

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение высшего образования  
Санкт-Петербургский национальный исследовательский университет информационных технологий  
механики и оптики  
Мегафакультет трансляционных информационных технологий  
Факультет информационных технологий и программирования

**Лабораторная работа № 5**

**По дисциплине «Операционные системы»**

**Управление памятью в ОС Linux**

Выполнил студент группы №М3203  
*Папикян Сергей Седракович*

Преподаватель:  
*Титова Анастасия Витальевна*

***САНКТ-ПЕТЕРБУРГ***

***2020***

# Отчет

## Данные о текущей конфигурации ОС:

1. Общий объем оперативной памяти: 512 Мб
2. Объем раздела подкачки: 820 Мб
3. Размер страницы виртуальной памяти: 4 Кб
4. Объем свободной физической памяти в ненагруженной системе: 102.4 Мб
5. Объем свободного пространства в разделе подкачки в ненагруженной системе: 820 Мб

## Эксперимент I

### Подготовительный этап:

1. Уменьшил объем оперативной памяти с 4096 Мб на 512 Мб
2. Был написан скрипт mem.bash, который на каждом шаге цикла в конец массива добавлял последовательность из 100 элементов. Далее, каждую 10000-ый шаг добавлял в файл report.log строка с текущим значением размера массива

```
#!/bin/bash

declare -a array
declare -a numbers=(1,2,3,4,5,6,7,8,9,10,11,12,13)

while true;
do
    array+=(${numbers[@]})
    let counter++
    if [[ $counter == 10000 ]]
    then
        echo "${#array[@]}" >> report.log
        counter=0
    fi
done
```

## Первый этап:

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

```
[ 6539.174950] [ 845] 0 845 77886 15 180224 148 0 gssproxy
[ 6539.178679] [ 856] 70 856 20655 3 155648 105 0 avahi-daemon
[ 6539.182310] [ 857] 991 857 40057 19 208896 184 0 rngd
[ 6539.185557] [ 864] 0 864 107202 19 438272 629 0 sssd_be
[ 6539.189296] [ 867] 0 867 124680 0 466944 5470 0 firewalld
[ 6539.193109] [ 868] 0 868 107778 96 479232 313 0 sssd_nss
[ 6539.210323] [ 874] 0 874 24455 219 212992 443 0 systemd-logind
[ 6539.222020] [ 877] 0 877 170191 211 409600 439 0 NetworkManager
[ 6539.225193] [ 885] 0 885 158766 126 454656 3582 0 tuned
[ 6539.228307] [ 893] 0 893 23242 0 208896 235 -1000 sshd
[ 6539.236270] [ 1106] 0 1106 61463 62 114688 158 0 crond
[ 6539.239634] [ 1107] 0 1107 30419 16 258048 205 0 login
[ 6539.242775] [ 1108] 0 1108 10656 22 114688 29 0 atd
[ 6539.245329] [ 1133] 0 1133 23438 66 212992 296 0 systemd
[ 6539.255049] [ 1138] 0 1138 58456 19 299008 1316 0 (sd-pam)
[ 6539.257914] [ 1144] 0 1144 59183 1 94208 553 0 bash
[ 6539.260326] [ 11276] 0 11276 38226 29 307200 292 0 sshd
[ 6539.268574] [ 11280] 0 11280 38882 579 307200 422 0 sshd
[ 6539.271811] [ 11281] 0 11281 55655 47 77824 50 0 bash
[ 6539.273358] [ 11325] 0 11325 55622 1 81920 60 0 sh
[ 6539.276113] [ 11333] 0 11333 221896 7079 917504 1345 0 node
[ 6539.278689] [ 11407] 0 11407 199017 1617 704512 1703 0 node
[ 6539.291208] [ 11428] 0 11428 222700 3819 1765376 13821 0 node
[ 6539.293768] [ 11462] 0 11462 3593850 75 1372160 9064 0 node
[ 6539.303667] [ 11509] 0 11509 59123 2 94208 479 0 bash
[ 6539.308090] [ 11610] 0 11610 55622 52 77824 28 0 sh
[ 6539.310856] [ 11612] 0 11612 287462 68072 1937408 163917 0 sh
[ 6539.322617] [ 19973] 0 19973 54261 17 61440 0 0 sleep
[ 6539.324947] [ 21028] 0 21028 55622 52 69632 28 0 sh
[ 6539.327491] [ 21029] 0 21029 65339 96 147456 0 0 top
[ 6539.337999] [ 21030] 0 21030 54264 16 61440 0 0 head
[ 6539.341131] [ 21031] 0 21031 54270 16 65536 0 0 tail
[ 6539.343378] [ 21032] 0 21032 58791 76 102400 0 0 awk
[ 6539.353911] Out of memory: Killed process 11612 (sh) total-vm:1149848kB, anon-rss:271880kB, file-
rss:408kB, shmem-rss:0kB, UID:0
handler.sh: line 12: 11612 Killed sh mem.bash
```

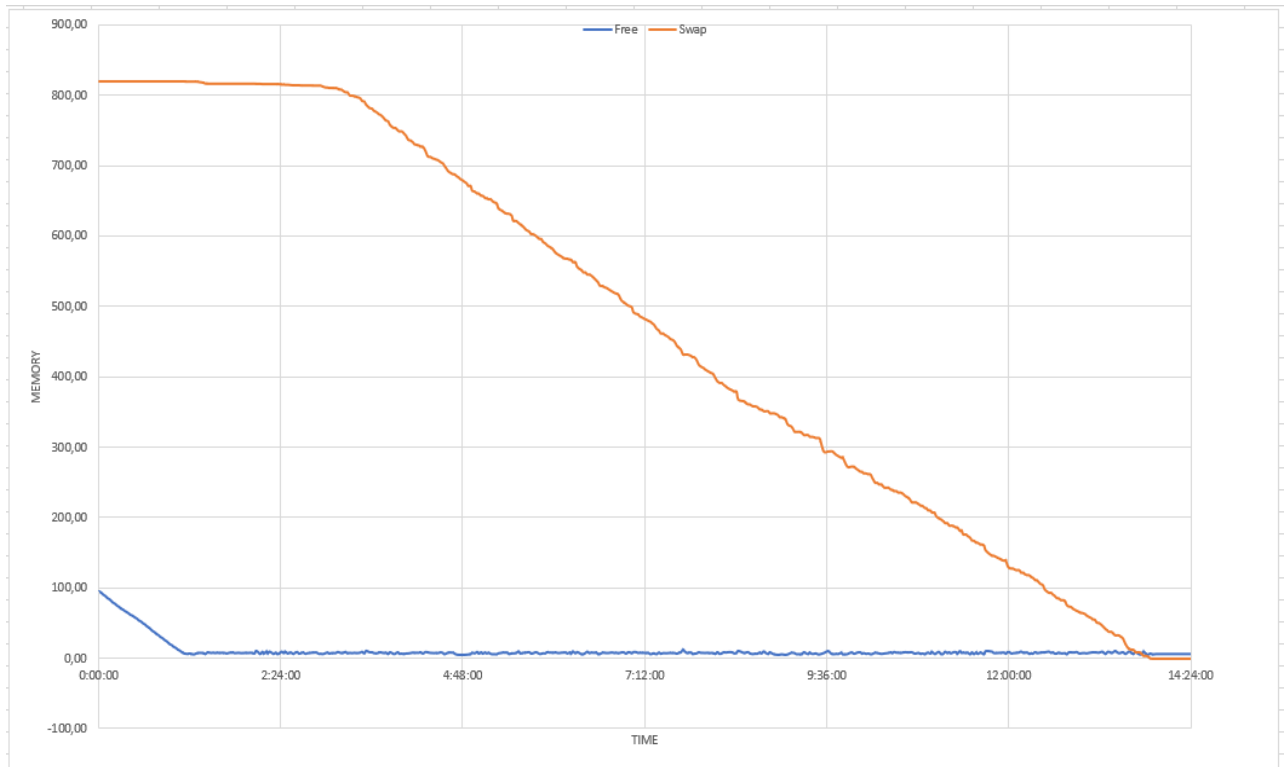
2. Значение в последней строке файла report.log: 49100000
3. Данные который получены во время наблюдения (парсер top): [ITMO-Operating-system-3-sem/data1\\_st1 at master · I-SER-I/ITMO-Operating-system-3-sem \(github.com\)](https://github.com/ITMO-Operating-system-3-sem/data1_st1_at_master_I-SER-I/ITMO-Operating-system-3-sem)

```
runner.bash data1 report.log mem.bash
lab5 > experiment1 > data1
1 TIME MEM VIRT RES SHR CPU FREE SWAP
2 0:00.44 0.8 223256 3816 2860 93.8 40.9 730.0
3 0:02.02 1.3 225688 6248 2860 99.9 38.4 730.0
4 0:03.58 1.8 228120 8680 2860 99.9 35.7 730.0
5 0:05.13 2.3 230552 11112 2860 88.2 34.0 730.0
6 0:06.70 2.8 232984 13544 2860 88.2 31.4 730.0
7 0:08.26 3.3 235544 16104 2860 99.9 28.9 730.0
8 0:09.81 3.8 237976 18536 2860 99.9 26.3 730.0
9 0:11.37 4.3 240408 20968 2860 99.9 24.8 730.0
10 0:12.94 4.8 242712 23272 2860 93.8 22.4 730.0
11 0:14.52 5.3 245144 25704 2860 99.9 19.8 730.0
12 0:16.07 5.8 247576 28008 2860 99.9 17.2 730.0
13 0:17.61 6.3 250008 30568 2860 99.9 14.4 730.0
14 0:19.16 6.8 252440 33000 2860 99.9 11.9 730.0
15 0:20.71 7.3 254872 35432 2860 94.4 9.4 730.0
16 0:22.25 7.8 257432 37992 2860 94.1 7.9 730.0
17 0:23.82 8.3 259864 40296 2860 94.1 5.2 730.0
18 0:25.36 8.8 262040 42600 2860 94.1 5.7 730.0
19 0:26.92 9.3 264472 45032 2860 94.1 5.3 730.0
20 0:28.49 9.7 266776 47336 2860 99.9 5.6 730.0
21 0:30.10 10.3 269336 49896 2860 93.3 5.1 730.0
22 0:31.71 10.8 271768 52328 2860 94.1 5.0 730.0
23 0:33.28 11.2 273816 54376 2860 93.8 5.4 730.0
24 0:34.85 11.6 276120 56588 2768 93.8 5.8 730.0
```

#### 4. Две записи о скрипте в системном журнале

```
[root@localhost experiment1]# dmesg | grep "mem.bash"
[ 8507.498564] [28156] 0 28156 258055 36709 1695744 165785 0 mem.bash
[ 8507.503969] [28712] 0 28712 81286 25725 290816 0 0 mem.bash
[ 8507.506117] [29700] 0 29700 61478 5917 122880 0 0 mem.bash
[ 8507.512185] Out of memory: Killed process 28156 (mem.bash) total-vm:1032220kB, anon-rss:146836kB,
file-rss:0kB, shmem-rss:0kB, UID:0
[ 9162.528233] [28712] 0 28712 157990 23428 901120 78994 0 mem.bash
[ 9162.530810] [29700] 0 29700 137926 23341 737280 59020 0 mem.bash
[ 9162.548482] [30208] 0 30208 104710 23496 471040 25651 0 mem.bash
[ 9162.568608] Out of memory: Killed process 28712 (mem.bash) total-vm:631960kB, anon-rss:93712kB, f
ile-rss:0kB, shmem-rss:0kB, UID:0
```

#### 5. График зависимости времени от памяти полученный на основе данных из (3)



#### Второй этап:

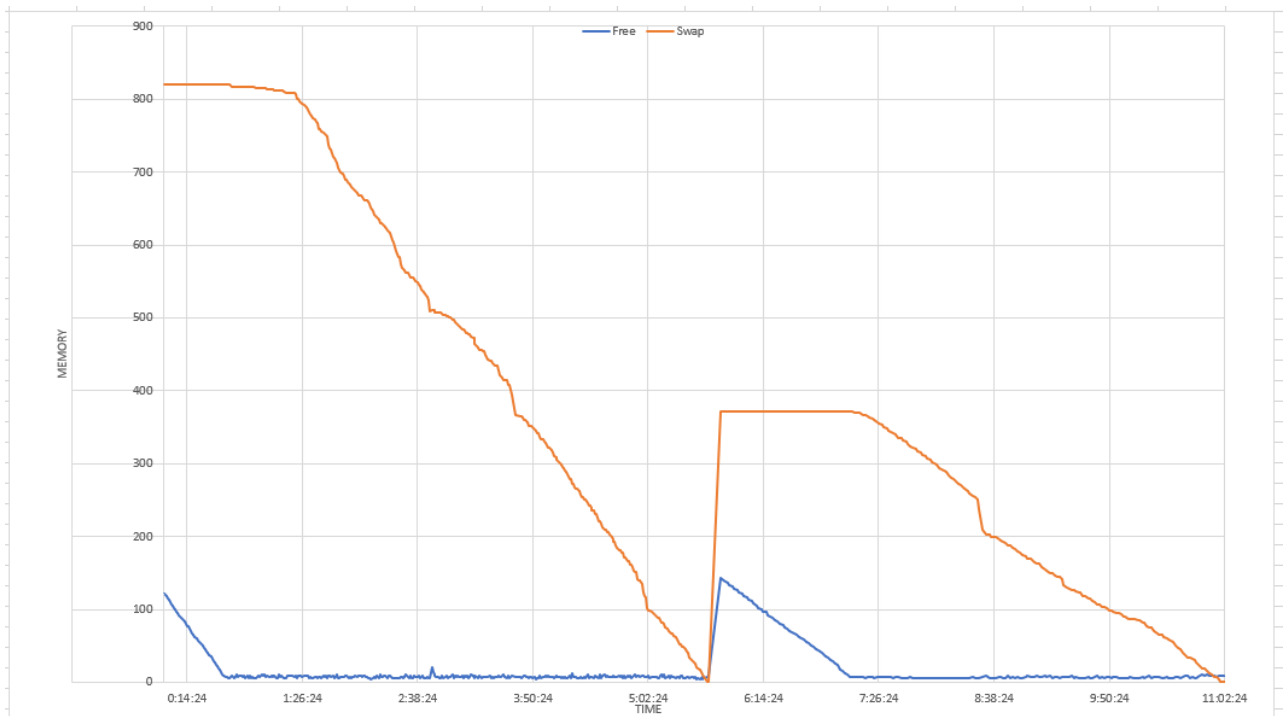
1. Производим аналогичные выше действия только с еще одним скриптом
2. Данные который получены во время наблюдения (парсер top): [ITMO-Operating-system-3-sem/data1\\_st2 at master · I-SER-I/ITMO-Operating-system-3-sem \(github.com\)](https://github.com/ITMO-Operating-system-3-sem/data1_st2_at_master)

3. Значение в последней строке файла report.log: 23200000 и 44600000

4. Две записи о скрипте в системном журнале

```
[root@localhost experiment1]# dmesg | grep "mem[2]*.bash"
[ 1209.369757] mem.bash invoked oom-killer: gfp_mask=0x6200ca(GFP_HIGHUSER_MOVABLE), nodemask=(null), order=0, oom_score_adj=0
[ 1209.372977] mem.bash cpuset=/ mems_allowed=0
[ 1209.374743] CPU: 0 PID: 1217 Comm: mem.bash Tainted: G                -----r-   - 4.18.0-193.el8.x86_64 #1
[ 1209.640073] [ 1217]      0 1217 194950  43803 1183744   95573          0 mem.bash
[ 1209.641926] [ 1219]      0 1219 194982  43828 1200128   95579          0 mem.bash
[ 1209.658295] Out of memory: Killed process 1219 (mem.bash) total-vm:779928kB, anon-rss:175312kB, file-rss:0kB, shmem-rss:0kB, UID:0
[ 1622.814216] [ 1217]      0 1217 329798  83064 2265088  191166          0 mem.bash
[ 1622.840241] Out of memory: Killed process 1217 (mem.bash) total-vm:1319192kB, anon-rss:332256kB, file-rss:0kB, shmem-rss:0kB, UID:0
[16336.591087] [371015]      0 371015 172582  33410 1015808   83615          0 mem.bash
[16336.593222] [371017]      0 371017 172582  33621 1015808   83405          0 mem.bash
[16336.633507] Out of memory: Killed process 371017 (mem.bash) total-vm:690328kB, anon-rss:134484kB, file-rss:0kB, shmem-rss:0kB, UID:0
[16744.988237] [371015]      0 371015 282502  62044 1892352  164902          0 mem.bash
[16745.023545] Out of memory: Killed process 371015 (mem.bash) total-vm:1130008kB, anon-rss:248152kB, file-rss:24kB, shmem-rss:0kB, UID:0
[root@localhost experiment1]# _
```

5. График зависимости времени от памяти полученный на основе данных из (3)



*Наблюдения:*

При низкой величине физической памяти (7 – 10 Мб) происходит переход на файлы подкачки. Как только их значение станет критично, то произойдет аварийное завершение программы.

Если запусти 2 скрипта одновременно, то из графика можно заметить, что один из них продолжил работу, это обусловлено тем, что во время аварийного завершения первого скрипта у нас восстанавливается физическая память, далее идет сценарий как из первого этапа.

## Эксперимент II

1. Подкорректирован newmem.bash – добавлено условие на прекращение скрипта

```
1  #!/bin/bash
2
3  declare -a array
4  declare -a numbers=(1,2,3,4,5,6,7,8,9,10,11)
5
6  rm report.log
7  while true
8  do
9      array+=(${numbers[@]})
10     let counter++
11     if [[ $counter == 10000 ]]
12     then
13         echo "${#array[@]}" >> report.log
14         counter=0
15         if [[ "${#array[@]}" == "$#" ]]
16         then
17             exit 0
18         fi
19     fi
20 done
```

2. Был написан обработчик, через который будет производиться запуск

```
b5 > experiment2 > ex2.sh > ...
1  #!/bin/bash
2
3  K=$1
4  N=$2
5
6  for i in $(seq 1 $K); do
7      ./newmem.bash $N&
8  done
```

3. Установлено значение K = 10 и N = 2500000
4. Убеждаемся, что было запущено 10 скриптов

```
top - 08:31:10 up 4:54, 2 users, load average: 17.29, 6.56, 4.30
Tasks: 135 total, 16 running, 119 sleeping, 0 stopped, 0 zombie
%Cpu(s): 92.6 us, 6.1 sy, 0.0 ni, 0.0 id, 0.0 wa, 1.3 hi, 0.0 si, 0.0 st
MiB Mem : 474.8 total, 31.4 free, 337.7 used, 105.7 buff/cache
MiB Swap: 820.0 total, 725.4 free, 94.6 used, 122.4 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
454596 root       20   0 233240 12768 2548 R   8.9   2.6   0:06.37 newmem.bash
454598 root       20   0 233240 12768 2544 R   8.9   2.6   0:06.37 newmem.bash
454595 root       20   0 233112 12640 2516 R   8.6   2.6   0:06.37 newmem.bash
454597 root       20   0 233240 12632 2532 R   8.6   2.6   0:06.36 newmem.bash
454599 root       20   0 233112 12696 2596 R   8.6   2.6   0:06.37 newmem.bash
454600 root       20   0 233112 12600 2516 R   8.6   2.6   0:06.36 newmem.bash
454602 root       20   0 233112 12700 2596 R   8.6   2.6   0:06.37 newmem.bash
454603 root       20   0 233112 12648 2544 R   8.6   2.6   0:06.36 newmem.bash
454594 root       20   0 233112 12704 2588 R   8.3   2.6   0:06.36 newmem.bash
454601 root       20   0 233240 12700 2480 R   8.3   2.6   0:06.36 newmem.bash
428209 root       20   0 797940 38116 23200 S   0.7   7.8   0:05.28 node
456490 root       20   0 274292 4452 3808 R   0.7   0.9   0:00.07 top
428100 root       20   0 869120 70400 26300 S   0.3  14.5   0:05.49 node
```

### *Наблюдения:*

При  $K = 10$  и  $N = 2500000$  программа успешно завершила свое выполнение.

При  $K = 30$  и  $N = 2500000$  программа аварийно завершила свое выполнение.

При  $K = 30$  и  $N = 2000000$  программа успешно завершила свое выполнение.

При  $K = 30$  и  $N = 2100000$  программа успешно завершила свое выполнение.

При  $K = 30$  и  $N = 3000000$  программа аварийно завершила свое выполнение.

При  $K = 30$  и  $N = 2500000$  программа успешно завершила свое выполнение.

При  $K = 30$  и  $N = 2600000$  программа аварийно завершила свое выполнение.

### **Вывод**

На практике убедился, что во время работы с оперативной памятью данные заполняют физическую память, пока не дойдет до критического значения. Далее ОС начинает заполнять файлы подкачки.