9272 S    0.0   0.7   0:01.83  systemd  
    2 root       20   0         0         0         0 S    0.0   0.0   0:00.00  kthreadd
```

```

MiB Mem : 2025.1 total, 68.8 free, 1440.6 used, 515.6 buff/cache
MiB Swap: 820.0 total, 813.7 free, 6.3 used. 418.4 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 2777 yanyu     20   0 295968 286252 2836 R  95.7  13.8   0:09.62 bash
   47 root      20   0      0      0      0 S   2.3   0.0   0:00.11 kswapd0
  896 rngd      20   0 160228  4436  3704 S   0.3   0.2   0:04.54 rngd
 2032 yanyu     20   0 2920456 273020 110608 S   0.3  13.2   0:12.12 gnome-s+
 2695 yanyu     20   0 973764  65392  39612 S   0.3   3.2   0:01.22 nautilus

MiB Mem : 2025.1 total, 66.6 free, 1517.3 used, 441.1 buff/cache
MiB Swap: 820.0 total, 803.7 free, 16.3 used. 345.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 2777 yanyu     20   0 383352 373636 2836 R  95.7  18.0   0:12.50 bash
   47 root      20   0      0      0      0 S   2.3   0.0   0:00.18 kswapd0
 1819 gdm        20   0 473972 12428 10672 S   0.3   0.6   0:00.05 gsd-sma+
 2180 root      20   0 204924  32012 10088 S   0.3   1.5   0:00.26 sssd_kcm
    1 root      20   0 245392 13964  9272 S   0.0   0.7   0:01.83 systemd

```

Последние две записи о скрипте в системном журнале
(с помощью команды `dmesg | grep "mem"`)

```

[ 198.215957] Out of memory: Killed process 2777 (bash) total-virt:1655700kB, anon-rss:1
248876kB, file-rss:0kB, shmem-rss:0kB, UID:1000
[ 198.292975] oom_reaper: reaped process 2777 (bash), now anon-rss:0kB, file-rss:0kB,
shmem-rss:0kB

```

Второй этап:

Наблюдайте за следующими значениями (и фиксируйте их изменения во времени в отчете):

- 1) значения параметров памяти системы (верхние две строки над основной таблицей);
- 2) значения параметров в строке таблицы, соответствующей работающему скрипту;
- 3) изменения в верхних пяти процессах (как меняется состав и позиции этих процессов).

Полные данные находятся в файле **top_history.txt**

```

MiB Mem : 2025.1 total, 995.0 free, 660.8 used, 369.3 buff/cache
MiB Swap: 820.0 total, 251.7 free, 568.3 used. 1206.4 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 4076 yanyu     20   0 46884  37112  2904 R  37.9   1.8   0:01.14 bash
 4075 yanyu     20   0 46620  36692  2744 R  37.5   1.8   0:01.13 bash
 2032 yanyu     20   0 2948516 194708 74732 S   2.0   9.4   2:14.69 gnome-s+
 2695 yanyu     20   0 977308  46880 28740 S   0.3   2.3   0:10.07 nautilus
 4071 yanyu     20   0 64508   4832  4008 R   0.3   0.2   0:00.03 top

MiB Mem : 2025.1 total, 907.0 free, 748.7 used, 369.3 buff/cache
MiB Swap: 820.0 total, 251.7 free, 568.3 used. 1118.5 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 4075 yanyu     20   0 91764  81832  2744 R  49.5   3.9   0:02.62 bash
 4076 yanyu     20   0 91632  81992  2904 R  48.8   4.0   0:02.61 bash
    1 root      20   0 245392  5836  3876 S   0.0   0.3   0:02.29 systemd
    2 root      20   0      0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 rcu_gp

```

```

MiB Mem : 2025.1 total, 826.4 free, 829.3 used, 369.3 buff/cache
MiB Swap: 820.0 total, 251.7 free, 568.3 used, 1037.9 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
  4075 yanyu      20   0 133344 123544 2744 R   47.7   6.0   0:04.06  bash
  4076 yanyu      20   0 133080 123440 2904 R   47.7   6.0   0:04.05  bash
  2032 yanyu      20   0 2948532 193864 74732 R    2.6   9.3   2:14.77  gnome-s+
  3890 yanyu      20   0 521656 41004 29888 S    0.7   2.0   0:02.14  gnome-t+
  2180 root        20   0 204924 21540  888 S    0.3   1.0   0:09.54  sssd_kcm

```

```

MiB Mem : 2025.1 total, 748.7 free, 907.0 used, 369.3 buff/cache
MiB Swap: 820.0 total, 251.7 free, 568.3 used, 960.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
  4076 yanyu      20   0 172152 162512 2904 R   45.7   7.8   0:05.43  bash
  4075 yanyu      20   0 172548 162616 2744 R   45.4   7.8   0:05.43  bash
  2032 yanyu      20   0 2948516 193868 74732 S    5.0   9.3   2:14.92  gnome-s+
  3890 yanyu      20   0 521892 41132 29888 S    0.7   2.0   0:02.16  gnome-t+
    9 root        20   0         0         0     0 S    0.3   0.0   0:00.08  ksoftir+

```

Значения в последних строках файлов report.log и report2.log.

```

10000000      120000000
20000000      130000000
30000000      140000000
40000000      150000000
50000000      160000000
60000000      170000000
70000000      180000000
80000000      190000000
90000000      200000000
100000000     210000000
report.log      report2.log

```

Последние записи о скриптах в системном журнале

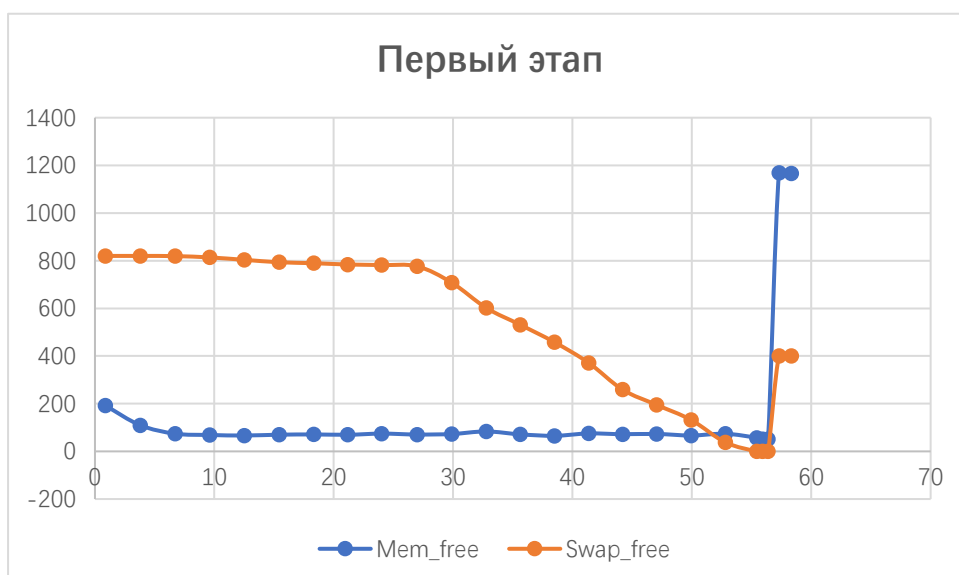
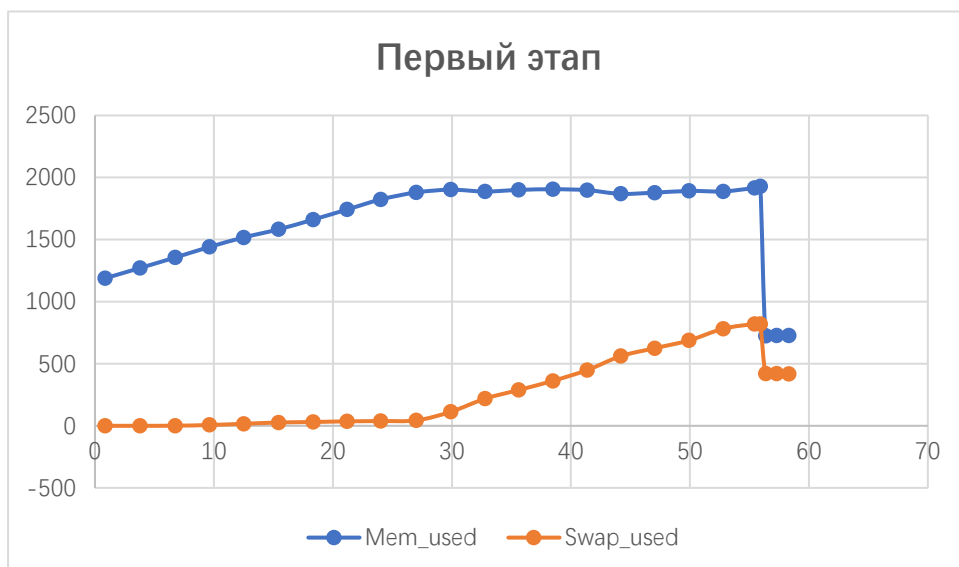
(с помощью команды dmesg | grep "bash[2]*")

```

[ 4733.486029] [ 3895] 1000 3895 6141 20 86016 474 0 ba
sh
[ 4733.486030] [ 3983] 1000 3983 6141 1 81920 490 0 ba
sh
[ 4733.486031] [ 4070] 1000 4070 3174 1 77824 66 0 ba
sh
[ 4733.486035] [ 4075] 1000 4075 205530 187764 1691648 14653 0 ba
sh
[ 4733.486036] [ 4076] 1000 4076 205134 187344 1679360 14665 0 ba
sh
[ 4733.486041] Out of memory: Killed process 4075 (bash) total-vm:822120kB, anon-rss:75
1056kB, file-rss:0kB, shmem-rss:0kB, UID:1000
[ 4767.674989] [ 3895] 1000 3895 6141 20 86016 474 0 ba
sh
[ 4767.674990] [ 3983] 1000 3983 6141 1 81920 490 0 ba
sh
[ 4767.674991] [ 4070] 1000 4070 3174 1 77824 66 0 ba
sh
[ 4767.674995] [ 4076] 1000 4076 413595 384386 3350528 26084 0 ba
sh
[ 4767.675002] Out of memory: Killed process 4076 (bash) total-vm:1654380kB, anon-rss:1
537544kB, file-rss:0kB, shmem-rss:0kB, UID:1000
[ 4767.833275] oom_reaper: reaped process 4076 (bash), now anon-rss:0kB, file-rss:0kB,
shmem-rss:0kB

```

Обработка результатов:

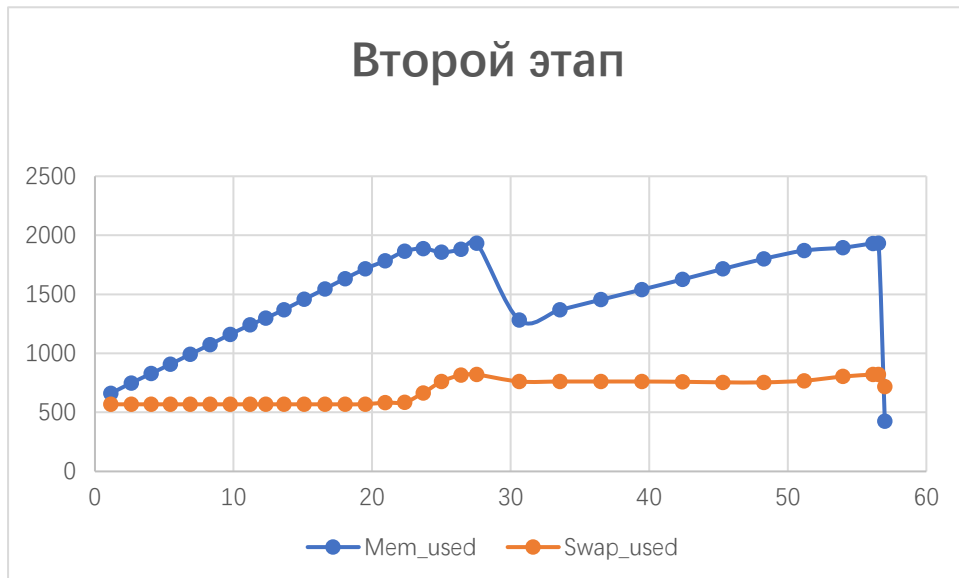


Письменные выводы:

Первоначально, в оперативной памяти есть место, поэтому процесс пользуется RAM, а swap не используется.

Потом, когда место оперативной памяти не хватает, происходит копирование страниц из RAM на swap-раздел жёсткого диска.

Затем, произошла аварийная остановка процесса, то есть процесс удалён системой, эти страницы удаляются из swap'a и оперативная память освобождается.



Письменные выводы:

Первоначально, в оперативной памяти есть место, поэтому процесс пользуется RAM. Когда место RAM не хватает, то свар копируются страницы виртуальной памяти. И количество доступной памяти раздела подкачки уменьшалось.

Затем, произошла аварийная остановка процесса первого скрипта. При этом размеры свободной оперативной памяти и памяти раздела подкачки увеличивались.

В оперативной памяти опять есть место, поэтому процесс пользуется RAM.....

Эксперимент No2

Подготовительный этап:

Создайте копию скрипта mem.bash в файл newmem.bash. Измените копию таким образом, чтобы она завершала работу, как только размер создаваемого массива превысит значение N, передаваемое в качестве параметра скрипту.

```
#!/bin/bash

N=$1

#step=0
my_array=()
size_array=0

touch report.log

while true
do
    #step=$(( $step + 1 ))
    my_array+=(1 2 3 4 5 6 7 8 9 10)
    size_array=$(( $size_array + 10 ))

    if [[ "$size_array" -ge "$N" ]];then
        echo $size_array >> report.log
        break
    fi
done
```

Основной этап:

Скрипт, который будет запускать newmem.bash каждую секунду, используя один и тот же параметр N так, что всего будет осуществлено K запусков.

Возьмите в качестве значения N, величину, в 10 раз меньшую, чем размер массива, при котором происходила аварийная остановка процесса в первом этапе предыдущего эксперимента. $N = 21000000/10 = 2100000$

Возьмите в качестве K значение 10. $K = 10$

```
#!/bin/bash

N=2100000

for(( i = 0; i < 10; i++ ))
do
    bash newmem "$N" &
    sleep 1
done
```

Убедитесь, что все K запусков успешно завершились, и в системном журнале нет записей об аварийной остановке

```
[yanyu@localhost Test2]$ dmesg | grep "bash"
[yanyu@localhost Test2]$
```


Измените значение K на 30 и снова запустите скрипт.

```
[yanyu@localhost ~]$ dmesg | grep "bash"
[ 303.385140] bash invoked oom-killer: gfp_mask=0x6200ca(GFP_HIGHUSER_MOVABLE),
nodemask=(null), order=0, oom_score_adj=0
[ 303.385141] bash cpuset=/ mems_allowed=0
[ 303.385145] CPU: 0 PID: 2876 Comm: bash Kdump: loaded Tainted: G
-----r- - 4.18.0-193.el8.x86_64 #1
[ 303.385460] [ 2648] 1000 2648 6141 130 90112 364
0 bash
[ 303.385461] [ 2798] 1000 2798 41388 21767 372736 16484
0 bash
[ 303.385462] [ 2802] 1000 2802 33303 21245 311296 8889
0 bash
[ 303.385463] [ 2805] 1000 2805 29772 21690 286720 4945
0 bash
[ 303.385464] [ 2808] 1000 2808 26637 21352 253952 2174
0 bash
[ 303.385465] [ 2811] 1000 2811 24789 21587 241664 34
0 bash
[ 303.385467] [ 2814] 1000 2814 23469 20364 233472 0

[ 336.261452] [ 2864] 1000 2864 19113 16248 196608 15 0 ba
sh
[ 336.261453] [ 2867] 1000 2867 18849 16032 200704 5 0 ba
sh
[ 336.261454] [ 2870] 1000 2870 17826 14922 188416 0 0 ba
sh
[ 336.261455] [ 2873] 1000 2873 16869 14003 180224 0 0 ba
sh
[ 336.261456] [ 2876] 1000 2876 16935 14083 180224 2 0 ba
sh
[ 336.261457] [ 2879] 1000 2879 17793 14886 188416 3 0 ba
sh
[ 336.261458] [ 2882] 1000 2882 16803 13916 184320 0 0 ba
sh
[ 336.261459] [ 2885] 1000 2885 16308 13474 176128 0 0 ba
sh
[ 336.261460] [ 2888] 1000 2888 15879 12776 176128 0 0 ba
sh
[ 336.261461] [ 2891] 1000 2891 14856 12053 163840 0 0 ba
sh
[ 336.261462] [ 2894] 1000 2894 15384 12509 167936 0 0 ba
sh
[ 336.261467] Out of memory: Killed process 2805 (bash) total-vm:146148kB, anon-rss:66
516kB, file-rss:1112kB, shmem-rss:0kB, UID:1000
[ 336.308976] oom_reaper: reaped process 2805 (bash), now anon-rss:0kB, file-rss:0kB,
shmem-rss:0kB
```

Q: Объясните, почему ряд процессов завершился аварийно.

A: Потому что $N * K >$ предельное значение, полученное в первом эксперименте.

Q: Подберите такое максимальное значение N, чтобы при K=30 не происходило аварийных завершений процессов.

A: $N = 2100000/3 = 700000$

```
[yanyu@localhost Test2]$ ls
newmem report.log start
[yanyu@localhost Test2]$ cat start
#!/bin/bash

N=700000

for(( i = 0; i < 30; i++ ))
do
    bash newmem "$N" &
    sleep 1
done
[yanyu@localhost Test2]$ bash start
[yanyu@localhost Test2]$ dmesg | grep "bash"
[yanyu@localhost Test2]$
```