

Предварительные данные:

```
maybebabyenjoyer@LAPTOP-EOQIRLHS:/mnt/d/OS-Lite/HW-5$ top -b -n 1 | head -n 5 | tail -n 2
MiB Mem : 1906.1 total, 1381.7 free, 327.4 used, 197.0 buff/cache
MiB Swap: 1024.0 total, 1024.0 free, 0.0 used. 1437.3 avail Mem
```

+ Размер страницы виртуальной памяти: 4096

(Дал WSL 2гб ОЗУ и 1гб свапа чтобы дело шло быстрее)

1.1

У нас есть скрипт `xd.sh` (`mem.sh`) который делает то, что написано в тз (Создайте скрипт `mem.bash`, реализующий следующий сценарий. Скрипт выполняет бесконечный цикл. Перед началом выполнения цикла создается пустой массив и счетчик шагов, инициализированный нулем. На каждом шаге цикла в конец массива добавляется последовательность из 10 элементов, например, (1 2 3 4 5 6 7 8 9 10). Каждый 100000-ый шаг в файл `report.log` добавляется строка с текущим значением размера массива (перед запуском скрипта, файл обнуляется).

```
#!/bin/bash

echo $$ >pid_mem

a=()
c=0

while ;; do
    a+=(1 2 3 4 5 6 7 8 9 10)
    ((c++))
    if [[ $c -eq 100000 ]]; then
        echo "${#a[*]}" >>report.log
        c=0
    fi
done
```

И он умер

```
maybebabyenjoyer@LAPTOP-EOQIRLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -3
[ 243.268919] [ 335] 0 335 618 15 45056 16 0 Relay
[ 243.269340] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task-bash,pid=331,uid=0
[ 243.269984] Out of memory: Killed process 331 (bash) total-vm:2611928kB, anon-rss:1632208kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:5140kB oom_score_adj:0
```

Последнее значение в логe: 33000000

Дальше есть скрипт который следит за топом и выводит данные в файл (xdd.sh):

```
#!/bin/bash
```

```

target="log_1"

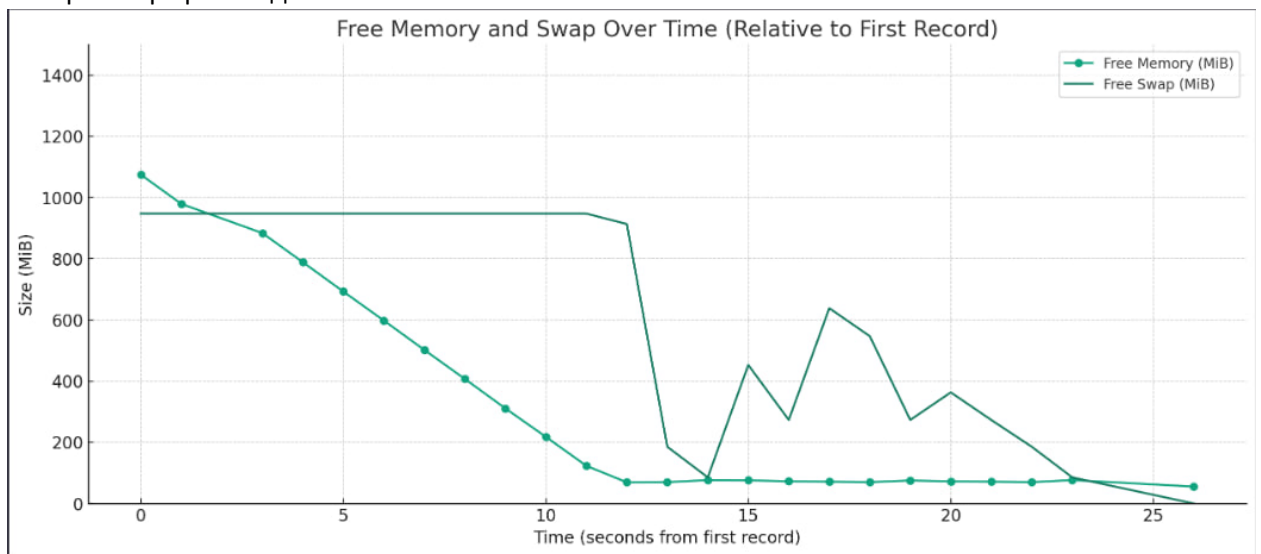
pid=$(cat pid_mem)

while ;; do
    a=$(top -b -n 1)
    b=$(grep "$pid" <<<"$a")
    if [[ -z $b ]]; then
        break
    fi
    date +%d.%m.%Y\ %H:%M:%S >>$target
    awk '{print $0}' <<<"$a" | head -n 5 | tail -n 2 >>$target
    echo "$b" >>$target
    awk '{print $0}' <<<"$a" | head -n 12 | tail -n 5 >>$target
    echo >>$target
    sleep 5
done

sudo dmesg | grep "xd.sh"

```

Построим график по данным из топа:



Ну скрипт закономерно надувался по памяти со временем (пока память не кончилась, потом дуться уже некуда стало).

Вывод: после того как оперативка кончилась, произошел страничный обмен с подкачкой и ее трата, а когда и подкачка закончилась, скрипт свалился с OOM (ожидаемо).

1.2

Создадим еще один скрипт (такой же как первый)

```
#!/bin/bash

echo $$ >pid1_mem

a=()
c=0

while ;; do
    a+=(1 2 3 4 5 6 7 8 9 10)
    ((c++))
    if [[ $c -eq 100000 ]]; then
        echo "${#a[*]}" >>report.log
        c=0
    fi
done
```

Также сделаем новый скрипт для отслеживания топа(xdd1.sh):

```
#!/bin/bash

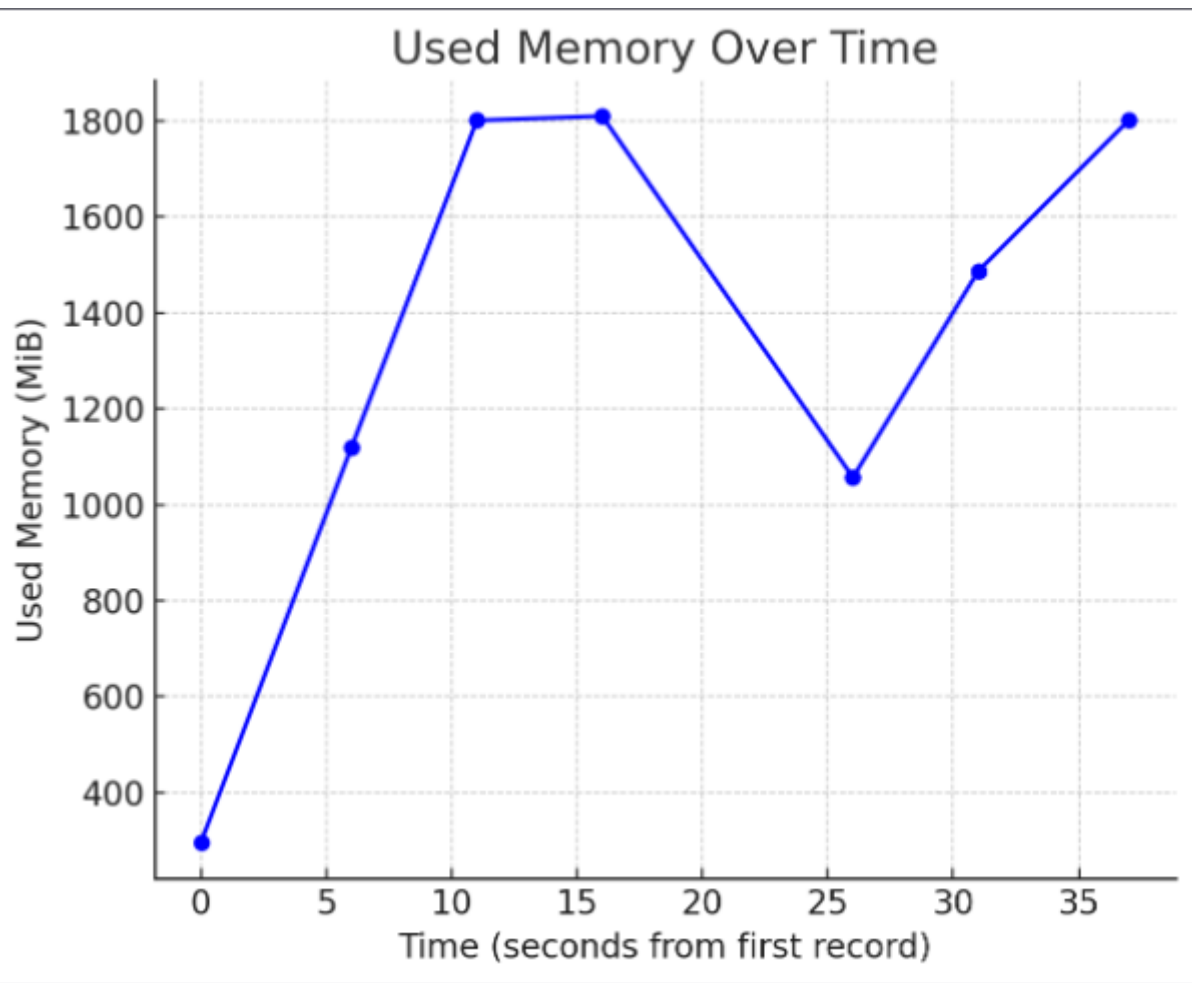
target="log_2"

pid=$(cat pid_mem)
pid2=$(cat pid1_mem)

while true; do
    a=$(top -b -n 1)
    b=$(grep "$pid" <<<"$a")
    b2=$(grep "$pid2" <<<"$a")
    if [[ -z $b && -z $b2 ]]; then
        break
    fi
    date +%d.%m.%Y\ %H:%M:%S >>$target
    awk '{print $0}' <<<"$a" | head -n 5 | tail -n 2 >>$target
    echo "$b" >>$target
    echo "$b2" >>$target
    awk '{print $0}' <<<"$a" | head -n 12 | tail -n 5 >>$target
    echo >>$target
    sleep 5
done

sudo dmesg | grep "xd.sh"
```

И по данным топа построим новый график:



Что тут интересного? Ну в какой-то момент количество памяти резко возросло. Это умер первый скрипт, и поэтому память освободилась (ее потом забрал второй да)

```
maybebabyenjoyer@LAPTOP-EQIURLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -6
[ 2211.869292] [ 3312]  0 3312 2758    0 57344 177      0 sudo
[ 2211.869732] [ 3313]  0 3313 2189    48 45056 70      0 bash
[ 2211.870166] [ 3314]  0 3314 647075 405501 5214208 239495 0 bash
[ 2211.870638] [ 3396]  0 3396 1374    55 45056 0      0 top
[ 2211.871077] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task=bash,pid=3314,uid=0
[ 2211.871741] Out of memory: Killed process 3314 (bash) total-vm:2588300kB, anon-rss:1622004kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:5092kB oom_score_adj:0
```

Вывод: Сначала у нас память кончилась в момент когда каждый из скриптов делил ее +- 50/50 (так как последняя запись в логе для первого – 16000000, потом он умер, и его память забрал второй (и его крайнее значение было 32000000)). А потом он тоже умер. Ну, вполне ожидаемое поведение.

Наш новый скрипт (из 1 задания (ну почти))

```
#!/bin/bash

a=()

while :
do
    a+=(1 2 3 4 5 6 7 8 9 10)
    if [[ "${#a[*]}" -ge $1 ]];
    then
        exit 0
    fi
done
```

Ну короче оно делает то, что просят в тз (Измените копию таким образом, чтобы она завершала работу, как только размер создаваемого массива превысит значение N, передаваемое в качестве параметра скрипту. Уберите запись данных в файл.)

Проблемы начались достаточно быстро ведь при запуске каждую секунду и условия из тз никто не умирал. Каждые 0.5 тоже. Поэтому имеем каждые 0.3

```
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ sudo killall bash
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ sudo bash runxd.sh 30 3300000
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -3
[ 3079.275672] [ 3582] 0 3582 2189 54 49152 66 0 bash
[ 3079.276091] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task=bash,pid=3485,uid=0
[ 3079.276731] Out of memory: Killed process 3485 (bash) total-vm:2577872kB, anon-rss:1622020kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:5080kB oom_score_adj:0
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$
```

(никто не умер)

```
#!/bin/bash

for (( i = 0; i < $1; i++ ));
do
    ./newxd.sh "$2" &
    sleep 0.3
done
```

Потом поделал что просят в тз (да при k = 10, n = 3300000) никто не умирает. При k = 30:

```
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -3
[ 3079.275672] [ 3582] 0 3582 2189 54 49152 66 0 bash
[ 3079.276091] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task=bash,pid=3485,uid=0
[ 3079.276731] Out of memory: Killed process 3485 (bash) total-vm:2577872kB, anon-rss:1622020kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:5080kB oom_score_adj:0
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ sudo bash runxd.sh 30 3300000
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -3
[ 3079.275672] [ 3582] 0 3582 2189 54 49152 66 0 bash
[ 3079.276091] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task=bash,pid=3485,uid=0
[ 3079.276731] Out of memory: Killed process 3485 (bash) total-vm:2577872kB, anon-rss:1622020kB, file-rss:0kB, shmem-rss:0kB, UID:0 pgtables:5080kB oom_score_adj:0
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ sudo bash runxd.sh 30 3300000
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$ dmesg | tail -3
[ 4642.944254] [ 3852] 0 3852 27863 25692 266240 0 0 newxd.sh
[ 4642.944954] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/,task=newxd.sh,pid=3822,uid=0
[ 4642.945939] Out of memory: Killed process 3822 (newxd.sh) total-vm:266156kB, anon-rss:100000kB, file-rss:4kB, shmem-rss:0kB, UID:0 pgtables:560kB oom_score_adj:0
maybebabyenjoyer@LAPTOP-EQOIRLHS:/mnt/d/OS-Lite/HW-5$
```

(кто-то умер)

Напишем скрипт для бинпоиска (вообще можно было бы руками, и возможно более честные результаты бы получились, но..)

```
#!/bin/bash

check_oom() {
    dmesg -C
    ./runxd.sh "$1" "$2"
    wait

    if dmesg | grep -q 'Out of memory: Killed process'; then
        return 1
    else
        return 0
    fi
}

binary_search() {
    local low=1
    local high=3300000
    local mid
    local best=1

    while [ $low -le $high ]; do
        mid=$((low + high) / 2)

        echo "Testing with n = $mid"

        if check_oom "$1" $mid; then
            best=$mid
            low=$((mid + 1))
        else
            high=$((mid - 1))
        fi
    done

    echo "Maximum safe value close to 3300000 found: $best"
}

binary_search "$1"
```

(Если кто-то умер берем поменьше, если никто не умер берем побольше)

```

maybebabyenjoyer@LAPTOP-EQIIRLHS:/mnt/d/OS-Lite/HW-5$ sudo bash xdsearch.sh 30
Testing with n = 1650000
Testing with n = 2475000
Testing with n = 2887500
Testing with n = 3093750
./runxd.sh: line 7: 4092 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4090 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4094 Killed                  ./newxd.sh $2
./runxd.sh: line 8: 4096 Killed                  ./newxd.sh $2
Testing with n = 2990625
./runxd.sh: line 7: 4154 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4156 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4160 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4164 Killed                  ./newxd.sh $2
Testing with n = 2939062
./runxd.sh: line 7: 4218 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 4220 Killed                  ./newxd.sh $2
Testing with n = 2913281
./runxd.sh: line 7: 4284 Killed                  ./newxd.sh $2
Testing with n = 2900390
./runxd.sh: line 7: 4346 Killed                  ./newxd.sh $2
Testing with n = 2893945
Testing with n = 2890722
./runxd.sh: line 7: 4474 Killed                  ./newxd.sh $2
Testing with n = 2889111
Testing with n = 2888305
./runxd.sh: line 7: 4602 Killed                  ./newxd.sh $2
./runxd.sh: line 8: 4606 Killed                  ./newxd.sh $2
Testing with n = 2887902
Testing with n = 2887701
Testing with n = 2887600
Testing with n = 2887550
Testing with n = 2887525
Testing with n = 2887512
Testing with n = 2887506
./runxd.sh: line 7: 5050 Killed                  ./newxd.sh $2
./runxd.sh: line 7: 5052 Killed                  ./newxd.sh $2
Testing with n = 2887503
Testing with n = 2887501
Maximum safe value close to 3300000 found: 2887500

```

Получилось 2887500. Круто. Возможно, это не совсем правда, но лучше перейдем к

Вывод: если мы запускаем несколько процессов через какой-то интервал, но предыдущие не успели завершиться, можем словить OOM, и тогда кто-то умирает, чтобы другие процессы не свалились. А если мы последовательно запускаем, они успевают отработать, и не едят всю память, то у нас никто умирать не будет. Это хорошо.