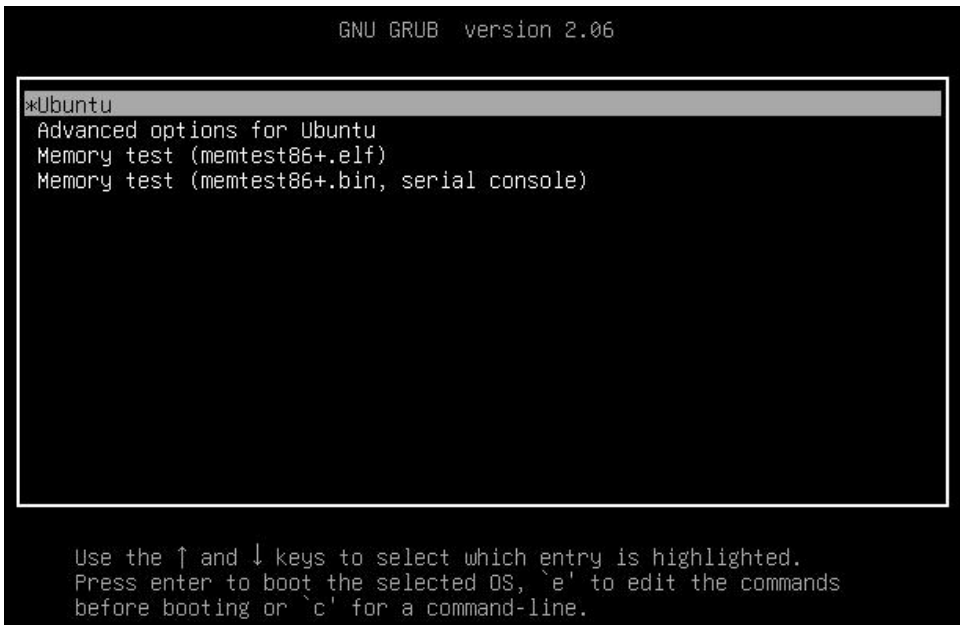
1.Reboot your system and pause at the GRUB menu. Enter your GRUB password to access the maintenance functions.

首先重启 ubuntu 系统：



2.Select the modify the kernel arguments option, and add the following to the end of the line: mem=256M

利用按键 e 进入编辑模式，在内核参数部分的行尾添加该指令：

系统报错，由于本次在 VMware 虚拟机进行实验，不能支持 256M 内存，故本次实验后续继续以 2G 内存作为替代。

3.Start top and identify the following pieces of information:

首先使用 top 指令进入 top 工具：

```
tux1@tux1-virtual-machine:~$ top
```

-The time, the up time, the number of users, and the load information on the first line

```
top - 23:25:41 up 1 min,  1 user,  load average: 0.86, 0.39, 0.14
```

-The number of processes on the second line

```
任务: 342 total,   1 running, 341 sleeping,   0 stopped,   0 zombie
```

目前有 342 个进程，1 个正在运行，341 个正在休眠，无终止进程与僵尸进程；

-The CPU breakdown on the third line

```
%Cpu(s):  0.1 us,  0.1 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
```

目前用户空间占用 CPU 百分比为 0.1，内核空间占用 CPU 百分比为 0.1，用户进程空间内改变过优先级的进程占用 CPU 百分比为 0.0，空闲 CPU 百分比为 99.8，等待输入输出的 CPU 时间百分比为 0.0，硬中断占用 CPU 时间百分比为 0.0，软中断占用 CPU 时间百分比为 0.0，窃取时间百分比为 0.0；

-The real memory breakdown on the fourth line

```
MiB Mem :   1940.6 total,     75.1 free,   1465.4 used,    400.1 buff/cache
```

物理内存总量为 1940.6MB，空闲内存量为 75.1MB，使用的物理内存量为 1465.4MB，用作内核缓存的内存量为 400.1MB；

-The swap space breakdown on the fifth line

```
MiB Swap:   2140.0 total,   2051.6 free,     88.4 used.    305.4 avail Mem
```

交换空间总量为 2140.0 MB，空闲交换区总量为 2051.6MB，使用的交换区总量为 88.4MB，缓冲的交换区总量为 305.4MB。

-The processes (sorted by CPU-time) on the next lines

目前运行的进程：

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1099 tux1      20   0 3759496 184520  63356 S   0.3   9.3   0:13.00 gnome-+
    1 root      20   0  167920  10432   6772 S   0.0   0.5   0:00.92 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthrea+
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_pa+
    5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_f+
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworke+
   10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_per+
   11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_ta+
   12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_ta+
   13 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_ta+
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.05 ksofti+
   15 root      20   0       0      0      0 I   0.0   0.0   0:00.71 rcu_pr+
   16 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migrat+
   17 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_i+
   19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
```

Verify that Linux detected only 128 MB.

4.top automatically refreshes itself after 10 seconds. To increase this to one second, enter the command s1.

在 top 工具运行时按 s 进入交互模式，输入 1 使 top 工具 1 秒后自动刷新：

```
Change delay from 3.0 to 1
```

5.top is by default does not show the amount of swap space used by each processes. To show this amount too, call up the Field Order screen with the f command, and enable the swap space display.

在 top 工具运行时按 f 调用 Field Order 屏幕，并在光标指向“SWAP”时按 d 启用交换空间显示：

```
Fields Management for window 1:Def, whose current sort field is %CPU
   Navigate with Up/Dn, Right selects for move then <Enter> or Left commits,
    'd' or <Space> toggles display, 's' sets sort.  Use 'q' or <Esc> to end!

* PID     = Process Id      TIME    = CPU Time        nsUTS   = UTS namespace
* USER    = Effective Use   SWAP    = Swapped Size    LXC     = LXC container
* PR      = Priority        CODE    = Code Size (Ki   RSan    = RES Anonymous
* NI      = Nice Value      DATA    = Data+Stack (K   RSfd    = RES File-base
* VIRT    = Virtual Image   nMaj    = Major Page Fa   RSlk    = RES Locked (K
* RES     = Resident Size   nMin    = Minor Page Fa   RSsh    = RES Shared (K
* SHR     = Shared Memory   nDRT    = Dirty Pages C   CGNAME  = Control Group
* S       = Process Statu   WCHAN   = Sleeping in F   NU      = Last Used NUM
* %CPU    = CPU Usage       Flags   = Task Flags <s
* TIME+   = CPU Time, hun   CGROUPS = Control Group
* COMMAND = Command Name/   SUPGIDS = Supp Groups I
  PPID    = Parent Proces   SUPGRPS = Supp Groups N
  UID     = Effective Use   TGID    = Thread Group
  RUID    = Real User Id  * %MEM    = Memory Usage
  RUSER   = Real User Nam   OOMa    = OOMEM Adjustm
  SUID    = Saved User Id   OOMs    = OOMEM Score c
  SUSER   = Saved User Na   ENVIRON = Environment v
  GID     = Group Id        vMj     = Major Faults
  GROUP   = Group Name      vMn     = Minor Faults
  PGRP    = Process Group   USED    = Res+Swap Size
  TTY     = Controlling T   nsIPC   = IPC namespace
  TPGID   = Tty Process G   nsMNT   = MNT namespace
  SID     = Session Id      nsNET   = NET namespace
  nTH     = Number of Thr   nsPID   = PID namespace
  P       = Last Used Cpu   nsUSER  = USER namespac
```

6.To sort processes in a different order use the M, P, or T command.

当使用 M 命令时将根据驻留内存大小对进程进行排序：

```
  PR  NI    VIRT    RES    SHR    %CPU  %MEM     TIME+ COMMAND
  20   0 4342992 268224 129520 S   0.0   6.7   0:11.82 gnome-shell
  20   0  226220  78476  54592 S   0.0   2.0   0:00.26 Xwayland
  20   0  549348  76728  57884 S   0.0   1.9   0:00.29 gsd-xsettings
  20   0 2906736  67500  51072 S   0.0   1.7   0:00.25 gjs
  20   0  926976  63340  47284 S   0.0   1.6   0:00.31 evolution-alarm
  20   0  646904  61280  45816 S   0.3   1.5   0:02.20 gnome-terminal-
  20   0 1392504  38848   8676 S   0.0   1.0   0:16.44 snapd
  20   0  299456  37128  26844 S   0.0   0.9   0:00.86 vmtoolsd
  20   0  573456  35208  28088 S   0.0   0.9   0:00.18 goa-daemon
  20   0  542636  33004  23732 S   0.0   0.8   0:00.13 update-notifier
  20   0  681580  29884  24128 S   0.0   0.8   0:00.19 evolution-addre
  20   0  849496  27804  23948 S   0.0   0.7   0:00.12 evolution-calen
  20   0  357144  26712  15828 S   0.0   0.7   0:01.05 ibus-extension-
  20   0 2612532  26648  21240 S   0.0   0.7   0:00.08 gjs
  20   0  671804  26068  17724 S   0.0   0.7   0:00.21 xdg-desktop-por
  39  19  644344  25892  17468 S   0.0   0.7   0:00.83 tracker-miner-f
  20   0 2678064  25832  21008 S   0.0   0.6   0:00.07 gjs
  20   0  392572  25468  17852 S   0.0   0.6   0:00.06 gnome-terminal.
   9 -11 1169092  24724  19216 S   0.0   0.6   0:01.74 pulseaudio
  20   0  652392  24520  18936 S   0.0   0.6   0:00.13 gsd-media-keys
```

当使用 P 命令时，将根据 CPU 使用百分比大小对进程进行排序：

```
  PR  NI    VIRT    RES    SHR    %CPU  %MEM     TIME+ COMMAND
  20   0 1392504  38848   8676 S   1.7   1.0   0:16.42 snapd
  20   0 4342992 268216 129520 S   0.3   6.7   0:11.72 gnome-shell
  20   0   22024   4184   3220 R   0.3   0.1   0:01.31 top
  20   0  168124  13272   8132 S   0.0   0.3   0:03.01 systemd
  20   0       0      0      0 S   0.0   0.0   0:00.02 kthreadd
   0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
   0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
   0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
   0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
   0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_kthread
  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
  20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
  20   0       0      0      0 S   0.0   0.0   0:00.06 ksoftirqd/0
  20   0       0      0      0 I   0.0   0.0   0:00.22 rcu_preempt
  rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
 -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
  20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
  20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
```

当使用 T 命令时，将根据时间/累计时间对进程进行排序：

```
  PR  NI    VIRT    RES    SHR    %CPU  %MEM     TIME+ COMMAND
  20   0 1392504  38848   8676 S   0.0   1.0   0:16.44 snapd
  20   0 4342992 268164 129520 S   0.0   6.7   0:11.89 gnome-shell
  20   0  168124  13272   8132 S   0.0   0.3   0:03.01 systemd
  20   0  646904  61280  45816 S   0.0   1.5   0:02.24 gnome-terminal-
   9 -11 1169092  24724  19216 S   0.0   0.6   0:01.74 pulseaudio
  20   0   22024   4184   3220 R   0.3   0.1   0:01.40 top
  20   0  357144  26712  15828 S   0.0   0.7   0:01.05 ibus-extension-
  20   0  299292  37100  26844 S   0.0   0.9   0:00.90 vmtoolsd
  39  19  644344  25892  17468 S   0.0   0.7   0:00.83 tracker-miner-f
  20   0   19192  12152   8192 S   0.0   0.3   0:00.82 systemd
  20   0       0      0      0 S   0.0   0.0   0:00.81 kswapd0
  20   0  252552   8248   6936 S   0.0   0.2   0:00.79 vmtoolsd
  20   0   14828   6180   5384 S   0.0   0.2   0:00.67 systemd-oomd
  20   0   26972   6936   4356 S   0.0   0.2   0:00.55 systemd-udevd
  20   0   10884   6056   3804 S   0.0   0.2   0:00.55 dbus-daemon
  rt   0       0      0      0 S   0.0   0.0   0:00.53 migration/2
  rt   0       0      0      0 S   0.0   0.0   0:00.52 migration/1
  rt   0       0      0      0 S   0.0   0.0   0:00.52 migration/3
  20   0    9972   5780   3792 S   0.0   0.1   0:00.46 dbus-daemon
```

7.Leave top running in a separate window.

开启另一个终端窗口，在执行其他命令时使 top 工具运行在单独的窗口中：

Checking swap space

8.There should be one active swap space (check the fifth line of top). Deactivate this swap space. (If you do not know which partition is used as swapspace, check /proc/swaps.)

首先使用 cat 命令检查/proc/swaps 中的内容：

```
tux1@tux1-virtual-machine:~$ cat /proc/swaps
Filename                                Type            Size            Used  P
riority
/swapfile                               file            2191356         91548 -
2
```

如上图所示，交换分区在文件/swapfile 中，接下来执行 swapoff 命令停用此交换空间。

9.Go back to your top window and check what happened.

返回 top 工具窗口发现：交换空间总量、空闲交换区总量和使用的交换区总量均变为 0MB。

10.Reactivate the swap space.

执行 swapon 命令可以重新激活交换空间：

```
tux1@tux1-virtual-machine:~$ sudo swapon /swapfile
```

再次返回 top 工具窗口可以发现：

```
%Cpu(s):  0.0 us,  0.5 sy,  0.0 ni, 99.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1940.6 total,     73.4 free,   1573.8 used,    293.4 buff/cache
MiB Swap:   2140.0 total,   2140.0 free,      0.0 used.    180.7 avail Mem
```

交换空间总量和空闲交换区总量重新变为 2140MB，不再为 0MB，交换空间成功重新激活。

Using the cache

11.Caching is done automatically. If you have performed the exercises so you're your cache usage should be about 20 MB. Let's see what happens if we really start accessing the disk. Start a very disk-intensive program.

使用 dd 命令创建一个大量占用磁盘的程序，此处在/tmp 目录下创建一个 2G 的文件：

```
tux1@tux1-virtual-machine:/var$ dd if=/dev/zero of=/tmp/test bs=1M count=2048
2048+0 records in
2048+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 2.20752 s, 973 MB/s
```

12.Watch what happens to the amount of cached data in top. This number should increase until your cache occupies almost all of your real memory. Only about 4 MB will be left over for application use. However, also note that virtually no processes will be swapped out to disk.

在运行 dd 命令时，观察 top 工具输出，此时 top 工具中内存显示结果：

```
MiB Mem :   1940.6 total,     62.3 free,   1556.3 used,    322.0 buff/cache
```

缓存大小出现增加（由 164.2MB 变为 322.0MB），空闲内存大小减小（由 186.8MB 变为 62.3MB）。

Using the swap space

The swap space is only used if there is a real shortage in memory. First the amount of cached data decreases to about zero before processes are swapped to disk. Therefore, you need to have a program which uses a lot of real memory. The easiest solution is to write that ourselves.

13.Create the directory /root/bin and cd into it.

执行 cd 命令进入/root/bin 目录：

```
tux1@tux1-virtual-machine:~$ sudo su
root@tux1-virtual-machine:/home/tux1# cd /root/bin
root@tux1-virtual-machine:~/bin#
```

当前目录切换至/root/bin，说明该目录成功创建并成功进入。

14.Create a program called usemem, with contents:

```
!/usr/bin/perl
print "Allocating $ARGV[0] megabytes of memory...\n";
$big = "";
for( $i=0; $i<$ARGV[0]; $i++ )
{
$big .= "1234567890"x104858;
}
print "Press ENTER to release memory...\n";
```

根据要求创建 usemem 程序：

```
#!/usr/bin/perl
print "Allocating $ARGV[0] megabytes of memory...\n";
$big = "";
for($i=0;$i<$ARGV[0];$i++)
{
$big .="1234567890"x104858;
}
print "Press ENTER to release memory...\n";
<STDIN>;
undef $big;
print "Memory released.\n";
```

启动 usemen 程序时，将逐渐分配在命令行中指定的内存量，然后进行等待。当按 Enter 键时，程序退出，并释放其内存。

When you start this program, it will gradually allocate the amount of memory you specify on the command line and then wait for you to press Enter. When you press Enter, the program exits and thus will deallocate its memory.

Note: Your instructor might have this program available already.

15.Do a chmod 755 of this program and start it with parameter 1. This will ensure that the program allocates about 1 MB of memory.

对 usemem 程序执行 chmod 755 命令，赋予其 755 权限：

```
root@tux1-virtual-machine:~/bin# chmod 755 usemem
```

利用 perl 命令调用 perl 解释器执行 usemen 程序：

```
root@tux1-virtual-machine:~/bin# perl usemem 1
Allocating 1 megabytes of memory...
Press ENTER to release memory...
```

程序执行成功。

16.Now, go back to your top screen and select the sort by memory usage mode by issuing the M command. Note how much memory is used by the usemem program.

返回 top 工具，利用 M 命令使进程按内存使用大小进行排序，观察 usemem 进程的相关情况：

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1091	tux1	20	0	3848184	167012	52512	S	0.0	8.4	0:21.71	gnome-+
2972	tux1	20	0	3025352	57392	42368	S	0.0	2.9	0:00.22	gjs
2690	tux1	20	0	571340	52832	40104	S	1.0	2.7	0:02.92	gnome-+
1574	tux1	20	0	1059836	22808	5252	S	0.0	1.1	0:03.90	snap-s+
1663	tux1	20	0	2866184	15572	12816	S	0.0	0.8	0:00.07	gjs
794	root	20	0	949184	14620	6464	S	0.0	0.7	0:08.30	snapd
361	root	19	-1	49188	12792	12180	S	0.0	0.6	0:00.58	system+
2385	tux1	20	0	502032	12704	7704	S	0.0	0.6	0:00.24	update+
951	tux1	20	0	16352	9392	1896	S	0.0	0.5	0:00.59	dbus-d+
944	tux1	9	-11	139984	8012	5552	S	0.0	0.4	0:01.50	pipewi+
1	root	20	0	169512	7908	4912	S	0.0	0.4	0:02.23	systemd
3034	root	20	0	24036	7212	4952	S	0.0	0.4	0:00.00	perl

17.Stop usemem and start it with about half the amount of memory which is in your system. Thus, if you have 256 MB of real memory or decreased the amount to 256 MB, use 99 as parameter. Watch what happens in top. First you should see the amount of cached data decrease, and maybe see some processes already being swapped out.

停止 usemem 程序：

```
root@tux1-virtual-machine:~/bin# perl usemem 1
Allocating 1 megabytes of memory...
Press ENTER to release memory...

Memory released.
```

此时 top 工具中的结果如下图所示：

```
%Cpu(s):  0.2 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1940.6 total,    110.4 free,   1532.1 used,    298.1 buff/cache
MiB Swap:   2140.0 total,   1750.5 free,    389.5 used.    244.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1130 tux1      20   0 3830316 158232  44404 S   0.0   8.0   0:32.19 gnome-+
   1629 tux1      20   0 1190308  91632   4652 S   0.0   4.6   0:06.43 snap-s+
   4842 tux1      20   0 2951620  56276  41232 S   0.0   2.8   0:00.32 gjs
   4903 tux1      20   0  571096  52588  40008 S   0.0   2.6   0:00.49 gnome-+
   1904 root      20   0  442744  27140   3600 S   0.0   1.4   0:01.07 fwupd
   1398 root      20   0  382040  15360   4912 S   0.0   0.8   0:05.61 packag+
    362 root      19  -1  114732  13224  12500 S   0.0   0.7   0:02.79 system+
   1709 tux1      39  19  637648  12984   5628 S   0.0   0.7   0:01.42 tracke+
   2610 tux1      20   0  502040  11568   8224 S   0.0   0.6   0:00.36 update+
   1421 tux1      20   0 2866244  11412   9068 S   0.0   0.6   0:00.14 gjs
    836 root      20   0  875452  11220   2992 S   0.0   0.6   0:02.76 snapd
   1718 tux1      20   0 2866244   9032   6596 S   0.0   0.5   0:00.16 gjs
    886 root      20   0  268944   8108   6868 S   0.0   0.4   0:00.51 Networ+
      1 root      20   0  169356   7708   4696 S   0.0   0.4   0:01.45 systemd
```

执行命令 perl usemem，此时 top 中的结果如下图所示：

```
%Cpu(s):  0.2 us,  0.5 sy,  0.0 ni, 99.2 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   1940.6 total,     78.7 free,   1759.2 used,    102.8 buff/cache
MiB Swap:   2140.0 total,   1005.7 free,   1134.3 used.     54.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   4955 root      20   0 1164524 486444    304 S   0.0  24.5   0:02.32 perl
   1130 tux1      20   0 3830372  91496  26840 S   0.0   4.6   0:33.10 gnome-+
   1629 tux1      20   0 1190308  30896    724 S   0.0   1.6   0:06.47 snap-s+
   4903 tux1      20   0  571096  17460  11772 S   0.3   0.9   0:00.87 gnome-+
    836 root      20   0  875452   5552      0 S   0.0   0.3   0:02.80 snapd
    886 root      20   0  268944   4532   3648 S   0.0   0.2   0:00.53 Networ+
   1516 tux1      20   0  355256   4084   2896 S   0.0   0.2   0:01.54 ibus-e+
   1438 tux1      20   0  396912   3788   2752 S   0.0   0.2   0:01.64 ibus-d+
    560 systemd+  20   0   19212   3368   3008 S   0.0   0.2   0:00.12 system+
   2610 tux1      20   0  502040   3312   2064 S   0.0   0.2   0:00.37 update+
   1904 root      20   0  442744   3244    352 S   0.0   0.2   0:01.07 fwupd
      1 root      20   0  169356   3072   1668 S   0.0   0.2   0:01.46 systemd
   4842 tux1      20   0 2951620   2840   2840 S   0.0   0.1   0:00.32 gjs
   1709 tux1      39  19  637648   2776      0 S   0.0   0.1   0:01.42 tracke+
   1398 root      20   0  382040   2716    512 S   0.0   0.1   0:05.61 packag+
   1441 tux1      20   0  349700   2612   2096 S   0.0   0.1   0:00.12 gsd-co+
   1474 tux1      20   0  385760   2060    932 S   0.0   0.1   0:00.14 gsd-po+
    802 message+  20   0   11752   1948    724 S   0.0   0.1   0:00.52 dbus-d+
   1626 tux1      20   0  171540   1640   1312 S   0.0   0.1   0:00.52 ibus-e+
```

可以发现当以系统中一半的内存启用 usemem 程序时，缓存大小减小（从 298.1MB 下降至 102.8MB）。

18.Now start usemem with about 80% of your memory and watch what happens.

停止 usemem 程序，然后再执行命令 perl usemem 1640，以 80%的内存启动 usemem 程序，返回 top 工具：

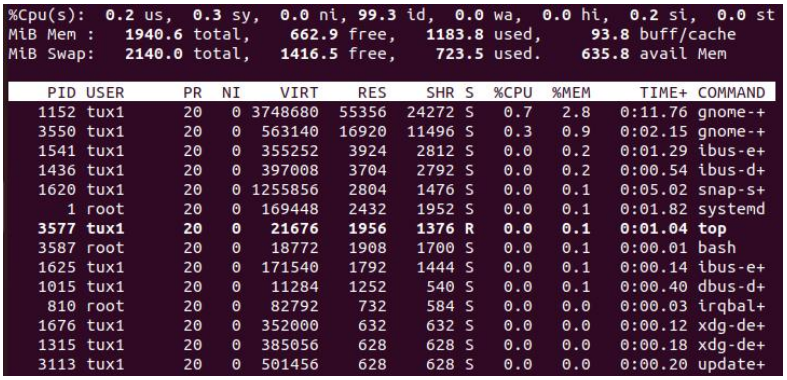
```
%Cpu(s):  0.0 us,  0.2 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1940.6 total,     70.3 free,   1776.8 used,     93.6 buff/cache
MiB Swap:   2140.0 total,    380.1 free,   1759.9 used.     43.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   4964 root      20   0 1804524 558396    272 S   0.0  28.1   0:03.77 perl
   1130 tux1      20   0 3830428  56772  22124 S   0.0   2.9   0:37.26 gnome-+
   4903 tux1      20   0  571096  17572  11876 S   0.3   0.9   0:03.06 gnome-+
   1516 tux1      20   0  355256   4060   3044 S   0.0   0.2   0:01.57 ibus-e+
   1438 tux1      20   0  396912   3472   2540 S   0.0   0.2   0:02.08 ibus-d+
   1327 tux1      20   0  672360   3300   2104 S   0.0   0.2   0:00.32 xdg-de+
   1699 tux1      20   0  352020   3052   2200 S   0.0   0.2   0:00.30 xdg-de+
   1629 tux1      20   0 1190308   2704    704 S   0.0   0.1   0:06.61 snap-s+
   1626 tux1      20   0  171540   2604   2188 S   0.0   0.1   0:00.66 ibus-e+
   4931 tux1      20   0   21676   1892   1272 R   0.3   0.1   0:03.71 top
   1001 tux1      20   0   16896   1560    668 S   0.0   0.1   0:00.94 dbus-d+
   1452 tux1      20   0  393668   1300   1120 S   0.0   0.1   0:00.21 gsd-ho+
```

可以发现，系统缓存大小进一步下降。

19.Stop usemem and watch what happens. Note that processes swapped out are not automatically swapped in as soon as memory comes available. They will be swapped in when needed.

按下回车键停止 usemem 程序，返回 top 工具，top 工具中内容如下图所示：



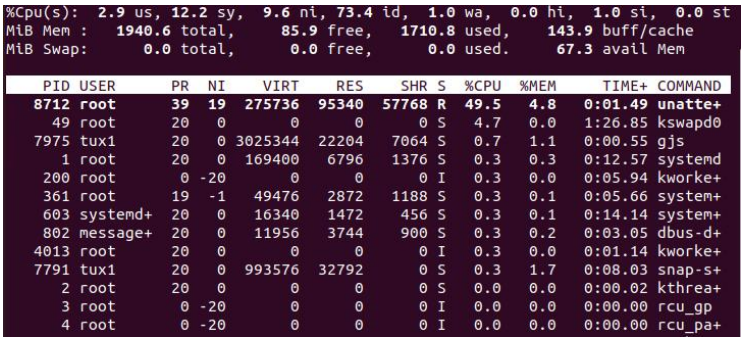
注意到此时缓存大小基本不变化，空闲内存大小明显增大，使用内存大小明显减小。

20.Disable your swap space with swapoff and watch what happens.

使用 swapoff 命令禁用交换空间：

```
root@tux1-virtual-machine:~/bin# swapoff /swapfile
```

返回 top 工具，top 工具中结果如下图所示：



可以发现，交换空间总量、空闲交换区总量和使用的交换区总量均变为 0MB。

21.Enable swap space again, and run usemem with about 90% of the total amount of memory (real + swap). If you exhaust your real memory, and exhaust the swap space, you will see that usemem is automatically killed when it tries to allocate even more memory.

使用 swapon 命令再次启用交换空间：

```
root@tux1-virtual-machine:~/bin# swapon /swapfile
```

使用约 90%的内存总量（实际内存+交换内存）运行 usemem 程序，运行结果如下：

```
root@tux1-virtual-machine:~/bin# perl usemem 3680
Allocating 3680 megabytes of memory...
Killed
```

程序被杀死。

Creating a swap file

If you suddenly need more swap space on a running system but you have no more empty partitions to spare, you can use swap files. Using swap files, however, is less efficient than swap partitions, so use this only in an emergency. It is even possible to put your swap file on an NFS-mounted directory, but there are easier ways of bringing down a network.

22.First, find a file system where you have room for a large swap file (at least 64 MB free). Locate a suitable directory on this file system.

首先使用 df 命令检查 Linux 系统上的可用磁盘空间，

```
tux1@tux1-virtual-machine:~$ df -Th
Filesystem     Type      Size  Used Avail Use% Mounted on
tmpfs          tmpfs     195M  1.8M  193M   1% /run
/dev/sda3      ext4       20G   13G  5.9G  68% /
tmpfs          tmpfs     971M     0  971M   0% /dev/shm
tmpfs          tmpfs     5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2      vfat      512M  5.3M  507M   2% /boot/efi
tmpfs          tmpfs     195M  2.4M  192M   2% /run/user/1000
/dev/sr0       iso9660   142M  142M     0 100% /media/tux1/CDROM
/dev/sr1       iso9660   3.8G  3.8G     0 100% /media/tux1/Ubuntu 22.10 amd64
```

其中/dev/sda3 文件系统，其文件系统类型为 ext4，有 5.9G 空闲空间可用，足够存放大型交换文件，故可以在/var 目录下创建交换文件。

23.Create the large file to be used as swap file.

首先执行 cd 命令进入目录/var 下：

```
tux1@tux1-virtual-machine:~$ cd /var
tux1@tux1-virtual-machine:/var$
```

接下来使用 fallocate 命令创建一个 2G 大小的文件，用于交换空间：

```
tux1@tux1-virtual-machine:/var$ sudo fallocate -l 2G /myswapfile
```

最后使用 chmod 命令和 chown 命令设置正确的文件系统权限，以确保只有 root 用户可以读写交换文件：

```
tux1@tux1-virtual-machine:/var$ sudo chmod 600 /myswapfile
tux1@tux1-virtual-machine:/var$ sudo chown root:root /myswapfile
```

24.Convert this file into a swap file.

使用 mkswap 命令将文件设置为交换文件：

```
tux1@tux1-virtual-machine:/var$ sudo mkswap /myswapfile
Setting up swapspace version 1, size = 2 GiB (2147479552 bytes)
no label, UUID=ec5af9a1-66a5-4544-bcab-4e4eefe9afeb
```

25.Activate it.

使用 swapon 命令激活该交换文件：

```
tux1@tux1-virtual-machine:/var$ sudo swapon /myswapfile
```

接下来使用 swapon --show 命令验证交换空间是否激活：

```
NAME        TYPE SIZE USED PRIO
/swapfile   file 2.1G 412M   -2
/myswapfile file   2G   0B   -3
```

经检验，该交换文件已成功激活。

26.Go to your top window and check whether the swap space has increased. Also, view the /proc/swaps file. What do you think is the meaning of the Priority field, and why is this different from the swap partition? Now try the usemem command that failed last time again.

返回 top 工具，此时 top 工具中结果如下图所示：

```
%Cpu(s):  0.0 us,  0.2 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1940.6 total,    571.9 free,   1228.0 used,    140.7 buff/cache
MiB Swap:   4188.0 total,   3634.7 free,    553.3 used.    568.2 avail Mem
```

可以发现，交换空间增加。接下来执行 cat 命令查看/proc/swaps 文件：

```
tux1@tux1-virtual-machine:~$ cat /proc/swaps
Filename                                Type            Size            Used  P
riority
/swapfile                               file            2191356         514992-
2
/myswapfile                             file            2097148         7304  -
3
```

再次以 3680MB 内存启动 usemem 程序：

```
root@tux1-virtual-machine:~/bin# perl usemem 3680
Allocating 3680 megabytes of memory...
Press ENTER to release memory...
```

usemem 程序不被终止，能够成功运行，此时 top 窗口结果如下图所示：

```
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1940.6 total,     67.0 free,   1769.2 used,    104.4 buff/cache
MiB Swap:   4188.0 total,    484.1 free,   3703.9 used.     45.2 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   3861 root      20   0 4367096 521780      0 S   0.0  26.3   0:11.54 perl
   1095 tux1      20   0 3833612  63252  22984 S   0.0   3.2   0:49.92 gnome-+
   3720 tux1      20   0  570976  23564  17512 S   0.0   1.2   0:00.75 gnome-+
    797 root      20   0  949184   4988    168 S   0.0   0.3   0:03.38 snapd
   3862 tux1      20   0   21676   4236   3372 R   0.0   0.2   0:00.03 top
   1399 tux1      20   0  323276   3580   2624 S   0.0   0.2   0:02.22 ibus-d+
   1497 tux1      20   0  355252   3516   2476 S   0.0   0.2   0:01.47 ibus-e+
   1569 tux1      20   0  171676   2808   2344 S   0.0   0.1   0:00.70 ibus-e+
   3738 tux1      20   0   19704   2044   1668 S   0.0   0.1   0:00.00 bash
   1577 tux1      20   0 1059860   1732   1256 S   0.0   0.1   0:04.46 snap-s+
      1 root      20   0  168024   1592   1068 S   0.0   0.1   0:01.41 systemd
    956 tux1      20   0   16124   1516    660 S   0.0   0.1   0:01.40 dbus-d+
   1400 tux1      20   0  349712   1516   1516 S   0.0   0.1   0:00.19 gsd-co+
    844 root      20   0  268928   1016    776 S   0.0   0.1   0:00.69 Networ+
```

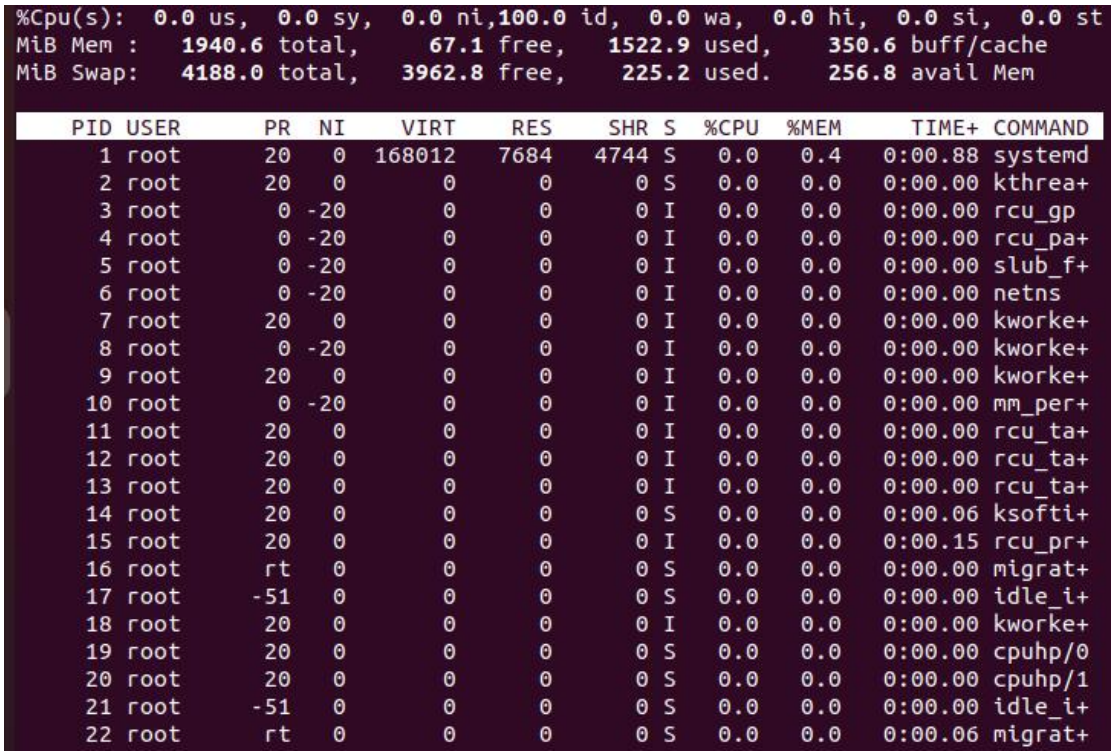
27.Add the swap file to your /etc/fstab file so that it is activated next time you reboot.

通过 echo 命令和 tee 命令将交换文件信息添加到/etc/fstab 文件的末尾：

```
tux1@tux1-virtual-machine:~$ echo '/myswapfile none swap sw 0 0' | sudo tee -a
/etc/fstab
/myswapfile none swap sw 0 0
```

28.Reboot your system to make sure that you are working with the correct amount of memory again.

重启系统，并打开 top 工具，得到显示结果如下图所示：



可以发现，内存总量和交换空间总量正确，说明已成功将 myswapfile 交换文件添加到/etc/fstab 文件中。