

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5

по дисциплине «**Операционные системы**»

Автор: Дмитриенко Никита Андреевич

Факультет: ИТиП

Группа: М3211

Санкт-Петербург
2020

Отчёт

Данные о текущей конфигурации операционной системы:

Общий объем оперативной памяти: 2048 Мб

Объем раздела подкачки: 820 Мб

Размер страницы виртуальной памяти: 4 Кб

Объем свободной физической памяти в ненагруженной системе: 966 Мб

Объем свободного пространства в разделе подкачки в ненагруженной системе: 328 Мб

Эксперимент 1

1) Использую следующий скрипт **mem.sh** для заполнения памяти:

```
#!/bin/bash

rm report.log

counter=0
index=0
array=()

while true
do
    for (( count=1; count<11; count++ ))
    do
        array[$index]=$count
        index=$((index+1))
    done
    counter=$((counter+1))
    if [[ $counter == 100000 ]]
    then
        echo "${#array[@]}" >> report.log
        counter=0
    fi
done
```

2) После запуска скрипт достаточно быстро заполняет оперативную память, после некоторого времени процесс аварийно прекращается.

Последняя строка файла **report.log**:

18000000

При использовании команды **dmesg | grep "mem.sh"** выводится следующая информация:

```
[120756.263358] [2311411]  0 2311411  426080  291556 3047424  78975      0
mem.sh
[120756.263368] Out of memory: Killed process 2311411 (mem.sh) total-vm:1704320kB,
anon-rss:1166224kB, file-rss:0kB, shmem-rss:0kB, UID:0
[120756.321697] oom_reaper: reaped process 2311411 (mem.sh), now anon-rss:0kB,
file-rss:0kB, shmem-rss:0kB
```

Данные, полученные в ходе выполнения скрипта:

1	TIME	MEM	VIRT	RES	SHR	CPU	FREE	SWAP
2	0:00.10	0.2	223676	4248	3016	58.8	1111.0	250.1
3	0:01.45	1.0	237800	18504	3016	76.5	1097.4	250.1
4	0:02.71	1.7	250868	31440	3016	47.1	1085.0	250.1
5	0:03.84	2.3	262748	43320	3016	37.5	1066.3	250.3
6	0:04.85	2.9	273176	53880	3016	60.0	1055.8	250.5
7	0:05.90	3.5	284000	64704	3016	64.7	1044.9	250.7
8	0:06.94	4.0	294824	75528	3016	31.2	1034.8	250.9
9	0:07.96	4.6	305516	86084	3016	93.3	1023.6	250.9
10	0:09.34	5.4	319772	100340	3016	76.5	1009.7	250.9
11	0:10.73	6.1	334160	114860	3016	80.0	993.0	251.1
12	0:12.10	6.9	348416	129116	3016	81.2	979.0	251.1
13	0:13.49	7.7	362804	143372	3016	81.2	965.0	251.1
14	0:14.86	8.4	377060	157628	3016	76.5	950.5	251.1
15	0:16.24	9.2	391184	171884	3016	88.2	936.8	251.1
16	0:17.61	9.9	405308	185876	3016	76.5	923.0	251.1
17	0:18.99	10.7	418904	199604	3016	72.2	909.8	251.1
18	0:20.37	11.4	433160	213860	3016	76.5	895.7	251.2
19	0:21.77	12.2	447680	228380	3016	80.0	881.6	251.2
20	0:23.11	12.9	461540	242108	3016	81.2	868.0	251.2
21	0:24.49	13.7	475796	256364	3016	80.0	853.9	251.2
22	0:25.85	14.4	488864	269564	3016	75.0	841.2	251.2
23	0:27.23	15.2	503120	283820	3016	76.5	827.1	251.2
24	0:28.63	15.9	517508	298076	3016	72.2	813.2	251.2
25	0:30.00	16.7	531632	312332	3016	76.5	799.3	251.2
26	0:31.37	17.4	545624	326324	3016	80.0	785.7	251.2
27	0:32.74	18.2	559616	340316	3016	82.4	772.0	251.2
28	0:34.11	18.9	573740	354308	3016	75.0	758.1	251.2
29	0:35.47	19.7	587336	368036	3016	81.2	744.9	251.2
30	0:36.87	20.4	601724	382292	3016	82.4	731.2	251.2

4) График зависимости, полученный на основе данных **top**



Этап 2.

1) Создаю копию скрипта **mem.sh** и файл для запуска сразу двух скриптов.

2) Последние строки файлов **report.log** и **report2.log**:
8000000 и **1800000000**

3) При использовании команды **dmesg | grep "mem.sh"** выводится следующая информация:

```
[124062.629040] [2330207] 0 2330207 273224 155897 1826816 61767 0 mem.sh
[124062.629043] [2330698] 0 2330698 136142 80585 724992 0 0 mem2.sh
[124062.629045] [2331040] 0 2331040 113207 57648 536576 0 0 mem.sh
[124062.629063] Out of memory: Killed process 2330207 (mem.sh) total-vm:1092896kB,
anon-rss:623588kB, file-rss:0kB, shmem-rss:0kB, UID:0
[124062.755770] oom_reaper: reaped process 2330207 (mem.sh), now anon-rss:0kB, file-rss:0kB,
shmem-rss:0kB
[124401.890863] [2335807] 0 2335807 415487 304990 2969600 54925 0 mem.sh
[124401.890887] Out of memory: Killed process 2335807 (mem.sh) total-vm:1661948kB,
anon-rss:1219960kB, file-rss:0kB, shmem-rss:0kB, UID:0
[127234.378487] [2393975] 0 2393975 230027 151682 1474560 22773 0 mem2.sh
[127234.378488] [2393984] 0 2393984 230192 151957 1478656 22691 0 mem.sh
```

```
[127234.378519] Out of memory: Killed process 2393984 (mem.sh) total-vm:920768kB,  
anon-rss:607828kB, file-rss:0kB, shmem-rss:0kB, UID:0  
[127451.647407] [2393975]  0 2393975  410735  307972  2924544  47191      0 mem2.sh  
[127451.647433] Out of memory: Killed process 2393975 (mem2.sh) total-vm:1642940kB,  
anon-rss:1231888kB, file-rss:0kB, shmem-rss:0kB, UID:0
```

Данные, полученные в ходе выполнения скрипта:

4) График зависимости, полученный на основе данных **top**



Вывод: при окончании свободной физической памяти начинает использоваться файл подкачки. На графике видно, как первый скрипт заканчивает работу, освобождается память, поэтому второй скрипт продолжает работать.

Эксперимент 2

1) Создан файл newmem.sh с условием для выхода из программы

```
#!/bin/bash

counter=0
index=0
array=()

if [[ $1 == 0 ]]
then
    N=1800000
else
    N=$1
fi

while true
do
    for (( count=1; count<11; count++ ))
    do
        array[$index]=$count
        index=$((index+1))
    done
    counter=$((counter+1))
    if [[ $counter == 100000 ]]
    then
        counter=0
        if [[ "${#array[@]}" > "$N" ]]
        then
            echo "$N"
            exit 0
        fi
    fi
done
```

Также написан скрипт для запуска:

```
1  #!/bin/bash
2
3  K=$1
4  N=$2
5
6  for i in $(seq 1 $K)
7  do
8      ./newmem.sh $N&
9  done
```

2) При запуске установлены значения $K=10$ $N=1800000$

Наблюдения:

При запуске с подобными значениями программа успешно завершилась.

Тоже самое происходит с параметрами ($K=30$ $N=1800000$), ($K=30$ $N=8000000$). При ($K=30$ $N=8400000$) процесс аварийно останавливается из-за нехватки памяти.

Вывод: в ходе выполнения лабораторной работы я увидел, что данные программ заполняют физическую память до предела, после чего начинает заполняться файл подкачки. Когда заполняется и он, тогда система аварийно останавливает процесс.