

## Отчет по лабораторной работе 5 ОС

- Общий объем оперативной памяти: 1870684 kB
- Объем раздела подкачки: 839676 kB
- Размер страницы виртуальной памяти: 4 kB
- Объем свободной физической памяти в ненагруженной системе: 1400952 kB
- Объем свободного пространства в разделе подкачки в ненагруженной системе: 839676 kB

### Эксперимент 1:

#### Первый этап эксперимента:

Скрипт mem.bash:

```
GNU nano 2.9.8 mem.bash

#!/bin/bash

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

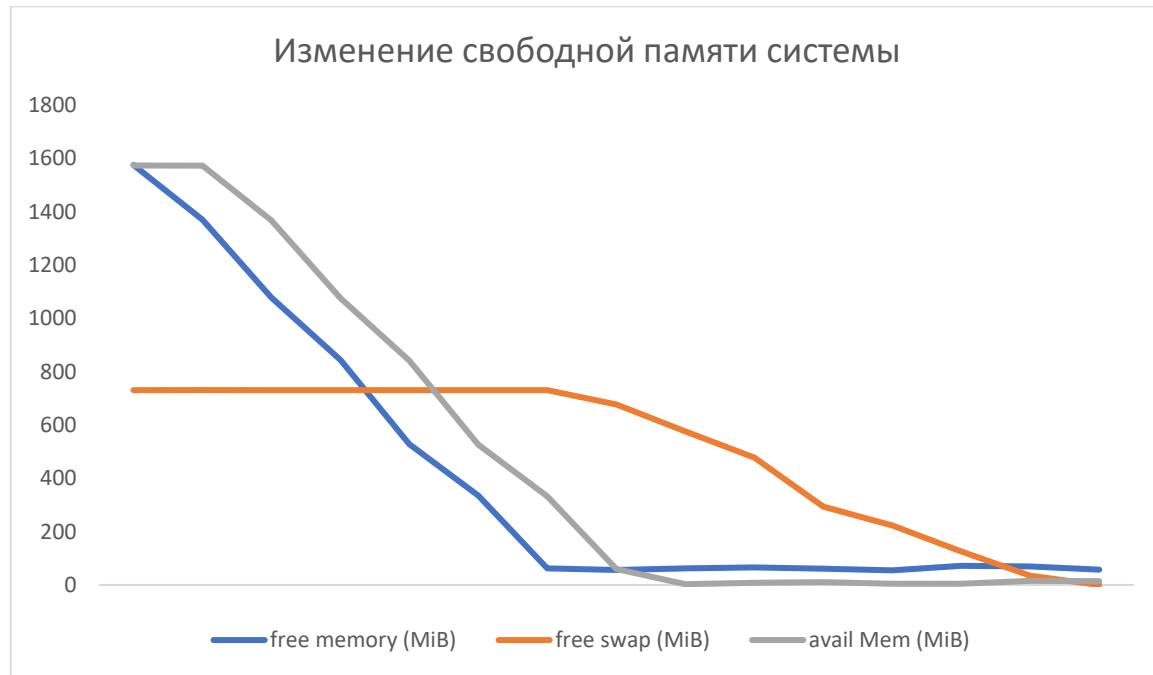
- 1) Последняя запись журнала о скрипте mem.bash:

```
[62627.435888] [ 4202] 1000 4202 656222 414729 4882432 185939 0 mem.bash
[62627.436229] Out of memory: Killed process 4202 (mem.bash) total-vm:2624888kB, anon-rss:1658916kB,
file-rss:0kB, shmem-rss:0kB, UID:1000
[62627.510445] oom_reaper: reaped process 4202 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
Killed
```

- 2) Значение в последней строке файла report.log: 300000000
- 3) Последние две записи о скрипте в системном журнале с помощью “dmesg | grep “mem.bash””:

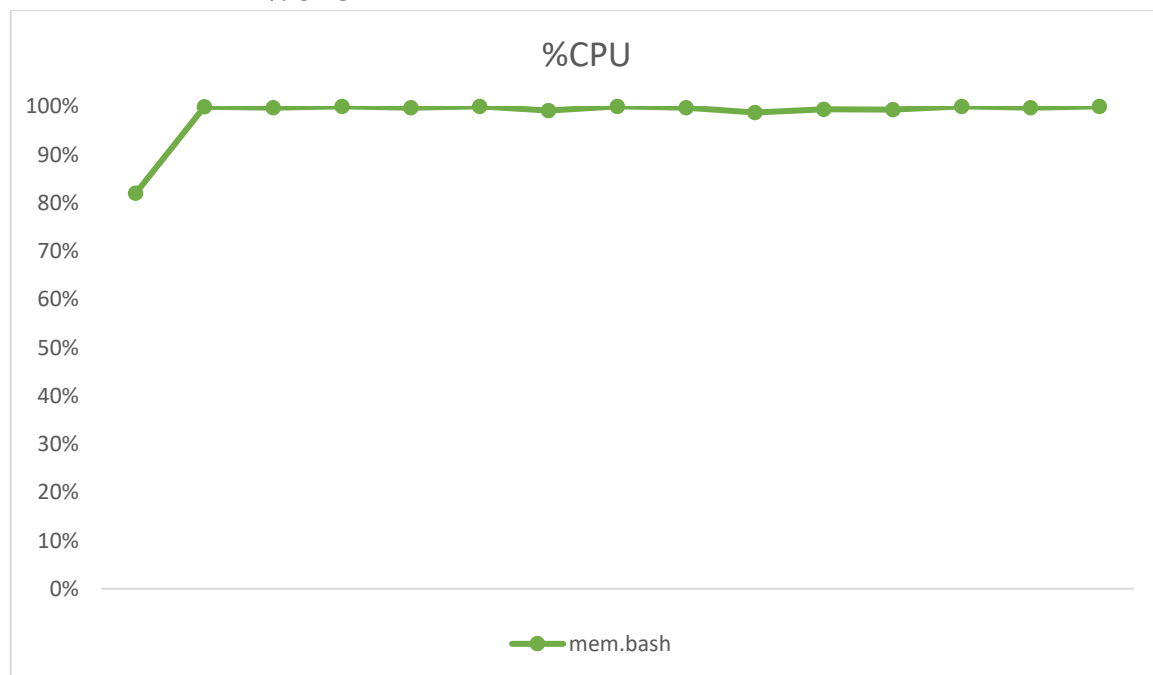
```
[64891.421912] Out of memory: Killed process 4331 (mem.bash) total-vm:2610632kB, anon-rss:1644760kB,
file-rss:0kB, shmem-rss:0kB, UID:1000
[64891.511609] oom_reaper: reaped process 4331 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
[user@localhost lab5]$
```

4) Диаграмма изменения состояния памяти:



Можно увидеть, что система сначала использовала оперативную память, а когда оперативная закончилась, система начала использовать файлы подкачки. Когда вся свободная память заканчивается, система вынуждена убить процесс.

5) Диаграмма с показанием %CPU



%CPU почти все время выполнения скрипта mem.bash составляет 100% от общего объема нагрузки процессов.

6) Изменения в пяти верхних процессах:

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4396	user	20	0	1184768	965272	3000	R	99.7	51.6	0:21.61	mem.bash
4371	root	20	0	0	0	0	R	0.3	0.0	0:00.48	kworker/0:3-events
1	root	20	0	187384	6556	4148	S	0.0	0.4	0:04.38	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.06	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp

За все время исполнения скрипта в топ скрипт mem.bash был всегда на первом месте.

### Второй этап эксперимента:

Создадим копию скрипты mem.bash с названием mem2bash и сделаем для него report2.log:

```
#!/bin/bash

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

Создадим скрипт launcher.bash, который будет запускать скрипт mem.bash и mem2.bash:

```
GNU nano 2.9.8 launcher.bash

#!/bin/bash

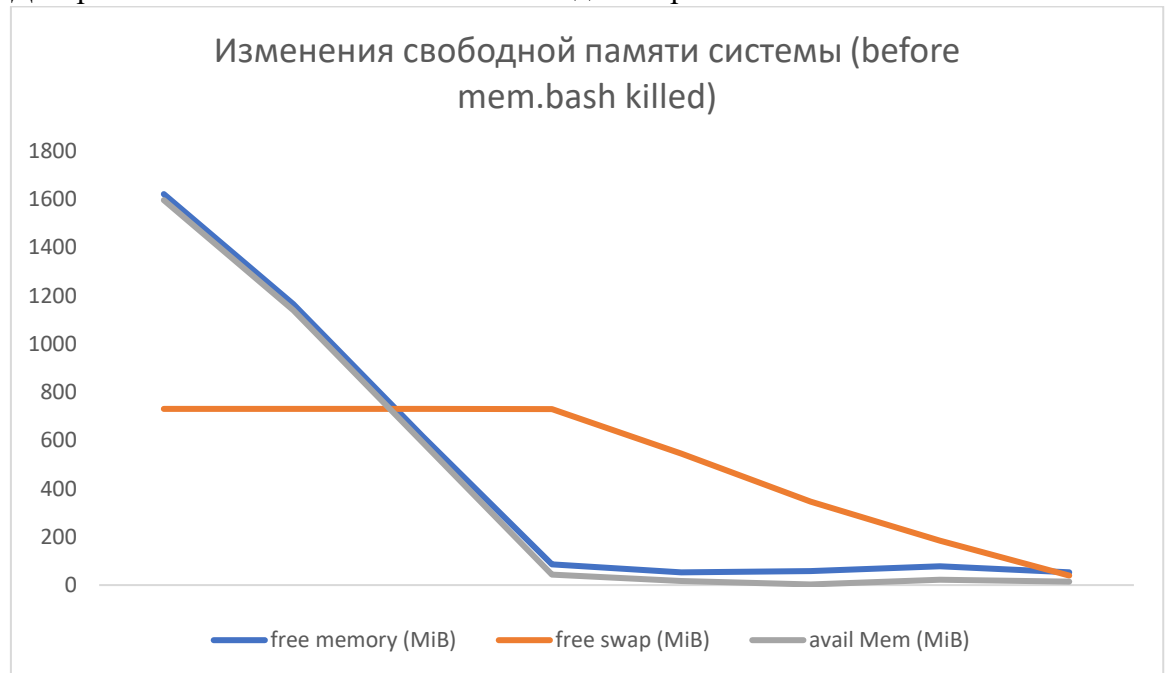
./mem.bash &
./mem2.bash
```

- 1) Значение в report.log: 15000000  
Значение в report2.log: 30000000

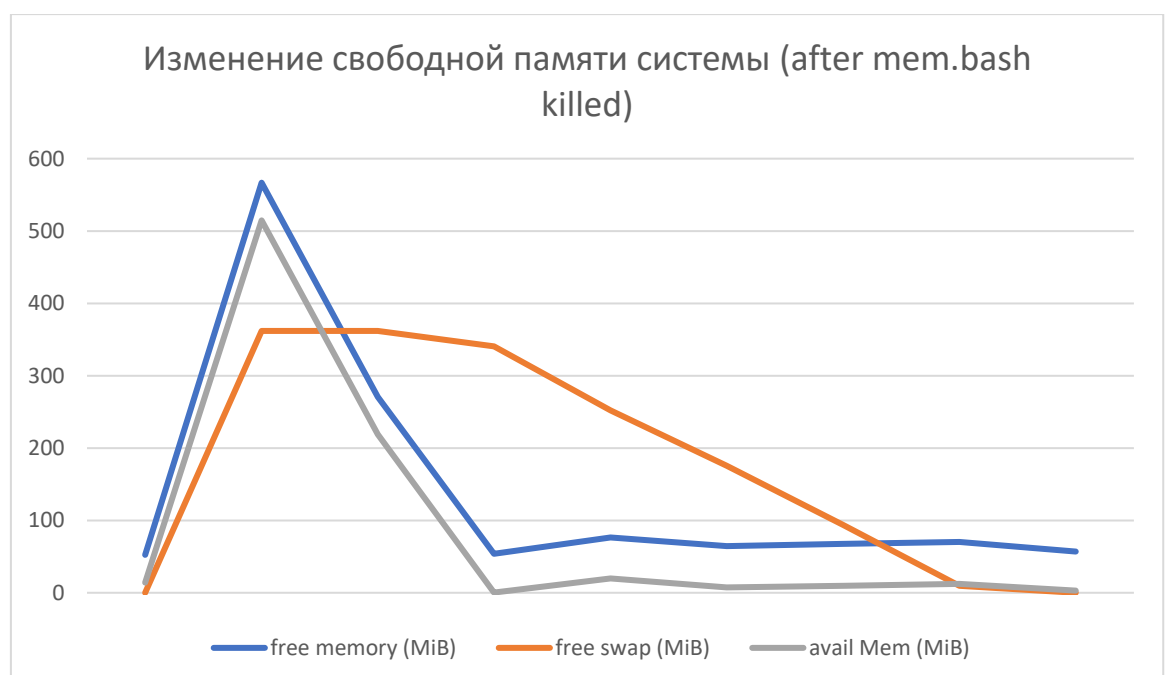
- 2) Последние две записи о скриптах в системной журнале:

```
[85290.152631] Out of memory: Killed process 4571 (mem.bash) total-vm:1428308kB, anon-rss:833756kB,
file-rss:0kB, shmem-rss:0kB, UID:1000
[85290.212072] oom_reaper: reaped process 4571 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:
0kB
[85316.390535] [ 4572] 1000 4572 656651 415504 4894720 185571 0 mem2.bash
[85316.391649] Out of memory: Killed process 4572 (mem2.bash) total-vm:2626604kB, anon-rss:1662016kB
, file-rss:0kB, shmem-rss:0kB, UID:1000
[85316.510363] oom_reaper: reaped process 4572 (mem2.bash), now anon-rss:0kB, file-rss:0kB, shmem-rs
s:0kB
```

- 3) Диаграмма изменения состояния памяти до завершения mem.bash:



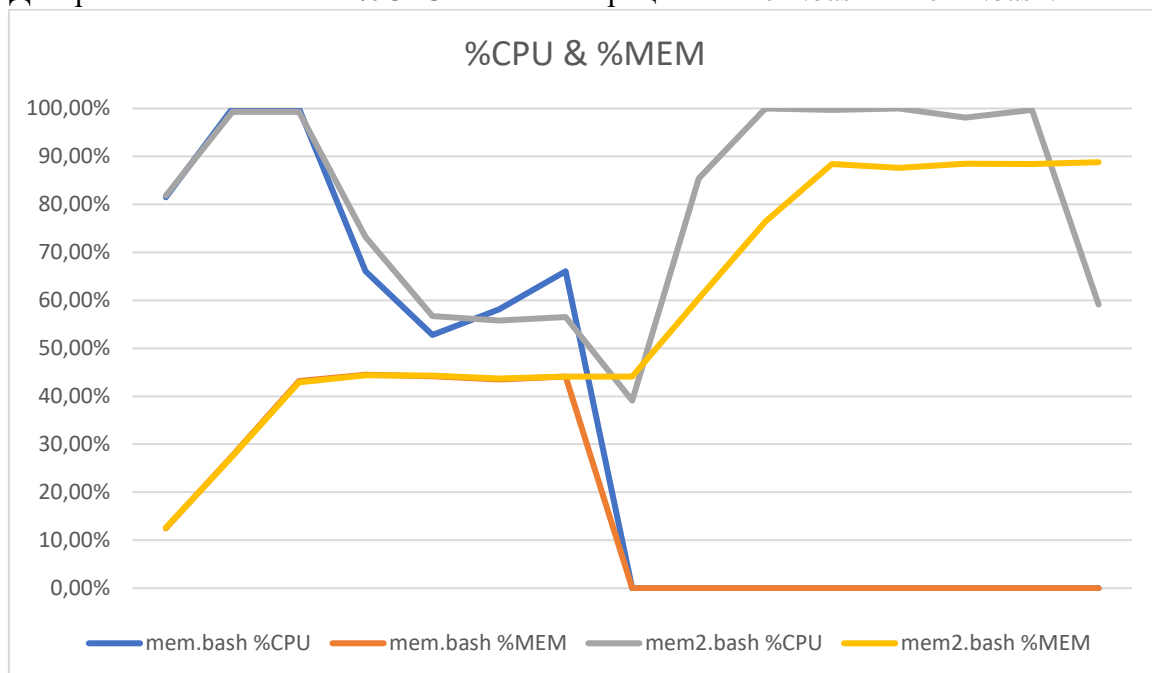
- 4) Диаграмма изменения состояния памяти после завершения mem.bash и до завершения mem2.bash:



Как и в прошлом этапе, система сначала использует оперативную память, потом файлы подкачки, а когда заканчивается свободная память, убивает первый процесс `mem.bash` и тем самым освобождает память для второго, но вскоре память кончится и у второго процесса `mem2.bash` и убивает его.

Можно заметить, что размер массива первой программы составляет примерно половину от размера второй, что подтверждает выше сказанные слова. В момент, когда заканчивается оперативная память, в списке процессов можно увидеть процесс `kswapd0`, который работает с оперативной памятью и ее выделением.

5) Диаграмма с показанием %CPU и %MEM процессов `mem.bash` и `mem2.bash`:



По диаграмме можно увидеть как работает выделение памяти и как оно влияет на процессы `mem.bash` и `mem2.bash`. То, что я написал выше полностью соответствует тому, что мы видим на диаграмме.

## Эксперимент 2:

### Подготовительный этап эксперимента:

- 1) Создадим копию `mem.bash` в файл `newmem.bash` с изменениями, а также создадим вспомогательный скрипт, который будет запускать скрипт `newmem.bash` с интервалом в 1 секунду:

```
GNU nano 2.9.8 newmem.bash

#!/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 launcher2.bash

#!/bin/bash

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

Основной этап эксперимента:

- 1)  $N = 3100000$  и  $K = 10$ : При запуске скрипта newmem.bash с ограничением на размер массива  $N = 4500000$  все скрипты завершились успешно. Ошибок в системном журнале нет.
- 2)  $N = 3100000$  и  $K = 30$ : При запуске нашего скрипта после нескольких выполненных итераций начали убиваться наши процессы. Если брать размер массива немного меньше, то проблем не возникает:

```

[93888.488216] newmem.bash invoked oom-killer: gfp_mask=0x6280ca(GFP_HIGHUSER_MOVABLE|__GFP_ZERO), n
odemask=(null), order=0, oom_score_adj=0
[93888.489279] newmem.bash cpuset=/ mems_allowed=0
[93888.489884] CPU: 1 PID: 6425 Comm: newmem.bash Kdump: loaded Tainted: G
r- - 4.18.0-193.el8.x86_64 #1
[93888.537995] [ 6411] 1000 6411 136076 19455 720896 60985 0 newmem.bash
[93888.538299] [ 6421] 1000 6421 118652 37363 581632 25738 0 newmem.bash
[93888.538613] [ 6423] 1000 6423 132743 54467 688128 22689 0 newmem.bash
[93888.538911] [ 6425] 1000 6425 114428 45618 540672 13217 0 newmem.bash
[93888.539207] [ 6427] 1000 6427 117794 57067 569344 5174 0 newmem.bash
[93888.539524] [ 6429] 1000 6429 105287 21711 479232 27997 0 newmem.bash
[93888.539815] [ 6431] 1000 6431 105386 48943 475136 848 0 newmem.bash
[93888.540106] [ 6433] 1000 6433 109445 40375 507904 13485 0 newmem.bash
[93888.540408] [ 6435] 1000 6435 120632 53268 602112 11766 0 newmem.bash
[93888.540695] [ 6437] 1000 6437 94034 35072 380928 3361 0 newmem.bash
[93888.540983] Out of memory: Killed process 6411 (newmem.bash) total-vm:544304kB, anon-rss:77816kB,
file-rss:4kB, shmem-rss:0kB, UID:1000
[93888.579881] oom_reaper: reaped process 6411 (newmem.bash), now anon-rss:0kB, file-rss:0kB, shmem-
rss:0kB

```

Некоторые процессы завершились без ошибок, а некоторые аварийно из-за того, что свободная память закончилась раньше, чем все процессы успели выполниться. Путем подбора значений было выявлено, что оптимальное N на данный момент равнялся 2900000. Дальше уже есть высокая вероятность того, что все 30 процессов не смогут выполниться и некоторые из них завершаться аварийно.