

Отчёт к лабораторной работе №5

- 1) Общий объём оперативной памяти - 3064624 байт
- 2) Объём раздела подкачки - 459260 байт
- 3) Размер страницы виртуальной памяти - 4096 байт
- 4) Объём свободной физической памяти в ненагруженной системе - 2010792 байт
- 5) Объём свободного пространства в разделе подкачки в ненагруженной системе - 459260 байт

Эксперимент 1

Подготовительный этап :

Скрипт mem.bash

```
1 #!/bin/bash
2
3 array=()
4 iter=1
5 echo "" > report.log
6 while true;
7 do
8     array+=(1 2 3 4 5)
9     if [[ "$iter" -eq "100000" ]]
10    then
11        iter=0
12        echo ${#array[*]} >> report.log
13    fi
14    let "iter=$iter+1"
15 done
```

Первый этап :

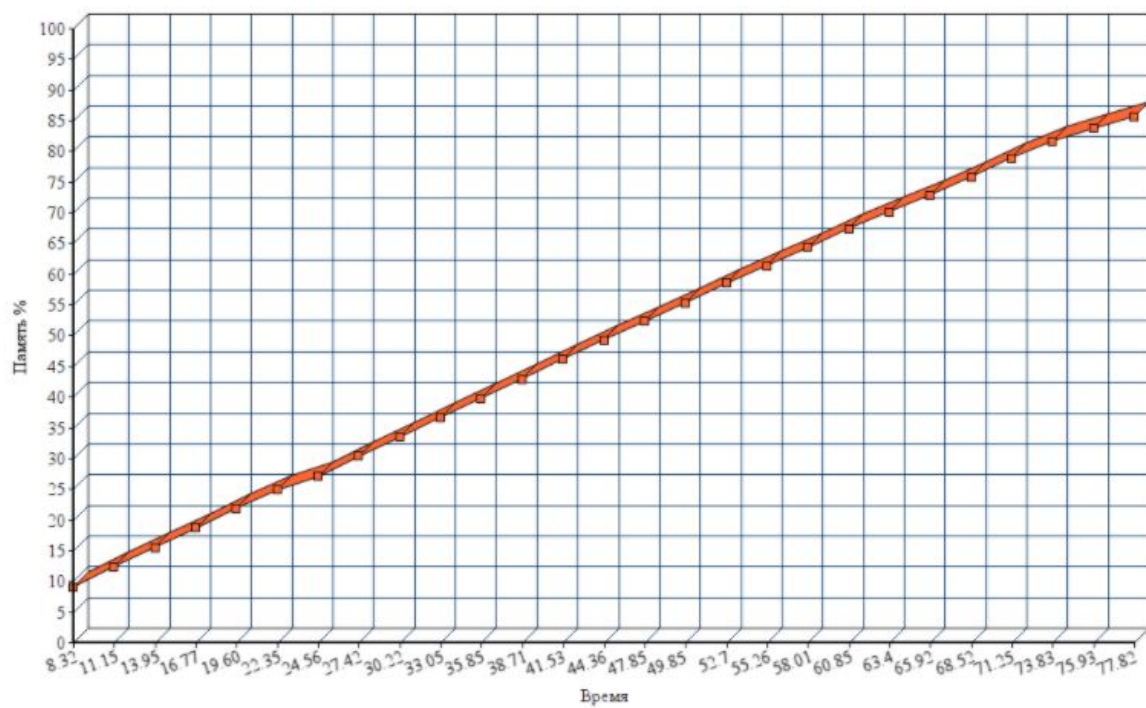
Параметры, с которыми произошла аварийная остановка:

total-vm:2762224kB
anon-rss:2752912kB
file-rss:0kB
shem-rss:0kB
pgtables:5440 kB

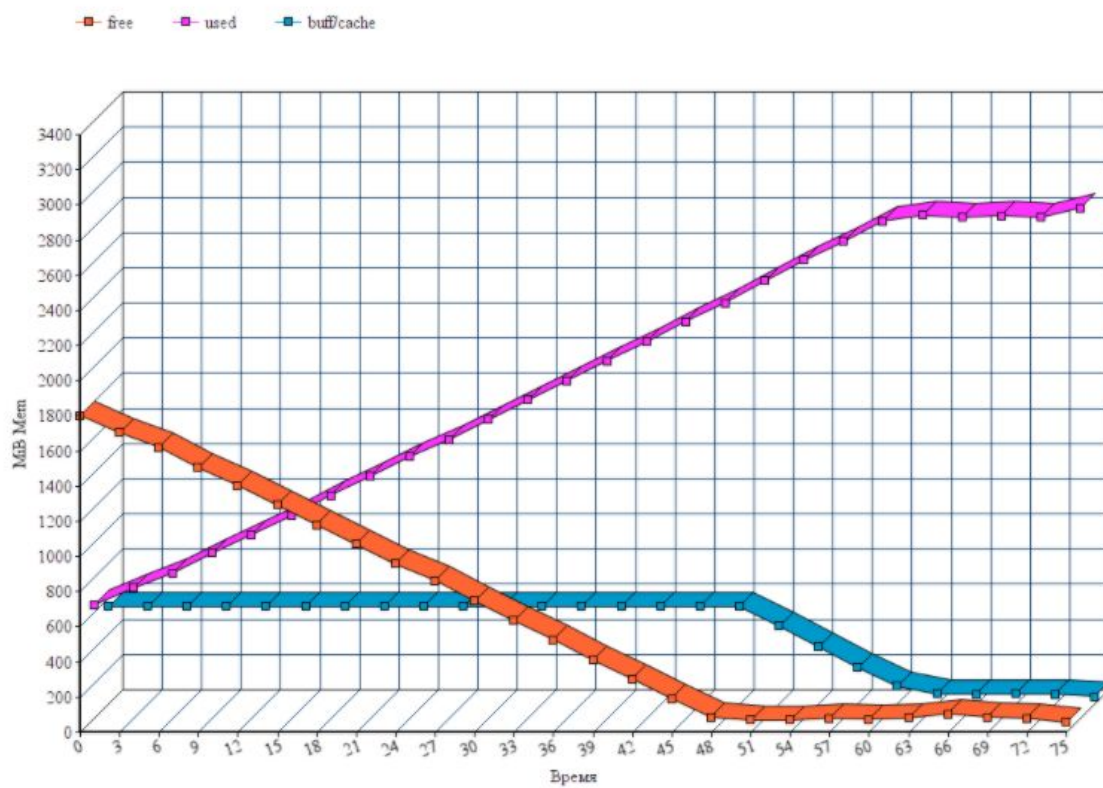
Последняя строка report.log - 34000000

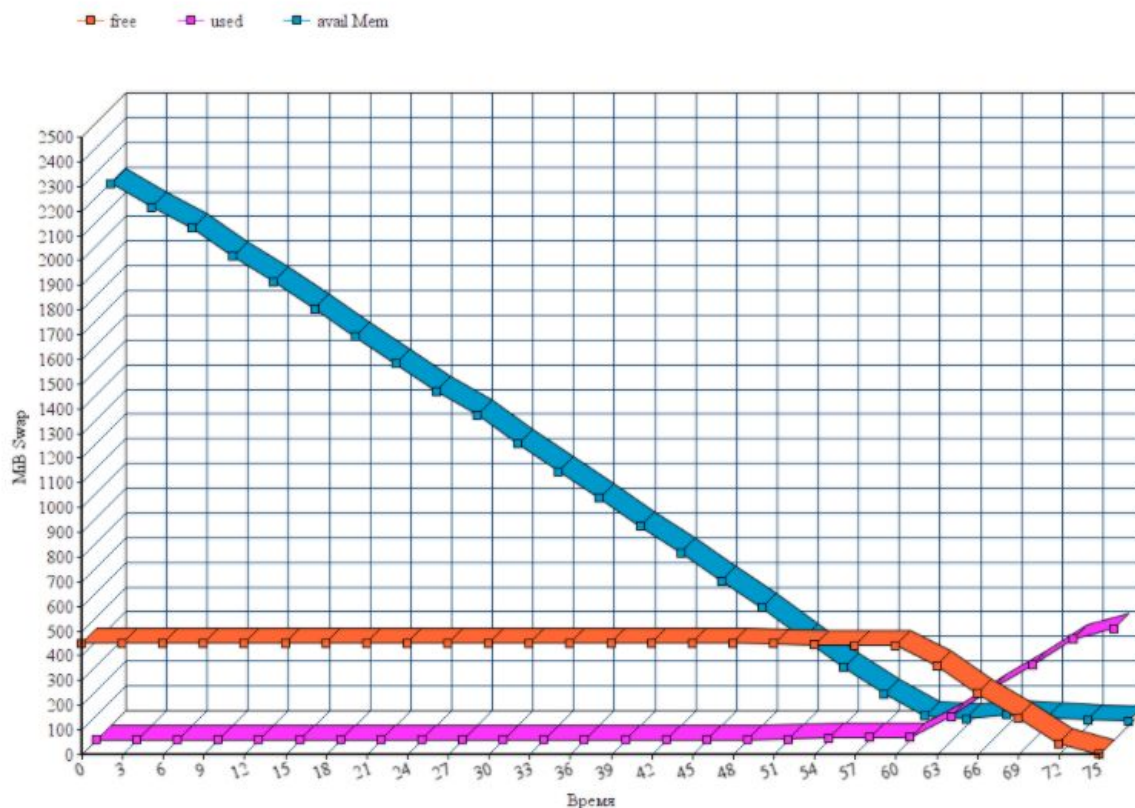
Изменения параметров в top в зависимости от времени

График изменения значения %MEM процесса mem.bash



Изменение значений MiB Mem и MiB Swap (интервал - 3 секунды):





Вывод по графикам :

Сначала система пытается выделить процессу mem.bash память из оперативной памяти, а как только она заканчивается пытается выделить ему память из раздела подкачки. Когда заканчивается и оперативная память и раздел подкачки процесс аварийно останавливается операционной системой.

Изменения верхних 5 процессов (слева направо, сверху вниз, интервал - 3 секунды)

mem.bash gnome-shell Xorg tracker-store systemd	mem.bash gnome-shell Xorg gnome-terminal- tracker-store	mem.bash tracker-store gnome-shell tracker-miner-f Xorg	mem.bash tracker-store tracker-miner-f tracker-extract Xorg	mem.bash Xorg gnome-terminal- gnome-shell tracker-store
mem.bash Xorg gnome-terminal- gnome-shell tracker-store	mem.bash Xorg gnome-terminal- gnome-shell tracker-store	mem.bash tracker-store tracker-extract kworker/u2:3-events_freezable_power_ tracker-miner-f dbus-daemon	mem.bash tracker-store tracker-miner-f gnome-shell tracker-extract systemd	
mem.bash tracker-store tracker-extract snapd Xorg	mem.bash tracker-store tracker-miner-f tracker-extract gnome-shell gnome-terminal- tracker-store	mem.bash tracker-store tracker-extract tracker-miner-f Xorg	mem.bash tracker-store tracker-extract dbus-daemon Xorg	mem.bash tracker-store tracker-miner-f gnome-shell gnome-terminal- tracker-store


```

mem.bash
tracker-store
tracker-extract
tracker-miner-f
Xorg

mem.bash
tracker-store
tracker-extract
gnome-shell
tracker-miner-f

mem.bash
tracker-store
tracker-miner-f
top
tracker-extract

mem.bash
kswapd0
tracker-store
tracker-extract
tracker-miner-f

mem.bash
kswapd0
tracker-store
tracker-miner-f
gnome-shell

mem.bash
kswapd0
tracker-store
kworker/0:1H-kblockd
tracker-miner-f
gnome-shell

kswapd0
mem.bash
tracker-miner-f
gnome-shell
accounts-daemon

```

Последние 2 записи в системном журнале :

- 1) [1262.694882] Out of memory: Killed process 2623 (mem.bash)
total-vm:2756812kB, anon-rss:2747504kB, file-rss:0kB, shmem-rss:0kB, UID:1000
pgtables:5432kB oom_score_adj:0
- 2) [1262.783503] oom_reaper: reaped process 2623 (mem.bash), now anon-rss:0kB,
file-rss:0kB, shmem-rss:0kB

report.log - 35000000

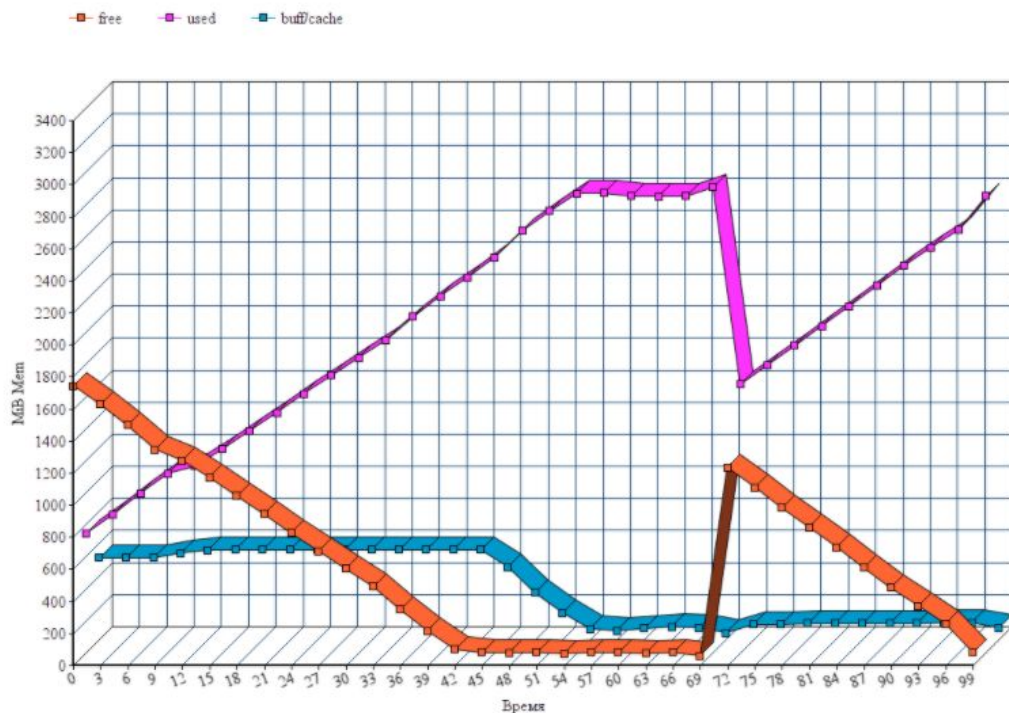
Второй этап :

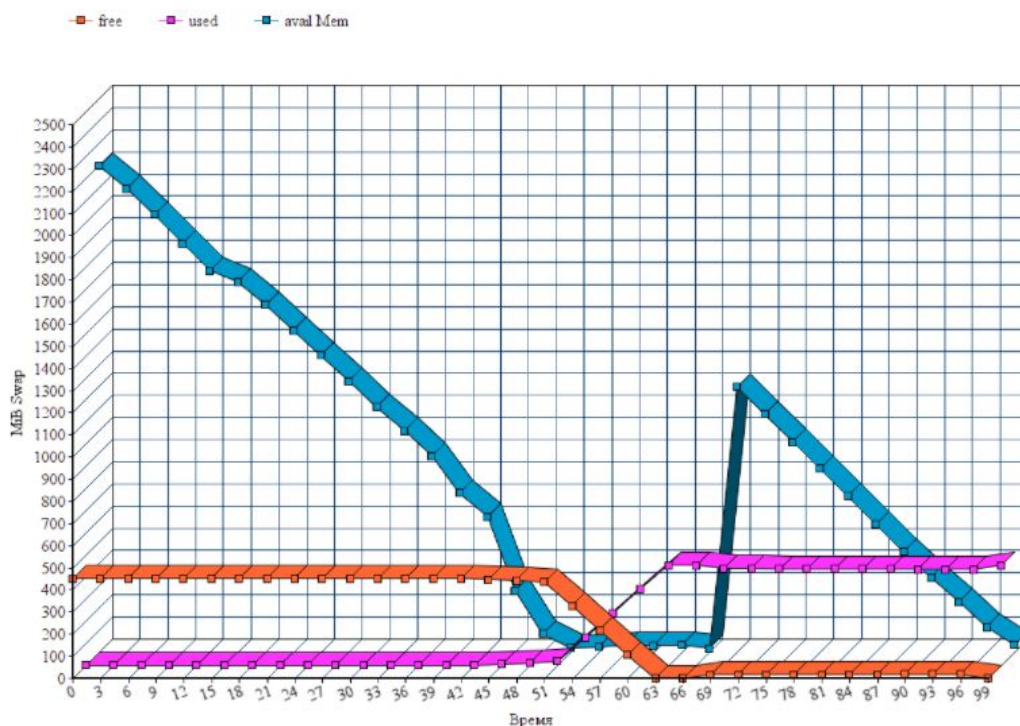
both.sh:

```

1 #!/bin/bash
2
3 ./mem.bash&
4 ./mem2.bash&

```





Вывод по графикам:

Изначально ОС выделяет двум процессам (mem.bash и mem2.bash) память из оперативной памяти, а в момент когда оперативной памяти перестаёт хватать, ОС задействует файлы подкачки. Как только стало невозможным выделить еще памяти в разделе подкачки, ОС аварийно останавливает первый процесс (mem.bash) и освобождает ту часть оперативной памяти, которую он использовал (скачок значения availMem на графике 2, скачок значения free и падение значения used на графике 1). Теперь для второго процесса mem2.bash есть откуда брать память. Но как только пропадает возможность выделения оперативной памяти и использования файла подкачки для процесса mem2.bash, он аналогично первому этапу аварийно завершается.

Записи в системной журнале:

```
[ 151.905262] Out of memory: Killed process 1499 (mem.bash) total-vml:1380316kB, anon-rss:1369144kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:2736kB
```

```
oom_score_adj:0
```

```
[ 151.970185] oom_reaper: reaped process 1499 (mem.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```

```
[ 185.013745]
```

```
[ 185.013752] Out of memory: Killed process 1500 (mem2.bash) total-vml:2728960kB, anon-rss:2717196kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:5368kB
```

```
oom_score_adj:0
```

```
[ 185.107702] oom_reaper: reaped process 1500 (mem2.bash), now anon-rss:0kB, file-rss:0kB, shmem-rss:0kB
```

report.log - 17500000

report2.log - 34500000

Эксперимент 2

Скрипт kmem.sh:

```
1 #!/bin/bash
2
3 count=$2
4 N=$1
5
6 for ((i=0;i<$count;i++))
7 do
8     ./newmem.bash $N&
9     sleep 1
10 done
```

Установим значение N как 3400000 (mem.bash аварийно останавливался на отметке ровно в 10 раз больше) и запустим скрипт командой ./kmem.sh 3400000 10.

Сообщений об аварийной остановке в консоли не было

```
andrew@andrew-VirtualBox:~/Desktop/lab5$ ./kmem.sh 3400000 10
andrew@andrew-VirtualBox:~/Desktop/lab5$ ./kmem.sh 3400000 10
andrew@andrew-VirtualBox:~/Desktop/lab5$
```

Системный журнал также не содержит информации об аварийных остановках

```
andrew@andrew-VirtualBox:~/Desktop/lab5$ dmesg | grep "newmem.bash"
andrew@andrew-VirtualBox:~/Desktop/lab5$
```

Так как размер массива в каждом из процессов в 10 раз меньше чем размер массива в скрипте mem.bash до аварийной остановки, то суммарно их объём памяти не более, чем последний фиксированный размер массива в скрипте mem.bash и всем 10 процессам хватает памяти, поэтому аварийных остановок не случилось.

Теперь запустим со значение K=30 и заглянем в системный журнал:

```
[ 3835.478412] Out of memory: Killed process 2862 (newmem.bash) total-vm:164332kB,
anon-rss:139328kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:356kB
oom_score_adj:0
[ 3850.521362] Out of memory: Killed process 2867 (newmem.bash) total-vm:168952kB,
anon-rss:142296kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:372kB
oom_score_adj:0
[ 3856.134057] Out of memory: Killed process 2883 (newmem.bash) total-vm:167500kB,
anon-rss:142920kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:360kB
oom_score_adj:0
[ 3876.988755] Out of memory: Killed process 2872 (newmem.bash) total-vm:179776kB,
anon-rss:153228kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:388kB
oom_score_adj:0
```

[3886.771741] Out of memory: Killed process 2878 (newmem.bash) total-vm:182416kB, anon-rss:157788kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:396kB
oom_score_adj:0
[3909.186825] Out of memory: Killed process 2888 (newmem.bash) total-vm:191260kB, anon-rss:164896kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:404kB
oom_score_adj:0
[3916.454035] Out of memory: Killed process 2893 (newmem.bash) total-vm:192184kB, anon-rss:166532kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:412kB
oom_score_adj:0
[3925.993958] Out of memory: Killed process 2898 (newmem.bash) total-vm:200500kB, anon-rss:172296kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:424kB
oom_score_adj:0
[3934.606227] Out of memory: Killed process 2913 (newmem.bash) total-vm:212380kB, anon-rss:185272kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:456kB
oom_score_adj:0
[3942.997170] Out of memory: Killed process 2908 (newmem.bash) total-vm:227032kB, anon-rss:198796kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:476kB
oom_score_adj:0
[3951.969608] Out of memory: Killed process 2922 (newmem.bash) total-vm:241552kB, anon-rss:213760kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:508kB
oom_score_adj:0
[3961.397839] Out of memory: Killed process 2927 (newmem.bash) total-vm:262012kB, anon-rss:232772kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:544kB
oom_score_adj:0

12 процессов завершились аварийной остановкой скрипта, так как теперь запускается сразу 30 процессов и в какой-то момент случается ситуация, что все запущенные скрипты хотят память, а памяти не хватает. Это обусловлено тем, что теперь суммарный размер всех массивов в каждом из процессов в 3 раза больше чем размер массива, при котором происходила аварийная остановка mem.bash

Уменьшим значение N в 3 раза и запустим заново скрипт ./kmem.sh 1130000 30, заглянем в системный журнал :

Последнее сообщение об аварийной остановке скрипта :

[3961.397839] Out of memory: Killed process 2927 (newmem.bash) total-vm:262012kB, anon-rss:232772kB, file-rss:0kB, shmem-rss:0kB, UID:1000 pgtables:544kB
oom_score_adj:0

Заметим, что это сообщение совпадает с последним сообщением выше, а следовательно при текущем запуске kmem аварийных остановок скриптов не происходило, так как суммарный объём памяти 30 данных процессов не больше чем объём памяти, занимаемый процессов mem.bash до аварийной остановки.

Попытаемся увеличить значение N:

1200000 - аварийных остановок всё ещё не происходит

1300000 - аварийных остановок всё ещё не происходит

1600000 - аварийных остановок всё ещё не происходит

1800000 - несколько скриптов завершились аварийной остановкой, следовательно искомое значение N в пределах от 1600000 до 1800000

1700000 - аварийных остановок не происходило

1750000 - аварийных остановок не происходило

1760000 - несколько скриптов завершились аварийной остановкой,
следовательно искомое значение N несущественно больше 1750000

Несмотря на то, что суммарный объём памяти 30 процессов при $N = 1750000$ больше объёма памяти занимаемого процессов `mem.bash` до аварийной остановки, аварийных остановок не происходит так как процессы запускаются не сразу, а раз в секунду, а следовательно от запуска первого скрипта до 30 ого прошло 29 секунд и некоторые процессы успели получить необходимое количество памяти и завершиться, освободив тем самым память для других процессов.