

Общий объем оперативки: 1860784 kB  
Подкачки: 839 MB  
Страницы: 4096 B  
Свободной физ памяти: 1440 MB  
Свободной в подкачке: 839 MB

## 1 TASK

### а.1) 31000000 - последнее в файле report.log

```
[ 248.774392] Out of memory: Killed process 1688 (mem.bash) total-virt:2645772kB, anon-rss:1638904kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:4816kB oom_score_adj:0
[ 248.849352] oom_reaper: reaped process 1688 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
user@localhost: ~$ cat /dev/null > /dev/null
```

### а.2) Скрипт мгновенно выходит на первую позицию по потреблению памяти, потребляя до 99.9 CPU и 95% MEM

```
mem(31000000) {
  MiB Mem : 45,8/1817,2 [|||||||||||||||||||||||||||||||||||||||||]
  MiB Swap:  6,4/820,0 [|||||]
}
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1702	user	20	0	829716	610464	2984	R	99,9	32,8	0:05.54	mem.bash
1700	user	20	0	275184	5216	4432	R	0,3	0,3	0:00.02	top
1	root	20	0	175184	3224	560	S	0,0	0,2	0:00.53	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	0	-20	0	0	0	T	0,0	0,0	0:00.00	rcu_gp

```
mem(31000000) {
  MiB Mem : 98,7/1817,2 [|||||||||||||||||||||||||||||||||||||||||]
  MiB Swap: 44,7/820,0 [|||||||||||||||||]
}
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1702	user	20	0	2189580	1,6g	1088	R	99,9	88,0	0:18.24	mem.bash
59	root	20	0	0	0	0	R	6,6	0,0	0:01.15	kswapd0
10	root	20	0	0	0	0	S	0,6	0,0	0:00.07	ksoftirqd/0
7	root	20	0	0	0	0	I	0,3	0,0	0:00.07	kworker/0:1-events_power_eff+
684	root	20	0	0	0	0	T	0,3	0,0	0:00.20	kworker/0:4-events

```
mem(31000000) {
  MiB Mem : 10,8/1817,2 [|||||]
  MiB Swap: 10,5/820,0 [|||||]
}
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1700	user	20	0	275184	352	0	R	0,7	0,0	0:00.06	top
1	root	20	0	175184	0	0	S	0,0	0,0	0:00.53	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
4	root	0	-20	0	0	0	T	0,0	0,0	0:00.00	rcu_gpd

Состав первых 5 скриптов: на первом месте mem.bash; на остальных четверых другие процессы, которые со временем сменяются на третьи (systemd почти всегда в первой пятерке)  
Шкала памяти (Mem) заполняется от ненагруженного состояние до конца + Swap заполняется до конца  
Когда обе шкалы заполняются, процесс убивается операционной системой

б) Аналогично оба скрипта сразу попадают в пятерку по памяти, потом один из них умирает и освобождает память – она начинает выдаваться второму скрипту, который вскоре тоже будет убит

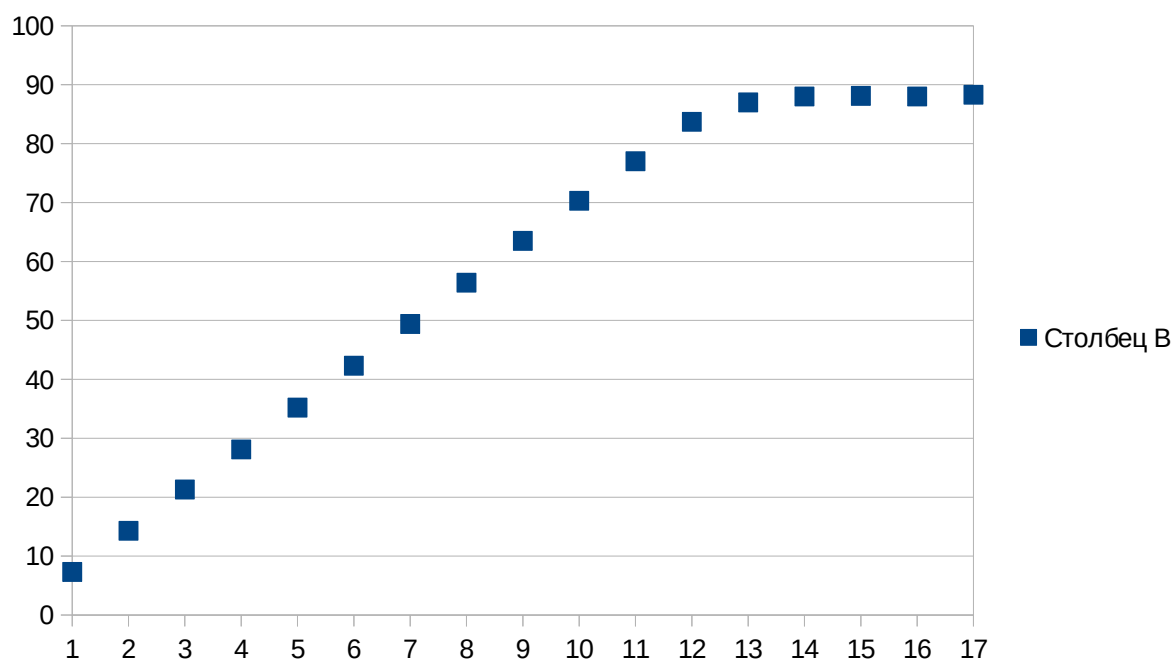
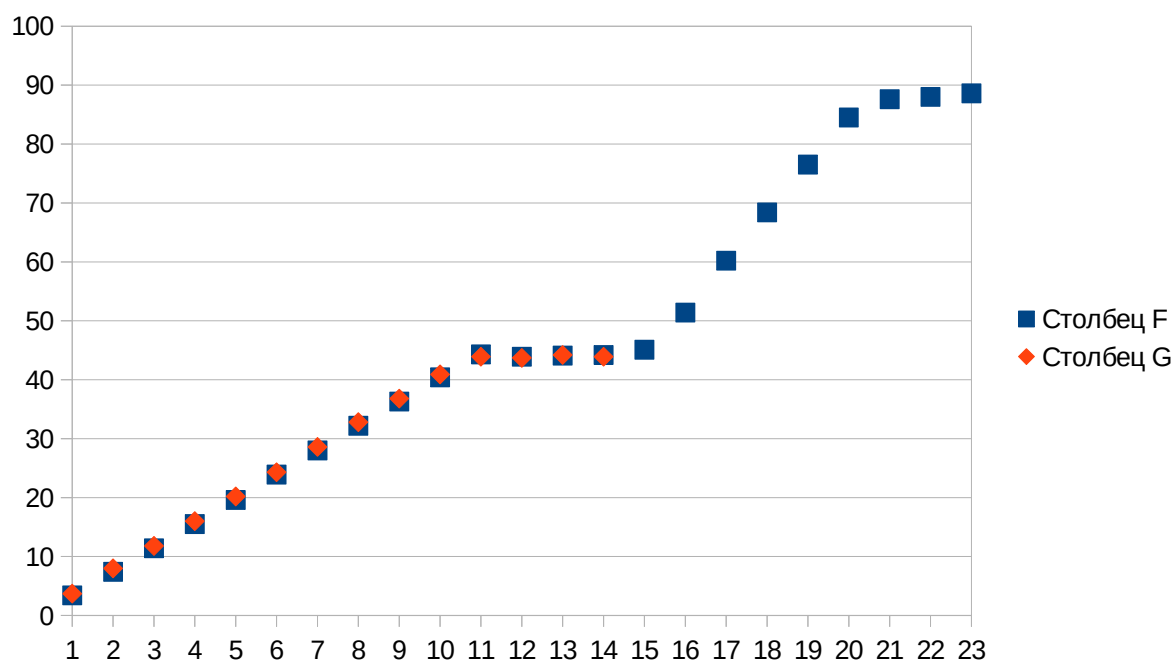
report.log: 15000000, report2.log: 31000000

Выводы сисжурнала:

```
[ 2996.077172] Out of memory: Killed process 3311 (mem2.bash) total-virt:2639040kB, anon-rss:1680452kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:4804kB oom_score_adj:0
[ 2996.151920] oom_reaper: reaped process 3311 (mem2.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
user@localhost: ~$ cat /dev/null > /dev/null
```

```
[ 2983.095554] Out of memory: Killed process 3310 (mem.bash) total-virt:1432428kB, anon-rss:841472kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:2444kB oom_score_adj:0
[ 2983.129964] oom_reaper: reaped process 3310 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
user@localhost: ~$ cat /dev/null > /dev/null
```

Графики



## Выводы

### Первый этап

Процессу выделялась память до того момента, пока оперативка не истощилась

Потом началось выделение с файла подкачки. Когда и там кончилось, то скрипт завершился.

### Второй этап

Память примерно в одинаковой пропорции выделялась обоим скриптам, потом один из них завершился аварийно, он отпустил выделенную память, она начала выделяться второму скрипту, до истощения ОЗУ и подкачки, потом и он завершился

2 TASK

```
[user@localhost lab5-k-kryukov]$ ./executor2.sh 2 1000000
Successful exit!
Successful exit!
```

```
^C
[user@localhost lab5-k-kryukov]$ ./executor2.sh 1 10000000
./executor2.sh: line 10: 4901 Killed                  ./newmem.bash $n
```

Теперь возьмём 10 по 1000000

```
^C
[user@localhost lab5-k-kryukov]$ ./executor2.sh 10 1000000
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
Successful exit!
```

Подберём для k=30:

(400000 – скрипты падают, 100000 – не падают)

Бинарный поиск...

```
[user@localhost lab5-k-kryukov]$ ./executor2.sh 30 250000
[user@localhost lab5-k-kryukov]$
```

```
^C
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
30
[user@localhost lab5-k-kryukov]$
```

Количество Success'в

270000 и 290000 провалилось

```
[user@localhost lab5-k-kryukov]$ ./newmem.bash: xmalloc: cannot allocate 2 bytes
./newmem.bash: xmalloc: cannot allocate 2 bytes
```

```
^C
[user@localhost lab5-k-kryukov]$ ./executor2.sh 30 260000
[user@localhost lab5-k-kryukov]$

3
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
3
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
4
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
17
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
30
[user@localhost lab5-k-kryukov]$ cat log_file | wc -l
30
[user@localhost lab5-k-kryukov]$
```

260000 успешно

265000 неуспешно, 262000 успешно

Примем  $N = 262000$

Важно отметить, что это скорее матожидание, чем конкретное значение, так как при планировании процессов очередь недетерминирована

### **Выводы**

При запуске процессов раз в секунду некоторые не успевают завершиться (следовательно, отдать ресурс) до того момента, как суммарная память сета процессов превысит лимит => происходит kill процесса. Если же процессы отпускают ресурсы быстрее, чем рождаются ИЛИ буфер взятых ресурсов растёт, но не успевает дорасти до лимита к моменту рождения последнего процесса, то все завершаются успешно

Конкретные значения  $N$  предельного определяются размером оперативной памяти