

данные о текущей конфигурации операционной системы

общий объем оперативной памяти `/proc/meminfo memtotal`: 3869080kB

объем раздела подкачки `/proc/meminfo swaptotal`: 839676kB

размер страницы виртуальной памяти `getconf PAGESIZE`: 4096bytes

объем свободной физической памяти в ненагруженной системе `/proc/meminfo memfree`: 3442228kB

объем свободного пространства в разделе подкачки в ненагруженной системе `/proc/meminfo swapfree`: 839676kB

эксперимент 1

первый этап

mem.bash:

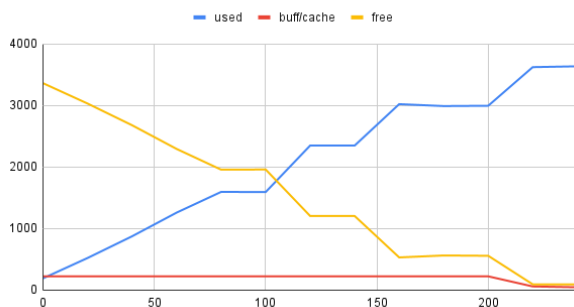
```
1 #!/bin/bash
2 > report.log
3 i=0
4 a=(1)
5 echo "starting at $(date +%F.%T)" >> report.log
6 while true; do
7 a+=(${a[@]})
8 echo "size = ${#a[@]}" >> report.log
9 ((i=i+1))
10 done
```

скрипт чтобы достать данные из `top`:

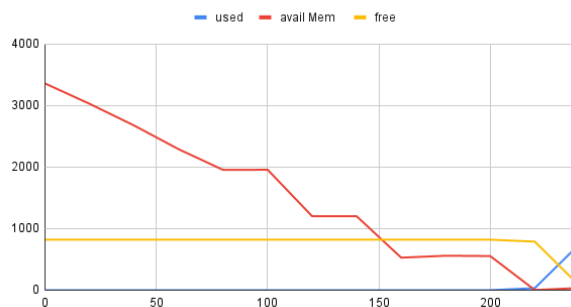
```
1 #!/bin/bash
2 > topdata.txt
3 while true; do
4 echo "" >> topdata.txt
5 top -b -n 1 | head -n 12 | tail -9 >> topdata.txt
6 sleep 20
7 done
```

изменения значений параметров памяти системы (верхние две строки над основной таблицей)

memory



swap



изменения значений параметров, соответствующих скрипту `mem.bash` (он всегда на первом месте был):

seconds	PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
0	1401	root	20	0	245036	25680	2892	R	64	0.7	0:01.50	bash
20	1401	root	20	0	582956	363600	2892	R	75	9.4	0:21.76	bash
40	1401	root	20	0	943448	724048	2892	R	84.2	18.7	0:42.00	bash
60	1401	root	20	0	1336544	1.1g	2892	R	72.7	28.9	1:02.10	bash
80	1401	root	20	0	1680688	1.4g	2892	R	71.4	37.8	1:22.23	bash
100	1401	root	20	0	1678292	1.4g	2892	R	72.7	37.7	1:42.38	bash
120	1401	root	20	0	2450756	2.1g	2892	R	76.2	57.7	2:02.48	bash
140	1401	root	20	0	2450756	2.1g	2892	R	71.4	57.7	2:22.70	bash
160	1401	root	20	0	3138908	2.8g	2892	R	77.3	75.5	2:42.78	bash
180	1401	root	20	0	3106136	2.8g	2892	R	76.2	74.6	3:02.94	bash
200	1401	root	20	0	3106136	2.8g	2892	R	78.9	74.6	3:21.23	bash
220	1401	root	20	0	3772868	3.4g	916	R	66.7	91.4	3:40.77	bash
240	1401	root	20	0	4472468	3.4g	560	R	40	91.5	3:57.10	bash

вывод `dmesg` (последние две записи о скрипте в системном журнале):

```
1 | Out of memory: Killed process 1387 (bash) total-vm:4606844kB, anon-res:3631460kB, file-rss:0kB, shmem-rss:0kB, UID:0
2 | oom-reaper: reaped process 1387 (bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```

значение в последней строке файла `report.log`: 16777216

вывод:

если принимать значение занимаемой одним интом памяти за 8(?), то процесс в итоге справился с $(16777216 * 8) / 1024 = 131072 \text{ kB}$, и упал на 262144kB, это почти в 15 раз меньше, чем размер доступной оперативной памяти. вероятно, последние несколько итераций пытались выполниться, но не записались в `report.log`, так как оперативная память все таки закончилась (интервал 220-240 сек с момента запуска), и `mem.bash` начал брать память из раздела подкачки. там ему памяти тоже не хватило, он запросил расширения адр. пространства, и был остановлен, так как столько памяти выделить на него было невозможно

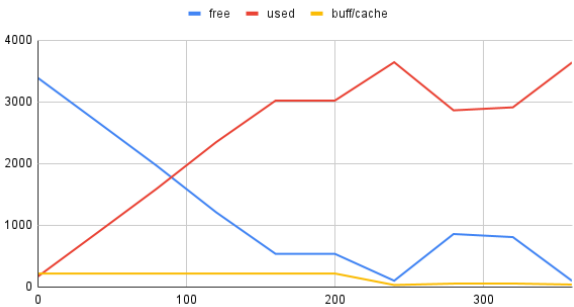
второй этап

скрипт чтобы достать данные из `top` (я не стала отдельно писать о местах `mem.bash` и `mem2.bash` в таблице, они очевидно на 1-2 начиная с момента запуска, но ниже есть таблица, какой из них на первом, какой на втором месте по %CPU):

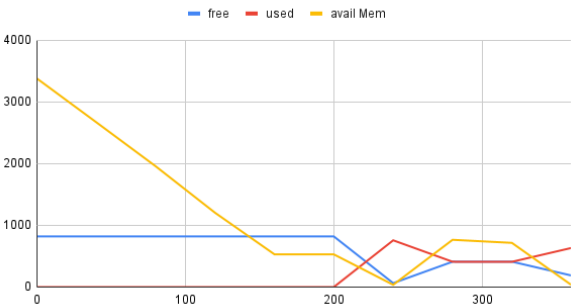
```
1 | #!/bin/bash
2 | > topdata.txt
3 | i=0
4 | while true; do
5 |   echo "$i" >> topdata.txt
6 |   top -b -n 1 | head -n 5 | tail -2 >> topdata.txt
7 |   top -b -n 1 | grep ".sh" >> topdata.txt
8 |   ((i=i+20))
9 |   sleep 20
10 | done
```

изменения значений параметров памяти системы (верхние две строки над основной таблицей)

memory2

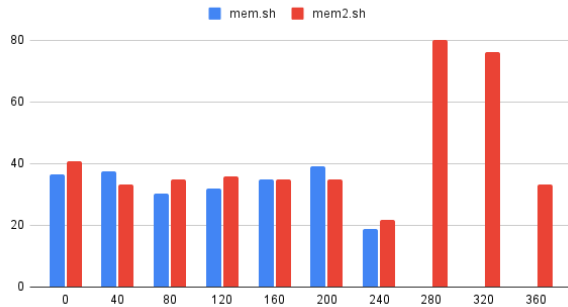


swap2

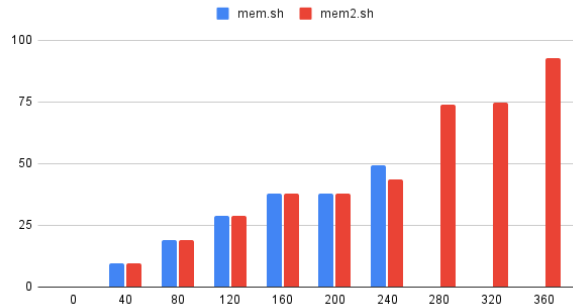


изменения значений параметров, соответствующих скриптам `mem.bash` и `mem2.bash`:

%CPU



%MEM



вывод `dmesg` (последние две записи о скриптах в системном журнале):

```

3 | Out of memory: Killed process 1384 (mem.sh) total-vm:2412212kB, anon-res:1920280kB, file-rss:0kB, shmem-rss:0kB, UID:0
4 | oom-reaper: reaped process 1384 (mem.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
5 | Out of memory: Killed process 1385 (mem2.sh) total-vm:4567112kB, anon-res:3594484kB, file-rss:0kB, shmem-rss:0kB, UID:0
6 | oom-reaper: reaped process 1385 (mem2.sh), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB

```

значение в последней строке файла `report.log`: 8388608

значение в последней строке файла `report2.log`: 16777216

вывод:

до примерно 200 секунды оба процесса пользовались оперативной памятью, потом попросили еще памяти из раздела подкачки, когда оперативная закончилась. когда раздел подкачки тоже закончился, первый процесс попробовал попросить расширить адресное пространство. такое расширение было невозможно (то же самое, что в эксп1), его работа была аварийно остановлена. теперь второму процессу досталась вся память, которую занимал первый, но ее хватило только на одно удвоение массива (что логично, первый процесс столько и успел занять), поэтому он так же был остановлен.

эксперимент 2

newmem.bash:

```

1 | #!/bin/bash
2 | a=(1)
3 | while true; do
4 |   a+=(${a[@]})
5 |   if [[ ${#a[@]} -ge $1 ]]; then
6 |     echo "done"
7 |     break
8 |   fi
9 | done

```

run newmem.bash (x10, норм работает, x30 11 погибло):

```

1 | #!/bin/bash
2 | chmod +x newmem.sh
3 | for ((i=0;i<$1;i++)); do
4 |   ./newmem.sh 1677721 &
5 | done

```

16777216 : 30 = 559240

на x30: (в качестве N ставим степени двойки, так как скрипт удваивает массив):

2^{18} : 262144 – работает

2^{19} : 524288 – работает

2^{20} : 1048576 – работает

2^{21} : 2097152 – начали вымирать, следовательно 1048576 – максимальное значение N