

Общий объем оперативной памяти : 2048 MB
Объем раздела подкачки : 448 MB
Размер страницы виртуальной памяти : 4096 B
Объем свободной физической памяти в ненагруженной системе: 213 MB
Объем свободного пространства в разделе подкачки в ненагруженной системе: 448 MB

Эксперимент №1

Первый этап:

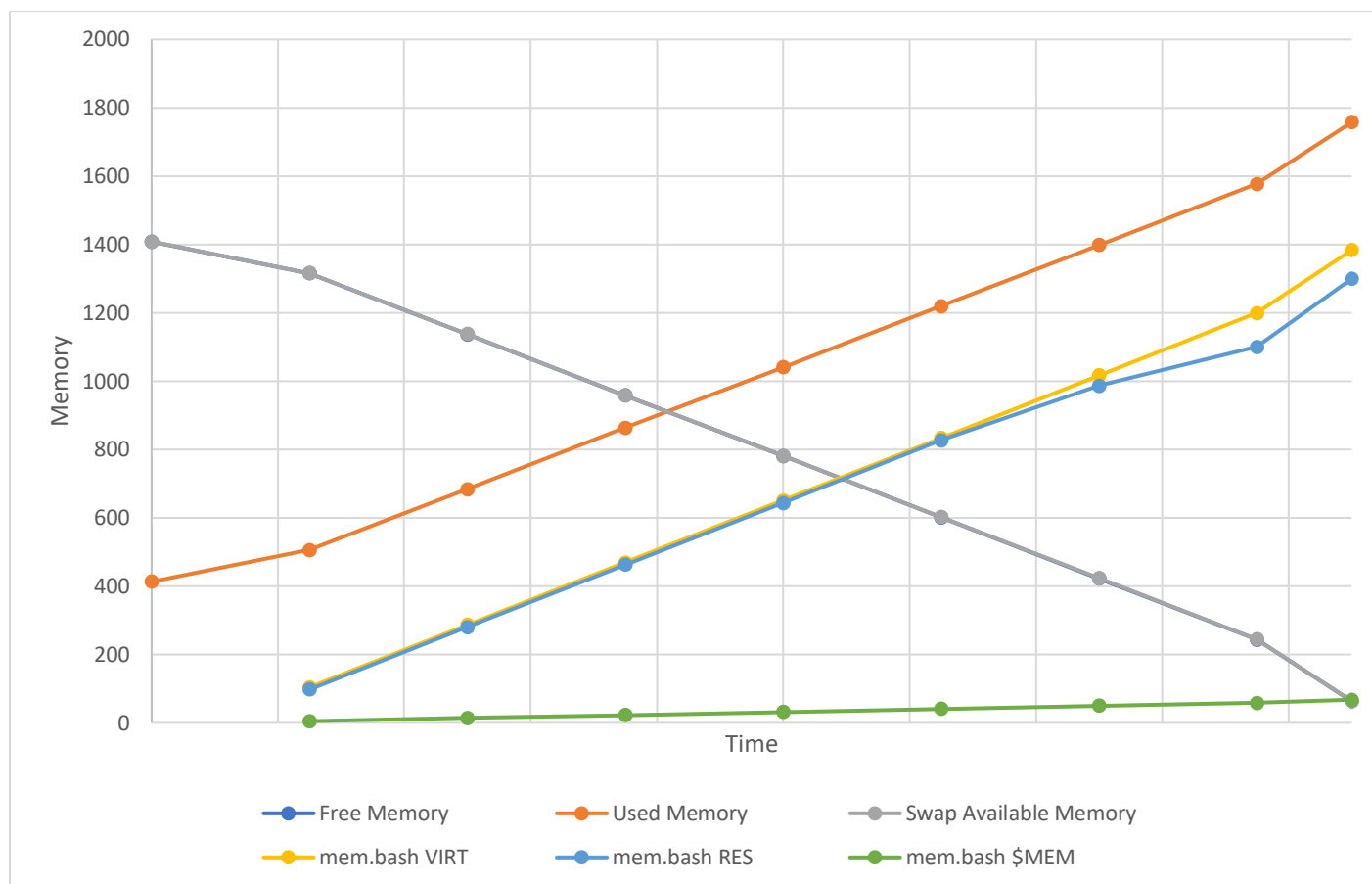
Максимальный размер массива:

```
17000000
17500000
18000000
18500000
19000000
root@user-VirtualBox:/home/user/Lab5#
```

Системные сообщения от dmesg:

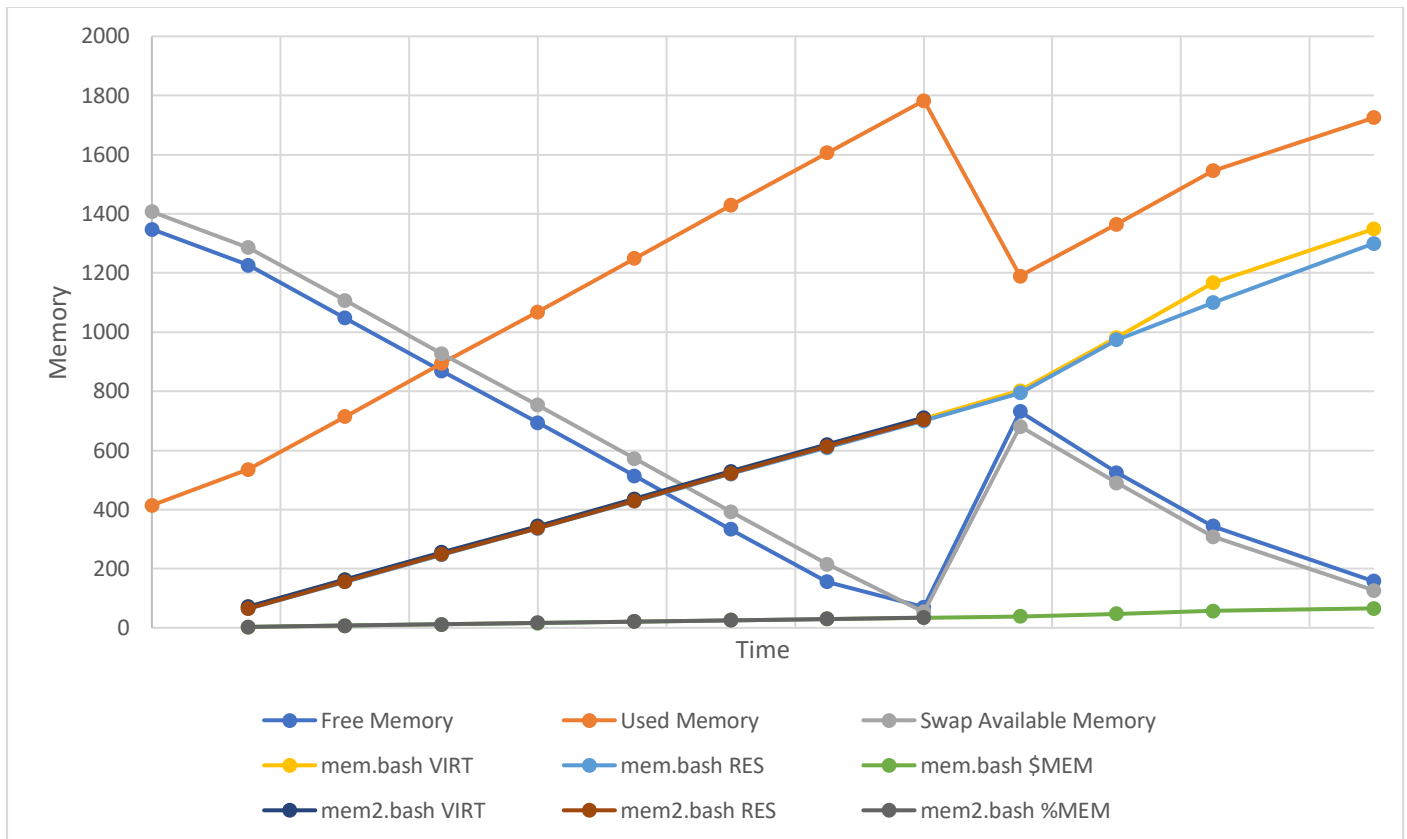
```
Dec  6 11:14:31 user-VirtualBox kernel: [ 4261.089919] [ 12870]      0 12870      2516      176   57344      0      0 bas
h
Dec  6 11:14:31 user-VirtualBox kernel: [ 4261.089919] [ 12880]      0 12880   377740   375411  3063808      0      0 mem
.bash
Dec  6 11:14:31 user-VirtualBox kernel: [ 4261.089920] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0
,global_oom,task_memcg=/user.slice/user-1000.slice/user@1000.service,task=mem.bash,pid=12880,uid=0
Dec  6 11:14:31 user-VirtualBox kernel: [ 4261.089926] Out of memory: Killed process 12880 (mem.bash) total-vm:1510960kB, anon-rss
:1501644kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:2992kB oom_score_adj:0
Dec  6 11:14:31 user-VirtualBox kernel: [ 4261.134458] oom_reaper: reaped process 12880 (mem.bash), now anon-rss:0kB, file-rss:0kB
, shmem-rss:0kB
root@user-VirtualBox:/home/user/Lab5#
```

Memory			Swap			mem.bash Process		
Free	Used	buff/cache	Free	Used	avail mem	VIRT	RES	%MEM
1408.3	413	166.5	0.8	447.7	1408.7			
1315.6	505.6	166.7	0.8	447.7	1316.2	103.972	97.48	4.8
1136.9	684.2	166.7	0.8	447.7	1137.5	286.66	280.168	13.8
957.7	863.4	166.7	0.8	447.7	958.4	469.48	463.12	22.8
780.8	1040.3	166.7	0.8	447.7	781.4	650.452	643.96	31.6
601.6	1219.5	166.7	0.8	447.7	602.3	833.404	826.912	40.6
422.2	1398.8	166.7	0.8	447.7	422.9	1016.884	986.7	49.6
243.6	1577.5	166.7	0.8	447.7	244.2	1199.176	1100	58.6
63.2	1757.9	166.7	0.8	447.7	63.8	1383.58	1300	67.7



Второй этап:

Memory			Swap			mem.bash Process			mem2.bash Process		
Free	Used	buff/cache	Free	Used	avail mem	VIRT	RES	%MEM	VIRT	RES	%MEM
1348	414.5	225.3	0.1	448.4	1406.9						
1226	535.9	225.4	0.1	448.4	1285.5	71.5	65.04	3.2	71.632	65.04	3.2
1047.8	714.6	225.4	0.1	448.4	1106.8	162.316	155.856	7.7	162.976	156.648	7.7
868.9	893.5	225.4	0.1	448.4	927.9	253.264	246.936	12.1	254.98	248.52	12.2
693.7	1068.7	225.4	0.1	448.4	752.7	343.024	336.432	16.5	344.476	338.016	16.6
513.8	1248.6	225.4	0.1	448.4	572.8	434.764	428.304	21	436.348	429.888	21.1
333.6	1428.7	225.4	0.1	448.4	392.7	526.504	520.176	25.6	528.88	522.288	25.7
156.2	1606.2	225.4	0.1	448.4	215.3	616.264	609.672	30	620.224	613.896	30.2
70.1	1782.9	134.8	0	448.5	53.4	707.212	700.008	34.4	710.908	703.66	34.6
731.4	1189.5	66.9	0	448.5	680.8	802.252	794.192	39			
524.9	1365.3	97.6	0	448.5	489.7	981.772	973.712	47.8			
343.8	1546	98	0	448.5	308.8	1166.572	1100	56.9			
157.5	1724.7	105.5	0	448.5	126.2	1348.996	1300	65.9			



Вывод:

На первом этапе мы можем понять, что процесс завершается, когда в системе заканчивается память, так чтобы система может продолжала работать. Процесс `mem.bash` завершается при превышении 75% использования памяти (%MEM) или размера массива 19000000.

На втором этапе мы понимаем, что когда память заполнена, только один процесс убивается, давая другим процессам больше оперативной памяти для работы, и в конце концов этот процесс также завершается, когда исчерпывается системную память.

Эксперимент №2

В этом эксперименте нам нужно было найти максимальный размер массива, на котором эти 30 процессов может быть запущен последовательно, и не будет никакой аварии (завершение процесса). После дальнейшего тестирования (Использованный ручной бинарный поиск) максимум в моей ситуации был найден 1780190.