

**Министерство науки и высшего образования РФ**  
**Федеральное государственное автономное**  
**образовательное учреждение высшего образования**  
**«Национальный исследовательский университет ИТМО»**

Отчет по дисциплине “Операционные системы”.

*Управление памятью в ОС Linux.*

**Выполнил:**

Лопатенко Георгий Валентинович

М32021

**Преподаватель:**

Хегай Максим Вилорьевич

Санкт-Петербург

2022

## Отчет к лабораторной работе

```
[user@localhost proc]$ free
```

	total	used	free	shared	buff/cache	available
Mem:	1870900	314096	1216900	8956	339904	1399132
Swap:	839676	0	839676			

```
[user@localhost proc]$ cat meminfo
```

MemTotal:	1870900	kB
MemFree:	1232072	kB
MemAvailable:	1413268	kB
Buffers:	16056	kB
Cached:	287464	kB
SwapCached:	0	kB
Active:	284644	kB
Inactive:	219656	kB
Active(anon):	201272	kB
Inactive(anon):	8456	kB
Active(file):	83372	kB
Inactive(file):	211200	kB
Unevictable:	0	kB
Mlocked:	0	kB
SwapTotal:	839676	kB
SwapFree:	839676	kB
Dirty:	0	kB
Writeback:	0	kB
AnonPages:	198768	kB
Mapped:	116956	kB
Shmem:	8956	kB

### Начальные данные системы:

Общий объем RAM: 1827 МБ

Общий объем раздела подкачки: 817.4 МБ

Размер страницы виртуальной памяти: 4 КБ

### В ненагруженной системе:

Объем свободной RAM: 1480.2 КБ

Объем раздела подкачки: 666.9 КБ

## Эксперимент 1

### Этап 1

```
[user@localhost experiment1]$ ./topparse.sh&pid0=$!  
[1] 1669  
[user@localhost experiment1]$ ./topparse.sh: line 10: 1670 Killed  
./mem.sh
```

```
820.991082] [ 4148] 1000 4148 65926 139 151552 0 0 top  
820.991368] [ 4149] 1000 4149 54264 16 65536 0 0 head  
820.991642] [ 4150] 1000 4150 54270 16 65536 0 0 tail  
820.991925] [ 4151] 1000 4151 58791 65 102400 0 0 awk  
820.992207] Out of memory: Killed process 3850 (mem.sh) total-vm:2529452kB, anon-rss:1639480kB, file-rss:4kB, shmem-rss:0kB, UID:1000  
821.115201] oom_reaper: reaped process 3850 (mem.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB  
[user@localhost experiment1]$
```

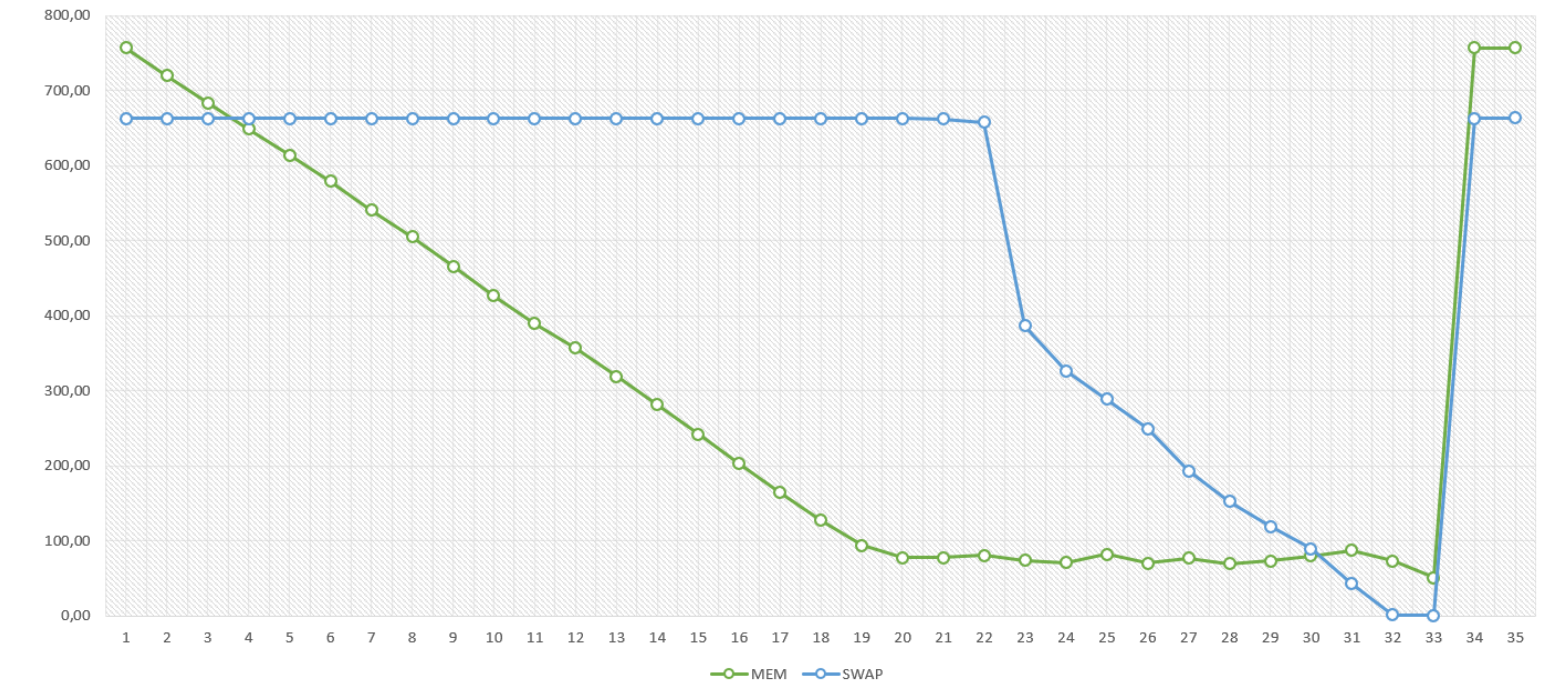
```
labs > blab5 > experiment1 > report.log
```

1	1000000
2	2000000
3	3000000
4	4000000
5	5000000
6	6000000
7	7000000
8	8000000
9	9000000
10	10000000
11	11000000
12	12000000
13	13000000
14	14000000
15	15000000
16	16000000
17	17000000
18	18000000
19	19000000
20	20000000
21	

```
labs > blab5 > experiment1 > topdata
```

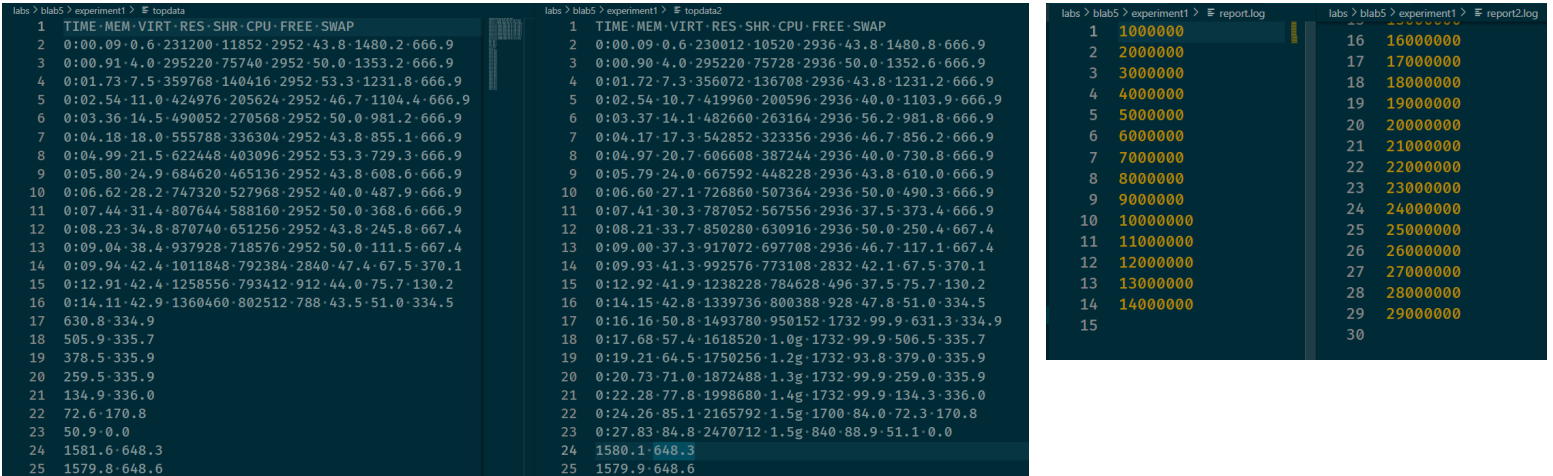
1	TIME	MEM	VIRT	RES	SHR	CPU	FREE	SWAP
2	0:00.15	0.7	232124	12672	2988	81.2	757.5	662.8
3	0:00.69	2.7	270140	50688	2988	93.8	720.4	662.8
4	0:01.24	4.7	307496	88172	2988	93.3	683.6	662.8
5	0:01.79	6.5	341420	121964	2988	87.5	649.2	662.8
6	0:02.33	8.5	378116	158660	2988	93.8	614.2	662.8
7	0:02.87	10.3	412040	192716	2988	93.3	579.3	662.8
8	0:03.41	12.4	450584	231260	2988	99.9	540.7	662.8
9	0:03.97	14.3	487544	268220	2988	93.8	505.4	662.8
10	0:04.52	16.4	526220	306764	2988	93.8	466.1	662.8
11	0:05.06	18.5	566216	346892	2988	99.9	427.0	662.8
12	0:05.62	20.6	604760	385436	2988	87.5	389.5	662.8
13	0:06.16	22.6	642908	423452	2988	93.3	357.0	662.8
14	0:06.72	24.7	682244	462788	2988	93.3	319.2	662.8
15	0:07.26	26.8	720656	501332	2988	93.8	281.3	662.8
16	0:07.81	28.9	759596	540140	2988	93.3	242.0	662.8
17	0:08.36	31.0	799724	580268	2988	99.9	203.0	662.8
18	0:08.91	33.1	838400	619076	2988	99.9	164.5	662.8
19	0:09.46	35.2	878660	659204	2988	93.8	127.4	663.1
20	0:09.99	37.2	915356	695900	2988	93.3	93.9	663.1
21	0:10.53	39.1	950468	731012	2988	93.3	77.8	663.3
22	0:11.12	41.3	992576	773252	2988	99.9	77.9	662.4
23	0:11.80	43.9	1040096	820772	2988	99.9	80.8	657.9
24	0:14.13	52.7	1208132	986808	968	75.0	74.0	386.4
25	0:16.62	62.3	1387388	1.1g	860	70.6	70.7	326.4
26	0:17.18	64.4	1426724	1.1g	860	75.0	82.3	288.7
27	0:17.76	66.6	1468568	1.2g	860	99.9	70.6	249.4
28	0:18.34	68.8	1509092	1.2g	860	75.0	76.9	193.6
29	0:19.01	71.5	1558592	1.3g	860	86.7	69.5	152.9
30	0:19.55	73.5	1597004	1.3g	860	81.2	73.4	119.5
31	0:20.06	75.4	1631852	1.3g	860	86.7	80.3	89.5
32	0:20.53	77.2	1664984	1.4g	860	56.2	87.4	43.2
33	0:21.40	80.5	1727024	1.4g	860	86.7	73.3	1.4
34	0:21.97	82.5	1764908	1.5g	860	87.5	51.1	0.8
35	1524.9	1.0						

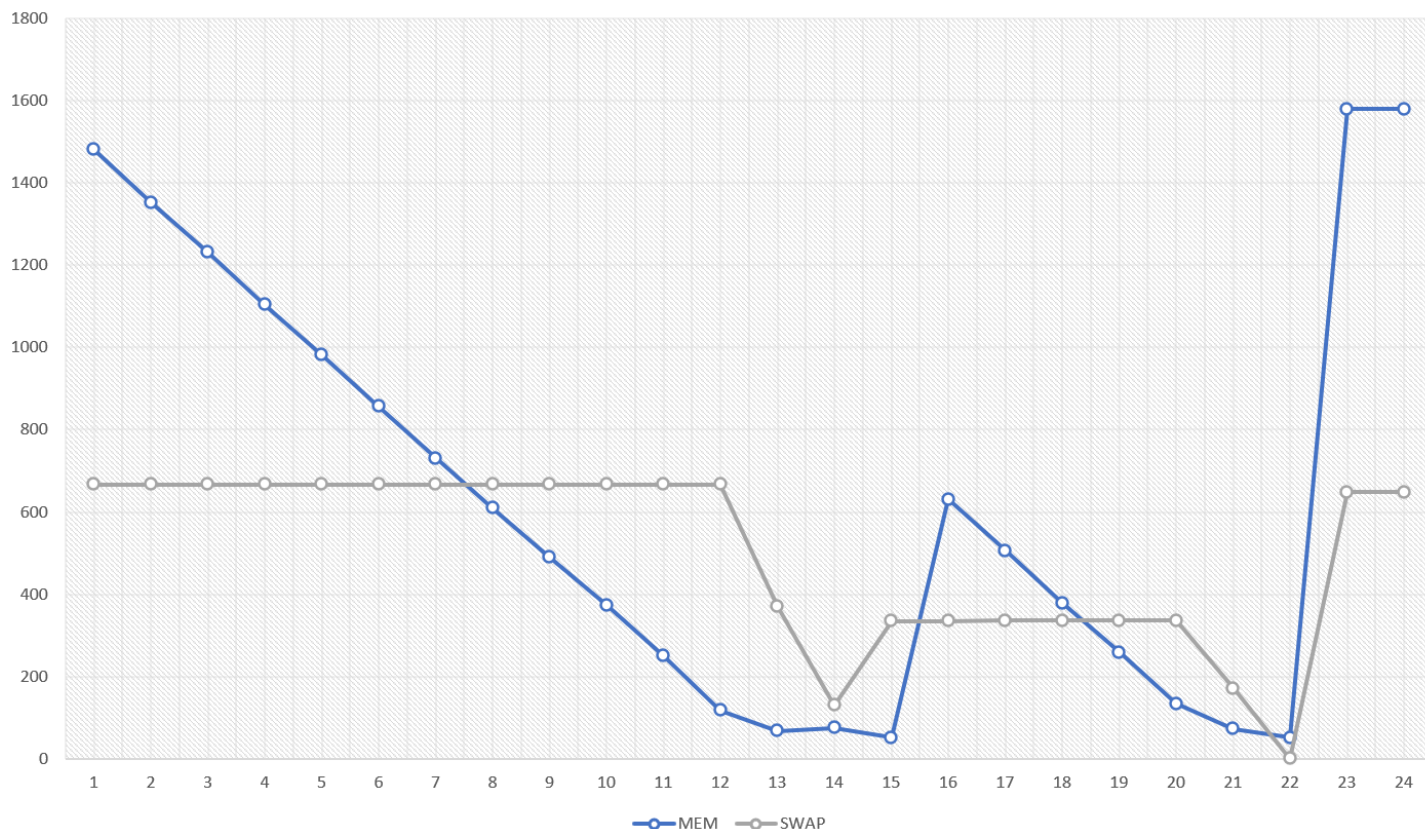
После запуска `mem.bash` этот процесс занимал первую строку `top` и оставался там до своего аварийного завершения. Также появлялся процесс-обработчик результатов `top`. После того, как свободная физическая память закончилась, появился демон-процесс `kswapd0`, который отвечает за управление разделом подкачки, соответственно, с этого момента значения доступной памяти `swar` начали уменьшаться до состояния полностью исчерпанного ресурса - в этот момент процесс `mem.bash` был убит и `swar` затерся.



Этап 2

Если же запустить 2 тестируемых процесса одновременно, то несложно убедиться, что сначала завершился процесс `mem.bash`, а после - `mem2.bash`. В моменте перехода из состояния выполнения двух процессов в состояние одного тестового процесса освободился некоторый объем `swar` памяти.





## Эксперимент 2

Запустим  $K$  процессов, для которых предел аллоцируемого массива будет составлять  $N$ .

Проверяем, что для  $N = \frac{N_{max}}{10}$  скрипт не завершается аварийно, а выходит при *reached array limit* инструкции.

```
[user@localhost experiment2]$ ./runner.sh 10 2900000
reached array limit and stopped
[user@localhost experiment2]$ reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
```

```
./runner.sh 40 2900000
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
./runner.sh: line 3: 3604 Killed                  ./newmem.bash $2
./runner.sh: line 3: 3606 Killed                  ./newmem.bash $2
./runner.sh: line 3: 3608 Killed                  ./newmem.bash $2
./runner.sh: line 3: 3610 Killed                  ./newmem.bash $2
./runner.sh: line 3: 3612 Killed                  ./newmem.bash $2
```

```
./runner.sh 33 2900000
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
./runner.sh: line 3: 3894 Killed
[user@localhost experiment2]$ ./newmem.bash $2
```

```
./runner.sh 33 2650000
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
./runner.sh: line 3: 4191 Killed
[user@localhost experiment2]$ ./newmem.bash $2
```

```
./runner.sh 33 2600000
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
[user@localhost experiment2]$ reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
reached array limit and stopped
```

Такие настройки конфигурации можно объяснить тем, что процессы успевают выйти по инструкции, высвобождая при этом swar.