

Отчёт по лабораторной работе №5

Управление памятью в ОС Linux

Жунусов Данияр М3233

Данные о текущей конфигурации операционной системы

- Общий объем оперативной памяти: 4010504 kB
- Объем раздела подкачки: 2744316 kB
- Размер страницы виртуальной памяти: 4 kB
- Объем свободной физической памяти в ненагруженной системе: 1367908 kB
- Объем свободного пространства в разделе подкачки в ненагруженной системе: 2744316 kB

Эксперимент №1

Первый этап

Запущенный скрипт

```
#!/bin/bash

> report.log

array=()
counter=0

while true; do
    array+=({1..10})

    ((counter++))

    if ((counter % 100000 == 0)); then
        echo "Step: $counter, Array size: ${#array[@]}" >> report.log
    fi
done
```

Рис. 1: mem.bash

Наблюдение после запуска скрипта

```

top - 22:28:04 up 1 min, 1 user, load average: 0,75, 0,45, 0,18
Tasks: 200 total, 2 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 39,4 us, 14,1 sy, 0,0 ni, 42,1 id, 1,2 wa, 0,0 hi, 3,2 si, 0,0 st
МиБ Mem : 3916,5 total, 1790,2 free, 1395,0 used, 731,4 buff/cache
МиБ Swap: 2680,0 total, 2680,0 free, 0,0 used. 2264,9 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2247	danial	20	0	660608	653824	3200	R	100,0	16,3	0:07.08	mem.bash
1578	danial	20	0	3759352	259160	132480	S	2,0	6,5	0:05.68	gnome-s+
321	root	0	-20	0	0	0	I	1,0	0,0	0:00.25	kworker+
1493	danial	39	19	650096	38988	23264	S	0,7	1,0	0:00.77	tracker+
1659	danial	20	0	997948	74624	52728	S	0,7	1,9	0:01.02	nautilus
11	root	20	0	0	0	0	I	0,3	0,0	0:00.87	kworker+
33	root	20	0	0	0	0	I	0,3	0,0	0:00.09	kworker+
178	root	20	0	0	0	0	I	0,3	0,0	0:00.10	kworker+
230	root	20	0	0	0	0	I	0,3	0,0	0:00.06	kworker+
2011	danial	20	0	557468	56156	42600	S	0,3	1,4	0:00.82	gnome-t+
2251	danial	39	19	400276	32556	25644	S	0,3	0,8	0:00.06	tracker+
1	root	20	0	101188	11504	8176	S	0,0	0,3	0:01.38	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_wo+
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker+

Рис. 2: Запущенный скрипт появился в top

```

top - 22:28:25 up 2 min, 1 user, load average: 0,90, 0,51, 0,20
Tasks: 200 total, 2 running, 198 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8,3 us, 16,7 sy, 0,0 ni, 75,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
МиБ Mem : 3916,5 total, 211,7 free, 2971,2 used, 733,6 buff/cache
МиБ Swap: 2680,0 total, 2680,0 free, 0,0 used. 688,6 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2247	danial	20	0	2407496	2,3g	3200	R	75,0	59,9	0:27.05	mem.bash
1578	danial	20	0	3766656	259480	132800	S	16,7	6,5	0:06.18	gnome-shell
2011	danial	20	0	557756	56664	42980	S	16,7	1,4	0:00.96	gnome-terminal-
16	root	20	0	0	0	0	S	8,3	0,0	0:00.09	ksoftirqd/0
51	root	20	0	0	0	0	S	8,3	0,0	0:00.09	kswapd0
1	root	20	0	101188	11504	8176	S	0,0	0,3	0:01.38	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_r+
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-netns
8	root	20	0	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0-even+
9	root	20	0	0	0	0	I	0,0	0,0	0:00.02	kworker/0:1-even+
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-eve+
11	root	20	0	0	0	0	I	0,0	0,0	0:00.87	kworker/u4:0-eve+
12	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-mm_pe
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_kthread
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_rude_k+

Рис. 3: Спустя 20 секунд скрипт занял около 2,3 Гб физической памяти


```

top - 22:28:58 up 2 min, 1 user, load average: 1,40, 0,67, 0,27
Tasks: 201 total, 4 running, 197 sleeping, 0 stopped, 0 zombie
%Cpu(s): 31,3 us, 44,2 sy, 0,2 ni, 17,4 id, 1,2 wa, 0,0 hi, 5,7 si, 0,0 st
МиБ Mem : 3916,5 total, 122,6 free, 3651,1 used, 142,8 buff/cache
МиБ Swap: 2680,0 total, 1227,3 free, 1452,7 used. 70,4 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2247	daniar	20	0	4423796	3,1g	2688	R	99,3	81,9	1:00.15	mem.bash
51	root	20	0	0	0	0	R	64,6	0,0	0:20.76	kswapd0
17	root	20	0	0	0	0	I	1,0	0,0	0:00.75	rcu_preempt
782	root	20	0	1320812	11616	5760	S	1,0	0,3	0:01.40	snappd
24	root	20	0	0	0	0	S	0,7	0,0	0:00.38	ksoftirqd/1
50	root	0	-20	0	0	0	I	0,3	0,0	0:00.15	kworker/0:1H-kbl+
60	root	20	0	0	0	0	R	0,3	0,0	0:00.20	kworker/u4:4-loo+
321	root	0	-20	0	0	0	I	0,3	0,0	0:00.51	kworker/1:2H-kbl+
482	systemd+	20	0	14836	6272	6016	S	0,3	0,2	0:00.25	systemd-oomd
1493	daniar	39	19	650096	25576	18784	S	0,3	0,6	0:01.16	tracker-miner-f
1578	daniar	20	0	3766672	86292	46528	S	0,3	2,2	0:06.42	gnome-shell
2256	daniar	20	0	13292	3840	3200	R	0,3	0,1	0:00.21	top
1	root	20	0	101188	10224	8176	S	0,0	0,3	0:01.38	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_r+
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-netns

Рис. 4: Спустя 1 минуту скрипт занял всю доступную физическую память и начался процесс подкачки (swap)

```

top - 22:29:26 up 3 min,  1 user,  load average: 1,61, 0,80, 0,33
Tasks: 201 total,   8 running, 193 sleeping,   0 stopped,   0 zombie
%Cpu(s): 25,4 us, 44,6 sy,  0,0 ni, 12,0 id,  6,0 wa,  0,0 hi, 12,0 si,  0,0 st
МиБ Mem :  3916,5 total,   103,9 free,  3787,5 used,   25,1 buff/cache
МиБ Swap:  2680,0 total,    0,0 free,  2680,0 used.   31,3 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2247	danlar	20	0	5812040	3,3g	1280	R	91,3	87,1	1:26.41	mem.bash
51	root	20	0	0	0	0	R	65,1	0,0	0:41.17	kswapd0
1659	danlar	20	0	997948	10368	2936	R	2,9	0,3	0:01.45	nautilus
50	root	0	-20	0	0	0	I	1,9	0,0	0:00.25	kworker/0:1H-kbl+
482	systemd+	20	0	14836	1152	896	R	1,6	0,0	0:00.32	systemd-oomd
1493	danlar	39	19	650096	7016	864	S	1,3	0,2	0:01.33	tracker-miner-f
1578	danlar	20	0	3766672	38036	6592	R	1,3	0,9	0:06.65	gnome-shell
1924	danlar	20	0	2793640	9000	1152	R	1,3	0,2	0:00.78	gjs
17	root	20	0	0	0	0	I	1,0	0,0	0:01.03	rcu_preempt
2256	danlar	20	0	13292	1536	1024	R	1,0	0,0	0:00.30	top
321	root	0	-20	0	0	0	I	0,6	0,0	0:00.60	kworker/1:2H-kbl+
16	root	20	0	0	0	0	S	0,3	0,0	0:00.11	ksoftirqd/0
24	root	20	0	0	0	0	R	0,3	0,0	0:00.45	ksoftirqd/1
33	root	20	0	0	0	0	I	0,3	0,0	0:00.11	kworker/u4:2-flu+
178	root	20	0	0	0	0	I	0,3	0,0	0:00.18	kworker/0:2-even+
1449	danlar	20	0	16344	1536	128	S	0,3	0,0	0:00.65	dbus-daemon
2011	danlar	20	0	557756	7896	4196	S	0,3	0,2	0:01.08	gnome-terminal-
1	root	20	0	101188	2032	112	S	0,0	0,1	0:01.38	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd

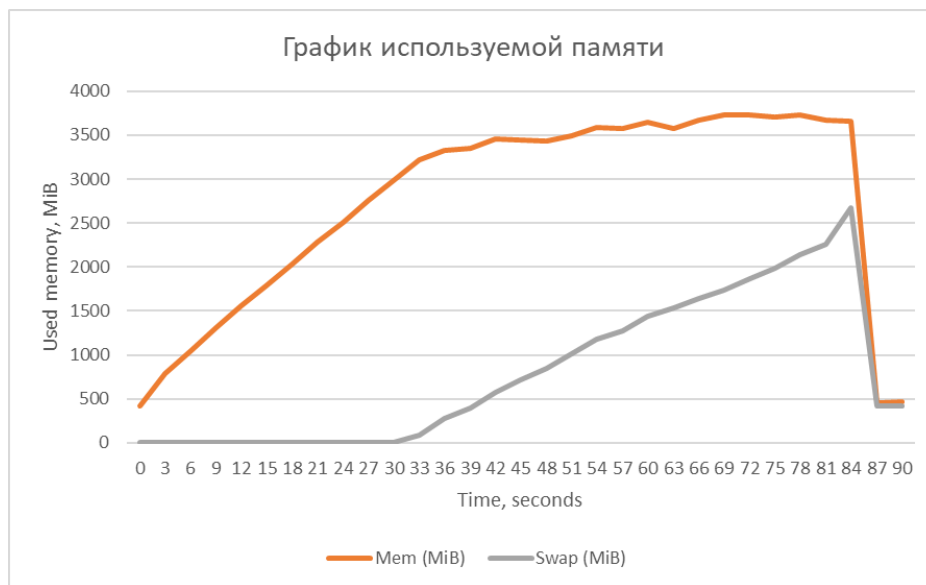
Рис. 5: Спустя 1.5 минуты скрипт всю свободную память после чего аварийно завершился

Последние две записи о скрипте в системном журнале:

```
[ 192.131240] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),
cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-1000.
slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-
spawn-83e765f0-27da-46d5-a532-06d4429f1941.scope,task=mem.bash,pid
=2247,uid=1000

[ 192.131262] Out of memory: Killed process 2247 (mem.bash) total-vm
:5835536kB, anon-rss:3515776kB, file-rss:0kB, shmem-rss:0kB, UID
:1000 pgtables:11460kB oom_score_adj:0
```

Значение в последней строке файла report.log: 74'000'000



На графике видно, что использование физической памяти перестало увеличиваться одновременно с началом заполнения раздела подкачки. Это произошло из-за исчерпания доступной физической памяти, после чего система начала использовать swap для хранения неактивных данных. Рост обеих метрик прекратился, когда физическая память достигла своего предела в 3.7 Гб, а раздел подкачки — 2.7 Гб. По завершении заполнения памяти система аварийно завершила работу скрипта, так как свободной памяти больше не осталось.

Второй этап

Запущенный скрипт

```
#!/bin/bash

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

Рис. 6: run12.bash

```
#!/bin/bash

> report.log

array=()
counter=0

while true; do
    array+=({1..10})

    ((counter++))

    if ((counter % 100000 == 0)); then
        echo "Step: $counter, Array size: ${#array[@]}" >> report2.log
    fi
done
```

Рис. 7: mem2.bash

```
top - 19:20:34 up 19 min, 1 user, load average: 0,38, 0,21, 0,19
Tasks: 188 total, 4 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us,100,0 sy, 0,0 ni, 0,0 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 3916,5 total, 1661,0 free, 1477,5 used, 778,0 buff/cache
MiB Swap: 2680,0 total, 2680,0 free, 0,0 used. 2179,2 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2456	danlar	20	0	376808	370304	3200	R	66,7	9,2	0:03.42	mem.bash
2457	danlar	20	0	376940	370304	3200	R	33,3	9,2	0:03.42	mem2.bash
1023	root	0	-20	0	0	0	I	16,7	0,0	0:00.01	kworker/u5:13-ttm
1486	danlar	20	0	3703284	258796	133164	S	16,7	6,5	0:07.32	gnome-shell
2126	danlar	20	0	558156	56768	42508	R	16,7	1,4	0:01.54	gnome-terminal-
1	root	20	0	167916	12980	8244	S	0,0	0,3	0:01.79	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-netns
10	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H-events_highpri
11	root	20	0	0	0	0	I	0,0	0,0	0:00.96	kworker/u4:0-events_unbound
12	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-mm_pe
13	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu_tasks_kthread
14	root	20	0	0	0	0	I	0,0	0,0	0:00.00	rcu tasks rude kthread

Рис. 8: После запуска run12.bash оба скрипта появились в top


```
top - 19:20:58 up 20 min, 1 user, load average: 1,26, 0,42, 0,26
Tasks: 188 total, 4 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 43,2 us, 46,0 sy, 0,3 ni, 0,3 id, 0,0 wa, 0,0 hi, 10,2 si, 0,0 st
MiB Mem : 3916,5 total, 127,1 free, 3613,2 used, 176,2 buff/cache
MiB Swap: 2680,0 total, 1310,3 free, 1369,7 used. 89,9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2456	danial	20	0	2156168	1,5g	2688	R	74,1	40,3	0:22.26	mem.bash
2457	danial	20	0	2149172	1,5g	2816	R	68,1	40,2	0:21.94	mem2.bash
51	root	20	0	0	0	0	R	48,8	0,0	0:10.33	kswapd0
2482	danial	39	19	400280	32496	25584	S	2,7	0,8	0:00.08	tracker-extract
1405	danial	39	19	649052	26656	18072	S	1,3	0,7	0:00.86	tracker-miner-f
17	root	20	0	0	0	0	I	1,0	0,0	0:00.65	rcu_preempt
306	root	0	-20	0	0	0	I	0,7	0,0	0:00.34	kworker/1:2H-kblockd
24	root	20	0	0	0	0	S	0,3	0,0	0:00.23	ksoftirqd/1
50	root	0	-20	0	0	0	I	0,3	0,0	0:00.21	kworker/0:1H-kblockd
165	root	20	0	0	0	0	S	0,3	0,0	0:00.17	jbd2/sda3-8
1363	danial	20	0	12668	5388	4096	S	0,3	0,1	0:00.52	dbus-daemon
1374	danial	20	0	240912	7680	7040	S	0,3	0,2	0:00.04	gvfsd
1502	danial	20	0	316508	10112	8704	S	0,3	0,3	0:00.10	gvfs-udisks2-vo
1828	danial	20	0	2801876	46348	33000	S	0,3	1,2	0:00.79	gjs
2084	danial	20	0	939732	45608	33088	S	0,3	1,1	0:03.07	nautilus
1	root	20	0	167916	10036	8244	S	0,0	0,3	0:01.79	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd

Рис. 9: Спустя 20 секунд каждый скрипт занял по 1.5 Гб физической памяти

```
top - 19:21:20 up 20 min, 1 user, load average: 2,39, 0,76, 0,37
Tasks: 187 total, 3 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 8,7 us, 3,4 sy, 0,0 ni, 43,3 id, 31,7 wa, 0,0 hi, 12,9 si, 0,0 st
MiB Mem : 3916,5 total, 947,9 free, 2850,6 used, 118,0 buff/cache
MiB Swap: 2680,0 total, 1023,5 free, 1656,5 used. 890,0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2457	danial	20	0	3789800	2,4g	1536	R	98,7	63,2	0:36.18	mem2.bash
1486	danial	20	0	3699824	51180	21164	S	1,3	1,3	0:08.77	gnome-shell
2496	danial	39	19	382644	23296	19712	R	1,3	0,6	0:00.04	tracker-extract
50	root	0	-20	0	0	0	I	0,7	0,0	0:00.40	kworker/0:1H-kblockd
306	root	0	-20	0	0	0	I	0,7	0,0	0:00.88	kworker/1:2H-kblockd
1405	danial	39	19	649052	13600	7960	S	0,7	0,3	0:01.38	tracker-miner-f
2084	danial	20	0	939732	18856	10688	S	0,7	0,5	0:04.02	nautilus
17	root	20	0	0	0	0	I	0,3	0,0	0:00.82	rcu_preempt
24	root	20	0	0	0	0	S	0,3	0,0	0:00.32	ksoftirqd/1
1507	danial	20	0	237652	3200	2816	S	0,3	0,1	0:00.02	gvfs-gphoto2-vo
2126	danial	20	0	558156	16448	11404	S	0,3	0,4	0:02.27	gnome-terminal-
1	root	20	0	167916	3636	2996	S	0,0	0,1	0:01.87	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_

Рис. 10: Спустя 35 секунд скрипты заняли всю память после чего mem.bash был аварийно остановлен

```

top - 19:21:37 up 20 min, 1 user, load average: 2,54, 0,87, 0,42
Tasks: 187 total, 3 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 39,5 us, 49,5 sy, 0,0 ni, 6,3 id, 0,3 wa, 0,0 hi, 4,4 si, 0,0 st
МИБ Mem : 3916,5 total, 98,7 free, 3669,7 used, 148,1 buff/cache
МИБ Swap: 2680,0 total, 487,6 free, 2192,3 used. 55,9 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2457	danlar	20	0	5178308	3,2g	1536	R	98,3	84,0	0:52.34	mem2.bash
51	root	20	0	0	0	0	R	77,4	0,0	0:27.51	kswapd0
306	root	0	-20	0	0	0	I	6,0	0,0	0:01.33	kworker/1:2H-kblockd
50	root	0	-20	0	0	0	I	3,0	0,0	0:00.59	kworker/0:1H-kblockd
17	root	20	0	0	0	0	I	0,7	0,0	0:00.94	rcu_preempt
1405	danlar	39	19	649052	13856	8088	S	0,7	0,3	0:01.48	tracker-miner-f
1486	danlar	20	0	3699824	53996	23596	S	0,3	1,3	0:08.92	gnome-shell
2084	danlar	20	0	939732	19752	11200	S	0,3	0,5	0:04.09	nautilus
2126	danlar	20	0	558156	17088	11788	S	0,3	0,4	0:02.32	gnome-terminal-
2454	danlar	20	0	13144	2048	1664	R	0,3	0,1	0:00.32	top
1	root	20	0	167916	3636	2996	S	0,0	0,1	0:01.87	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_
7	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-netns

Рис. 11: Из-за аварийной остановки скрипта mem.bash память, выделенная ему, была освобождена. После этого освобожденная память стала доступна другим процессам, включая запущенный параллельно скрипт mem2.bash, который занял её для выполнения своих операций.

```

top - 19:21:49 up 21 min, 1 user, load average: 2,70, 0,99, 0,46
Tasks: 187 total, 3 running, 184 sleeping, 0 stopped, 0 zombie
%Cpu(s): 23,4 us, 45,6 sy, 0,0 ni, 19,2 id, 3,6 wa, 0,0 hi, 8,2 si, 0,0 st
МИБ Mem : 3916,5 total, 108,3 free, 3784,6 used, 23,6 buff/cache
МИБ Swap: 2680,0 total, 0,0 free, 2680,0 used. 3,2 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2457	danlar	20	0	5801480	3,3g	768	R	85,6	87,1	1:03.75	mem2.bash
51	root	20	0	0	0	0	R	63,6	0,0	0:36.59	kswapd0
441	systemd+	20	0	14836	2048	1792	D	3,5	0,1	0:01.80	systemd-oomd
2084	danlar	20	0	939732	9768	1216	D	2,6	0,2	0:04.20	nautilus
1405	danlar	39	19	649052	6176	408	D	1,3	0,2	0:01.57	tracker-miner-f
2454	danlar	20	0	13144	1408	1024	R	1,3	0,0	0:00.38	top
50	root	0	-20	0	0	0	I	0,6	0,0	0:00.86	kworker/0:1H-kblockd
306	root	0	-20	0	0	0	I	0,6	0,0	0:02.09	kworker/1:2H-kblockd
17	root	20	0	0	0	0	I	0,3	0,0	0:01.04	rcu_preempt
695	root	20	0	16504	128	0	S	0,3	0,0	0:00.02	wpa_supplicant
1486	danlar	20	0	3699808	33132	2988	S	0,3	0,8	0:08.98	gnome-shell
1	root	20	0	167916	820	180	S	0,0	0,0	0:01.87	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.00	pool_workqueue_release
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_g
5	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-rcu_p
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/R-slub_

Рис. 12: Спустя 1 минуту скрипт mem2.bash занял всю память, после чего был аварийно остановлен.

Последние записи в системном журнале:

```
[ 1230.694152] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),
cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-1000.
slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-
spawn-19c30427-8b18-46e7-8dba-ab7787912078.scope,task=mem.bash,pid
=2456,uid=1000
[ 1230.694168] Out of memory: Killed process 2456 (mem.bash) total-vm
:2925860kB, anon-rss:1797120kB, file-rss:0kB, shmem-rss:0kB, UID
:1000 pgtables:5768kB oom_score_adj:0
[ 1268.604826] [ 2457] 1000 2457 1457861 880320 880320 0
0 11726848 575040 0 mem2.bash
[ 1268.604831] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),
cpuset=/,mems_allowed=0,global_oom,task_memcg=/user.slice/user-1000.
slice/user@1000.service/app.slice/app-org.gnome.Terminal.slice/vte-
spawn-19c30427-8b18-46e7-8dba-ab7787912078.scope,task=mem2.bash,pid
=2457,uid=1000
[ 1268.604849] Out of memory: Killed process 2457 (mem2.bash) total-vm
:5831444kB, anon-rss:3521280kB, file-rss:0kB, shmem-rss:0kB, UID
:1000 pgtables:11452kB oom_score_adj:0
```

Значение в последней строке файла report.log: 37'000'000

Значение в последней строке файла report2.log: 74'000'000

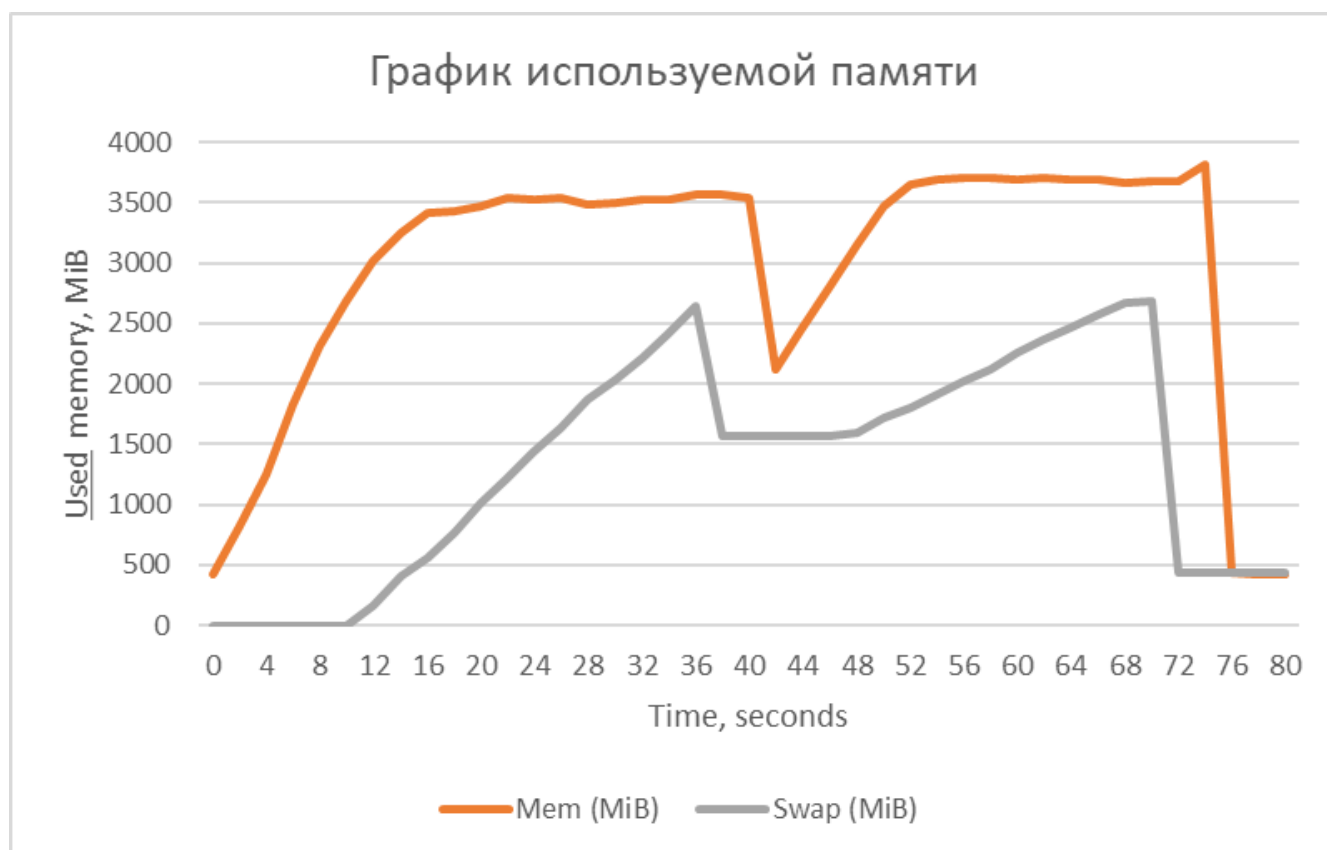
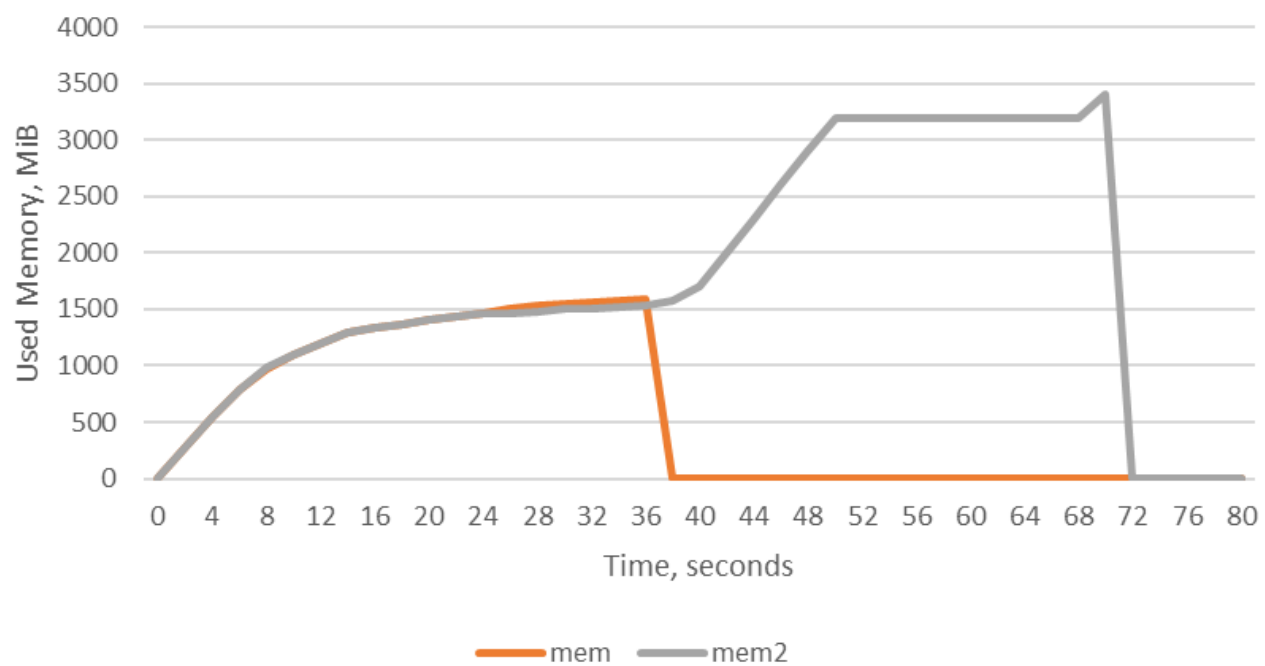


График используемой памяти скриптами



Выводы: На первом этапе эксперимента работающий процесс постепенно заполнил всю доступную физическую память, что привело к активации механизма подкачки, после чего был заполнен и раздел подкачки. Как только ресурсы виртуальной памяти оказались исчерпаны, процесс завершился аварийно, поскольку система больше не могла выделить новые страницы ни в физической, ни в виртуальной памяти. Во втором этапе процессы завершались поочередно: первым завершился `mem.sh`, так как на момент исчерпания памяти он использовал её больше, чем `mem2.sh`. После освобождения памяти первым процессом, `mem2.sh` продолжил выполнение, используя доступные ресурсы, однако позже также был завершён системой.

Эксперимент №2

Сначала запускаем скрипт с параметрами $n = 7400000$ (размер массива в 10 раз меньший при котором происходила аварийная остановка) и $k = 10$. Все запуски успешно завершились и в системном журнале нет записей об аварийной остановке. Увеличиваем значение k до 30. При этом некоторые процессы завершились аварийно из-за того, что закончилась свободная память. Используя скрипт `find2.bash`, который выполняет бинарный поиск, определяем максимальное значение N , при котором все процессы успешно завершаются. В результате выясняем, что $N = 5875000$.

Выводы: В ходе эксперимента был проведен анализ влияния размера массива (N) и количества запусков скрипта (k) на возможность аварийных завершений процессов. При значении $k = 10$ все запуски скрипта завершились успешно, без аварийных остановок. При увеличении значения k до 30 запусков, наблюдались аварийные завершения некоторых процессов. Далее было найдено максимальное значение N , которое позволяет избежать аварийных завершений процессов при $k = 30$. Это значение составляет 5875000.