

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего
образования

Санкт-Петербургский национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа 5. Управление памятью в ОС Linux

По дисциплине «Операционные системы»

Выполнил:

студент группы №М3212

Тимофеев Вячеслав [REDACTED]

Проверил:

Трусилов [REDACTED]



УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2025

Системные показатели памяти текущей ОС:

Общий объем оперативной памяти (`grep MemTotal /proc/meminfo`) : 9962856 kB

Объем раздела подкачки (`grep SwapTotal /proc/meminfo`) : 0 kB

Размер страницы виртуальной памяти (`getconf PAGESIZE`): 4096 B

Объем свободной физической памяти в ненагруженной системе (`grep MemFree /proc/meminfo`): 6908668 kB

Объем свободного пространства в разделе подкачки в ненагруженной системе (`grep SwapFree /proc/meminfo`) : 0 kB

Эксперимент 1

Первый этап

mem.bash:

```
#!/bin/bash
if [ ! -f report.log ]; then
    echo "creating report.log"
    touch report.log
fi
>report.log
arr=()
i=0
while true
do
    for j in {1..10}
    do
        arr+=($i)
    done
    ((i++))
    if (( $i % 100000 == 0 )); then
        echo "${#arr[*]}" >> report.log
    fi
done
```

Последние записи о скрипте mem.bash в системном журнале (без даты):

```
[DATE] 123 kernel: 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-3800724a-acae-40b8-85a1-  
891ce9a32e3f.scope,task=mem.bash,pid=3515,uid=1000
```

```
[DATE] 123 kernel: Out of memory: Killed process 3515 (mem.bash) total-vm:8882024kB, anon-rss:8872064kB, file-rss:1768kB, shmem-rss:0kB, UID:1000 pgtables:17424kB  
oom_score_adj:200
```

Последняя запись в report.log: 113000000

Вспомогательный скрипт для отслеживания `top` при запущенном `mem.bash`:

```

#!/bin/bash

> mem.log
> scriptinfo.log
> top5proc.log

sleep 5s
pid=$(pgrep -f mem.bash)

while true; do
    output=$(top -b -n 1)

    echo "$output" | sed '4,5!d' >> mem.log
    echo "" >> mem.log

    ps -p $pid -o pid,pcpu,pmem,etime,vsz,rss,cmd >> scriptinfo.log
    echo "" >> scriptinfo.log

    echo "$output" | sed '8,12!d' >> top5proc.log
    echo "" >> top5proc.log

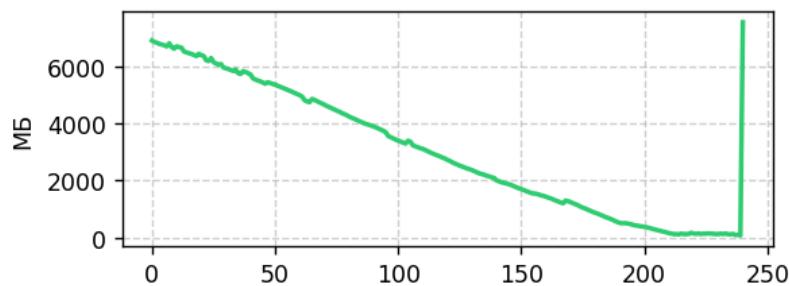
    if ! ps -p $pid > /dev/null; then
        exit
    fi

    sleep 1s
done

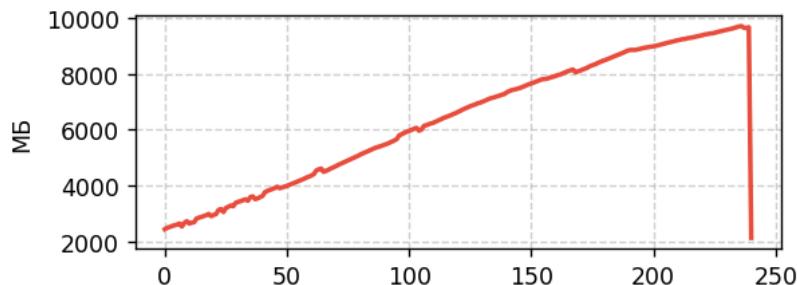
```

Графики изменений использования памяти:

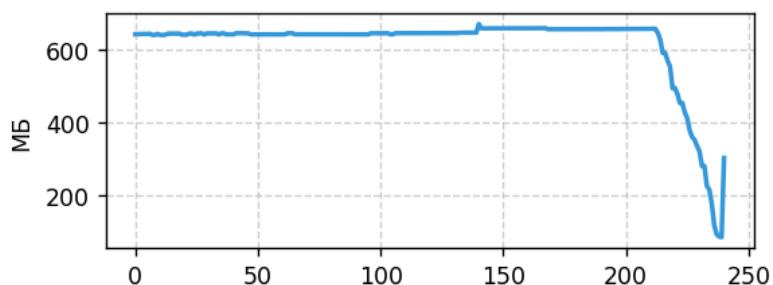
Свободная память (Mem Free)



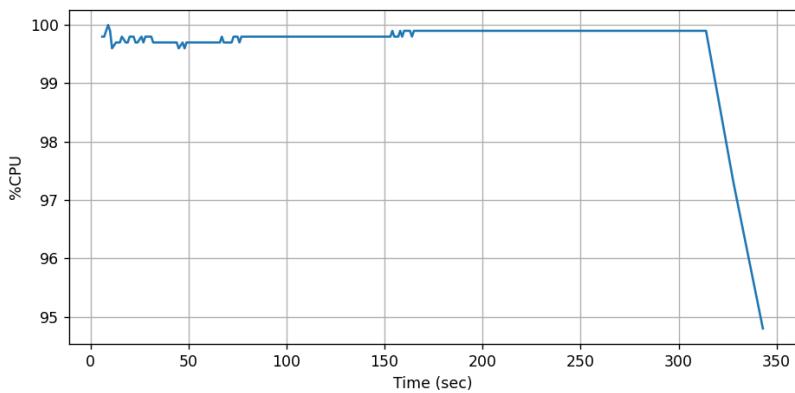
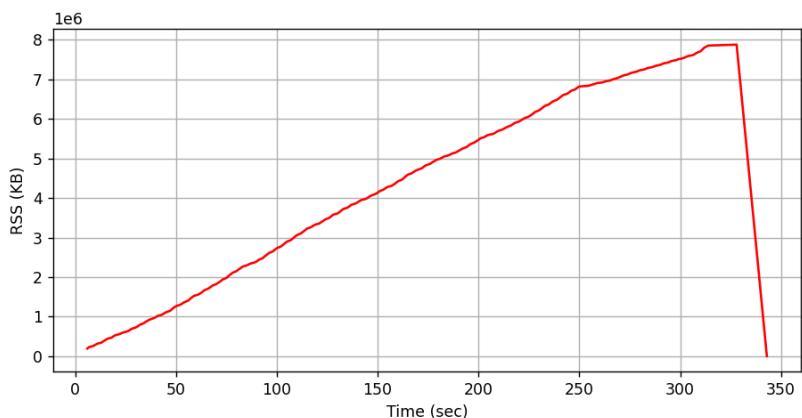
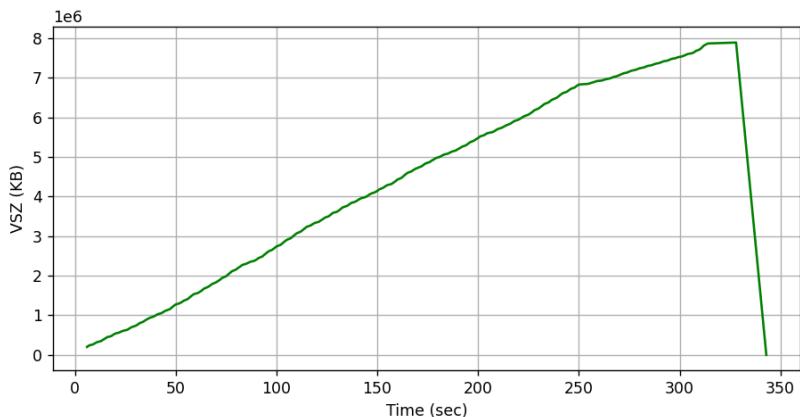
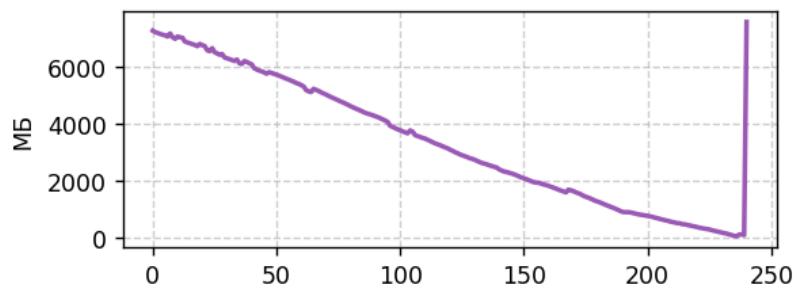
Используемая память (Mem Used)

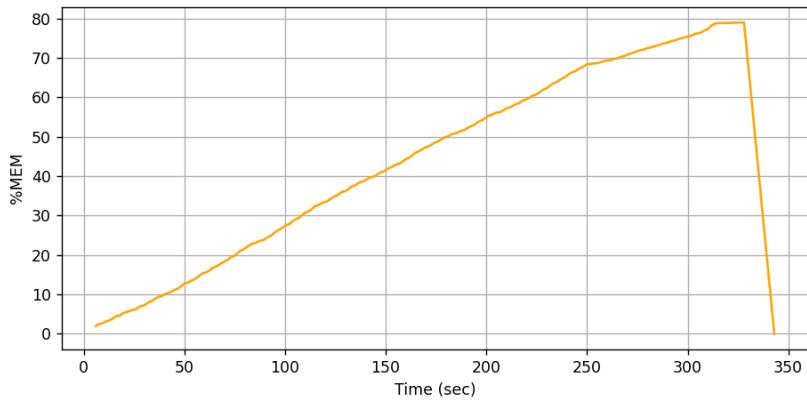


Буферизованная память (Buff/Cache)



Доступная память (Swap Avail Mem)





Итог: бесконечный цикл складывания элементов в массив вызвал линейный рост использования физической (RSS) и виртуальной (VSZ) памяти. Когда RSS & VSZ начали потреблять много памяти, подключается использование свободной оперативной памяти (RAM). Когда закончилась свободная RAM, происходит очистка кэша (Buff/Cache).

Конечные значения

- Размер массива (после убийства процесса): 113000000
- Использовано памяти:

total-vm ≈ 8,88 Гб

anon-rss ≈ 8,87 Гб

Второй этап

Создадим копию скрипта mem.bash (**mem2.bash**):

```
#!/bin/bash
if [ ! -f report2.log ]; then
    echo "creating report2.log"
    touch report2.log
fi
>report2.log
arr=()
i=0
while true
do
    for j in {1..10}
    do
        arr+=($i)
    done
    ((i++))
    if (( $i % 100000 == 0 )); then
        echo "${#arr[*]}" >> report2.log
    fi
done
```

и соответствующий ему вспомогательный скрипт для отслеживания top (**mem2top.bash**):

```

#!/bin/bash

> mem.log
> scriptinfo.log
> top5proc.log

sleep 5s
pid=$(pgrep -f mem.bash)

while true; do
    output=$(top -b -n 1)

    echo "$output" | sed '4,5!d' >> mem.log
    echo "" >> mem.log

    ps -p $pid -o pid,pcpu,pmem,etime,vsz,rss,cmd >> scriptinfo.log
    echo "" >> scriptinfo.log

    echo "$output" | sed '8,12!d' >> top5proc.log
    echo "" >> top5proc.log

    if ! ps -p $pid > /dev/null; then
        exit
    fi

    sleep 1s
done

```

Создадим скрипт (**start.bash**), запускающий имеющиеся 4 файла в фоновом режиме:

```

#!/bin/bash
./mem.bash &
./mem2.bash &

./memtop.bash &
./mem2top.bash &

wait

```

Последние записи о скрипте mem.bash в системном журнале (без даты):

```

[DATE] 123 kernel: 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-e0e69f7c-e6b8-4d43-a7d3-fd94b1ec5f90.scope,task=mem.bash,pid=22378,uid=1000

[DATE] 123 kernel: Out of memory: Killed process 22378 (mem.bash) total-vm:8684816kB, anon-rss:8674560kB, file-rss:144kB, shmem-rss:0kB, UID:1000 pgtables:17040kB oom_score_adj:200

```

Последние записи о скрипте mem2.bash в **системном журнале** (без даты):

```

[DATE] 123 kernel: 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-e0e69f7c-e6b8-4d43-a7d3-fd94b1ec5f90.scope,task=mem2.bash,pid=22379,uid=1000

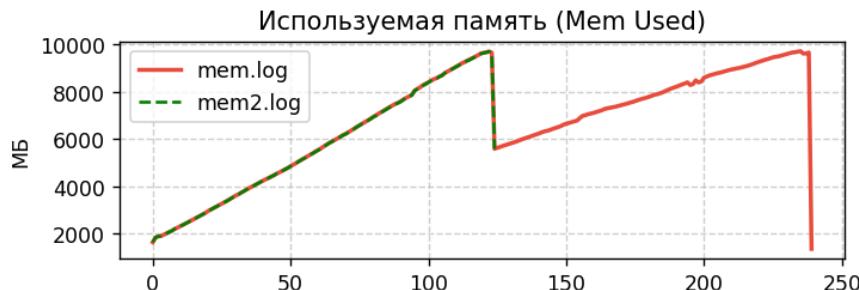
```

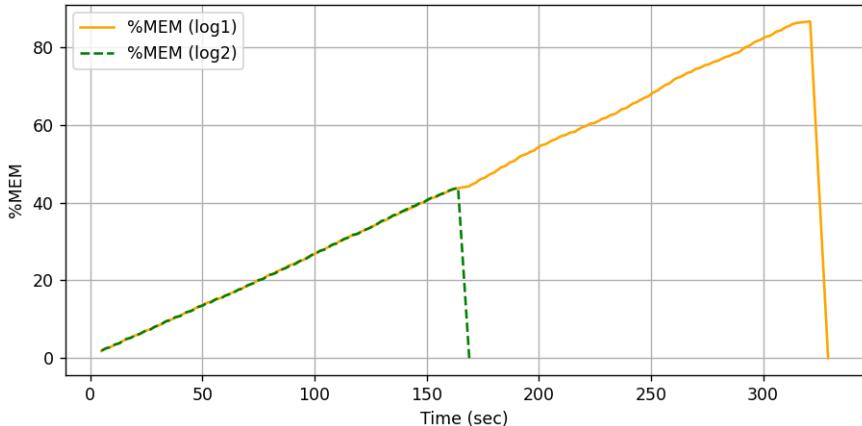
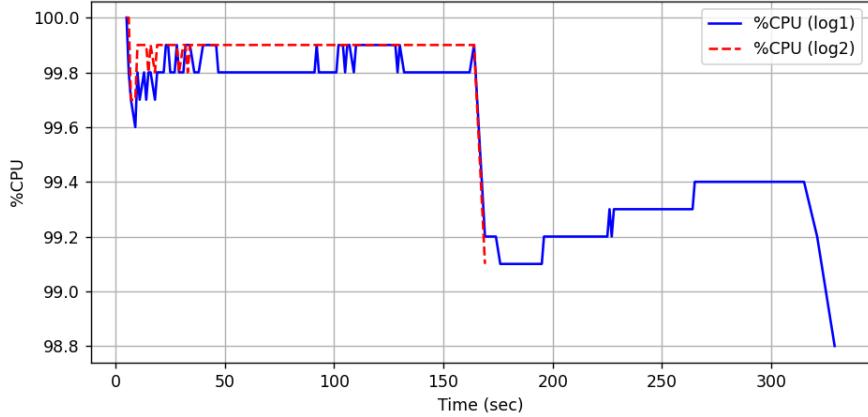
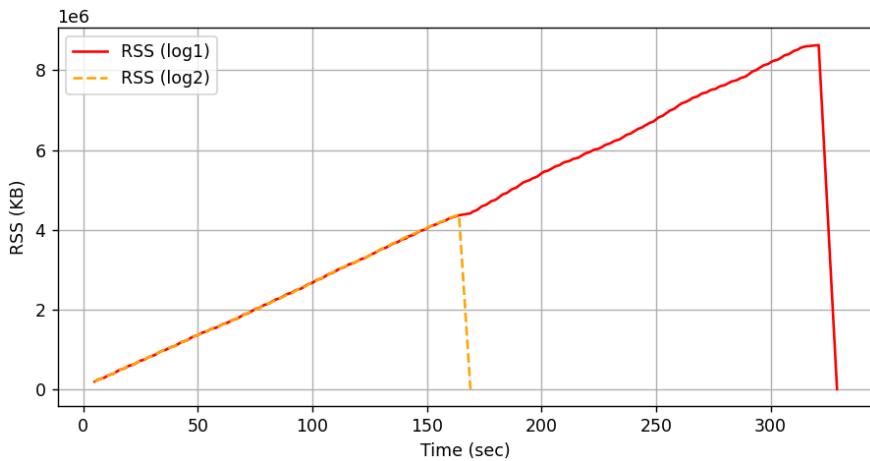
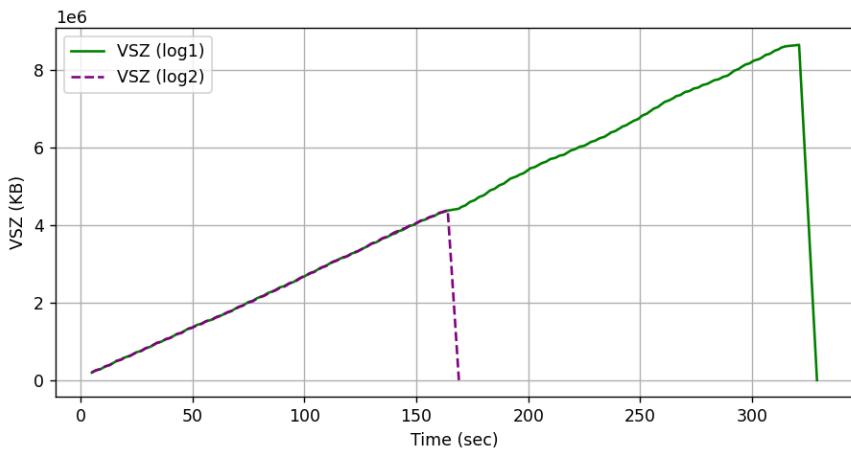
```
[DATE] 123 kernel: Out of memory: Killed process 22379 (mem2.bash) total-vm:4431644kB,  
anon-rss:4421504kB, file-rss:712kB, shmem-rss:0kB, UID:1000 pgtables:8708kB  
oom_score_adj:200
```

Последняя запись в **report.log**: 111000000

Последняя запись в **report2.log**: 560000000

Графики изменений использования памяти:





Итог: бесконечный цикл складывания элементов в массив вызвал линейный рост использования физической (RSS) и виртуальной (VSZ) памяти. Когда RSS & VSZ начали

потреблять много памяти, подключается использование свободной оперативной памяти (RAM). Когда закончилась свободная RAM, происходит очистка кэша (Buff/Cache).

Конечные значения

- Размер первого массива (после убийства процесса): 111000000; второго - 56000000

- Использовано памяти для первого скрипта:

total-vm ≈ 8,684 Гб

anon-rss ≈ 8,674 Гб

для второго скрипта:

total-vm ≈ 4,431 Гб

anon-rss ≈ 4,421 Гб

По записям из файлов **5proc.log** и **5proc2.log** видно, что процессу по началу конкурировали между собой за первое место в top.

Эксперимент 2

newmem.bash:

```
#!/bin/bash
arr=()
while true
do
  for j in {1..10}
  do
    arr+=($j)
    if [ ${#arr[*]} -gt $1 ]; then
      exit
    fi
  done
done
```

newmemstart.bash (скрипт запуска):

```
#!/bin/bash
k=$1
for ((i=0;i<k;i++))
do
  ./newmem.bash 11300000 &
  sleep 1s
done
wait
```

Набор параметров 1

N = 11300000; K = 10

Итог: успешное завершение всех процессов, в системном журнале нет записей об аварийных остановках процессов

Набор параметров 2

N = 11300000; K = 30

Итог: завершение процессов ООМ killer'ом, в системном журнале аварийная остановка

Набор параметров 3

N = 10105000; K = 30

Итог: успешное завершение всех процессов, в системном журнале нет записей об аварийных остановках процессов

Экспериментальным путем нашли N, при котором множество параллельных процессов завершаются с успехом.