IT Scripting and Automation

System Monitoring

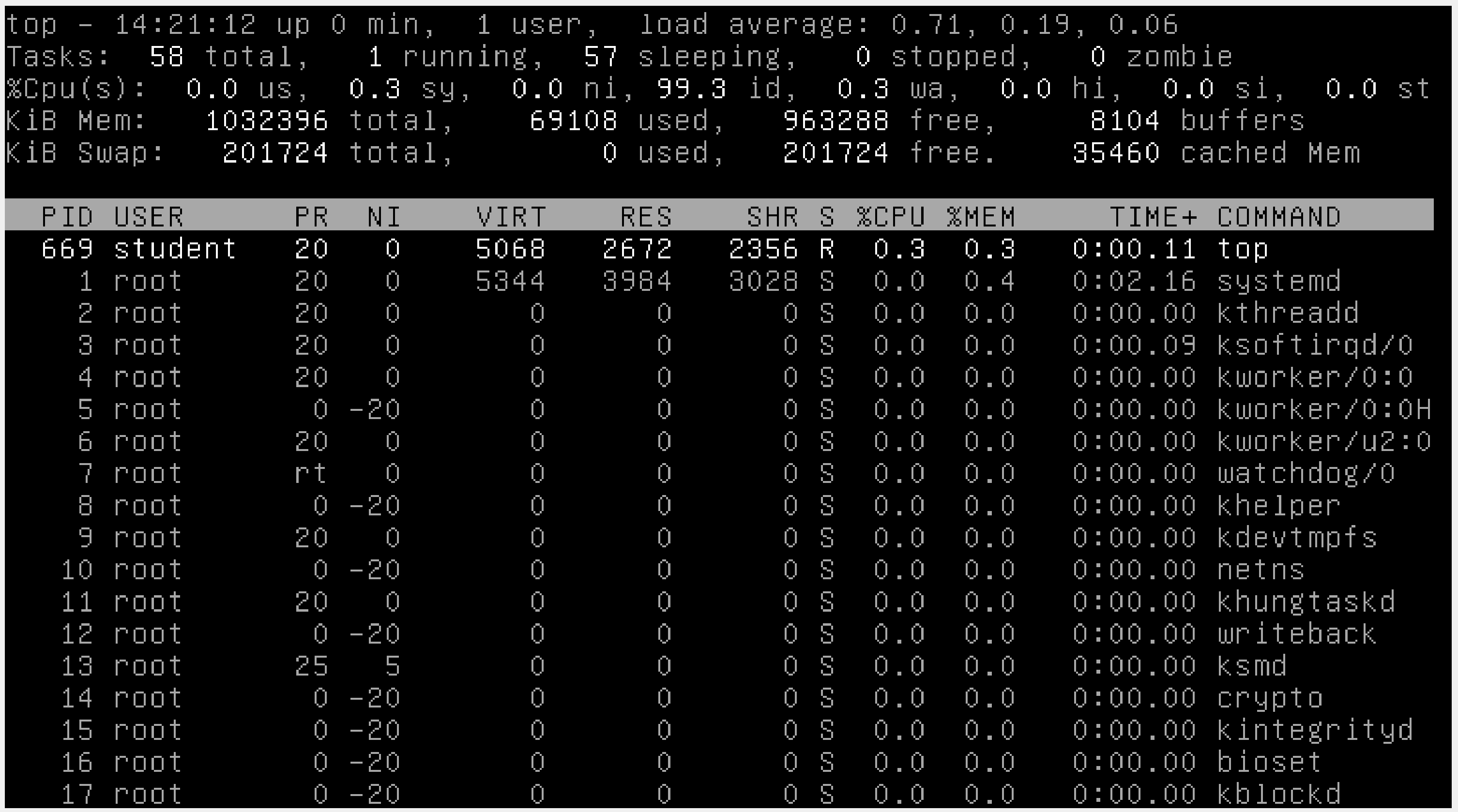
Lecturer: Art Ó Coileáin

System Monitoring

- One of the most important responsibilities as a system administrator is monitoring their systems.
- A system administrator must have the ability to find out what is happening on your system at any given time:
  - Whether it is the percentage of system resources currently used
  - What commands are being run
  - Who is logged on
- If system resources become to low it can cause a lot of problems.
- System resources can be used by individual users, or by services your system may host e.g. as email or web pages.

System Monitoring: commands

- The most common command is **top**. This command displays a continually updating report of system resource usage.
- The top portion of the report list information: system time, uptime, CPU usage, swap memory usage, and number of processes.



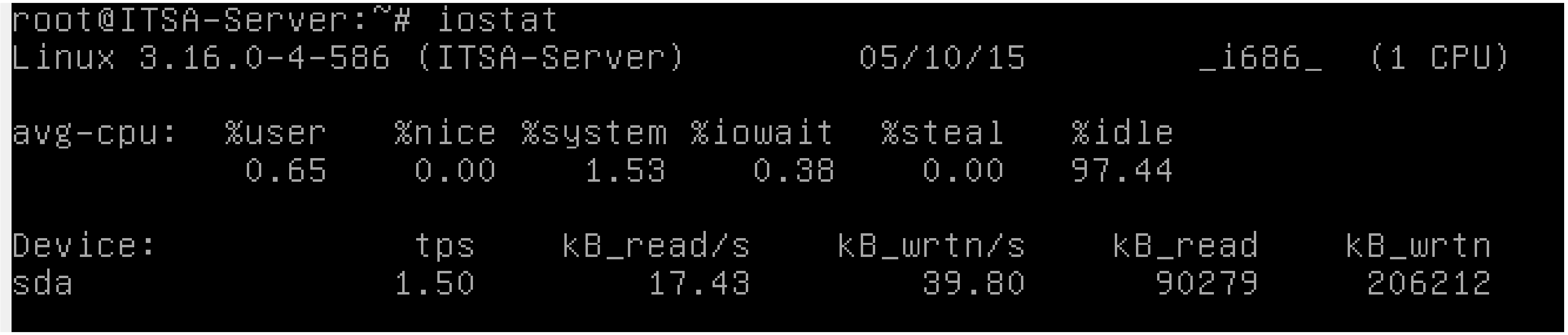
```
top - 14:21:12 up 0 min,  1 user,  load average: 0.71, 0.19, 0.06
Tasks:  58 total,   1 running,  57 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.3 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1032396 total,    69108 used,   963288 free,     8104 buffers
KiB Swap:   201724 total,        0 used,   201724 free.    35460 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  669 student   20   0    5068   2672   2356 R  0.3  0.3   0:00.11 top
    1 root      20   0    5344   3984   3028 S  0.0  0.4   0:02.16 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S  0.0  0.0   0:00.09 ksoftirqd/0
    4 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0
    5 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kworker/u2:0
    7 root      rt   0       0      0      0 S  0.0  0.0   0:00.00 watchdog/0
    8 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 khelper
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   10 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 netns
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.00 khungtaskd
   12 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 writeback
   13 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
   14 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 crypto
   15 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kintegrityd
   16 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 bioset
   17 root       0 -20       0      0      0 S  0.0  0.0   0:00.00 kblockd
```

Command ‘top’

- The output of **top** can be modified while it is running.
- **i**: If you hit an **i**, top will no longer display ide processes. Hit **i** again to see them again.
- **M**: Hitting **M** will sort by memory usage, and
- **P**: **P** will sort by CPU usage.
- **u**: You can use **u** to view processes owned by a specific user,
- **k**: **k** to kill processes, and
- **r**: **r** to renice them.
- **h**: For more information hit **h**.
- For more in-depth information about processes you can look in the **/proc** filesystem. In the **/proc** filesystem you will find a series of sub-directories with numeric names. These directories are associated with the process ids of currently running processes.

Command ‘iostat’

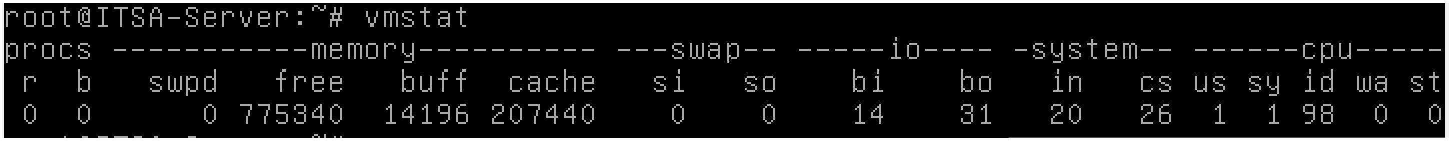
- The iostat will display the current CPU load average and disk I/O information.



```
root@ITSA-Server:~# iostat
Linux 3.16.0-4-586 (ITSA-Server)        05/10/15        _i686_  (1 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.65    0.00    1.53    0.38    0.00   97.44

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               1.50        17.43        39.80      90279     206212
```

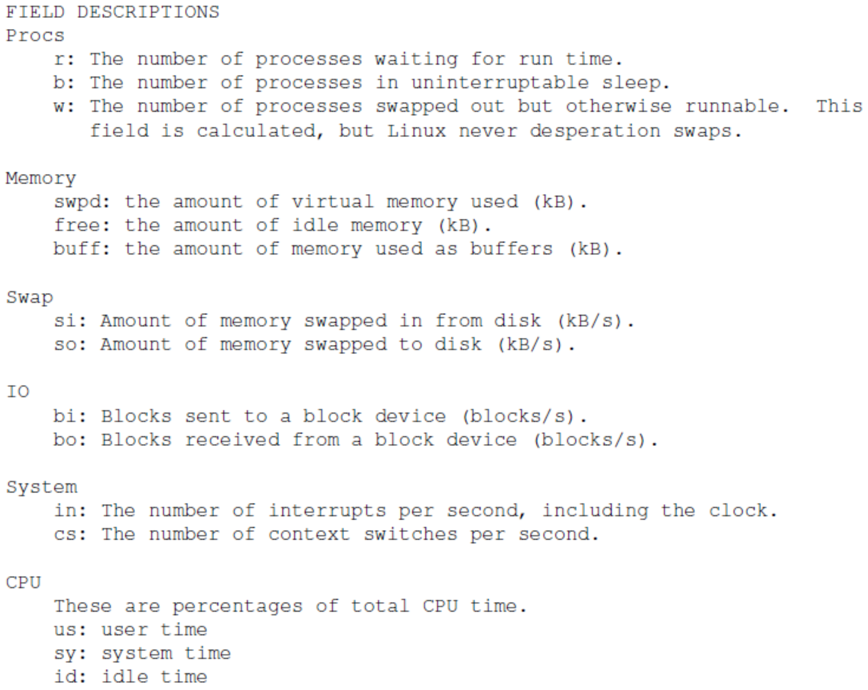
Command ‘vmstat’

- The **vmstat** command will provide a report showing statistics for system processes, memory, swap, I/O, and the CPU. These statistics are generated using data from the last time the command was run to the present. If the command never being run, the data will be from the last reboot until the present.



```
root@ITSA-Server:~# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0      0 775340  14196 207440    0    0    14    31   20   26  1  1 98  0  0
```

- Use the man page for more information.



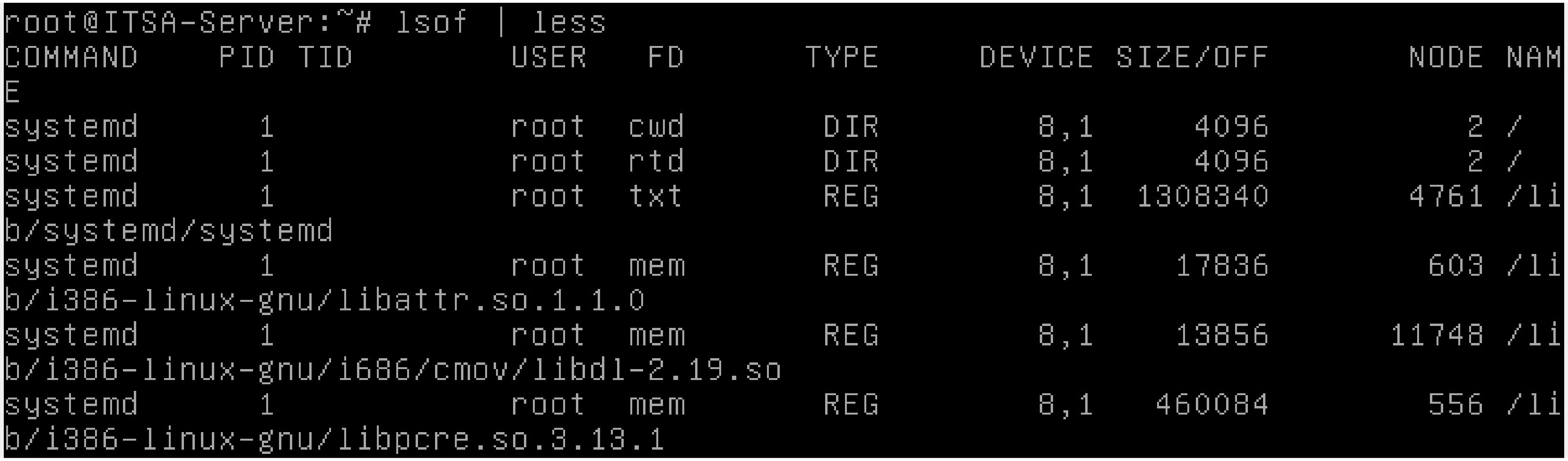
```
FIELD DESCRIPTIONS
Procs
    r: The number of processes waiting for run time.
    b: The number of processes in uninterruptible sleep.
    w: The number of processes swapped out but otherwise runnable.  This
       field is calculated, but Linux never desperation swaps.

Memory
    swpd: the amount of virtual memory used (kB).
    free: the amount of idle memory (kB).
    buff: the amount of memory used as buffers (kB).

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

IO
    bi: Blocks sent to a block device (blocks/s).
    bo: Blocks received from 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: user time
    sy: system time
    id: idle time
```

Command ‘lsof’

- The **lsof** command will print out a list of every file that is in use.

Usage example:

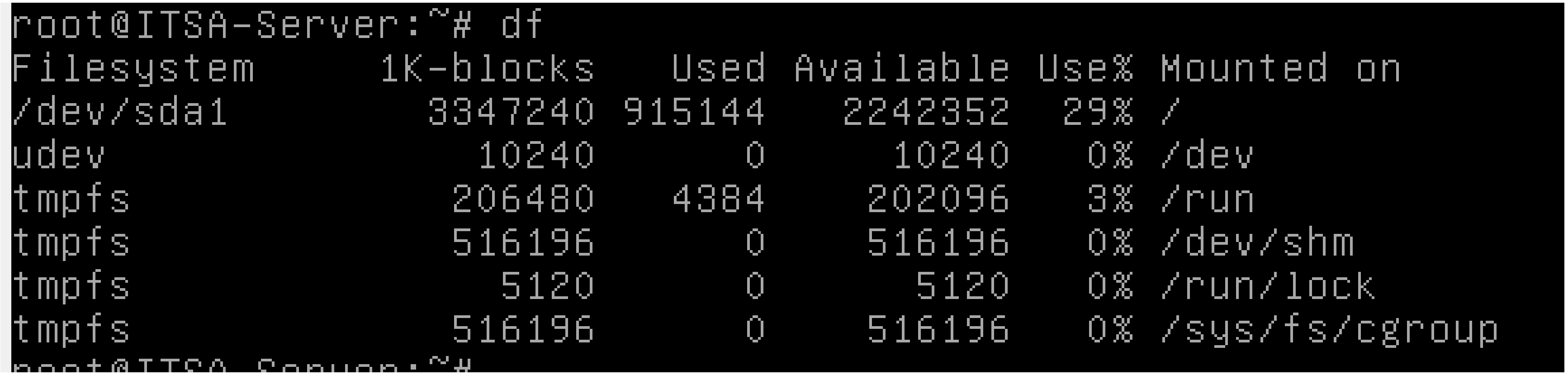
- An example of use is if you wish to unmount a filesystem, but you are being told that is in use. This command and **grep** for the name of the filesystem to see who is using it.



```
root@ITSA-Server:~# lsof | less
COMMAND    PID TID          USER   FD       TYPE       DEVICE SIZE/OFF        NODE NAM
E
systemd      1              root  cwd        DIR          8,1     4096           2 /
systemd      1              root  rtd        DIR          8,1     4096           2 /
systemd      1              root  txt        REG          8,1  1308340        4761 /li
b/systemd/systemd
systemd      1              root  mem        REG          8,1    17836         603 /li
b/i386-linux-gnu/libattr.so.1.1.0
systemd      1              root  mem        REG          8,1    13856       11748 /li
b/i386-linux-gnu/i686/cmov/libdl-2.19.so
systemd      1              root  mem        REG          8,1   460084         556 /li
b/i386-linux-gnu/libpcre.so.3.13.1
```

Command ‘df’

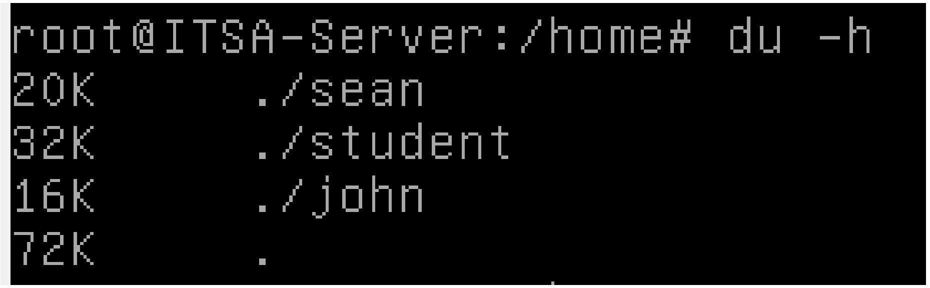
- The command **df** is the simples tool available to view disk usage.
- It will show the disk usage for all mounted filesystems in 1K blocks.
- **df -h** will display output in “human-readable” format (K, Megs, Gigs depending on the size of the filesystem).



```
root@ITSA-Server:~# df
Filesystem     1K-blocks   Used Available Use% Mounted on
/dev/sda1        3347240 915144   2242352  29% /
udev               10240      0     10240   0% /dev
tmpfs             206480   4384    202096   3% /run
tmpfs             516196      0    516196   0% /dev/shm
tmpfs               5120      0      5120   0% /run/lock
tmpfs             516196      0    516196   0% /sys/fs/cgroup
root@ITSA-Server:~#
```

Command ‘du’

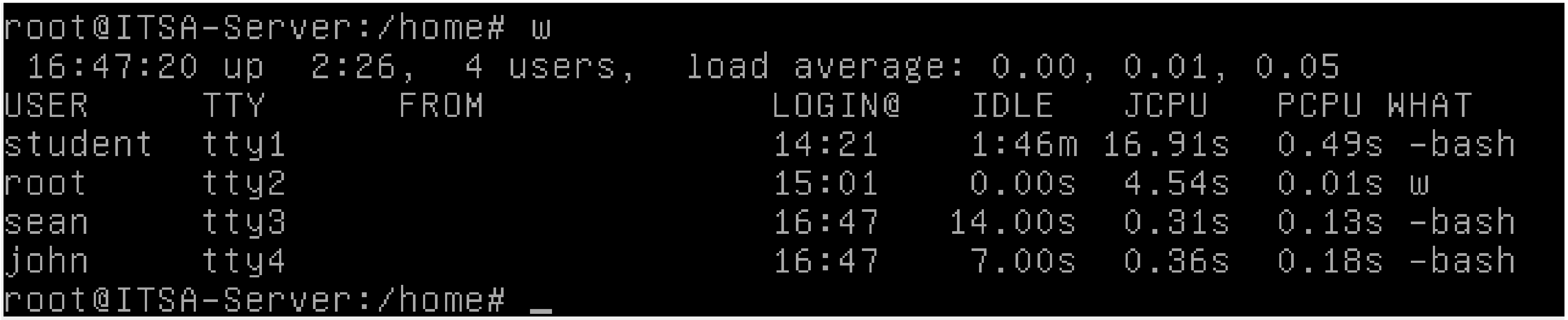
- To view usage by a directory or file, the command **du** is used; **du** command will act recursively.



```
root@ITSA-Server:/home# du -h
20K     ./sean
32K     ./student
16K     ./john
72K     .
```

Command ‘w’

- The command **w** will print out not only who is on the system, but also the commands they are running.



```
root@ITSA-Server:/home# w
 16:47:20 up  2:26,  4 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
student  tty1                      14:21    1:46m 16.91s  0.49s -bash
root     tty2                      15:01    0.00s  4.54s  0.01s w
sean     tty3                      16:47   14.00s  0.31s  0.13s -bash
john     tty4                      16:47    7.00s  0.36s  0.18s -bash
root@ITSA-Server:/home# _
```

Command ‘shutdown’

- The command **shutdown** will quit all running programs, log out on all virtual consoles.

  e.g.: **$ shutdown -h now**

- It will shutdown the system immediately.

Time delay and message:

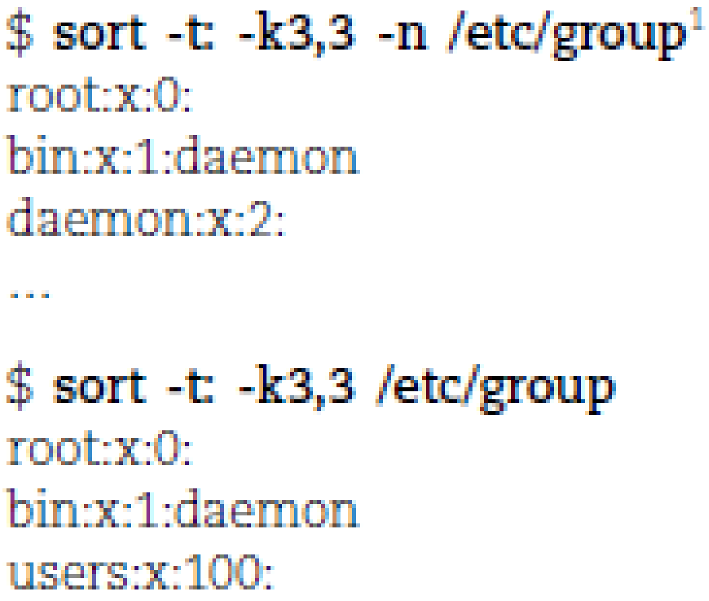
- Alternatively, the command **shutdown -h + time message**, where time is the time in minutes until the system is halted, and message is a short explanation of why the system is shutting down.

Example:

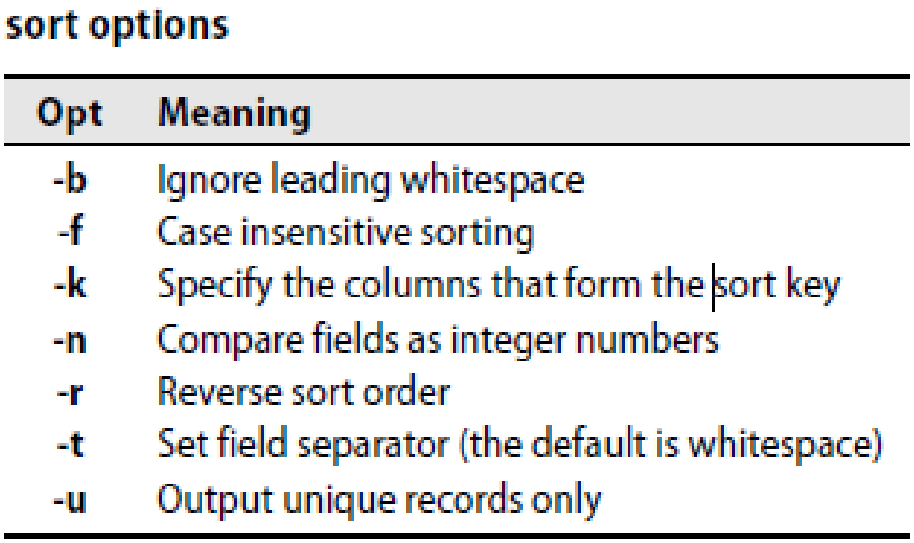
$ shutdown -h +10 ‘The system requires to reboot. It will be restarted in 10 minutes.’

Command ‘sort’

- This command sorts its input lines.



```
$ sort -t: -k3,3 -n /etc/group[1]
root:x:0:
bin:x:1:daemon
daemon:x:2:
...

$ sort -t: -k3,3 /etc/group
root:x:0:
bin:x:1:daemon
users:x:100:
```

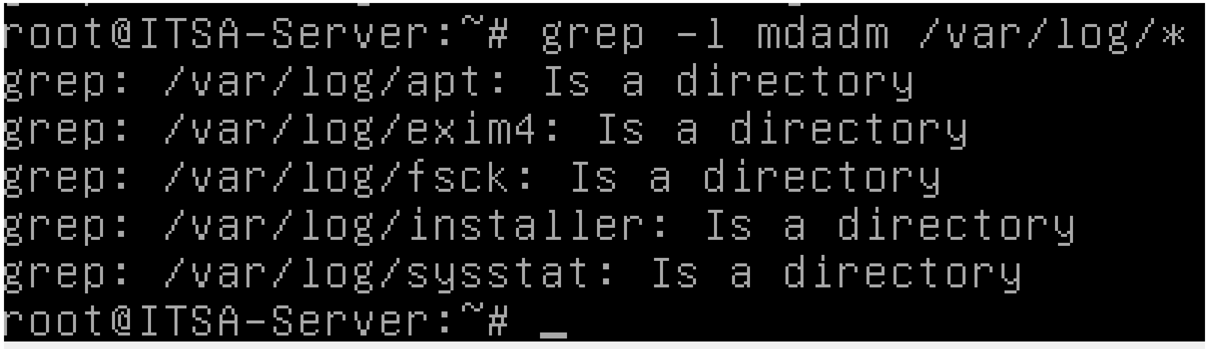


sort options

Opt	Meaning
-b	Ignore leading whitespace
-f	Case insensitive sorting
-k	Specify the columns that form the sort key
-n	Compare fields as integer numbers
-r	Reverse sort order
-t	Set field separator (the default is whitespace)
-u	Output unique records only

Command ‘grep’

- Command **grep** searches its input text and prints the lines that match a given pattern.
- **grep** has many options including :
- **-c** to print a count of matching lines,
- **-i** to ignore case when matching, and
- **-v** to print nonmatching lines
- **-l** which makes grep print only the names of matching files rather than printing each line that matches



```
root@ITSA-Server:~# grep -l mdadm /var/log/*
grep: /var/log/apt: Is a directory
grep: /var/log/exim4: Is a directory
grep: /var/log/fsck: Is a directory
grep: /var/log/installer: Is a directory
grep: /var/log/sysstat: Is a directory
root@ITSA-Server:~# _
```

Command ‘find’

- The command **find** is one of the most important and much used command in Linux systems. Find can be used in variety of conditions like you can find files by permission, users, groups, file type, date, size and other possible criteria.
- To find files whose name is file.txt in a current directory:

```
$ find . -name ‘file.txt’
```

- To find all files under /home directory with name file.txt:

```
$ find /home -name ‘file.txt’
```

- To find all the files whose name is file.txt and contains both capital and small letter in the directory (*ignore case*)

```
$ find /home -iname ‘file.txt’
```

Command ‘find’ (2)

- To find all the files whose permissions are 755:

```
$ find . -type f -perm 0755 -print
```

- To find all read only files:

```
$ find / -perm /u=r
```

- To find all executable files:

```
$ find / -perm /a=x
```

- To find and remove multiple files such as .avi

```
$ find . -type f -name “*.avi” -exec rm -rf{} \;
```

- To find all empty files

```
$ find /tmp -type f -empty
```

- To find all files that belongs to user Sean under /tmp directory

```
$ find /tmp -user sean
```

Command ‘find’ (3)

- To find all files which are modified 20 days back:

```
$ find / -mtime 20
```

- To find all files which are changed in last 1 hour:

```
$ find / -cmin -60
```

- To find all 50 Mb files, use:

```
$ find / -size 50M
```

- To find all the files which are greater than 50M and less than 100MB:

```
$ find / -size +50M -size -100M
```

- To find all the .sh files which contain “*keyword*”:

```
find . -iname "*.sh"  -exec grep -l "keyword" {} \;
```