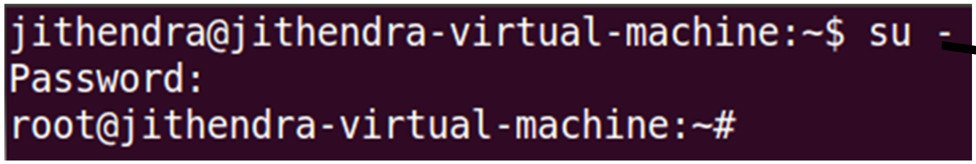
IT 6204
Section 3.0

Host Management

3.1 Root Privileges

Root Privileges

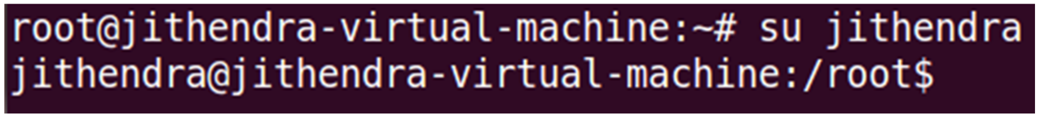
- **su**
  - Create a shell with the effective user ID. If no user is specified, create a shell for a privileged user.
  - su [option] [user]



```
jithendra@jithendra-virtual-machine:~$ su -
Password:
root@jithendra-virtual-machine:~#
```

" – " switch will allow you to login as root

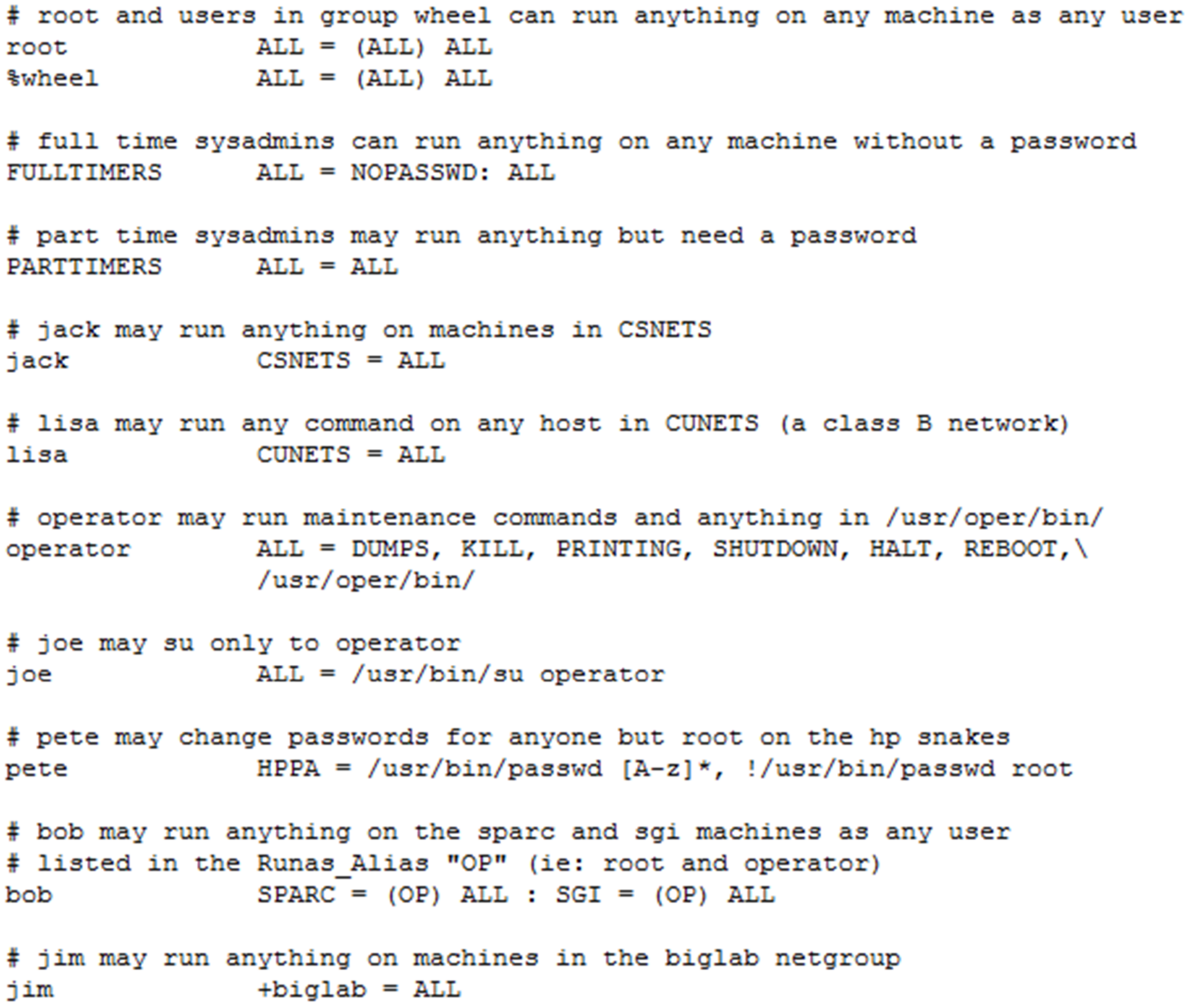
  - Login as a user.



```
root@jithendra-virtual-machine:~# su jithendra
jithendra@jithendra-virtual-machine:/root$
```

Root Privileges

- **sudo**
  - If you have privileges, “sudo” allows you to execute commands as superuser.
  - sudo [options] [command]
  - Main advantage of sudo is you can create policies for users and limit their access to execute programs.
  - These policies are located in /etc/sudoers file

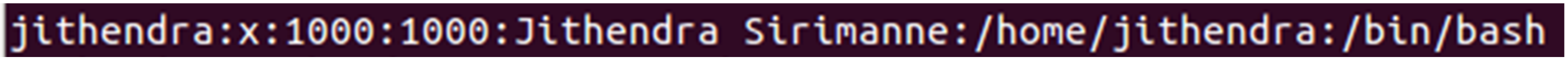


```
# root and users in group wheel can run anything on any machine as any user
root            ALL = (ALL) ALL
%wheel          ALL = (ALL) ALL

# full time sysadmins can run anything on any machine without a password
FULLTIMERS      ALL = NOPASSWD: ALL

# part time sysadmins may run anything but need a password
PARTTIMERS      ALL = ALL

# jack may run anything on machines in CSNETS
jack            CSNETS = ALL

# lisa may run any command on any host in CUNETS (a class B network)
lisa            CUNETS = ALL

# operator may run maintenance commands and anything in /usr/oper/bin/
operator        ALL = DUMPS, KILL, PRINTING, SHUTDOWN, HALT, REBOOT,\
                /usr/oper/bin/

# joe may su only to operator
joe             ALL = /usr/bin/su operator

# pete may change passwords for anyone but root on the hp snakes
pete            HPPA = /usr/bin/passwd [A-z]*, !/usr/bin/passwd root

# bob may run anything on the sparc and sgi machines as any user
# listed in the Runas_Alias "OP" (ie: root and operator)
bob             SPARC = (OP) ALL : SGI = (OP) ALL

# jim may run anything on machines in the biglab netgroup
jim             +biglab = ALL
```

3.2 User Management

User Management

- **Passwd file**
  - Locates in /etc/passwd
  - When we create a user and a password it will store these user information in the passwd file. There are seven fields of information. Each record consists of seven fields separated by colons ' : ' symbol.

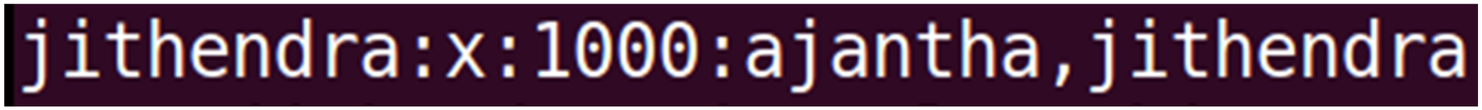


```
jithendra:x:1000:1000:Jithendra Sirimanne:/home/jithendra:/bin/bash
```

Username : Password : User Identifier(UID) : Group Identifier(GID) : Name of the User : Home Directory : Program or Shell

User Management

- **Group file**
  - Locates in /etc/group
  - Group file is text file, it defines the groups on the system.
  - In the group file there are three data fields.
  - Groupname : Password : Group ID: Users



```
jithendra:x:1000:ajantha,jithendra
```

User Management

- **Home Directory**
  - Personal workspace of the user. Only the user and super user has the access to this personal directory. User directories are stored under /home/[user]
  - Eg: /home/saman
- **Setting permission and ownership**
  - Linux has three types of permissions.

    Read Write Execute
  - We can allocate permissions by using binary numbers.

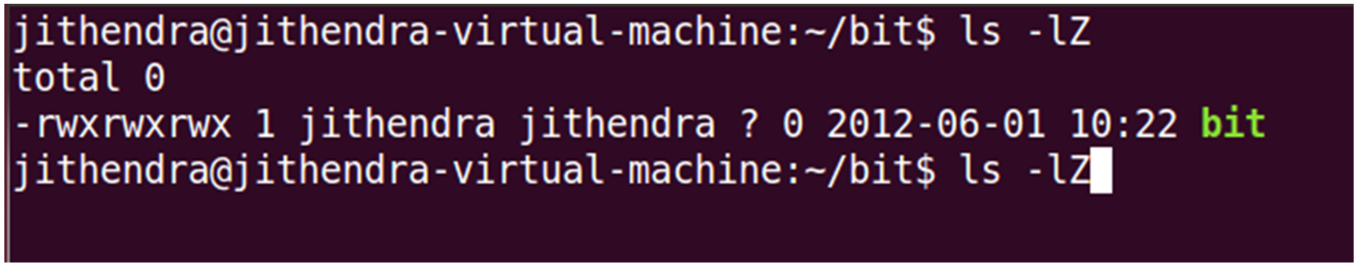
User Management

Triplet for u: rwx => 4 + 2 + 1 = **7**

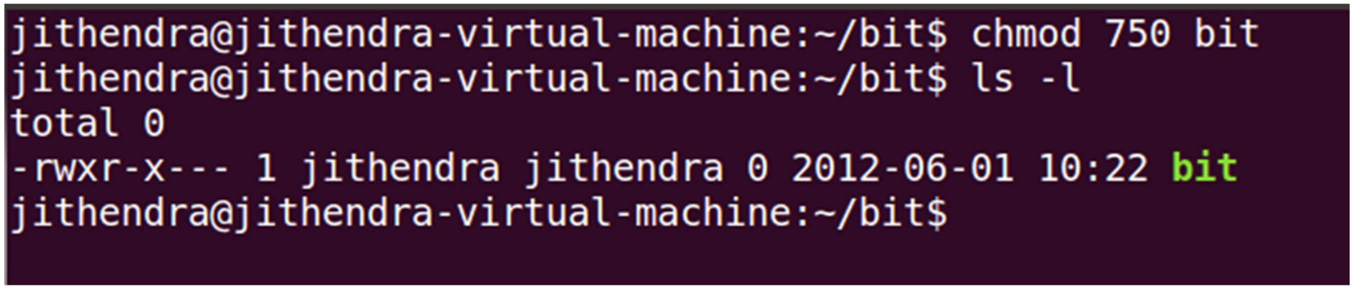
Triplet for g: r-x => 4 + 0 + 1 = **5**

Tripler for o: r-x => 4 + 0 + 1 = **5**

Which makes : 755



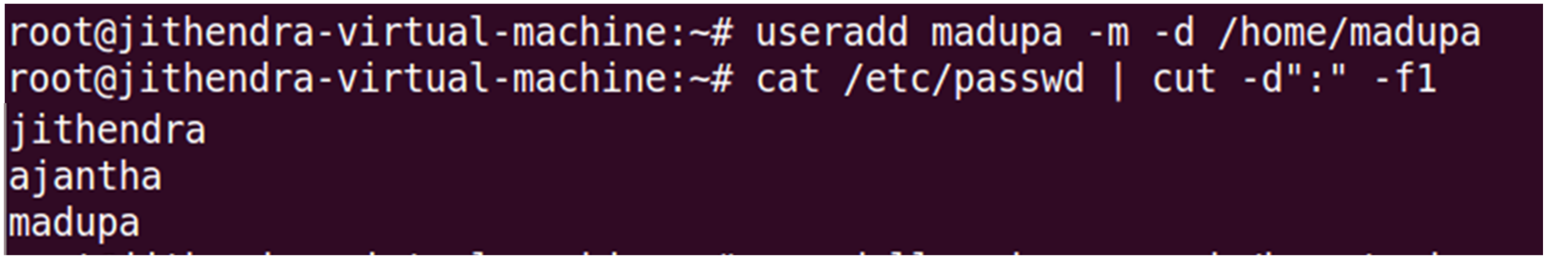
```
jithendra@jithendra-virtual-machine:~/bit$ ls -lZ
total 0
-rwxrwxrwx 1 jithendra jithendra ? 0 2012-06-01 10:22 bit
jithendra@jithendra-virtual-machine:~/bit$ ls -lZ
```



```
jithendra@jithendra-virtual-machine:~/bit$ chmod 750 bit
jithendra@jithendra-virtual-machine:~/bit$ ls -l
total 0
-rwxr-x--- 1 jithendra jithendra 0 2012-06-01 10:22 bit
jithendra@jithendra-virtual-machine:~/bit$
```

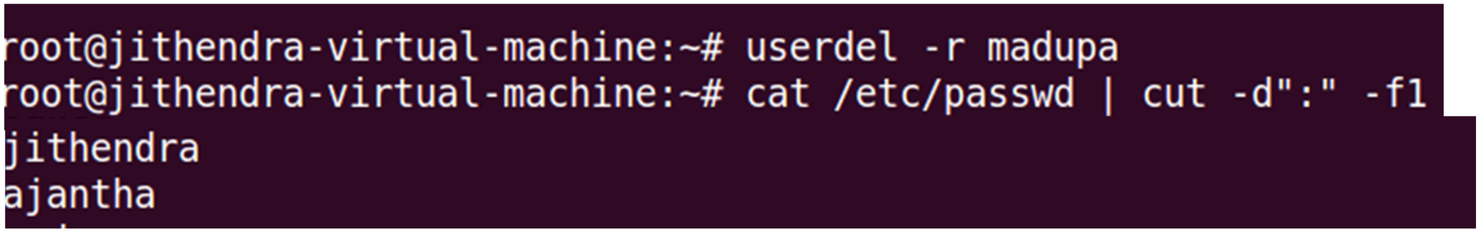
User Management

- **Adding/deleting users**
  - Superuser or privileged user can add or remove users from the system.
  - useradd [user] [options]



```
root@jithendra-virtual-machine:~# useradd madupa -m -d /home/madupa
root@jithendra-virtual-machine:~# cat /etc/passwd | cut -d":" -f1
jithendra
ajantha
madupa
```

userdel [user] [options]



```
root@jithendra-virtual-machine:~# userdel -r madupa
root@jithendra-virtual-machine:~# cat /etc/passwd | cut -d":" -f1
jithendra
ajantha
```

User Management

- **Modify user account information**
  - usermod [options] [user]
- **Disabling logins**
  - System administrator can block users temporary without deleting their account using “pw lock [user]” command.
  - To unlock you have to use “pw unlock [user]”
  - Another way to block user is
    - “usermod –L [user] “

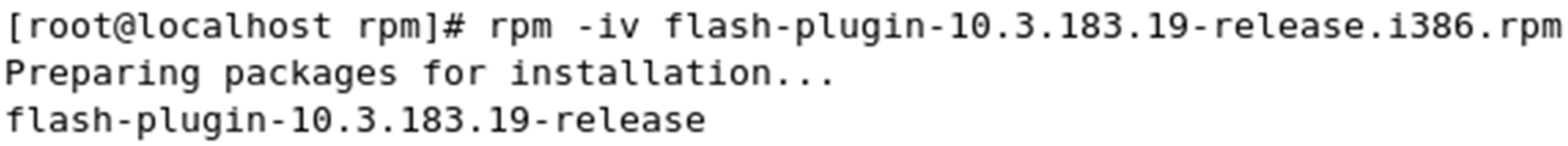
3.3 Software Installation & Management

Software Installation & Management

- **rpm**
  - rpm [*options*]
  - A freely available packaging system for software distribution and installation. RPM packages are built, installed, and queried with the **rpm** and **rpmbuild** . The rpm command options are grouped into three subgroups for: Querying and verifying packages Installing, upgrading, and removing packages Performing miscellaneous functions

Software Installation & Management

- **Rpm install**

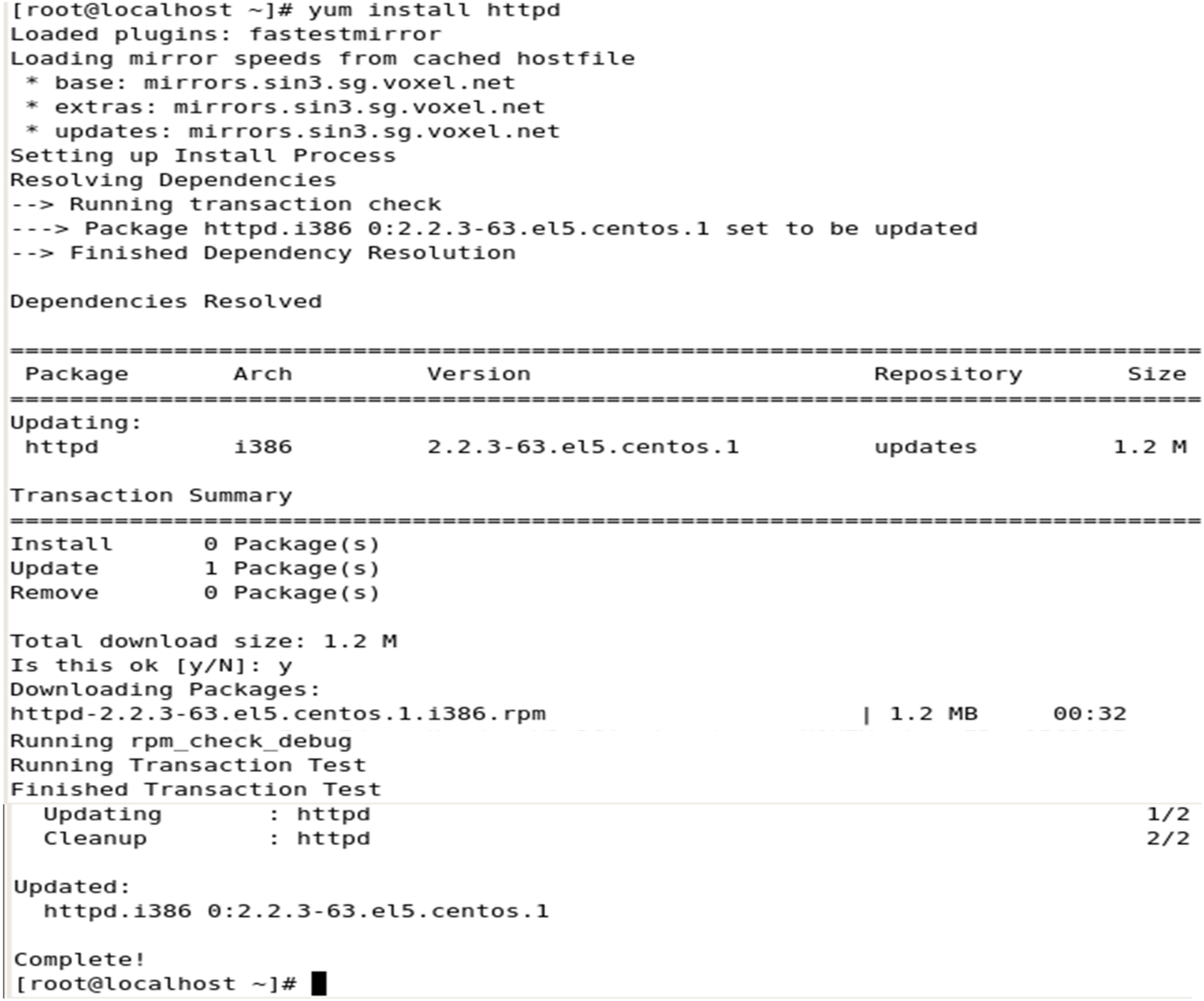


```
[root@localhost rpm]# rpm -iv flash-plugin-10.3.183.19-release.i386.rpm
Preparing packages for installation...
flash-plugin-10.3.183.19-release
```

- **RPM remove**
  - To remove you can use the –e option
  - Rpm –ev [package]

Software Installation & Management

- **yum (Yellowdog Updater Modified)**
  - *Yum [command] [package name/s]*
  - yum will automatically attempt to check all configured repositories to resolve all package dependencies during an installation/upgrade.
  - You can add new yum software repository url into the end of the file /etc/yum.conf or in a separate file named [anyname].repo in /etc/yum.repos.d/ directory.



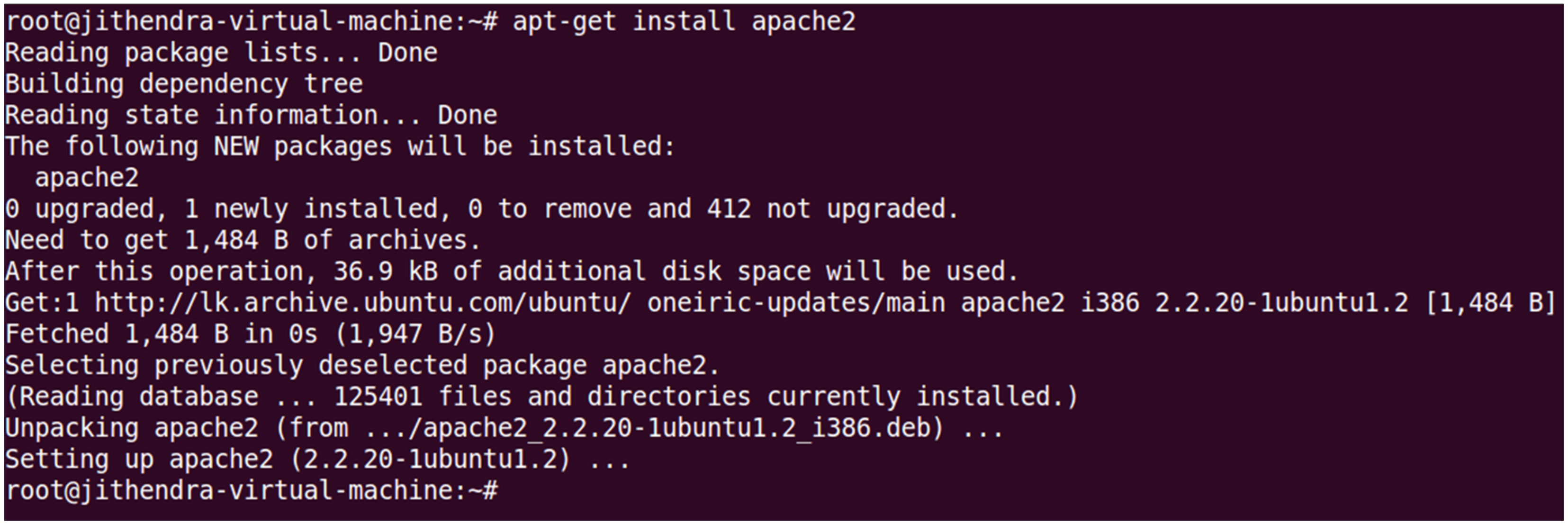
```
[root@localhost ~]# yum install httpd
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.sin3.sg.voxel.net
 * extras: mirrors.sin3.sg.voxel.net
 * updates: mirrors.sin3.sg.voxel.net
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package httpd.i386 0:2.2.3-63.el5.centos.1 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package       Arch         Version                       Repository       Size
================================================================================
Updating:
 httpd         i386         2.2.3-63.el5.centos.1         updates         1.2 M

Transaction Summary
================================================================================
Install       0 Package(s)
Update        1 Package(s)
Remove        0 Package(s)

Total download size: 1.2 M
Is this ok [y/N]: y
Downloading Packages:
httpd-2.2.3-63.el5.centos.1.i386.rpm                     | 1.2 MB     00:32
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
  Updating       : httpd                                                    1/2
  Cleanup        : httpd                                                    2/2

Updated:
  httpd.i386 0:2.2.3-63.el5.centos.1

Complete!
[root@localhost ~]# █
```

Software Installation & Management

- **apt**
  - apt *[command] [package name/s]*
  - The Advanced Package Tool. A freely available packaging system for software distribution and installation.
  - You can add new apt software repository url to /etc/apt/sources.list file or in a separate files named [anyname].list in /etc/apt/sources.list.d directory.

Software Installation & Management

- Apt install



```
root@jithendra-virtual-machine:~# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 412 not upgraded.
Need to get 1,484 B of archives.
After this operation, 36.9 kB of additional disk space will be used.
Get:1 http://lk.archive.ubuntu.com/ubuntu/ oneiric-updates/main apache2 i386 2.2.20-1ubuntu1.2 [1,484 B]
Fetched 1,484 B in 0s (1,947 B/s)
Selecting previously deselected package apache2.
(Reading database ... 125401 files and directories currently installed.)
Unpacking apache2 (from .../apache2_2.2.20-1ubuntu1.2_i386.deb) ...
Setting up apache2 (2.2.20-1ubuntu1.2) ...
root@jithendra-virtual-machine:~#
```

3.4 Disk Storage

Storage Hardware Interfaces

- **ATA (Advanced Technology Attachment)**

  Known in earlier revisions as IDE, was developed as a simple, low-cost interface for PCs. It was originally called Integrated Drive Electronics because it put the hardware controller in the same box as the disk platters and used a relatively high-level protocol for communication between the computer and the disks.

Storage Hardware Interfaces

- **PATA(parallel ATA)**

This style of disk is nearly obsolete, but the installed base is enormous. PATA disks are often labeled as “IDE” to distinguish them from SATA drives. PATA disks are medium to fast in speed, generous in capacity, and unbelievably cheap

Storage Hardware Interfaces

- **SATA(Serial ATA)**

  SATA is the successor to PATA. In addition to supporting much higher transfer rates, SATA simplifies connectivity with tidier cabling and a longer maximum cable length.

- **SCSI**

  SCSI is one of the most widely supported disk interfaces. It comes in several flavors, all of which support multiple disks on a bus and various speeds and communication styles.

Disk Partitioning

- **RAID(Redundant Array of Inexpensive disks)**

  RAID is normally used to spread data among several physical hard drives with enough redundancy that should any drive fail the data will still be intact. Once created a RAID array appears to be one device which can be used pretty much like a regular partition.

Disk Partitioning

There are several kinds of RAID but there are two most common here.

RAID-1 (mirroring) With **RAID-1** it's basically done with two essentially identical drives, each with a complete set of data.

RAID-5 which is set up using three or more drives with the data spread in a way that any one drive failing will not result in data loss.

Disk Partitioning

- **LVM (Logical Volume Manager)**

  LVM is a way of grouping drives and/or partition in a way where instead of dealing with hard and fast physical partitions the data is managed in a virtual basis where the virtual partitions can be resized.

File Systems and Mounting.

- **File system types**
  - Ext: This is like UNIX file system. It has the concepts of blocks, inodes and directories.
  - Ext3: It is ext2 filesystem enhanced with journalling capabilities. Journalling allows fast file system recovery.
  - Isofs (iso9660): Used by CDROM file system.
  - Sysfs: It is a ram-based filesystem initially based on ramfs. It is use to exporting kernel objects so that end user can use it easily.
  - Procfs: The proc file system acts as an interface to internal data structures in the kernel.

File Systems and Mounting.

- **Mounting File Systems**
  - To access any file system, it is first necessary to *mount* it. Likewise, when access to a particular file system is no longer desired, it is necessary to *unmount* it. To mount any file system, two pieces of information must be specified:
  - A means of uniquely identifying the desired disk drive and partition, such as device file name, file system label, or devlabel-managed symbolic link
  - A directory under which the mounted file system is to be made available.

3.5 Controlling Processes

Controlling Processes

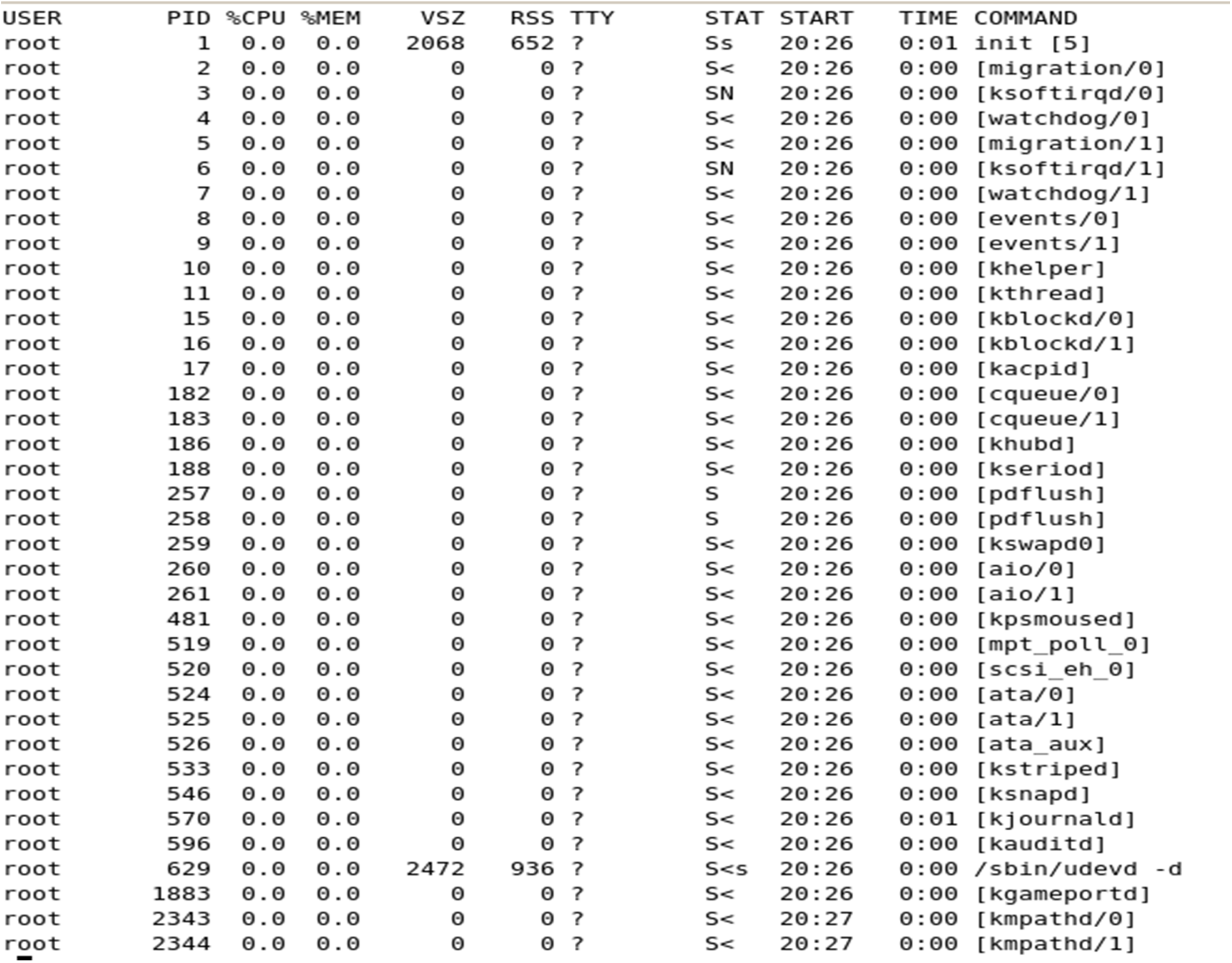
Process Attributes

- PID or process ID, an integer.
- PPID or parent process ID, an integer.
- Nice number, the degree of friendliness of the process towards other processes (process priority is calculated from nice numbers and recent CPU usage).
- TTY, the terminal to which the process is connected
- RUID, or real user ID. The user issuing the command.
- EUID, or effective user ID. The one determining access permissions to system resources.

Controlling Processes

Process Attributes ctd:

- EGID, or effective group owner. Different from RGID when SGID has been applied to a file.
- RGID, or real group owner. The group of the user who started the process



```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2068   652 ?        Ss   20:26   0:01 init [5]
root         2  0.0  0.0      0     0 ?        S<   20:26   0:00 [migration/0]
root         3  0.0  0.0      0     0 ?        SN   20:26   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S<   20:26   0:00 [watchdog/0]
root         5  0.0  0.0      0     0 ?        S<   20:26   0:00 [migration/1]
root         6  0.0  0.0      0     0 ?        SN   20:26   0:00 [ksoftirqd/1]
root         7  0.0  0.0      0     0 ?        S<   20:26   0:00 [watchdog/1]
root         8  0.0  0.0      0     0 ?        S<   20:26   0:00 [events/0]
root         9  0.0  0.0      0     0 ?        S<   20:26   0:00 [events/1]
root        10  0.0  0.0      0     0 ?        S<   20:26   0:00 [khelper]
root        11  0.0  0.0      0     0 ?        S<   20:26   0:00 [kthread]
root        15  0.0  0.0      0     0 ?        S<   20:26   0:00 [kblockd/0]
root        16  0.0  0.0      0     0 ?        S<   20:26   0:00 [kblockd/1]
root        17  0.0  0.0      0     0 ?        S<   20:26   0:00 [kacpid]
root       182  0.0  0.0      0     0 ?        S<   20:26   0:00 [cqueue/0]
root       183  0.0  0.0      0     0 ?        S<   20:26   0:00 [cqueue/1]
root       186  0.0  0.0      0     0 ?        S<   20:26   0:00 [khubd]
root       188  0.0  0.0      0     0 ?        S<   20:26   0:00 [kseriod]
root       257  0.0  0.0      0     0 ?        S    20:26   0:00 [pdflush]
root       258  0.0  0.0      0     0 ?        S    20:26   0:00 [pdflush]
root       259  0.0  0.0      0     0 ?        S<   20:26   0:00 [kswapd0]
root       260  0.0  0.0      0     0 ?        S<   20:26   0:00 [aio/0]
root       261  0.0  0.0      0     0 ?        S<   20:26   0:00 [aio/1]
root       481  0.0  0.0      0     0 ?        S<   20:26   0:00 [kpsmoused]
root       519  0.0  0.0      0     0 ?        S<   20:26   0:00 [mpt_poll_0]
root       520  0.0  0.0      0     0 ?        S<   20:26   0:00 [scsi_eh_0]
root       524  0.0  0.0      0     0 ?        S<   20:26   0:00 [ata/0]
root       525  0.0  0.0      0     0 ?        S<   20:26   0:00 [ata/1]
root       526  0.0  0.0      0     0 ?        S<   20:26   0:00 [ata_aux]
root       533  0.0  0.0      0     0 ?        S<   20:26   0:00 [kstriped]
root       546  0.0  0.0      0     0 ?        S<   20:26   0:00 [ksnapd]
root       570  0.0  0.0      0     0 ?        S<   20:26   0:01 [kjournald]
root       596  0.0  0.0      0     0 ?        S<   20:26   0:00 [kauditd]
root       629  0.0  0.0   2472   936 ?        S<s  20:26   0:00 /sbin/udevd -d
root      1883  0.0  0.0      0     0 ?        S<   20:26   0:00 [kgameportd]
root      2343  0.0  0.0      0     0 ?        S<   20:27   0:00 [kmpathd/0]
root      2344  0.0  0.0      0     0 ?        S<   20:27   0:00 [kmpathd/1]
```

Controlling Processes

- **Signals**
  - Signals are a way of sending simple messages to processes.
  - Most of these messages are already defined and can be found in <linux/signal.h>.
  - signals can only be processed when the process is in user mode.
  - If a signal has been sent to a process that is in kernel mode, it is dealt with immediately on returning to user mode.
  - Signals are one of the oldest inter-process communication methods used by Unix TM systems.

Controlling Processes

Process states in Linux:

- **Running:** Process is either running or ready to run
- **Interruptible:** a Blocked state of a process and waiting for an event or signal from another process
- **Uninterruptible:** a blocked state. Process waits for a hardware condition and cannot handle any signal
- **Stopped:** Process is stopped or halted and can be restarted by some other process
- **Zombie:** process terminated, but information is still there in the process table.

Controlling Processes

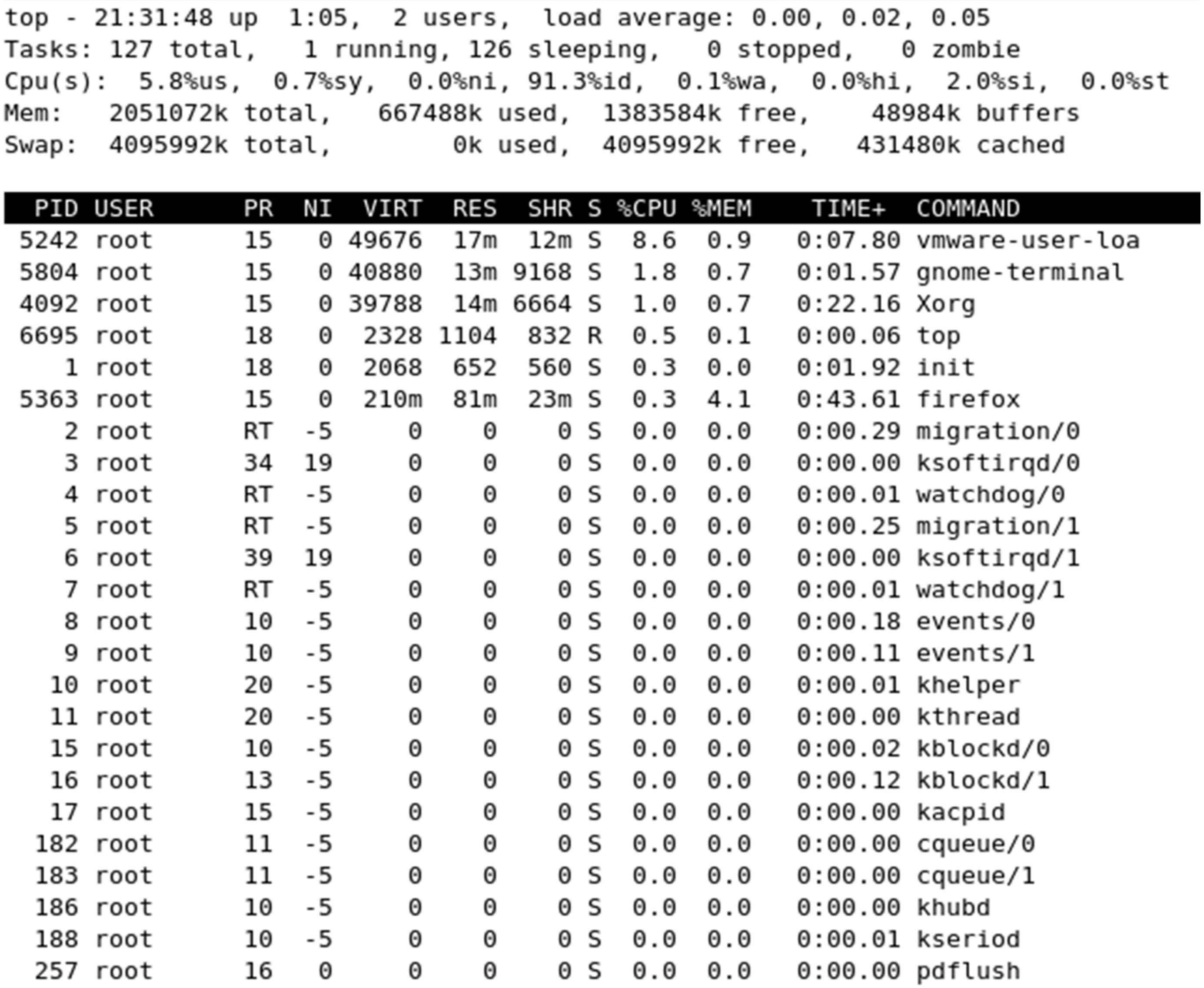
Commands

top - display top CPU processes

top [-] [d *delay*] [p *pid*]

top provides an ongoing look at processor activity in real time. It displays a listing of the most CPU-intensive tasks on the system, and can provide an interactive interface for manipulating processes. It can sort the tasks by CPU usage, memory usage and runtime.

Controlling Processes



```
top - 21:31:48 up  1:05,  2 users,  load average: 0.00, 0.02, 0.05
Tasks: 127 total,   1 running, 126 sleeping,   0 stopped,   0 zombie
Cpu(s):  5.8%us,  0.7%sy,  0.0%ni, 91.3%id,  0.1%wa,  0.0%hi,  2.0%si,  0.0%st
Mem:   2051072k total,   667488k used,  1383584k free,    48984k buffers
Swap:  4095992k total,        0k used,  4095992k free,   431480k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5242 root      15   0 49676  17m  12m S  8.6  0.9   0:07.80 vmware-user-loa
 5804 root      15   0 40880  13m 9168 S  1.8  0.7   0:01.57 gnome-terminal
 4092 root      15   0 39788  14m 6664 S  1.0  0.7   0:22.16 Xorg
 6695 root      18   0  2328 1104  832 R  0.5  0.1   0:00.06 top
    1 root      18   0  2068  652  560 S  0.3  0.0   0:01.92 init
 5363 root      15   0  210m  81m  23m S  0.3  4.1   0:43.61 firefox
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:00.29 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.01 watchdog/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:00.25 migration/1
    6 root      39  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/1
    7 root      RT  -5     0    0    0 S  0.0  0.0   0:00.01 watchdog/1
    8 root      10  -5     0    0    0 S  0.0  0.0   0:00.18 events/0
    9 root      10  -5     0    0    0 S  0.0  0.0   0:00.11 events/1
   10 root      20  -5     0    0    0 S  0.0  0.0   0:00.01 khelper
   11 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
   15 root      10  -5     0    0    0 S  0.0  0.0   0:00.02 kblockd/0
   16 root      13  -5     0    0    0 S  0.0  0.0   0:00.12 kblockd/1
   17 root      15  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
  182 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue/0
  183 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 cqueue/1
  186 root      10  -5     0    0    0 S  0.0  0.0   0:00.00 khubd
  188 root      10  -5     0    0    0 S  0.0  0.0   0:00.01 kseriod
  257 root      16   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
```

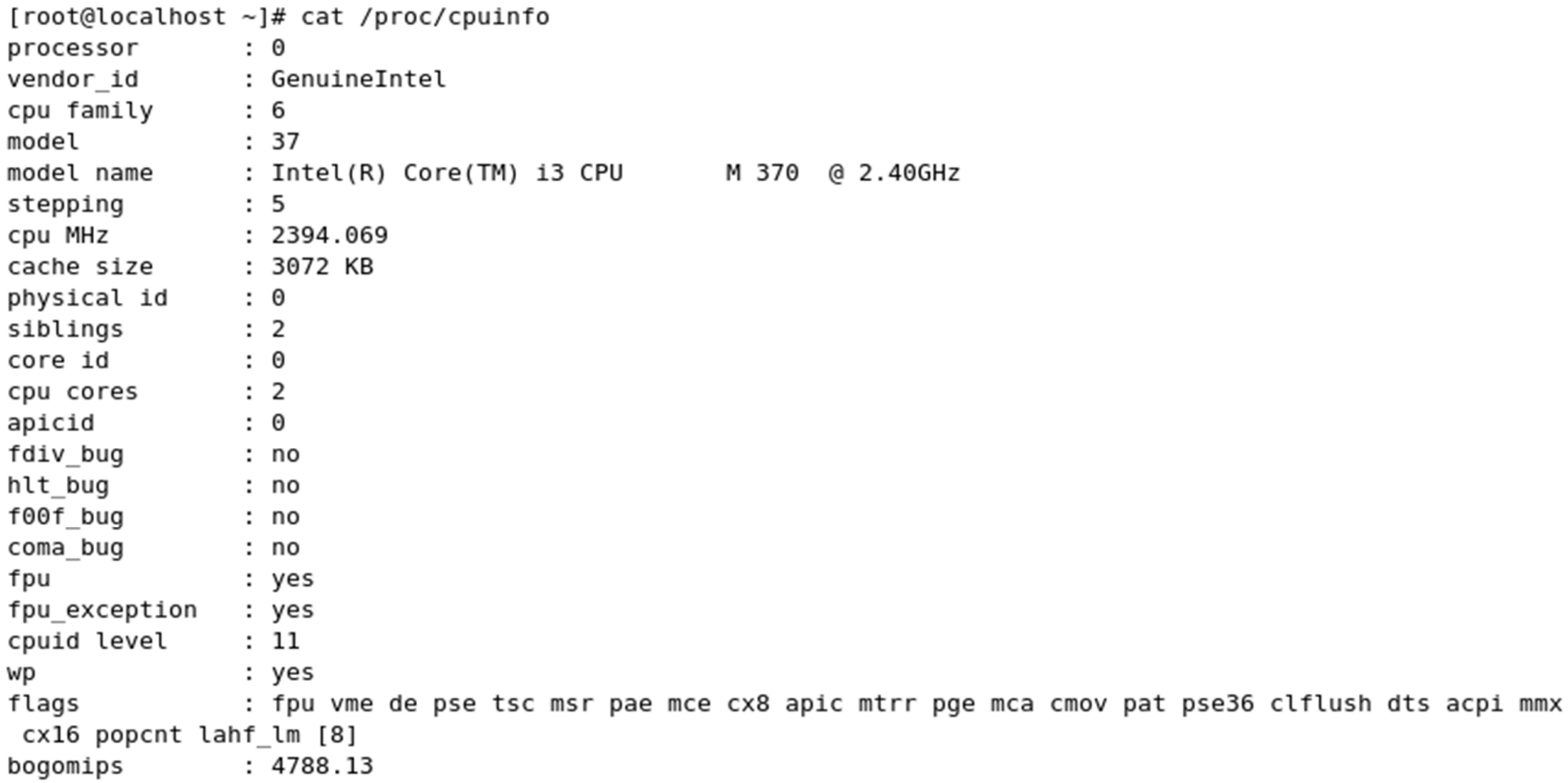
Controlling Processes

proc - process information pseudo-filesystem

/proc is a pseudo-filesystem which is used as an interface to kernel data structures rather than reading and interpreting /dev/kmem. Most of it is read-only, but some files allow kernel variables to be changed.

Controlling Processes

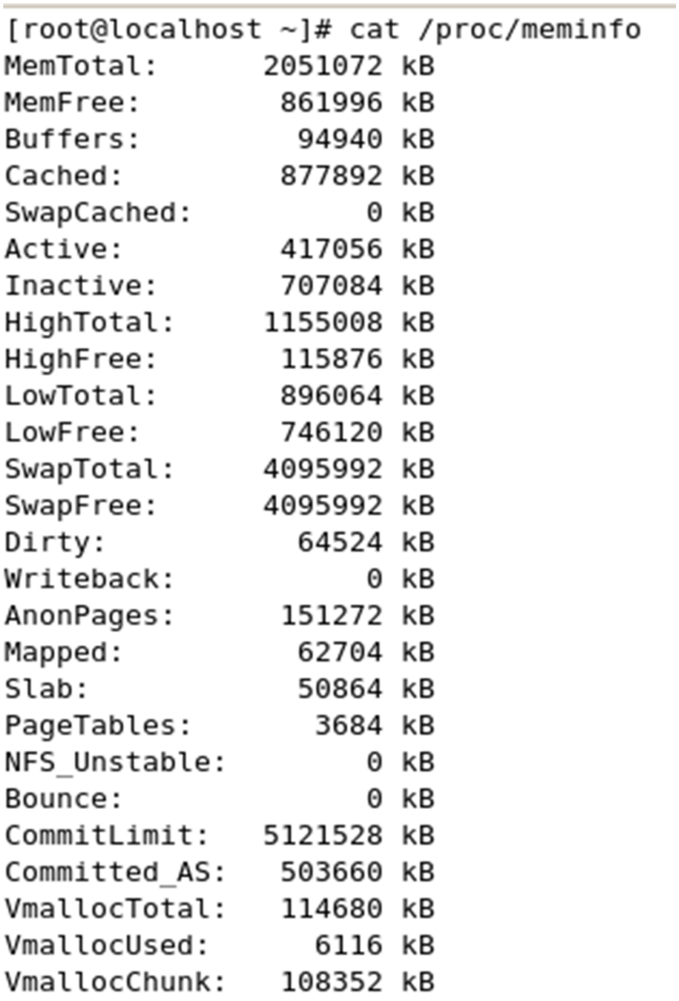
proc - cpuinfo



```
[root@localhost ~]# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 37
model name      : Intel(R) Core(TM) i3 CPU       M 370  @ 2.40GHz
stepping        : 5
cpu MHz         : 2394.069
cache size      : 3072 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
apicid          : 0
fdiv_bug        : no
hlt_bug         : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 11
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts acpi mmx
 cx16 popcnt lahf_lm [8]
bogomips        : 4788.13
```

Controlling Processes

proc - meminfo



```
[root@localhost ~]# cat /proc/meminfo
MemTotal:       2051072 kB
MemFree:         861996 kB
Buffers:          94940 kB
Cached:          877892 kB
SwapCached:           0 kB
Active:          417056 kB
Inactive:        707084 kB
HighTotal:      1155008 kB
HighFree:        115876 kB
LowTotal:        896064 kB
LowFree:         746120 kB
SwapTotal:      4095992 kB
SwapFree:       4095992 kB
Dirty:            64524 kB
Writeback:            0 kB
AnonPages:       151272 kB
Mapped:           62704 kB
Slab:             50864 kB
PageTables:        3684 kB
NFS_Unstable:         0 kB
Bounce:               0 kB
CommitLimit:    5121528 kB
Committed_AS:    503660 kB
VmallocTotal:    114680 kB
VmallocUsed:       6116 kB
VmallocChunk:    108352 kB
```

Controlling Processes

nice - run a program with modified scheduling priority

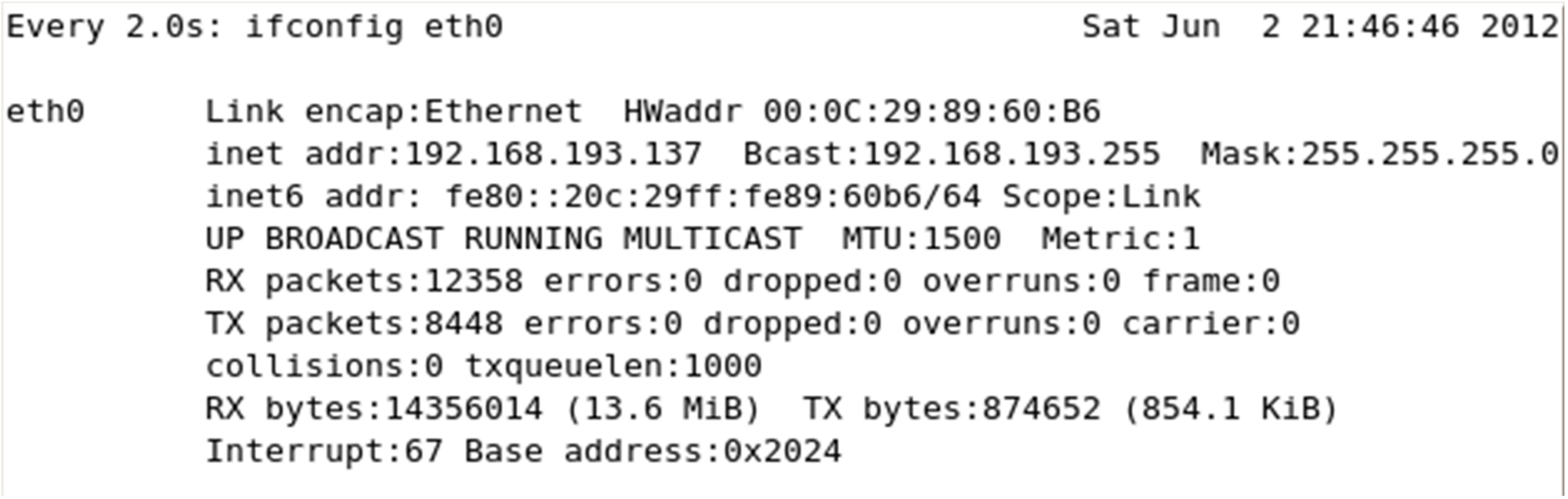
nice [*OPTION*] [*COMMAND* [*ARG*]...]

Run COMMAND with an adjusted scheduling priority. With no COMMAND, print the current scheduling priority. ADJUST is 10 by default. Range goes from **-20** (highest priority) to 19 (lowest).

Controlling Processes

Watch – this will execute a program periodically, showing output fullscreen

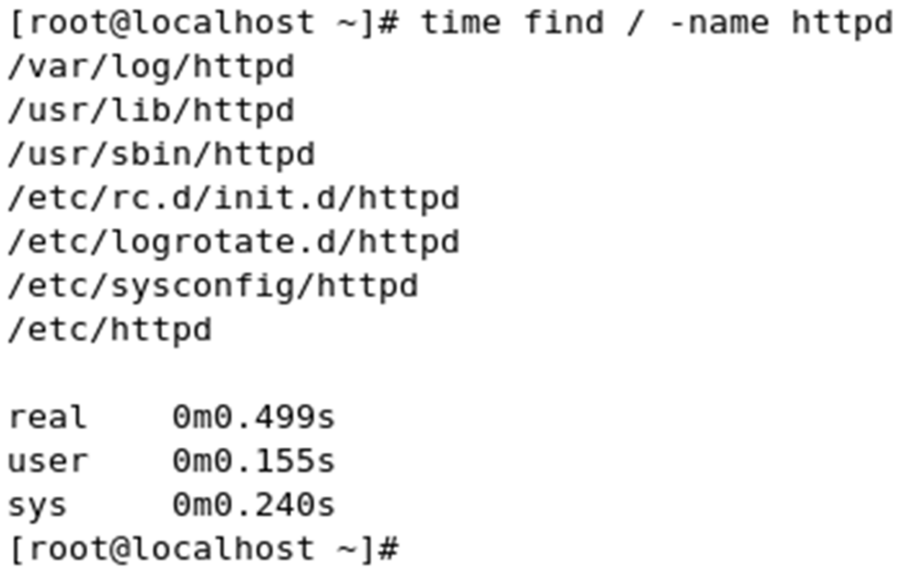
Watch [*OPTION*] <command>



```
Every 2.0s: ifconfig eth0                           Sat Jun  2 21:46:46 2012

eth0      Link encap:Ethernet  HWaddr 00:0C:29:89:60:B6
          inet addr:192.168.193.137  Bcast:192.168.193.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe89:60b6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12358 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8448 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:14356014 (13.6 MiB)  TX bytes:874652 (854.1 KiB)
          Interrupt:67 Base address:0x2024
```

Controlling Processes

- **time** – The time command runs the specified program command with the given arguments. When command finishes, time outputs giving timing statistics about this program run.
- **time** [*OPTION*] <command>



```
[root@localhost ~]# time find / -name httpd
/var/log/httpd
/usr/lib/httpd
/usr/sbin/httpd
/etc/rc.d/init.d/httpd
/etc/logrotate.d/httpd
/etc/sysconfig/httpd
/etc/httpd

real    0m0.499s
user    0m0.155s
sys     0m0.240s
[root@localhost ~]#
```

3.6 File System

File System

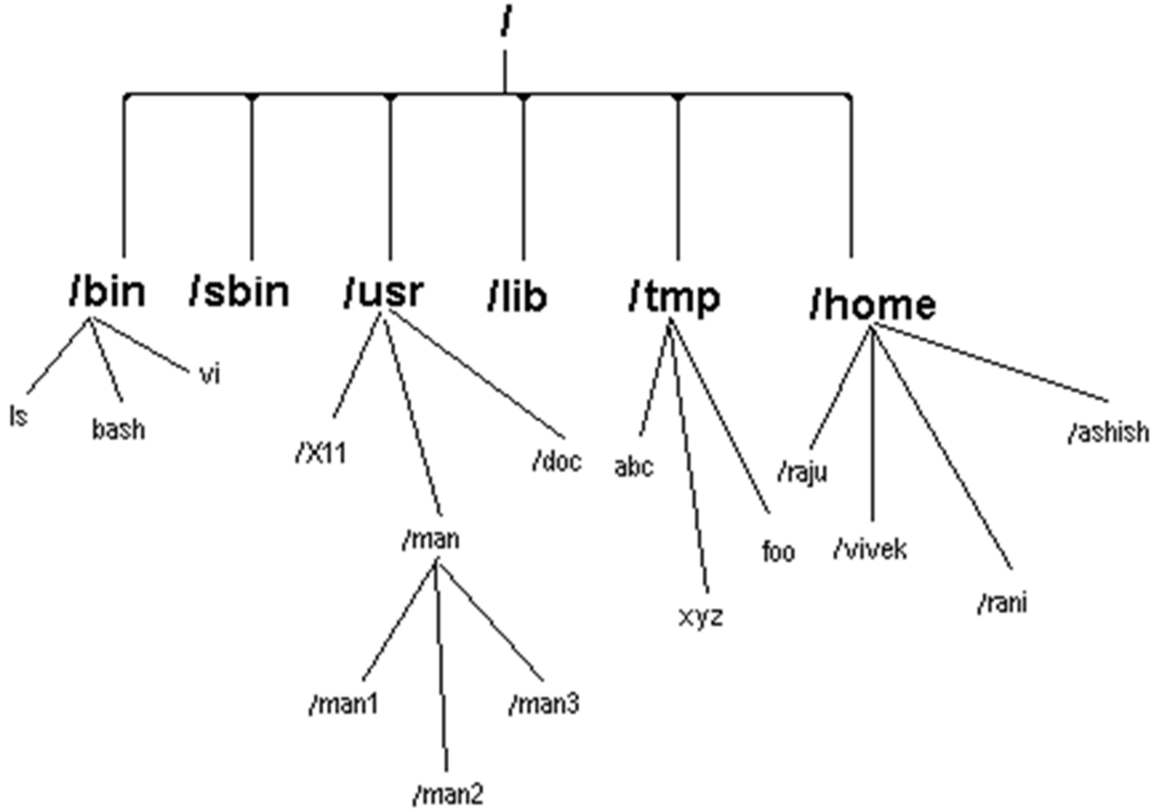
- **Path Names**
  - – In linux everything has a absolute path unlike windows.
  - – Everything starts from root (/)
  - – Paths use / as the separator.
    - Eg: /home/jithendra/Downloads

File System

- **File Names**
  - Filenames can contain any normal text character including spaces and special characters.
  - Filenames can be almost any length.
  - It is best to stick to a-z, A-Z,_, -, and numbers.
  - If a filename contains a special character or a space you may need to put quotes around the whole path.

File System

- File Tree



File System

- File Types
  - Regular File : It comes under the Normal File category.
  - Directory : These are special types of files that are lists of other files.
  - Symbolic Link : A symbolic link is a reference to another file (a shortcut to any file).

File System

- File Types
  - Socket : Special type of file that provides inter-process networking protected by the file system's access control
  - Named Pipe : A special type of file that acts more or less like sockets and form a way for processes to communicate with each other, without using network socket semantics.
  - Device File : Character devices and Block devices

File System

- File commands
  - Chmod: changes a permission of a file
    - Permissions
      - u - User who owns the file.
      - g - Group that owns the file.
      - o - Other.
      - a - All.
      - r - Read the file.
      - w - Write or edit the file.
      - x - Execute or run the file as a program.

File System

- chmod
  - Numeric Permissions: CHMOD can also to attributed by using Numeric Permissions:
    - 400 read by owner
    - 040 read by group
    - 004 read by anybody (other)
    - 200 write by owner
    - 020 write by group
    - 002 write by anybody
    - 100 execute by owner
    - 010 execute by group
    - 001 execute by anybody

http://www.oreillynet.com/linux/cmd/cmd.csp?path=c/chmod

File System

- File commands
  - Chown : change file owner and group
    - chown [OPTION] [OWNER][:[GROUP]] FILE
  - Chgrp : changes the group that has access to a file or directory.
    - chgrp newgroup filenames

File System

- Umask
  - The User Mask
  - Who determines the default permissions when a new file is created?
  - Default permissions before applying mask are completely insecure:
    - rw-rw-rw (octal 666) for files
    - rwxrwxrwx (octal 777) for directories

File System

– Umask

- System default can be changed by umask command (a shell builtin).
- umask statement placed in a startup script (typically,
- /etc/profile or /etc/bashrc).
- Reassigns default file and directory permissions.

File System

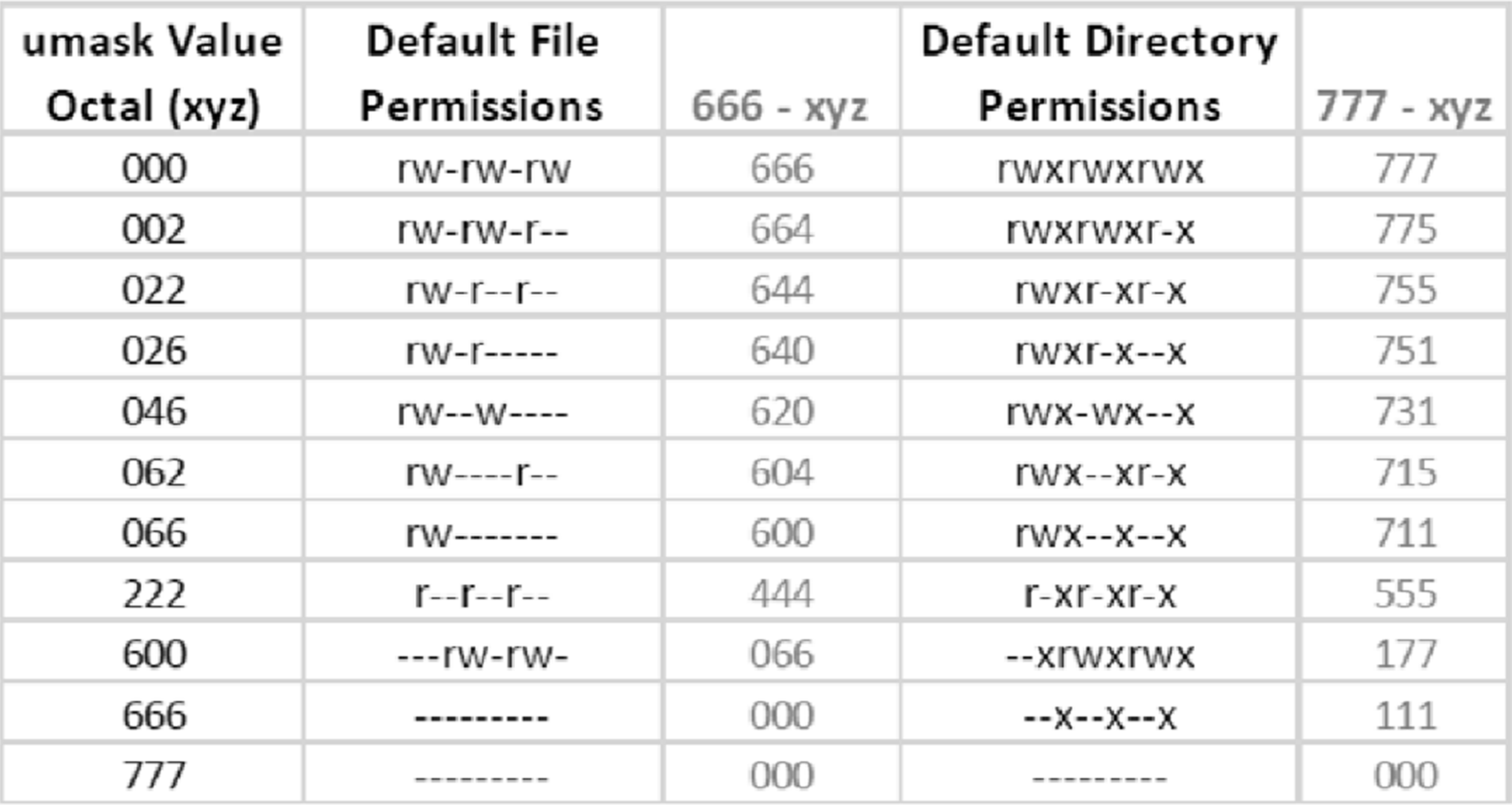
– Umask

Use umask w/o arguments to show your current permission setting

- Bash builtin shows 4 digit “0066” (inode actually stores 12 binary permission bits)
- /usr/bin/umask shows “0022”
- To change permission default setting:
- umask xyz (x, y, or z = octal digit)
- – x, y, or z doesn’t convert to binary meaning of rwx directly
- umask 0066

File System

– The User Mask Value Table



umask Value Octal (xyz)	Default File Permissions	666 - xyz	Default Directory Permissions	777 - xyz
000	rw-rw-rw	666	rwxrwxrwx	777
002	rw-rw-r--	664	rwxrwxr-x	775
022	rw-r--r--	644	rwxr-xr-x	755
026	rw-r-----	640	rwxr-x--x	751
046	rw--w----	620	rwx-wx--x	731
062	rw----r--	604	rwx--xr-x	715
066	rw-------	600	rwx--x--x	711
222	r--r--r--	444	r-xr-xr-x	555
600	---rw-rw-	066	--xrwxrwx	177
666	---------	000	--x--x--x	111
777	---------	000	---------	000

End of Section 3.0