#uni/semester3/Betriebssysteme/chapter21/homework

1. First, open two separate terminal connections to the same machine, so that you can easily run something in one window and the other. Now, in one window, run vmstat 1, which shows statistics about machine usage every second. Read the man page, the associated README, and any other information you need so that you can understand its output. Leave this window running vmstat for the rest of the exercises below. Now, we will run the program mem.c but with very little memory usage. This can be accomplished by typing ./mem 1 (which uses only 1 MB of memory). How do the CPU usage statistics change when running mem? Do the numbers in the user time column make sense? How does this change when running more than one instance of mem at once?

   What is the output of vmstat 1?

   ➜ vmstat reports information about processes, memory, paging, block IO, traps and cpu activity.

   ➜ first report gives averages since the last reboot. Additional reports are given on a delay, which is the 1 second. Process and memory reports are instantaneous

   ➜ default shows memory in KiB!

Field Description For Vm Mode

Procs

```
r: The number of processes waiting for run time.
b: The number of processes in uninterruptible sleep.
```

Memory

```
swpd: the amount of virtual memory used.
free: the amount of idle memory.
buff: the amount of memory used as buffers.
cache: the amount of memory used as cache.
inact: the amount of inactive memory. (-a option)
active: the amount of active memory. (-a option)
```

Swap

```
si: Amount of memory swapped in from disk (/s).
so: Amount of memory swapped to disk (/s).
```

IO

```
bi: Blocks received from a block device (blocks/s).
bo: Blocks sent to a block device (blocks/s).
```

System

```
in: The number of interrupts per second, including the clock.
cs: The number of context switches per second.
```

CPU

These are percentages of total CPU time.

```
us: Time spent running non-kernel code. (user time, including nice time)
sy: Time spent running kernel code. (system time)
id: Time spent idle. Prior to Linux 2.5.41, this includes IO-wait time.
wa: Time spent waiting for IO. Prior to Linux 2.5.41, included in idle.
st: Time stolen from a virtual machine. Prior to Linux 2.6.11, unknown.
```

cpu

The last five columns give the percentages of total CPU time:

us	Percentage of CPU cycles spent on user processes
sy	Percentage of CPU cycles spent on system (kernel) processes
id	Percentage of CPU cycles spent idle
wa	Percentage of CPU cycles spent waiting for I/O
st	Percentage of CPU cycles stolen from a virtual machine

How do the CPU usage statistics change when running mem? Do the numbers in the user

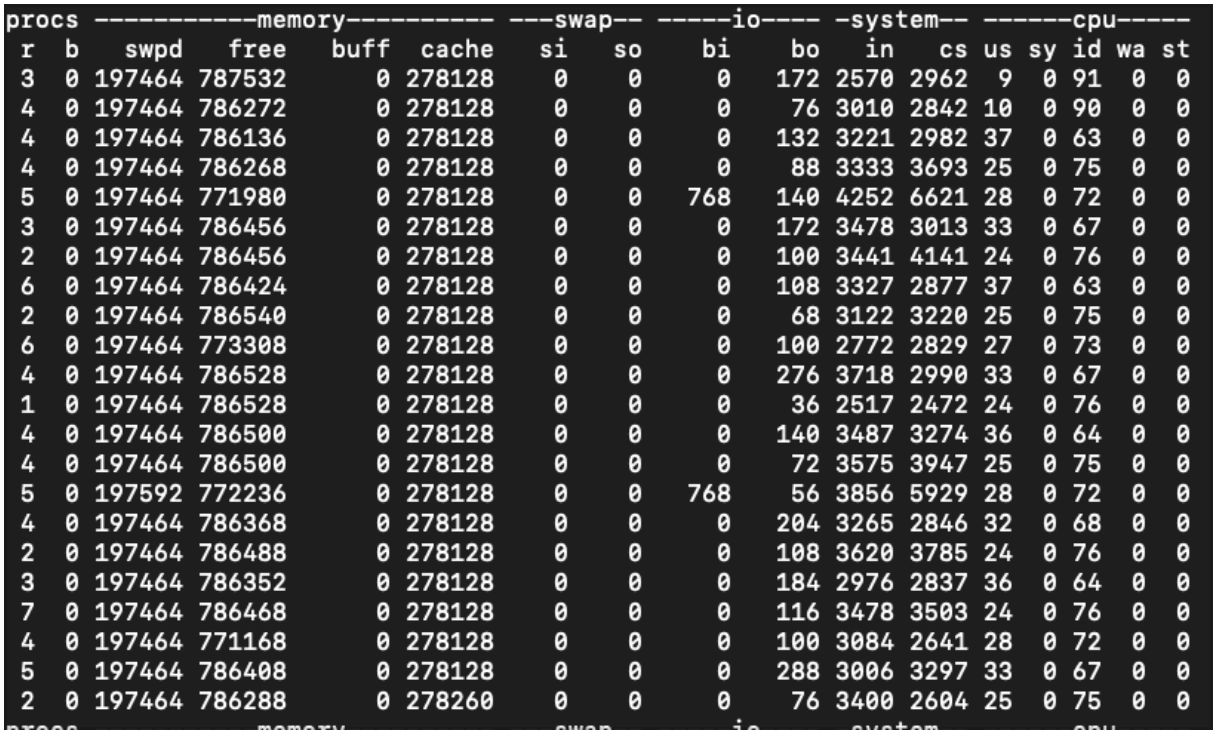
time column make sense?

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1
allocating 1048576 bytes (1.00 MB)
  number of integers in array: 262144
loop 0 in 0.86 ms (bandwidth: 1158.97 MB/s)
loop 957 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 1914 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 2871 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 3828 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 4785 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 5742 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 6699 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 7656 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 8613 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 9570 in 0.24 ms (bandwidth: 4240.95 MB/s)
loop 10527 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 11484 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 12440 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 13395 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 14352 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 15308 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 16265 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 17221 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 18178 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 19134 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 20091 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 21048 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 22005 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 22962 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 23919 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 24876 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 25833 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 26790 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 27747 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 28704 in 0.22 ms (bandwidth: 4604.07 MB/s)
```

➜ A) Before running ./mem 1

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0 197464 787708      0 278128    0    0     0   216 2966 3391  0  0 100  0  0
 4  0 197464 780832      0 278128    0    0     0    88 2610 2908  2  0 98  0  0
 4  0 197464 787868      0 278128    0    0     0   144 3402 3271 11  0 89  0  0
 1  0 197464 787868      0 278128    0    0     0    36 2516 2739  0  0 100  0  0
 5  0 197464 787840      0 278128    0    0     0   168 2827 2930 13  0 87  0  0
 1  0 197464 787840      0 278128    0    0     0   148 3621 3980  1  0 99  0  0
 3  0 197464 779304      0 278128    0    0   768   168 3124 5199  2  0 98  0  0
 5  0 197464 787816      0 278128    0    0     0    32 3815 4694 11  0 89  0  0
 1  0 197464 787816      0 278128    0    0     0    80 2551 3002  0  0 100  0  0
 4  0 197464 787772      0 278128    0    0     0   160 2774 3111 13  0 87  0  0
 2  0 197464 787680      0 278128    0    0     0   244 3415 3697  0  0 100  0  0
 4  0 197464 777572      0 278128    0    0     0   196 2622 2938  3  0 97  0  0
 4  0 197464 787580      0 278128    0    0     0    64 3072 3136 10  0 90  0  0
 3  0 197464 787608      0 278128    0    0     0    44 2897 3033  1  0 99  0  0
 3  0 197464 787584      0 278128    0    0     0     4 3392 4112 12  0 88  0  0
 3  0 197464 787580      0 278128    0    0     0   200 2944 3319  0  0 100  0  0
 4  0 197464 776288      0 278128    0    0   768   184 4089 5833  4  0 96  0  0
 2  0 197464 787544      0 278128    0    0     0   264 2801 3718 10  0 90  0  0
 4  0 197464 787572      0 278128    0    0     0    76 3732 4021  0  0 100  0  0
 1  0 197464 787480      0 278128    0    0     0    32 2827 2997 12  0 88  0  0
 3  0 197464 787540      0 278128    0    0     0     0 2683 3330  0  0 100  0  0
 3  0 197464 773892      0 278128    0    0     0   228 3129 2978  4  0 96  0  0
```

➜ B) During running ./mem 1



```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0 197464 787532      0 278128    0    0     0   172 2570 2962  9  0 91  0  0
 4  0 197464 786272      0 278128    0    0     0    76 3010 2842 10  0 90  0  0
 4  0 197464 786136      0 278128    0    0     0   132 3221 2982 37  0 63  0  0
 4  0 197464 786268      0 278128    0    0     0    88 3333 3693 25  0 75  0  0
 5  0 197464 771980      0 278128    0    0   768   140 4252 6621 28  0 72  0  0
 3  0 197464 786456      0 278128    0    0     0   172 3478 3013 33  0 67  0  0
 2  0 197464 786456      0 278128    0    0     0   100 3441 4141 24  0 76  0  0
 6  0 197464 786424      0 278128    0    0     0   108 3327 2877 37  0 63  0  0
 2  0 197464 786540      0 278128    0    0     0    68 3122 3220 25  0 75  0  0
 6  0 197464 773308      0 278128    0    0     0   100 2772 2829 27  0 73  0  0
 4  0 197464 786528      0 278128    0    0     0   276 3718 2990 33  0 67  0  0
 1  0 197464 786528      0 278128    0    0     0    36 2517 2472 24  0 76  0  0
 4  0 197464 786500      0 278128    0    0     0   140 3487 3274 36  0 64  0  0
 4  0 197464 786500      0 278128    0    0     0    72 3575 3947 25  0 75  0  0
 5  0 197592 772236      0 278128    0    0   768    56 3856 5929 28  0 72  0  0
 4  0 197464 786368      0 278128    0    0     0   204 3265 2846 32  0 68  0  0
 2  0 197464 786488      0 278128    0    0     0   108 3620 3785 24  0 76  0  0
 3  0 197464 786352      0 278128    0    0     0   184 2976 2837 36  0 64  0  0
 7  0 197464 786468      0 278128    0    0     0   116 3478 3503 24  0 76  0  0
 4  0 197464 771168      0 278128    0    0     0   100 3084 2641 28  0 72  0  0
 5  0 197464 786408      0 278128    0    0     0   288 3006 3297 33  0 67  0  0
 2  0 197464 786288      0 278260    0    0     0    76 3400 2604 25  0 75  0  0
```

➜ In B) (during /.mem 1) at one point there are 6 or 7 processes waiting for runtime, while in A) the max of this number is at 5

➜ the amount of virtual memory used (swpd) didn't change

➜ idle memory decreased with 1024 Bytes. In A) 787K and in B) 786K

➜ cache didn't change

➜ memory swapped in and out didn't change

➜ blocks received and blocks sent to a disk device didn't really change

➜ interrupts per second increased, almost the whole column is above 3000 in B) and before in A) most of it was under 3000

➜ I don't see a pattern in context switches per seconds. Noting really that changed

➜ User time definitely increased in B). The percentage 0% isn't reached anymore. So about 30% of the cpu cycles, are user time in B).

➜ Idle time decreased in B). The percentage 100 % isn't reached anymore. Now only about 75% of the cpu cycles are spent idle.

How does this change when running more than one instance of mem at once?

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1
& ./mem 1
[1] 12012
allocating 1048576 bytes (1.00 MB)
allocating 1048576 bytes (1.00 MB)
  number of integers in array: 262144
  number of integers in array: 262144
loop 0 in 0.86 ms (bandwidth: 1167.03 MB/s)
loop 0 in 0.87 ms (bandwidth: 1144.42 MB/s)
loop 930 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 957 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 1884 in 0.21 ms (bandwidth: 4788.02 MB/s)
loop 1899 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 2709 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 2723 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 3347 in 0.21 ms (bandwidth: 4809.98 MB/s)
loop 3360 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 3987 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 4019 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 4607 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 4658 in 0.34 ms (bandwidth: 2957.90 MB/s)
loop 5440 in 0.21 ms (bandwidth: 4804.47 MB/s)
loop 5512 in 0.21 ms (bandwidth: 4782.56 MB/s)
loop 6393 in 0.21 ms (bandwidth: 4788.02 MB/s)
```

➜ kill the second one with top

➜ Before running two instances

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 2  0 197464 772192      0 278588    0    0     0   136 2829 3027  4  0 96  0  0
 3  0 197464 786600      0 278588    0    0     0   164 2987 2970  9  0 91  0  0
 4  0 197464 786600      0 278588    0    0     0    36 3165 3287  0  0 100  0  0
 2  0 197464 786568      0 278588    0    0     0   336 2657 2567 12  0 88  0  0
 4  0 197464 786600      0 278588    0    0     0   104 3038 2979  0  0 100  0  0
 2  0 197464 769700      0 278588    0    0     0    96 2904 2999  5  0 95  0  0
 3  0 197464 786580      0 278588    0    0     0   100 2935 3311  8  0 92  0  0
 6  0 197464 786580      0 278588    0    0   768   112 4061 5572  0  0 100  0  0
 2  0 197464 786472      0 278588    0    0     0   180 3018 4087 12  0 88  0  0
 1  0 197464 786560      0 278588    0    0     0    72 3636 4290  0  0 100  0  0
 2  0 197464 767672      0 278588    0    0     0    72 3059 2989  5  0 95  0  0
 3  0 197464 786764      0 278588    0    0     0   108 2847 2799  7  0 93  0  0
 4  0 197464 786764      0 278588    0    0     0   176 3026 3661  0  0 100  0  0
 4  0 197464 786744      0 278588    0    0     0   168 3020 2622 12  0 88  0  0
 1  0 197464 786744      0 278588    0    0     0    48 2870 2911  0  0 100  0  0
 5  0 197464 765956      0 278588    0    0     0    36 2896 3295  6  0 94  0  0
 3  0 197464 786716      0 278588    0    0     0   144 3683 4041  7  0 93  0  0
 4  0 197464 786716      0 278588    0    0   768    36 3959 6214  0  0 100  0  0
 3  0 197464 786520      0 278588    0    0     0    80 3231 3077 12  0 88  0  0
 2  0 197464 786676      0 278588    0    0     0   192 3199 4221  0  0 100  0  0
 2  0 197464 764396      0 278588    0    0     0   104 3095 2982  6  0 94  0  0
 1  0 197464 786656      0 278588    0    0     0   104 3015 3027  6  0 94  0  0
```

➜ C) while running two instances of ./mem 1

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 4  0 197464 771572      0 278588    0    0     0   208 3157 2852 50  0 50  0  0
 7  0 197464 790812      0 277712    0    0     0   136 3616 2973 48  0 52  0  0
 3  0 197464 790736      0 277712    0    0     0    96 3588 3700 50  0 50  0  0
 7  0 197464 782968      0 277712    0    0   384    72 3662 4583 50  0 50  0  0
 7  0 197464 790440      0 277712    0    0   384   108 3579 3986 47  0 53  0  0
 4  0 197464 786672      0 277712    0    0     0    72 3735 4054 48  0 52  0  0
 4  0 197464 785620      0 277712    0    0     0   136 3625 3121 49  0 51  0  0
 6  0 197464 785032      0 277976    0    0     0    76 3401 2874 48  0 52  0  0
 6  0 197464 784144      0 277976    0    0     0     0 3513 3089 48  0 52  0  0
 4  0 197464 784800      0 278240    0    0     0   232 3272 3048 48  0 52  0  0
 4  0 197464 784884      0 278240    0    0     0    36 3244 2609 49  0 51  0  0
 5  0 197464 771124      0 278240    0    0     0   164 3369 2955 49  0 51  0  0
 4  0 197464 784724      0 278240    0    0     0    76 4067 4008 48  0 52  0  0
 5  0 197464 784724      0 278240    0    0   768    32 3799 4597 48  0 52  0  0
 6  0 197464 780384      0 278240    0    0     0   100 3623 4168 50  0 50  0  0
 5  0 197464 784804      0 278240    0    0     0   144 3846 4150 47  0 53  0  0
 7  0 197464 784804      0 278240    0    0     0   104 3444 2941 48  0 52  0  0
 4  0 197464 765228      0 278240    0    0     0   136 3596 3056 49  0 51  0  0
 6  0 197464 784740      0 278240    0    0     0    72 3410 3204 48  0 52  0  0
 6  0 197464 784740      0 278240    0    0     0    20 3294 2849 50  0 50  0  0
 4  0 197464 784616      0 278240    0    0     0   104 2892 2551 48  0 52  0  0
 4  0 197464 784736      0 278240    0    0     0    44 3478 2935 49  0 51  0  0
```

➜ r column, now there are more processes waiting for run time

➜ free column, decreased by 2000, now 784K

➜ cache column, at the beginning it decreased by 1000 for a time, then increased again

➜ bi and bo didn't really change

➜ in column, interrupts per second increased

➜ cs column, context switches per second didn't really change, maybe decreased for a little bit

➜ us time, increased even more as in B), because there are now two instances. It is now at 50%

➜ id column, idle time also decreased more. It is at 50%

2. Let's now start looking at some of the memory statistics while running mem. We'll focus on two columns: swpd (the amount of virtual memory used) and free (the amount of idle memory). Run ./mem 1024 (which allocates 1024 MB) and watch how these values change. Then kill the running program (by typing control-c) and watch again how the values change. What do you notice about the values? In particular, how does the free column change when the program exits? Does the amount of free memory increase by the expected amount when mem exits?

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1024
allocating 1073741824 bytes (1024.00 MB)
  number of integers in array: 268435456
loop 0 in 2337.29 ms (bandwidth: 438.11 MB/s)
loop 1 in 525.11 ms (bandwidth: 1950.05 MB/s)
loop 2 in 231.34 ms (bandwidth: 4426.33 MB/s)
loop 3 in 229.35 ms (bandwidth: 4464.75 MB/s)
loop 4 in 393.54 ms (bandwidth: 2602.05 MB/s)
loop 5 in 1054.20 ms (bandwidth: 971.35 MB/s)
loop 6 in 408.86 ms (bandwidth: 2504.51 MB/s)
loop 7 in 320.25 ms (bandwidth: 3197.47 MB/s)
loop 8 in 380.53 ms (bandwidth: 2690.98 MB/s)
loop 9 in 313.00 ms (bandwidth: 3271.55 MB/s)
loop 10 in 249.89 ms (bandwidth: 4097.75 MB/s)
loop 11 in 251.27 ms (bandwidth: 4075.30 MB/s)
loop 12 in 255.10 ms (bandwidth: 4014.19 MB/s)
loop 13 in 254.67 ms (bandwidth: 4020.86 MB/s)
loop 14 in 263.08 ms (bandwidth: 3892.37 MB/s)
loop 15 in 263.09 ms (bandwidth: 3892.19 MB/s)
loop 16 in 656.97 ms (bandwidth: 1558.67 MB/s)
loop 17 in 245.46 ms (bandwidth: 4171.83 MB/s)
loop 18 in 411.97 ms (bandwidth: 2485.62 MB/s)
loop 19 in 234.81 ms (bandwidth: 4360.99 MB/s)
loop 20 in 401.37 ms (bandwidth: 2551.26 MB/s)
```

➜ A) before running

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0 203508 1082324      0 172872    0    0     0   120 2873 2765  6  0 94  0  0
 4  0 203508 1082444      0 172872    0    0     0    68 2646 4051  0  0 100  0  0
 3  0 203508 1082404      0 172872    0    0     0   168 3502 2907 13  0 88  0  0
 0  0 203508 1082440      0 172872    0    0     0   144 2615 3058  0  0 100  0  0
 7  0 203508 1077800      0 172872    0    0     0    68 2807 3338  7  0 93  0  0
 3  0 203508 1082412      0 172872    0    0     0   176 3157 2904  5  0 95  0  0
 0  0 203508 1082412      0 172872    0    0     0    72 2511 3017  0  0 100  0  0
 4  0 203508 1082368      0 172872    0    0  1352   100 3409 3509 12  0 88  0  0
 2  0 203508 1082368      0 172872    0    0     0    72 3828 4182  0  0 100  0  0
 6  0 203508 1075744      0 172872    0    0     0    32 3159 5032  8  0 92  0  0
 6  0 203508 1082364      0 172872    0    0   768   132 3502 4457  5  0 95  0  0
 0  0 203508 1082364      0 172872    0    0     0   204 3487 4266  0  0 100  0  0
 1  0 203508 1082332      0 172872    0    0     0   160 2675 2760 12  0 88  0  0
 3  0 203508 1082332      0 172872    0    0     0   108 3157 3027  0  0 100  0  0
 4  0 203508 1073552      0 172872    0    0     0    64 2835 3316  9  0 91  0  0
 3  0 203508 1082328      0 172872    0    0     0    80 2815 3063  4  0 96  0  0
 3  0 203508 1082328      0 172872    0    0     0   128 3043 3062  0  0 100  0  0
 1  0 203508 1082276      0 172872    0    0     0   188 2932 2514 12  0 88  0  0
 3  0 203508 1082276      0 172872    0    0     0    72 2929 3661  0  0 100  0  0
 7  0 203508 1072144      0 172872    0    0   768    68 3884 4945  9  0 91  0  0
 1  0 203508 1082436      0 172872    0    0     0    36 2744 4096  4  0 96  0  0
 4  0 203508 1082436      0 172872    0    0     0    88 3408 4131  0  0 100  0  0
```

➜ B) during running

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0 203556 323436      0 172872   12    0    12   112 3179 2583 28  0 72  0  0
 3  0 203300  31592      0 172872    0    0     0   208 2583 2907 24  0 76  0  0
 6  0 203300  23056      0 172872    0    0     0   100 3830 3182 33  0 67  0  0
 2  0 203300  31600      0 172872    0    0     0    36 3108 3065 29  0 71  0  0
 5  0 203300  31600      0 172872    0    0     0    56 2634 2633 25  0 75  0  0
 3  0 203300  31544      0 172872    0    0     0   144 3500 2535 37  0 63  0  0
 4  0 203300  31560      0 172872    0    0     0    40 2995 3480 24  0 76  0  0
 8  0 203300  29208      0 172872    0    0   768   364 4088 5069 32  0 68  0  0
 3  0 203300  31556      0 172872    0    0     0   100 3559 4220 31  0 69  0  0
 4  0 203300  31556      0 172872    0    0     0    84 3328 4055 25  0 75  0  0
 6  0 203300  31420      0 172872    0    0     0   184 3260 2512 37  0 63  0  0
 1  0 203300  31528      0 172872    0    0     0    40 3174 2788 24  0 76  0  0
 3  0 203300  11888      0 172872    0    0     0    64 3081 3440 30  0 70  0  0
 6  0 203300  31472      0 172872    0    0     0   104 3566 2921 32  0 68  0  0
 3  0 203300  31472      0 172872    0    0     0    52 2659 2543 25  0 75  0  0
 5  0 203300  39396      0 171344    0    0     0   136 3000 2409 38  0 62  0  0
 2  0 203300  39200      0 171344    0    0     0   148 3883 3334 25  0 75  0  0
 4  0 203300  24448      0 171344    0    0   768   180 3937 5371 28  0 72  0  0
 6  0 203300  35244      0 171344    0    0     0    68 3568 3880 35  0 65  0  0
 4  0 203300  35352      0 171344    0    0     0   100 3701 4039 25  0 75  0  0
 3  0 203300  33956      0 171344    0    0     0    36 3012 2520 39  0 61  0  0
 3  0 203300  32608      0 172004    0    0     0   180 3354 2993 25  0 75  0  0
```

➜ C) after running

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 6  0 203300  21868      0 172400    0    0     0   140 3535 3235 27  0 73  0  0
 4  0 203300 1082564      0 172532    4    0     4   220 2879 2853 27  0 73  0  0
 6  0 203300 1082308      0 172532    0    0     0   104 3016 2841  0  0 100  0  0
 1  0 203300 1082408      0 172532    0    0     0    36 2884 2658 12  0 88  0  0
 2  0 203300 1082488      0 172532    0    0     0   144 2453 2956  0  0 100  0  0
 4  0 203300 1073820      0 172532    0    0   768   108 4287 6232  2  0 98  0  0
 2  0 203300 1082480      0 172532    0    0     0    96 2486 2904 10  0 90  0  0
 5  0 203300 1082480      0 172532    0    0     0    88 2760 3014  0  0 100  0  0
 1  0 203300 1082404      0 172532    0    0     0   108 3129 2536 13  0 87  0  0
 2  0 203300 1082480      0 172532    0    0     0    40 2475 3029  0  0 100  0  0
 7  0 203300 1071896      0 172532    0    0     0   176 3092 3280  3  0 97  0  0
 2  0 203300 1082464      0 172532    0    0     0    32 2981 2772 10  0 90  0  0
 2  0 203300 1082464      0 172532    0    0     0    88 2357 2900  0  0 100  0  0
 3  0 203300 1082428      0 172532    0    0     0   172 3307 2904 12  0 88  0  0
 0  0 203300 1082632      0 172532    0    0     0   108 2893 3462  0  0 100  0  0
 5  0 203300 1069892      0 172532    0    0   768   116 3798 6601  4  0 96  0  0
 3  0 203300 1082604      0 172532    0    0     0    64 3373 2994  9  0 91  0  0
 3  0 203300 1082604      0 172532    0    0     0    52 2203 3020  0  0 100  0  0
 7  0 203300 1082544      0 172532    0    0     0    56 2952 2676 12  0 88  0  0
 1  0 203300 1082544      0 172532    0    0     0   188 3042 3173  0  0 100  0  0
 5  0 203300 1068308      0 172532    0    0     0   260 2682 3355  4  0 96  0  0
 4  0 203300 1082472      0 172532    0    0     0   136 3343 3303  8  0 92  0  0
```

What do you notice about the values? In particular, how does the free column change when the program exits? Does the amount of free memory increase by the expected amount when mem exits?

➜ swpd is in other words, how much memory has been swapped out

➜ swpd decreases in B) and in C) it remains the same

➜ free column is the amount of currently unused memory

➜ In a) is is high at around 1GiB. In b) it decreased by 1024 * 1024 KiB, so it gets at

around 33000 KB. In c) it goes back up to 1GiB.

3. We'll next look at the swap columns (si and so), which indicate how much swapping is taking place to and from the disk. Of course, to activate these, you'll need to run mem with large amounts of memory. First, examine how much free memory is on your Linux system (for example, by typing cat *proc*meminfo; type man proc for details on the *proc file system and the types of information you can find there). One of the first entries in /proc*meminfo is the total amount of memory in your system. Let's assume it's something like 8 GB of memory; if so, start by running mem 4000 (about 4 GB) and watching the swap in/out columns. Do they ever give non-zero values? Then, try with 5000, 6000, etc. What happens to these values as the program enters the second loop (and beyond), as compared to the first loop? How much data (total) are swapped in and out during the second, third, and subsequent loops? (do the numbers make sense?)

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$
cat /proc/meminfo
MemTotal:        2097152 kB
MemFree:         1082288 kB
MemAvailable:    1254644 kB
Buffers:               0 kB
Cached:           172356 kB
SwapCached:            0 kB
Active:           237432 kB
Inactive:         173216 kB
Active(anon):     132524 kB
Inactive(anon):   151488 kB
Active(file):     104908 kB
Inactive(file):    21728 kB
Unevictable:           0 kB
Mlocked:          191260 kB
SwapTotal:       3145728 kB
SwapFree:        2942508 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:        238628 kB
Mapped:            85932 kB
Shmem:             45384 kB
```

```
KReclaimable:   15271420 kB
Slab:               0 kB
SReclaimable:          0 kB
SUnreclaim:            0 kB
KernelStack:       40800 kB
PageTables:       100176 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:    49861052 kB
Committed_AS:   27142548 kB
VmallocTotal:   34359738367 kB
VmallocUsed:      200880 kB
VmallocChunk:          0 kB
Percpu:            80000 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
FileHugePages:         0 kB
FilePmdMapped:         0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:               0 kB
DirectMap4k:     1645568 kB
DirectMap2M:    27387904 kB
DirectMap1G:     6291456 kB
```

➜ the proc filesystem provides an interface to kernel data structures.

➜ *proc*meminfo reports statistics about memory usage on the system. It used by free to report the amount of free and used memory (both physical and swap). Look up proc(5) - Linux manual page on meminfo for details about each row.

➜ MemTotal: 2097152 kB ➜ 2 GiB

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1000
allocating 1048576000 bytes (1000.00 MB)
  number of integers in array: 262144000
loop 0 in 872.08 ms (bandwidth: 1146.69 MB/s)
loop 1 in 227.43 ms (bandwidth: 4396.92 MB/s)
loop 2 in 224.42 ms (bandwidth: 4455.85 MB/s)
loop 3 in 224.07 ms (bandwidth: 4462.97 MB/s)
loop 4 in 223.38 ms (bandwidth: 4476.76 MB/s)
loop 5 in 225.22 ms (bandwidth: 4440.15 MB/s)
loop 6 in 240.98 ms (bandwidth: 4149.70 MB/s)
loop 7 in 230.25 ms (bandwidth: 4343.05 MB/s)
loop 8 in 236.23 ms (bandwidth: 4233.14 MB/s)
loop 9 in 223.38 ms (bandwidth: 4476.78 MB/s)
loop 10 in 239.52 ms (bandwidth: 4175.07 MB/s)
loop 11 in 230.27 ms (bandwidth: 4342.80 MB/s)
loop 12 in 240.76 ms (bandwidth: 4153.44 MB/s)
loop 13 in 224.12 ms (bandwidth: 4461.85 MB/s)
loop 14 in 231.77 ms (bandwidth: 4314.55 MB/s)
loop 15 in 222.17 ms (bandwidth: 4501.09 MB/s)
loop 16 in 223.37 ms (bandwidth: 4476.96 MB/s)
loop 17 in 237.07 ms (bandwidth: 4218.23 MB/s)
loop 18 in 228.88 ms (bandwidth: 4369.01 MB/s)
loop 19 in 234.89 ms (bandwidth: 4257.27 MB/s)
loop 20 in 225.82 ms (bandwidth: 4428.27 MB/s)
```

➜ during running

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 6  0 202816 384712      0 175536    0    0     0    36 3373 2745 22  0 78  0  0
 1  0 202688  52296      0 175536    0    0     0   124 3022 2831 25  0 75  0  0
 6  0 202688  52196      0 175536    0    0     0   208 3258 3107 38  0 62  0  0
 4  0 202688  52276      0 175536    0    0   768     0 4388 5710 25  0 75  0  0
 3  0 202688  38004      0 175536    0    0     0   212 2461 3570 29  0 71  0  0
 8  0 202688  52264      0 175536    0    0     0   148 3417 2619 34  0 66  0  0
 3  0 202688  52352      0 175536    0    0     0   208 3906 4645 25  0 75  0  0
 5  0 202688  52212      0 175536    0    0     0   108 2791 2941 38  0 62  0  0
 3  0 202692  52352      0 175536    0    0     0    36 3691 2784 25  0 75  0  0
 3  0 202688  42788      0 175536    0    0     0    68 2744 3010 27  0 73  0  0
 6  0 202688  52348      0 175536    0    0     0   124 3338 2853 33  0 67  0  0
 2  0 202688  53580      0 175536    0  392   252   544 3614 3155 24  0 76  0  0
 3  0 202688  54228      0 175536    0  288    28   360 3583 3920 36  0 64  0  0
 6  0 202688  54936      0 175536    0   92  1084   540 4056 5109 25  0 75  0  0
 8  0 202688  44476      0 175536    0  132    12   164 3184 3553 27  0 73  0  0
 3  0 202688  55052      0 175536    0    0     0   124 2901 2620 34  0 66  0  0
 3  0 202688  55792      0 175536    0  136     8   368 3553 3370 24  0 76  0  0
 3  0 202688  55700      0 175536    0    8     0    80 3600 3111 35  0 65  0  0
 3  0 202688  55784      0 175536    0    0     0     0 2566 2892 25  0 75  0  0
 5  0 202688  51356      0 175536    0   40     0   108 3470 2968 25  0 75  0  0
 3  0 202688  55864      0 175536    0    0     0    76 3118 2818 36  0 64  0  0
 5  0 202688  55864      0 175536    0    0     0   196 2993 2929 24  0 76  0  0
```

➜ there are non zero values for the so column, which means that during the running of mem, some pages are swapped out.

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1200
allocating 1258291200 bytes (1200.00 MB)
  number of integers in array: 314572800
loop 0 in 1655.87 ms (bandwidth: 724.69 MB/s)
loop 1 in 287.37 ms (bandwidth: 4175.74 MB/s)
loop 2 in 309.49 ms (bandwidth: 3877.38 MB/s)
loop 3 in 271.58 ms (bandwidth: 4418.56 MB/s)
loop 4 in 267.40 ms (bandwidth: 4487.73 MB/s)
loop 5 in 279.47 ms (bandwidth: 4293.77 MB/s)
loop 6 in 268.60 ms (bandwidth: 4467.56 MB/s)
loop 7 in 279.64 ms (bandwidth: 4291.16 MB/s)
loop 8 in 290.90 ms (bandwidth: 4125.14 MB/s)
loop 9 in 341.44 ms (bandwidth: 3514.56 MB/s)
```

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 2  0 202816 850856      0 175536    0    0     0    80 3297 3439 12  0 88  0  0
 3  0 202696    968      0 172636    0    8   768   104 3663 5984 25  0 75  0  0
 5  0 202712  26656      0 167212    0   16     0   100 3808 2809 41  0 60  0  0
 2  0 202712  26656      0 167212    0    0     0   284 3044 3811 25  0 75  0  0
 8  0 202712  22856      0 167212    0    0     0   220 3062 3287 26  0 74  0  0
 3  0 202712  26660      0 167212    0    0     0   192 3695 2858 36  0 64  0  0
 1  0 202712  26644      0 167212    0    0     0    32 3040 3336 24  0 76  0  0
 6  0 202712   7048      0 167212    0    0     0   120 2978 2898 37  0 63  0  0
 3  0 202712  26644      0 167212    0    0     0   128 3278 2655 26  0 74  0  0
 4  0 202712  26644      0 167212    0    0     0   232 2361 2763 25  0 75  0  0
 9  0 202712  26644      0 167212    0    0     0   216 4187 4517 37  0 63  0  0
 1  0 202712  26644      0 167212    0    0   768    92 3604 5233 25  0 75  0  0
 4  0 202712  13888      0 167212    0    0     0   180 2814 2620 35  0 65  0  0
 3  0 202712  26644      0 167212    0    0     0    84 4126 3953 27  0 73  0  0
 3  0 202712  26644      0 167212    0    0     0    56 2395 2629 25  0 75  0  0
 5  0 202712  26632      0 167212    0    0     0   172 3446 3007 38  0 62  0  0
 2  0 202712  26632      0 167212    0    0     0    68 2991 2637 25  0 75  0  0
 7  0 202712  19364      0 167212    0    0   656   172 3510 3798 33  0 67  0  0
 6  0 202712  26428      0 167212    0    0     0   216 2966 2562 29  0 71  0  0
 5  0 202712  26528      0 167212    0    0     0    72 2889 2811 24  0 76  0  0
 4  0 202712  26528      0 167212    0    0     0    64 3897 4720 38  0 62  0  0
 7  0 202712  26544      0 167212    0    0   768   216 3767 4934 24  0 76  0  0
```

➜ less pages swapped out

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1400
allocating 1468006400 bytes (1400.00 MB)
  number of integers in array: 367001600
loop 0 in 1883.45 ms (bandwidth: 743.32 MB/s)
loop 1 in 389.89 ms (bandwidth: 3590.72 MB/s)
loop 2 in 2102.22 ms (bandwidth: 665.96 MB/s)
loop 3 in 323.79 ms (bandwidth: 4323.80 MB/s)
loop 4 in 327.28 ms (bandwidth: 4277.67 MB/s)
loop 5 in 320.74 ms (bandwidth: 4364.91 MB/s)
loop 6 in 327.88 ms (bandwidth: 4269.87 MB/s)
loop 7 in 311.50 ms (bandwidth: 4494.41 MB/s)
loop 8 in 327.51 ms (bandwidth: 4274.63 MB/s)
loop 9 in 320.57 ms (bandwidth: 4367.15 MB/s)
```

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 2  0 202712 1264128      0 167672    0    0     0    96 2329 3282  0  0 100  0  0
 5  0 202840 614632       0 167672    0    0     8   144 3496 2979 26  0 74  0  0
 3  0 202860    140       0 161968    0  144     0   224 3034 2582 23  0 77  0  0
 7  0 202820     36       0 126292  540  704   564  1732 3888 3839 30  0 70  0  0
 4  0 202980  39724       0 125844    0   96     0   304 3662 3341 34  0 66  0  0
 2  0 202980  39736       0 125844    0    0   768    68 3877 4649 25  0 75  0  0
 3  0 202980  39704       0 125844   32    0    32    32 3627 4250 38  0 62  0  0
10  0 202980  39732       0 125844    0    0     0   136 3155 2926 25  0 75  0  0
 4  0 202980  32492       0 125844    0    0     0   108 3250 3221 27  0 73  0  0
 3  0 202980  39960       0 125844    0    0     0   108 2613 2549 36  0 64  0  0
 4  0 202980  39960       0 125844    0    0     0    68 3892 3880 25  0 75  0  0
 2  0 202980  39808       0 125844    0    0     0   100 3417 3196 37  0 63  0  0
 5  0 202980  39984       0 125844    0    0     0    72 2766 2725 25  0 75  0  0
 6  0 202980  37076       0 125844    0    0     0   120 3638 3377 26  0 74  0  0
 1  0 202980  39968       0 125844    0    0     0   152 3488 3668 36  0 64  0  0
 6  0 202980  39968       0 125844    0    0   768   156 3806 5420 25  0 75  0  0
 7  0 202980  20740       0 125844    0    0     0    68 3810 3807 37  0 63  0  0
 2  0 202980  39812       0 125844    0    0     0     0 2378 2708 26  0 74  0  0
10  0 202980  39912       0 125844    0    0     0   164 3354 3102 25  0 75  0  0
 1  0 202980  39912       0 125844    0    0     0   152 3422 3203 38  0 62  0  0
 2  0 202980  44624       0 126372    0    0   760   176 2753 3548 26  0 74  0  0
 6  0 202980  30540       0 126372    0    0     0   108 3865 3060 34  0 66  0  0
```

➜ now pages swapped in and swapped out

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1800
allocating 1887436800 bytes (1800.00 MB)
  number of integers in array: 471859200
loop 0 in 4042.16 ms (bandwidth: 445.31 MB/s)
loop 1 in 1736.27 ms (bandwidth: 1036.70 MB/s)
loop 2 in 426.00 ms (bandwidth: 4225.37 MB/s)
loop 3 in 424.73 ms (bandwidth: 4238.00 MB/s)
loop 4 in 461.62 ms (bandwidth: 3899.29 MB/s)
loop 5 in 469.11 ms (bandwidth: 3837.02 MB/s)
loop 6 in 503.88 ms (bandwidth: 3572.26 MB/s)
loop 7 in 431.19 ms (bandwidth: 4174.54 MB/s)
loop 8 in 411.47 ms (bandwidth: 4374.52 MB/s)
loop 9 in 426.37 ms (bandwidth: 4221.72 MB/s)
loop 10 in 403.01 ms (bandwidth: 4466.37 MB/s)
```

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 3  0 202980 1480996      0 126372    0    0   768   144 3181 5485  2  0 98  0  0
 7  0 203088 329308       0 126372   32    0    32   156 3301 3433 25  0 75  0  0
 2  0 203980     20       0 111312   32 1052   288  1180 3674 3009 45  0 55  0  0
 3  0 246948     44       0  34648   36 42896  5808 42964 5173 4439 32  0 68  0  0
 5  0 343300     84       0  30088    0 86308  8928 86428 7744 3632 28  0 72  0  0
 0  2 357680     48       0  33256 52904 48108 56988 48148 7878 10169 18  0 82  0  0
 8  1 376420  21436       0  36752 18068 28564 25884 28680 5663 5815 40  0 60  0  0
 2  0 375872  20880       0  37148  432    0   808   100 3434 3119 25  0 75  0  0
 4  0 375200  16080       0  38864  976    0  2836    36 2529 2589 25  0 75  0  0
 7  0 378540  22552       0  37600   76 2160   320  2352 4209 4282 39  0 61  0  0
 4  0 378532  22548       0  37600   24    0    24     0 2996 2724 26  0 74  0  0
 6  1 378480  11988       0  37600 1112  612  1904   692 3989 5473 35  0 65  0  0
 2  0 378588  22404       0  37600  732  532   732   764 3503 3440 29  0 71  0  0
 3  0 378264  22044       0  37732  436    0   436     0 3422 3863 25  0 75  0  0
 5  0 378952  22724       0  37732  296  604   300   768 3212 3047 38  0 62  0  0
 3  0 378952  22624       0  37732   60    0    60     0 3844 3524 24  0 76  0  0
 5  0 378896  19824       0  37732  116  144   116   288 2832 2448 31  0 69  0  0
 6  0 378888  22644       0  37732   48    0    48   104 3220 2901 31  0 69  0  0
 4  0 378820  22044       0  38128   84    0   528   312 2830 2449 25  0 75  0  0
 4  0 378932  22344       0  37996   60   92    60   232 3430 3863 38  0 62  0  0
 4  0 378748  22460       0  37996   24    0    24   100 3324 3019 25  0 75  0  0
 5  0 378624   2816       0  37996  272    0  1040    80 4160 5939 31  0 69  0  0
```

➜ a lot more pages swapped in and out

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 2048
allocating 2147483648 bytes (2048.00 MB)
  number of integers in array: 536870912
loop 0 in 4129.20 ms (bandwidth: 495.98 MB/s)
loop 1 in 31518.43 ms (bandwidth: 64.98 MB/s)
```

```
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 4  0 365668 1430436      0  45784  156    0   156    76 2242 2581 10  0 90  0  0
 5  0 365532 280288       0  45784    8    0     8   132 4435 4325 37  0 63  0  0
 5  1 407852    164       0  31008    0 41616  1088 41876 4545 3372 24  0 76  0  0
10  0 518720     32       0  35240   32 110752 12860 110864 12795 6033 34  0 66  0  0
 2  0 608668     64       0  34172 15824 102912 16992 103036 11727 4320 37  0 63  0  0
 2  0 608000     16       0  33512 59312 57880 59312 57952 4851 6594 19  0 81  0  0
 6  0 628420     92       0  33248 58036 74252 58200 74416 10216 6913 28  0 72  0  0
 1  0 607632      0       0  33248 86628 62344 86636 62472 10294 9167 27  0 73  0  0
 9  0 607808      0       0  33248 63664 63024 63804 63136 5990 6694 19  0 81  0  0
 4  0 621828   9052       0  32984 68048 80788 68428 80916 12360 7664 32  0 68  0  0
 3  0 608000    104       0  33380 75516 58980 76356 58984 9341 7636 25  0 75  0  0
 8  1 607632     52       0  32984 64808 61660 64808 61832 6512 8274 19  0 81  0  0
 6  0 620308     20       0  32720 71208 81484 71876 81592 12911 9531 34  0 66  0  0
 1  1 607528    124       0  32720 74204 58840 74588 58884 8925 9107 23  0 77  0  0
 5  0 607160     12       0  32588 65472 62804 65472 62972 5843 6988 20  0 80  0  0
 6  0 625660    148       0  32456 67836 83956 68068 83988 13688 7565 36  0 64  0  0
 2  0 607724    100       0  32456 77480 57032 77480 57124 6934 7984 20  0 80  0  0
 3  0 607448     20       0  32456 64816 63556 64816 63808 6401 8020 20  0 80  0  0
 2  0 623036  15536       0  32456 67036 82244 67300 82272 14435 7209 38  0 62  0  0
 3  0 607736    100       0  32456 72780 57680 72780 57840 6107 7590 20  0 80  0  0
 7  0 609856     32       0  32456 64172 66732 64256 66764 6402 6695 21  0 79  0  0
 3  0 607492    200       0  32588 74028 70812 74156 70908 13449 8922 34  0 66  0  0
```

➜ big increase in swap ins and swap outs

➜ on the first loop, the row at si column gets pretty big and in the next loops this value decreases dramatically. I got this with ./mem 1600
To get this behaviour you must give the system some idle time and then run it. You will get a spike for swap in on the first loop and the next loops go in direction 0. The swap in column is in KB, so yeah it does make sense.

➜ Nonzero **si** and **so** numbers indicate that there is not enough physical memory and that the kernel is swapping memory to disk.

4. Do the same experiments as above, but now watch the other statistics (such as CPU utilization, and block I/O statistics). How do they change when mem is running?
   ➜ for cpu utilisation see first exercise
   ➜ the bi, blocks received from a block device, increases when the allocated memory by mem increases. There will be more rows with non zero values. Theoretically when so increases, also bo should increase and the same for si and bi. But remember bi and bo show blocks received or sent to a disk device. So although memory is swapped, it may now show in block statistics, because I think the memory gets clustered first and then sent.

5. Now let's examine performance. Pick an input for mem that comfortably fits in memory (say 4000 if the amount of memory on the system is 8 GB). How long does loop 0 take (and subsequent loops 1, 2, etc.)? Now pick a size comfortably beyond the size of memory (say 12000 again assuming 8 GB of memory). How long do the loops take here? How do the bandwidth numbers compare? How different is performance when constantly swapping versus fitting everything comfortably in memory? Can you make a graph, with the size of memory used by mem on the x-axis, and the bandwidth of accessing said memory on the y-axis? Finally, how does the performance of the first loop compare to that of subsequent loops, for both the case where everything fits in memory and where it doesn't?

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1000
allocating 1048576000 bytes (1000.00 MB)
  number of integers in array: 262144000
loop 0 in 877.15 ms (bandwidth: 1140.05 MB/s)
loop 1 in 233.81 ms (bandwidth: 4276.90 MB/s)
loop 2 in 225.29 ms (bandwidth: 4438.76 MB/s)
loop 3 in 227.07 ms (bandwidth: 4403.89 MB/s)
loop 4 in 227.98 ms (bandwidth: 4386.35 MB/s)
loop 5 in 236.02 ms (bandwidth: 4236.89 MB/s)
loop 6 in 222.92 ms (bandwidth: 4485.81 MB/s)
loop 7 in 224.15 ms (bandwidth: 4461.22 MB/s)
loop 8 in 232.75 ms (bandwidth: 4296.42 MB/s)
loop 9 in 224.73 ms (bandwidth: 4449.82 MB/s)
loop 10 in 234.75 ms (bandwidth: 4259.81 MB/s)
```

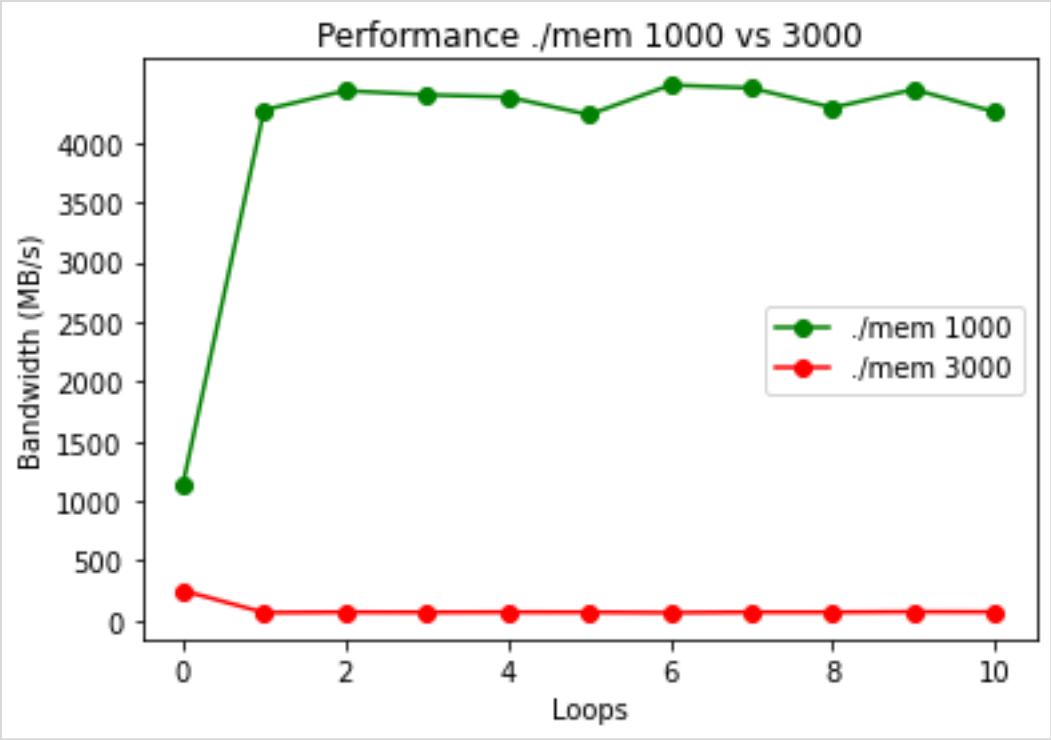
```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 3000
allocating 3145728000 bytes (3000.00 MB)
  number of integers in array: 786432000
loop 0 in 12063.20 ms (bandwidth: 248.69 MB/s)
loop 1 in 46904.24 ms (bandwidth: 63.96 MB/s)
loop 2 in 44353.43 ms (bandwidth: 67.64 MB/s)
loop 3 in 45090.48 ms (bandwidth: 66.53 MB/s)
loop 4 in 44183.05 ms (bandwidth: 67.90 MB/s)
loop 5 in 44751.04 ms (bandwidth: 67.04 MB/s)
loop 6 in 47400.83 ms (bandwidth: 63.29 MB/s)
loop 7 in 44539.24 ms (bandwidth: 67.36 MB/s)
loop 8 in 44624.18 ms (bandwidth: 67.23 MB/s)
loop 9 in 42557.00 ms (bandwidth: 70.49 MB/s)
loop 10 in 43569.75 ms (bandwidth: 68.86 MB/s)
```

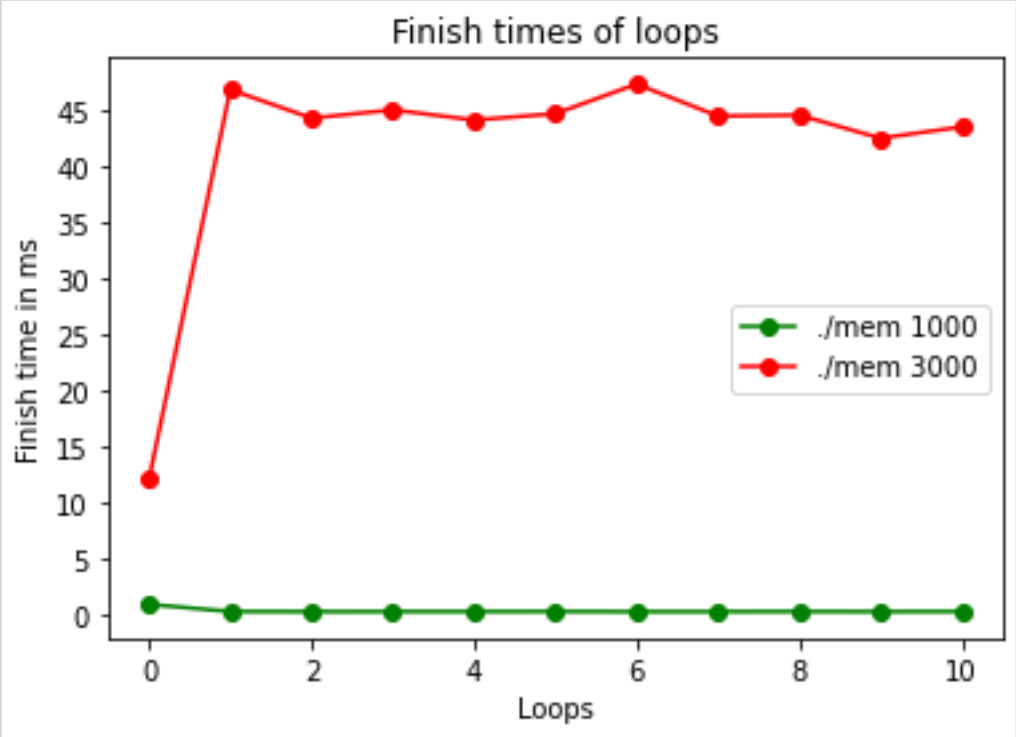
```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 500
allocating 524288000 bytes (500.00 MB)
  number of integers in array: 131072000
loop 0 in 438.68 ms (bandwidth: 1139.79 MB/s)
loop 2 in 115.58 ms (bandwidth: 4325.97 MB/s)
loop 4 in 113.97 ms (bandwidth: 4387.24 MB/s)
loop 6 in 112.47 ms (bandwidth: 4445.82 MB/s)
loop 8 in 113.04 ms (bandwidth: 4423.25 MB/s)
loop 10 in 113.90 ms (bandwidth: 4389.96 MB/s)
loop 12 in 113.91 ms (bandwidth: 4389.40 MB/s)
```

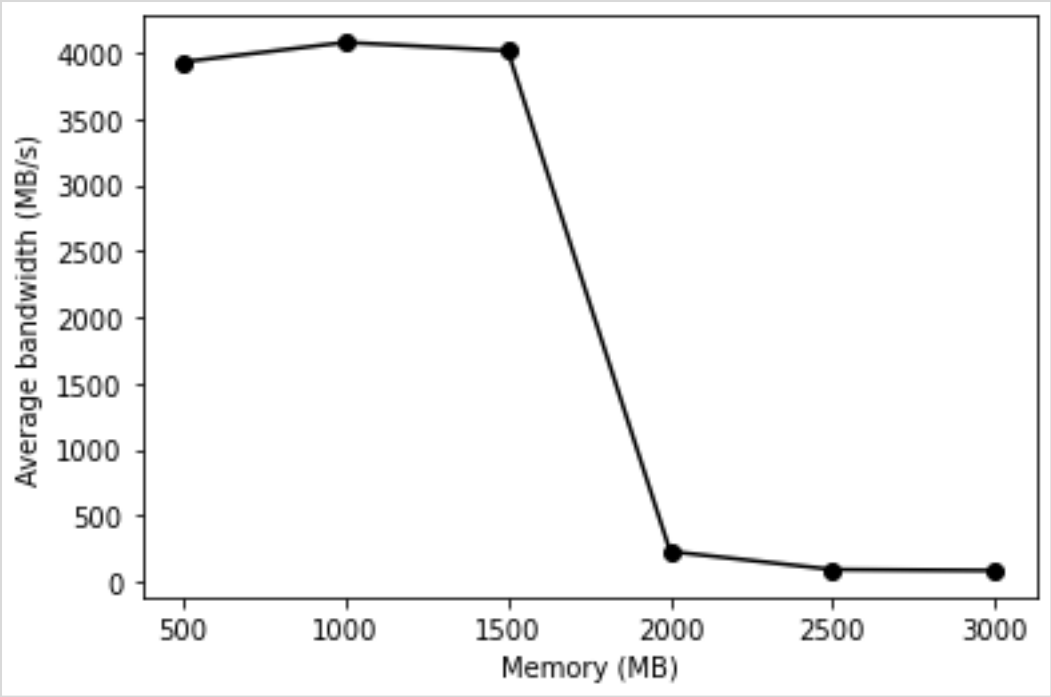
```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 1500
allocating 1572864000 bytes (1500.00 MB)
  number of integers in array: 393216000
loop 0 in 1318.61 ms (bandwidth: 1137.56 MB/s)
loop 1 in 340.31 ms (bandwidth: 4407.75 MB/s)
loop 2 in 336.83 ms (bandwidth: 4453.30 MB/s)
loop 3 in 348.67 ms (bandwidth: 4302.11 MB/s)
loop 4 in 350.41 ms (bandwidth: 4280.73 MB/s)
loop 5 in 348.38 ms (bandwidth: 4305.58 MB/s)
loop 6 in 355.87 ms (bandwidth: 4215.02 MB/s)
loop 7 in 345.05 ms (bandwidth: 4347.13 MB/s)
loop 8 in 360.21 ms (bandwidth: 4164.24 MB/s)
loop 9 in 362.42 ms (bandwidth: 4138.86 MB/s)
loop 10 in 341.30 ms (bandwidth: 4394.91 MB/s)
loop 11 in 352.15 ms (bandwidth: 4259.60 MB/s)
loop 12 in 389.51 ms (bandwidth: 3850.96 MB/s)
```

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 2000
allocating 2097152000 bytes (2000.00 MB)
  number of integers in array: 524288000
loop 0 in 3691.32 ms (bandwidth: 541.81 MB/s)
loop 1 in 14479.20 ms (bandwidth: 138.13 MB/s)
loop 2 in 18891.40 ms (bandwidth: 105.87 MB/s)
loop 3 in 13580.93 ms (bandwidth: 147.27 MB/s)
loop 4 in 10366.94 ms (bandwidth: 192.92 MB/s)
loop 5 in 18773.23 ms (bandwidth: 106.53 MB/s)
loop 6 in 4892.12 ms (bandwidth: 408.82 MB/s)
loop 7 in 17257.76 ms (bandwidth: 115.89 MB/s)
loop 8 in 4982.48 ms (bandwidth: 401.41 MB/s)
loop 9 in 22270.14 ms (bandwidth: 89.81 MB/s)
loop 10 in 7788.53 ms (bandwidth: 256.79 MB/s)
```

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ ./mem 2500
allocating 2621440000 bytes (2500.00 MB)
  number of integers in array: 655360000
loop 0 in 7854.91 ms (bandwidth: 318.27 MB/s)
loop 1 in 36969.00 ms (bandwidth: 67.62 MB/s)
loop 2 in 37522.77 ms (bandwidth: 66.63 MB/s)
loop 3 in 36812.00 ms (bandwidth: 67.91 MB/s)
loop 4 in 37116.18 ms (bandwidth: 67.36 MB/s)
loop 5 in 37581.88 ms (bandwidth: 66.52 MB/s)
loop 6 in 29837.11 ms (bandwidth: 83.79 MB/s)
loop 7 in 37401.14 ms (bandwidth: 66.84 MB/s)
loop 8 in 38946.78 ms (bandwidth: 64.19 MB/s)
loop 9 in 37780.56 ms (bandwidth: 66.17 MB/s)
loop 10 in 38297.47 ms (bandwidth: 65.28 MB/s)
```


Performance ./mem 1000 vs 3000
Bandwidth (MB/s)
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
Loops
./mem 1000
./mem 3000


Finish times of loops
Finish time in ms
45
40
35
30
25
20
15
10
5
0
0
2
4
6
8
10
Loops
./mem 1000
./mem 3000



6. Swap space isn't infinite. You can use the tool swapon with the -s flag to see how much swap space is available. What happens if you try to run mem with increasingly large values, beyond what seems to be available in swap? At what point does the memory allocation fail?

➜ swapon doesn't work on the container so I used free:

```
vl161bra@ct-bsys-ws20-12:~/htwg/S3/BS/Homeworks/HW21-Paging-BeyondPhys-Real$ free
              total        used        free      shared  buff/cache   available
Mem:        2097152      174440     1865132        3552       57580     1922712
Swap:       3145728      398808     2746920
```

➜ there is about 3GiB swap space available. Free shows in kb.

➜ We have about 2GiB RAM and 3GiB swap space so 5GiB in total

➜ Between 4550 and 4600. Thats roughly 90% of the total 5 GiB

```
vl161bra@ct-bsys-ws20-12:~$ /sbin/swapon -s
Filename                                Type            Size    Used    Priority
none                                    virtual         3145728 2857388 0
```

7. Finally, if you're advanced, you can configure your system to use different swap devices using swapon and swapoff. Read the man pages for details. If you have access to different hardware, see how the performance of swapping changes when swapping to a classic hard drive, a flash-based SSD, and even a RAID array. How much can swapping performance be improved via newer devices? How close can you get to in-memory performance?

➜ **swapon**is used to specify devices on which paging and swapping are to take place.

➜ **swapoff**disables swapping on the specified devices and files. When the **-a**flag is given, swapping is disabled on all known swap devices

and files (as found in proc*swaps* or etc*fstab*).

➜ Option -d enables swap discards if the swap backing device supports the trim operation. Trim operation allows an OS to inform a SSD which blocks of data are no longer considered in use can be be wiped internally. This may improve performance on some SSDs.

➜ Swap on SSD is much faster than HDD, but still pretty bad compared to RAM. Thrashing on SSD is still a problem. And you will not see the difference, unless you are severely starving for free memory and constantly swapping.
➜ However, the page file is often disabled on SSDs to limit the number of writes and thus increase lifespan a little bit.
➜ When your system starts to swap, you are already in trouble. You don't want to get there, ever.

➜ **RAID** is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both.
➜ Swapping on mirrored RAID (RAID 1) can help survive a failing disk. If a disk fails, then data for swapped processes would be inaccessible in a non-mirrored environment. If you run in a mirrored environment, then the system can go on running even if a disk fails in service. There's not much reason to use RAID0 (splits data evenly across two or more disks) for swap performance reasons.

➜ Übrigens: Ganz voll wird der Swap nicht geschrieben ... warum wohl ... ?
➜ I think it is because swap space is meant to be a buffer for RAM, and if memory is full and swap also, then the system gets into some kind of deadlock.