

Linux essentials

Table of Contents

Introduction:	3
Terminal Window:	3
Absolute basic commands:	3
Linux Folder structure:	4
Linux files and file system:	7
List command:	7
df command:	9
The df command only displays information about active non-swap partitions:	9
Your home directory:	9
\$HOME env variable will show the home directory of the	9
cd ~ will take you to the home folder of the logged user:	9
Cat command	9
Device files:	10
File command:	11
Creating and deleting files and directories:	11
Find command:	12
Grep command:	13
view file content:	13
Linking files:	14
Creating symbolic links:	14
File security:	15
Changing user and group ownership:	16
I/O redirection:	16
Sorting while displaying the file contents:	17
Package formats:	18
What is RPM?:	18
DEB (.deb) packages:	18
Installing extra packages from the installation CDs:	19
Tarring and untarring compressed files:	19
Compressing and unpacking with gzip or bzip2:	19
Using the dd command to create a ISO file from CDROM:	19
Linux Networking:	20
Linux TCP/IP Network Configuration Files:	20
Fedora / Red Hat Network Configuration Files:	20
Static IP address assignment:	20
Ubuntu / Debian IP Configuration Files:	21

Network IP aliasing:.....	21
Changing the host name:	22
Network Classes:	23
Address Resolution:.....	23
IP Forwarding.....	24
Case study:.....	24

Knowledge labs

Introduction:

In order to work on a Linux system directly, you will need to provide a user name and password. You always need to authenticate to the system.

Linux systems have two basic modes for a system to run in: either quick and sober in text console mode, which looks like DOS with mouse, multitasking and multi-user features, or in graphical mode, which looks better but eats more system resources.

Terminal Window:

The terminal window is your control panel for the system. Almost everything that follows is done using this simple but powerful text tool. A terminal window should always show a command prompt when you open one. This terminal shows a standard prompt, which displays the user's login name, and the current working directory, represented by the twiddle (~):

Another common form for a prompt is this one:

```
[user@host dir]
```

In the above example, *user* will be your login name, *hosts* the name of the machine you are working on, and *dir* an indication of your current location in the file system.

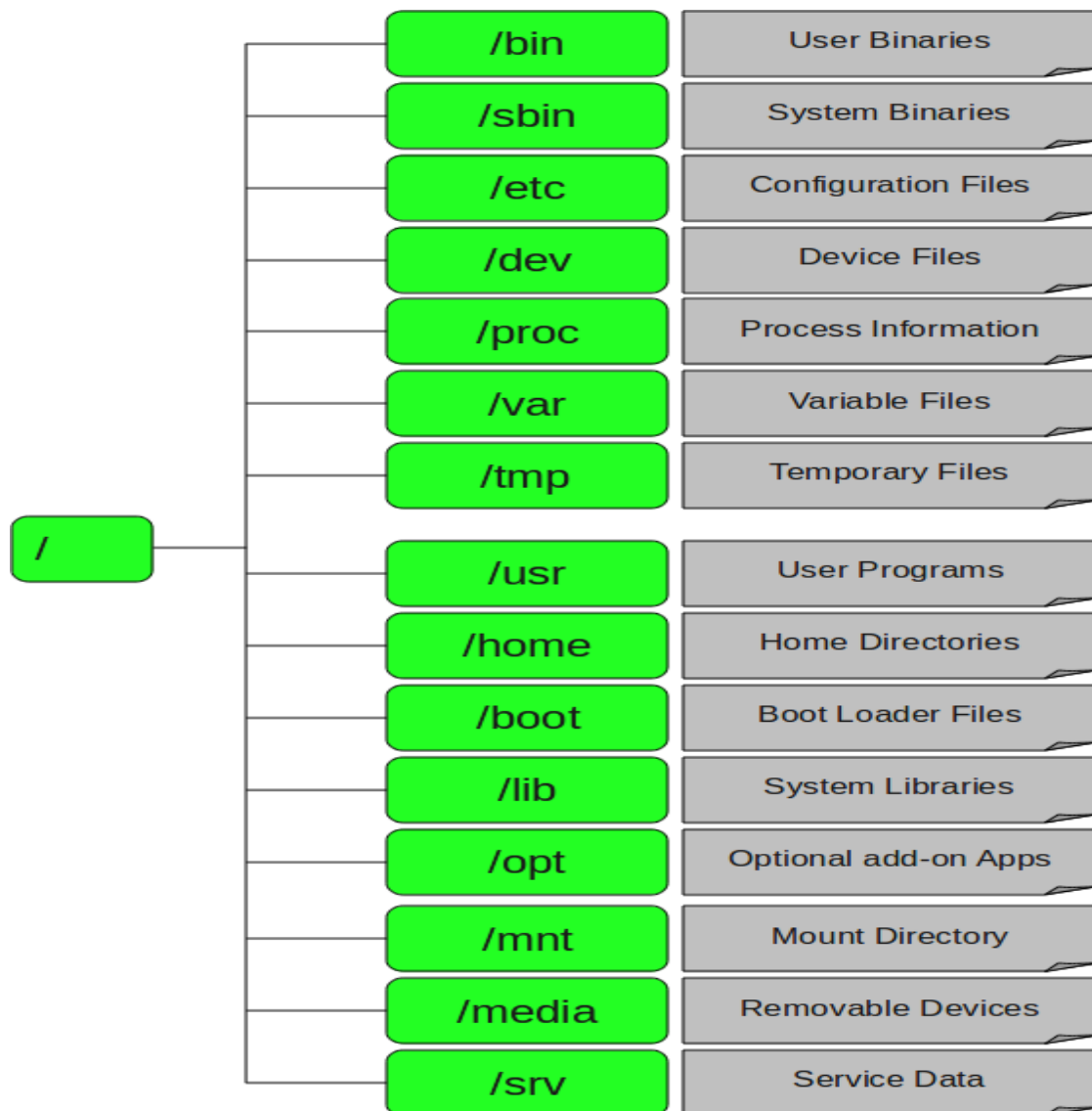
Absolute basic commands:

These are the quickies, which we need to get started

Command	Meaning
ls	Displays a list of files in the current working directory, like the dir command in DOS
cd directory	change directories
passwd	change the password for the current user
file filename	display file type of file with name filename
cat textfile	throws content of textfile on the screen
pwd	display present working directory
exit or logout	leave this session
man command	read man pages on command
info command	read Info pages on command
apropos string	search the whatis database for strings

- **--help**, which gives a short explanation about how to use the command
- **e.g cat --help**

Linux Folder structure:



1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, etc

4. /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

- Home directories for all users to store their personal files.
- For example: /home/bharath, /home/rajesh

11. /boot – Boot Loader Files

- Contains boot loader related files.
- Kernel initrd, vmlinuz, grub files are located under /boot
- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic
- 12. /lib – System Libraries
- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*
- For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

Linux files and file system

Everything you encounter on a Linux system is a file, still there are some *exceptions*

Directories: files that are lists of other files.

Special files: the mechanism used for input and output. Most special files are in /dev,

Links: a system to make a file or directory visible in multiple parts of the system's file tree.

(Domain) sockets: a special file type, similar to TCP/IP sockets, providing inter-process networking protected by the file system's access control.

Named pipes: act more or less like sockets and form a way for processes to communicate with each other, without using network socket semantics.

List command

The `-l` option to `ls` displays the file type, using the first character of each input line:

```
Last login: Sun Sep 30 18:27:55 2012 from anil
oneadmin@onehost:~$ ls -l
total 9032
drwxrwxr-x  2 oneadmin oneadmin  4096 Jul  19 00:52 bin
drwxrwxr-x  2 oneadmin oneadmin  4096 Sep 10 19:10 centos
drwxrwxr-x  2 oneadmin oneadmin  4096 Jun 22 00:24 debian
-rw-r--r--   1 oneadmin oneadmin 127380 Dec 17  2009 haizea-1.0.tar.gz
-rwxrwxr-x  1 oneadmin oneadmin  8183 Jun 22 13:56 poll
```

This table gives an overview of the characters determining the file type:

Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Special file
s	socket
p	Named Pipe
b	Block device

`ls -F`, which suffixes file names with one of the characters `"/=*|@"` to indicate the file type

Character	File type
nothing	regular file
/	directory
*	executable file
@	link
=	socket
	named pipe

```
Oneadmin@onehost:~$ ls -F
1/          ec2/          iso/          one-hyperv-3.2.0/    opennebula-3.6.0/
tmp/        ubuntu.net
2/          etc/          kvm.txt       one-hyperv-3.2.0.tar.gz  opennebula-3.6.0.tar.gz
ttylinux/   var/
bin/        haizea-1.0/   lib/          open-iscsi-2.0-873/  poll*
ttylinux1/
```

```
centos/  haizea-1.0.tar.gz  mountpoint/  open-iscsi-2.0-873.tar.gz  share/
ttylinux.tar.gz
debian/  images/          one/          opennebula-3.4.1/          template/
two/
debin    include/            one3.6/       opennebula-3.4.1.tar.gz    test1
ubuntu/
```

Here is list of most common colors: (RHEL and other linux distros)

- * Executable files: Green
- * Normal file : Normal
- * Directory: Blue
- * Symbolic link : Cyan
- * Pipe: Yellow
- * Socket: Magenta
- * Block device driver: Bold yellow foreground, with black background
- * Character device driver: Bold yellow foreground, with black background
- * Orphaned syminks : Blinking Bold white with red background
- * Missing links (- and the files they point to) : Blinking Bold white with red background
- * Archives or compressed : Red (.tar, .gz, .zip, .rpm)
- * Image files : Magenta (.jpg, gif, bmp, png, tif)

Comma separated display

```
oneadmin@onehost:~$ ls -m
1, 2, bin, centos, debian, debin, ec2, etc, haizea-1.0, haizea-1.0.tar.gz, images, include,
iso, kvm.txt, lib, mountpoint, one, one3.6, one-hyperv-3.2.0, one-hyperv-3.2.0.tar.gz,
open-iscsi-2.0-873, open-iscsi-2.0-873.tar.gz, opennebula-3.4.1, opennebula-3.4.1.tar.gz,
opennebula-3.6.0, opennebula-3.6.0.tar.gz, poll, share, template, test1, tmp, ttylinux,
ttylinux1, ttylinux.tar.gz, two, ubuntu, ubuntu.net, var
```

Sorting based on Create date/time

```
oneadmin@onehost:~$ ls -lSt
total 9032
drwxrwxr-x 180 oneadmin oneadmin 4096 Sep 30 23:57 var
drwxrwxr-x 2 oneadmin oneadmin 4096 Sep 28 23:01 template
drwxrwxr-x 2 oneadmin oneadmin 4096 Sep 27 00:21 iso
drwxrwxr-x 2 oneadmin oneadmin 4096 Sep 10 19:10 centos
drwxrwxr-x 2 oneadmin oneadmin 4096 Sep 10 16:53 images
-rw-rw-r-- 1 oneadmin oneadmin 0 Sep 9 21:08 kvm.txt
drwxrwxr-x 2 oneadmin oneadmin 4096 Sep 6 23:42 ubuntu
```

Sort based on Size

```
oneadmin@onehost:~$ ls -lSn
total 9032
-rw-rw-r-- 1 10000 10000 4906067 Jul 19 02:28 ttylinux.tar.gz
-rw-rw-r-- 1 10000 10000 1938228 Jul 12 20:24 opennebula-3.6.0.tar.gz
-rw-rw-r-- 1 10000 10000 1460564 Jun 20 21:51 opennebula-3.4.1.tar.gz
-rw-rw-r-- 1 10000 10000 667069 May 21 07:42 open-iscsi-2.0-873.tar.gz
-rw-r--r-- 1 10000 10000 127380 Dec 17 2009 haizea-1.0.tar.gz
-rw-rw-r-- 1 10000 10000 10395 Jul 18 05:27 one-hyperv-3.2.0.tar.gz
-rwxrwxr-x 1 10000 10000 8183 Jun 22 13:56 poll
drwxrwxr-x 3 10000 10000 4096 Jul 19 21:28 1
drwxrwxr-x 2 10000 10000 4096 Jul 19 21:28 2
```

Show files starting with . (all)

```
oneadmin@onehost:~$ ls -la
total 9120
drwxr-xr-x 38 oneadmin oneadmin 4096 Sep 27 00:13 .
drwxr-xr-x 67 root root 4096 Sep 27 00:15 ..
```



```
drwxrwxr-x   3 oneadmin oneadmin   4096 Jul 19 21:28 1
drwxrwxr-x   2 oneadmin oneadmin   4096 Jul 19 21:28 2
drwx-----   2 oneadmin oneadmin   4096 Jul 23 23:29 .aptitude
-rw-----   1 oneadmin oneadmin 13290 Sep 30 23:57 .bash_history
-rw-r--r--   1 oneadmin oneadmin   220 Apr  3 2012 .bash_logout
-rw-rw-r--   1 oneadmin oneadmin   542 Sep  5 22:14 .bash_profile
-rw-r--r--   1 oneadmin oneadmin 3486 Apr  3 2012 .bashrc
```

Print one entry per line of output

```
oneadmin@onehost:~$ ls -l
1
2
bin
centos
debian
debin
ec2
etc
```

ls */ - lists contents of the sub-directories

df command

The df command only displays information about active non-swap partitions

```
root@onehost:/var/lib/one# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       233G  145G   77G  66% /
udev            3.8G   12K   3.8G   1% /dev
tmpfs           1.6G  332K   1.6G   1% /run
none            5.0M     0   5.0M   0% /run/lock
none            3.8G     0   3.8G   0% /run/shm
cgroup          3.8G     0   3.8G   0% /sys/fs/cgroup
```

Your home directory:

\$HOME env variable will show the home directory of the

```
root@onehost:/var/lib/one# echo $HOME
/root
```

cd ~ will take you to the home folder of the logged user

```
oneadmin@onehost:~$ cd ~
oneadmin@onehost:~$ pwd
/var/lib/one
```

Cat command

File Content can be viewed using the **cat** command

```
root@onehost:/var/lib/one/template# cat system.ds
NAME      = ec2_ds
TM_MAD    = dummy
SYSTEM    = yesd
```

```

root@onehost:/var/lib/one/template# cat system.ds ds.conf
NAME      = ec2_ds
TM_MAD    = dummy
SYSTEM    = yes
NAME      = prod-iscsi-1
BASE_IQN="iqn.2012-02.org.opennebula"
DISK_TYPE="block"
DS_MAD="iscsi"
HOST="192.168.1.7"
TM_MAD="iscsi"
VG_NAME="vg-one"

```

Device files

Name	Device
cdrom	CD drive
console	Special entry for the currently used console.
cua*	Serial ports
dsp*	Devices for sampling and recording
fd*	Entries for most kinds of floppy drives, the default is /dev/fd0, a floppy drive for 1.44 MB floppies.
hd[a-t][1-16]	Standard support for IDE drives with maximum amount of partitions each.
ir*	Infrared devices
isdn*	Management of ISDN connections
js*	Joystick(s)
lp*	Printers
mem	Memory
midi*	midi player
mixer* and music	Idealized model of a mixer (combines or adds signals)
modem	Modem
mouse	(also msmouse, logimouse, psmouse,
input/mice, psaux)	All kinds of mouses
null	Bottomless garbage can
par*	Entries for parallel port support
pty*	Pseudo terminals
radio*	For Radio Amateurs (HAMs).
ram*	boot device
sd*	SCSI disks with their partitions
sequencer	For audio applications using the synthesizer features of
card (MIDI-device	controller)
tty*	Virtual consoles simulating vt100 terminals.
usb*	USB card and scanner
video*	For use with a graphics card supporting video.

e.g.

```

root@onehost:/var/lib/one/template# ls -l /dev
total 0
crw-----T 1 root root    10, 235 Oct  5 19:55 autofs
drwxr-xr-x 2 root root      640 Oct  5 19:55 block
drwxr-xr-x 2 root root       80 Oct  5 19:55 bsg
crw-----T 1 root root    10, 234 Oct  5 19:55 btrfs-control
drwxr-xr-x 3 root root       60 Oct  5 19:55 bus
lrwxrwxrwx 1 root root        3 Oct  5 19:55 cdrom -> sr0
lrwxrwxrwx 1 root root        3 Oct  5 19:55 cdrw -> sr0
drwxr-xr-x 2 root root    3360 Oct  5 19:55 char
crw----- 1 root root        5,   1 Oct  5 19:55 console
lrwxrwxrwx 1 root root       11 Oct  5 19:55 core -> /proc/kcore

```

```
drwxr-xr-x 2 root root          60 Oct  5 19:55 cpu
crw----- 1 root root        10,  60 Oct  5 19:55 cpu_dma_latency
drwxr-xr-x 5 root root       100 Oct  5 19:55 disk
lrwxrwxrwx 1 root root          3 Oct  5 19:55 dvd -> sr0
lrwxrwxrwx 1 root root          3 Oct  5 19:55 dvdrw -> sr0
crw----- 1 root root        10,  61 Oct  5 19:55 ecryptfs
```

File command:

Results with type of the object

```
root@onehost:/var/lib/one/template# file ~
/root: directory

oneadmin@onehost:~/template$ file one-149
one-149: ASCII text

oneadmin@onehost:~/template$ file image.png
image.png: PNG image data, 616 x 862, 8-bit grayscale, non-interlaced
```

Creating and deleting files and directories

```
$mkdir <dir_name> # creates specified directory
$ cd <dir_name> # switches into specified directory
$ cd .. # moves one directory up
$ cd ../../ # moves two directories up (and so on)
$ cd # brings you to highest level of your home directory
$ rmdir <dir_name> # removes empty directory
$ rm <file_name> # removes file name
$ rm -r <dir_name> # removes directory including its content, but asks for confirmation,
'f' argument turns confirmation off
$ mv <name1> <name2> # renames directories or files
$ mv <name> <path> # moves file/directory as specified in path
$ cp <name> <path> # copy file/directory as specified in path (-r to include content in
directories)
```

```
oneadmin@onehost:~/template$ mkdir myfolder

oneadmin@onehost:~/template$ ls -ld myfolder/
drwxrwxr-x 2 oneadmin oneadmin 4096 Oct  5 21:12 myfolder/

oneadmin@onehost:~/template$ rmdir myfolder/

oneadmin@onehost:~/template$ mkdir myfolder
oneadmin@onehost:~/template$ cd myfolder/
oneadmin@onehost:~/template/myfolder$ touch myzerosizefile
oneadmin@onehost:~/template/myfolder$ ls
myzerosizefile

oneadmin@onehost:~/template/myfolder$ cd ..
oneadmin@onehost:~/template$ mkdir my2ndfolder
oneadmin@onehost:~/template$ mv myfolder/myzerosizefile my2ndfolder/
oneadmin@onehost:~/template$ ls my2ndfolder/
myzerosizefile

oneadmin@onehost:~/template$ cp my2ndfolder/myzerosizefile myfolder/
oneadmin@onehost:~/template$ ls myfolder/
```

```
myzerosizefile
```

```
oneadmin@onehost:~/template$ rmdir myfolder/  
rmdir: failed to remove `myfolder/': Directory not empty
```

```
oneadmin@onehost:~/template$ rm -rf myfolder/  
oneadmin@onehost:~/template$ ls myfolder/  
ls: cannot access myfolder/: No such file or directory
```

Find command

```
Finding files, directories and applications  
$ find -name "*pattern*" # searches for *pattern* in and below current directory  
$ find /usr/local -name "*blast*" # finds file names *blast* in specfied directory  
$ find /usr/local -iname "*blast*" # same as above, but case insensitive  
additional useful arguments: -user <user name>, -group <group name>, -ctime <number of days  
ago changed>  
$ find ~ -type f -mtime -2 # finds all files you have modified in the last two days  
$ locate <pattern> # finds files and dirs that are written into update file  
$ which <application_name> # location of application  
$ whereis <application_name> # searches for executeables in set of directories  
$ dpkg -l | grep mypattern # find Debian packages and refine search with grep pattern
```

```
oneadmin@onehost:~/template$ find ~ -name myzerosizefile 2>/dev/null  
/var/lib/one/template/my2ndfolder/myzerosizefile
```

```
oneadmin@onehost:~/template$ which find  
/usr/bin/find
```

```
oneadmin@onehost:~/template$ whereis find  
find: /usr/bin/find /usr/bin/X11/find /usr/share/man/man1/find.1.gz
```

```
oneadmin@onehost:~/template$ dpkg -l | grep whois  
ii  whois          5.0.15ubuntu2      intelligent WHOIS client
```

```
oneadmin@onehost:~/iso$ ls -l && find -size +50000k  
total 16042932  
-rw-r--r-- 1 libvirt-qemu kvm      4423155712 Jul 22 00:03 centos63-dvd1.iso  
-rw-r--r-- 1 oneadmin    oneadmin 4185128960 Jun 21 23:03 centos.iso  
-rw-rw-r-- 1 oneadmin    oneadmin 677380096 Jun 21 22:20 debian-6.0.5-amd64-CD-1.iso  
-rw-r--r-- 1 oneadmin    oneadmin 726827008 Sep  5 01:27 ubuntu1010.iso  
-rw-r--r-- 1 libvirt-qemu kvm      672694272 Sep  5 21:20 ubuntu1010server.iso  
-rw-rw-r-- 1 libvirt-qemu kvm      717533184 Jul 24 10:45 ubuntu-12.04-server-amd64.iso  
-rw-rw-r-- 1 oneadmin    oneadmin 731164672 Jun 21 21:52 ubuntu.iso  
-rw-rw-r-- 1 libvirt-qemu kvm      45762560 Jul 15 21:39 virtio-win-0.1-30.iso  
-rw-rw-r-- 1 oneadmin    oneadmin 29574892 Sep 27 00:13 virtio-win-1.1.16-  
0.el6.noarch.rpm  
-rw-r--r-- 1 root        root      31719420 Sep 27 00:16 virtio-win_1.1.16-1_all.deb  
-rw-r--r-- 1 libvirt-qemu kvm      50112512 Sep 27 00:21 virtio-win-1.1.16.iso  
-rw-r--r-- 1 oneadmin    oneadmin 3529609216 Jul 14 01:00 win2008.iso  
-rw-rw-r-- 1 libvirt-qemu kvm      607250432 Sep 26 20:30 Win_XP_original.iso  
./ubuntu-12.04-server-amd64.iso  
./debian-6.0.5-amd64-CD-1.iso  
./Win_XP_original.iso  
./ubuntu1010.iso  
./ubuntu1010server.iso  
./ubuntu.iso  
./centos63-dvd1.iso  
./win2008.iso  
./centos.iso
```

```

oneadmin@onehost:~/iso$ ls ~/template/my2ndfolder/myzerosizefile
/var/lib/one/template/my2ndfolder/myzerosizefile
oneadmin@onehost:~/iso$ find ~ -name "myzerosizefile" -exec rm {} \;
oneadmin@onehost:~/iso$ ls ~/template/my2ndfolder/myzerosizefile
ls: cannot access /var/lib/one/template/my2ndfolder/myzerosizefile: No such file or
directory

oneadmin@onehost:~/iso$ locate whoami
/usr/bin/whoami
/usr/lib/byobu/whoami
/usr/share/man/man1/whoami.1.gz

```

Grep command

Used to filter

```

oneadmin@onehost:~/iso$ grep MAD ~/etc/oned.conf
IM_MAD = [
IM_MAD = [
#IM_MAD = [
#IM_MAD = [
IM_MAD = [
#IM_MAD = [
#IM_MAD = [ name="im_dummy", executable="one_im_dummy"]
#Hyper-v VMM_MAD
VM_MAD = [

oneadmin@onehost:~/iso$ cat ~/etc/oned.conf | grep MAD
IM_MAD = [
IM_MAD = [
#IM_MAD = [
#IM_MAD = [
IM_MAD = [
#IM_MAD = [
#IM_MAD = [ name="im_dummy", executable="one_im_dummy"]
#Hyper-v VMM_MAD

```

view file content

Head command – lists first 10 lines from top

```

oneadmin@onehost:~/iso$ head ~/etc/oned.conf
# *****
#
#                               OpenNebula Configuration file
# *****

# *****
# Daemon configuration attributes
# -----
#   MANAGER_TIMER: Time in seconds the core uses to evaluate periodical functions.
#   HOST_MONITORING_INTERVAL and VM_POLLING_INTERVAL can not have smaller values
#   than MANAGER_TIMER.

```

Tail comm and – lists 10 lines from bottom

```

oneadmin@onehost:~/iso$ tail ~/etc/oned.conf
# *****
# The following attributes are restricted to users outside the oneadmin group
# *****

```

```
VM_RESTRICTED_ATTR = "CONTEXT/FILES"
VM_RESTRICTED_ATTR = "NIC/MAC"
VM_RESTRICTED_ATTR = "NIC/VLAN_ID"
VM_RESTRICTED_ATTR = "RANK"

IMAGE_RESTRICTED_ATTR = "SOURCE"

oneadmin@onehost:~/iso$ tail -3 ~/etc/oned.conf
VM_RESTRICTED_ATTR = "RANK"

IMAGE_RESTRICTED_ATTR = "SOURCE"
```

Exercise:

Try **head -n 100** and **tail -n 100**

Also try **tail -f**

Linking files

A link is nothing more than a way of matching two or more file names to the same set of file data. There are two ways to achieve this

Hard link: Associate two or more file names with the same inode. Hard links share the same data blocks on the hard disk, while they continue to behave as independent files.

There is an immediate disadvantage: hard links can't span partitions, because inode numbers are only unique within a given partition.

Soft link or symbolic link (or for short: symlink): a small file that is a pointer to another file. A symbolic link contains the path to the target file instead of a physical location on the hard disk. Since inodes are not used in this system, soft links can span across partitions.

Creating symbolic links

The symbolic link is particularly interesting for beginning users: they are fairly obvious to see and you don't need to worry about partitions

ln -s targetfile linkname

```
oneadmin@onehost:~/iso$ cd ~/template/my2ndfolder/
oneadmin@onehost:~/template/my2ndfolder$ ls
oneadmin@onehost:~/template/my2ndfolder$ touch my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ echo "file content" > my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ cat my2ndfile
file content
oneadmin@onehost:~/template/my2ndfolder$ ls
my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ ln -s ~/template/my2ndfolder/my2ndfile
linktomyfile
oneadmin@onehost:~/template/my2ndfolder$ ls -l
total 4
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-r-- 1 oneadmin oneadmin 13 Oct  5 21:50 my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ cat linktomyfile
file content
```

File security

Access mode codes

Code	Meaning
0 or -	The access right that is supposed to be on this place is not granted.
4 or r	read access is granted to the user category defined in this place
2 or w	write permission is granted to the user category defined in this
1 or x	execute permission is granted to the user category defined in this place
3	exercise
5	exercise
6	exercise
7	exercise

User group codes

Code	Meaning
u	user permissions
g	group permissions
o	permissions for others

Creating a new group

e.g creating a group gid [10000]and a name [cloud]

```
sudo groupadd -g 10000 cloud
```

Creating a new user with uid 10000 , name oneadmin, home folder /srv/cloud/one and adding the user to an existing group [cloud]

```
mkdir /srv/cloud/one  
sudo useradd -u 10000 -m oneadmin -d /srv/cloud/one -s /bin/bash -g cloud
```

Set a password for the user oneadmin

```
sudo passwd oneadmin
```

Changing ownership of a folder and with a group

```
sudo chown -R oneadmin:cloud /srv/cloud/
```

Changing user

```
su -l oneadmin
```

```
oneadmin@onehost:~/template/my2ndfolder$ ls -l my2ndfile
```

```
-rw-rw-r-- 1 oneadmin oneadmin 13 Oct  5 21:50 my2ndfile

oneadmin@onehost:~/template/my2ndfolder$ chmod o+w my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ ls -l
total 4
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-rw- 1 oneadmin oneadmin 13 Oct  5 21:50 my2ndfile
```

Exercise :

1. try other mode changes
2. What does **chmod 600** results
3. what does **chmod 076** results

Changing user and group ownership

The **chown** command can be applied to change both user and group ownership of a file, while **chgrp** only changes group ownership.

Syntax: **chown newuser file**

```
neadmin@onehost:~/template/my2ndfolder$ ls -l
total 4
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-rw- 1 oneadmin oneadmin 13 Oct  5 21:50 my2ndfile

oneadmin@onehost:~/template/my2ndfolder$ sudo chown oneadmin:root my2ndfile
oneadmin@onehost:~/template/my2ndfolder$ ls -l
total 4
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-rw- 1 oneadmin root 13 Oct  5 21:50 my2ndfile
```

Exercise : Try chgrp

I/O redirection

Sometimes you will want to put output of a command in a file, or you may want to issue another command on the output of one command. This is known as redirecting output. Redirection is done using either the ">" (greater-than symbol), or using the "|" (pipe) operator which sends the standard output of one command to another command as standard input.

```
oneadmin@onehost:~/template/my2ndfolder$ cat my2ndfile
file content

oneadmin@onehost:~/template/my2ndfolder$ cat my2ndfile > my3rdfile
oneadmin@onehost:~/template/my2ndfolder$ ls -l
total 8
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-rw- 1 oneadmin root 13 Oct  5 21:50 my2ndfile
-rw-rw-r-- 1 oneadmin oneadmin 13 Oct  5 22:15 my3rdfile

oneadmin@onehost:~/template/my2ndfolder$ cat my3rdfile
file content

oneadmin@onehost:~/template/my2ndfolder$ cat my2ndfile my3rdfile > my4tfile
oneadmin@onehost:~/template/my2ndfolder$ cat my4tfile
file content
```



```

file content

oneadmin@onehost:~/template/my2ndfolder$ echo "this is a new line" >> my4tfile
oneadmin@onehost:~/template/my2ndfolder$ cat my4tfile
file content
file content
this is a new line

oneadmin@onehost:~/template/my2ndfolder$ > newfile
oneadmin@onehost:~/template/my2ndfolder$ ls -l
total 12
lrwxrwxrwx 1 oneadmin oneadmin 43 Oct  5 21:53 linktomyfile ->
/var/lib/one/template/my2ndfolder/my2ndfile
-rw-rw-rw- 1 oneadmin root      13 Oct  5 21:50 my2ndfile
-rw-rw-r-- 1 oneadmin oneadmin 13 Oct  5 22:15 my3rdfile
-rw-rw-r-- 1 oneadmin oneadmin 64 Oct  5 22:17 my4tfile
-rw-rw-r-- 1 oneadmin oneadmin  0 Oct  5 22:17 newfile

Filter for "content" and omit "new"
oneadmin@onehost:~/template/my2ndfolder$ grep content my4tfile | grep -v new
file content
file content

```

Exercise : try “less“ command

There are three types of I/O, which each have their own identifier, called a file descriptor:

- standard input: 0
- standard output: 1
- standard error: 2

In the following descriptions, if the file descriptor number is omitted, and the first character of the redirection operator is <, the redirection refers to the standard input (file descriptor 0). If the first character of the redirection operator is >, the redirection refers to the standard output (file descriptor 1).

Some practical examples will make this more clear:

ls > dirlist 2>&1

will direct both standard output and standard error to the file dirlist, while the command

ls 2>&1 > dirlist

will only direct standard output to dirlist. This can be a useful option for programmers.

Exercise : see what happens when you redirect the output of a find command to 2>/dev/null

Sorting while displaying the file contents

```

oneadmin@onehost:~/template/my2ndfolder$ cat my4tfile | sort
file content
file content
this is a new line
this is a new line
oneadmin@onehost:~/template/my2ndfolder$ echo "000 abc" >> my4tfile
oneadmin@onehost:~/template/my2ndfolder$ echo "z00 abc" >> my4tfile
oneadmin@onehost:~/template/my2ndfolder$ cat my4tfile
file content
file content
this is a new line
this is a new line
000 abc
z00 abc
oneadmin@onehost:~/template/my2ndfolder$ cat my4tfile | sort
000 abc
file content
file content
this is a new line

```

```
this is a new line
z00 abc
```

Changing the prompt: Try it out

```
oneadmin@onehost:~/template/my2ndfolder$ export PS1="[\t \j] "
[22:29:48 1] export PS1="[\d][\u@\h \w] : "
[Fri Oct 05][oneadmin@onehost ~/template/my2ndfolder] :
```

Package formats

What is RPM?

RPM, the RedHat Package Manager, is a powerful package manager that you can use to install, update and remove packages

syntax **rpm -Uvh /path/to/rpm-package(s)**

e.g. **rpm -Uvh totem-0.99.5-1.fr.i386.rpm**

you can find out which package contains a certain file or executable

```
rpm -qf /etc/profile
```

```
rpm -qf /bin/cat
```

Below is one last example, demonstrating how to uninstall a package using rpm:

```
rpm -e totem
```

Note that uninstalling is not that verbose by default, it is normal that you don't see much happening. When in

doubt, use **rpm -qa** again to verify that the package has been removed

DEB (.deb) packages

What are Debian packages :This package format is the default on Debian GNU/Linux

Checking whether a package is installed is done using the dpkg command. For instance, if you want to know which version of the Gallery software is installed on your machine:

```
dpkg -l *gallery*
```

Searching which package a file belongs to is done using the -S to dpkg

```
dpkg -S /bin/cat
```

Installing a debian package

syntax : apt-get install <package-name> or aptitude install <package-name>

update a system packages

```
apt-get update
```

Upgrade a system

```
apt-get upgrade
```

Installing extra packages from the installation CDs

1. Mount a CDROM

`mount /dev/cdrom /mnt/cdrom`

2. For automation purposes, the CD drive usually has an entry in `/etc/fstab`, which lists the file systems and their mount points, that make up your file system tree. This is such a line

`grep cdrom /etc/fstab`

This indicates that the system will understand the command `mount /mnt/cdrom`. The `noauto` option means that on this system, CDs are not mounted at boot time

3. `mount | grep cdrom`

4. once done eject the CD **Ejecting the CD**

`umount /mnt/cdrom`

Tarring and untarring compressed files

e.g. compress a folder to a tar file

`tar cvf images-in-a-dir.tar images/`

Decompress it:

`tar xvf images-in-a-dir.tar`

`exercise try : tar tvf images/images-without-dir.tar`

Compressing and unpacking with gzip or bzip2

e.g. `gzip images-without-dir.tar`

Using the dd command to create a ISO file from CDROM

`dd if=/dev/cdrom of=~/cdrom_image.iso`

Linux Networking

IP [Internet Protocol] requires a hardware-independent addressing scheme. This is achieved by assigning each host a unique 32-bit number called the *IP address*. An IP address is usually written as four decimal numbers, one for each 8-bit portion, separated by dots

Linux TCP/IP Network Configuration Files:

File	Description
/etc/resolv.conf	List DNS servers for internet domain name resolution. Manual page for: /etc/ resolv.conf
/etc/hosts	Lists hosts to be resolved locally (not by DNS). Manual page for: /etc/ hosts
Red Hat/Fedora/CentOS: /etc/sysconfig/network	Specify network configuration. eg. Static IP, DHCP, NIS, etc.
Red Hat/Fedora/CentOS: /etc/sysconfig/network-scripts/ifcfg-device	Specify TCP network information.
Ubuntu/Debian: /etc/network/interfaces	Specify network configuration and devices. eg. Static IP and info, DHCP, etc.

Fedora / Red Hat Network Configuration Files:

/etc/sysconfig/network

Red Hat network configuration file used by the system during the boot process.

- File: /etc/sysconfig/network-scripts/ifcfg-eth0

Configuration settings for your first ethernet port (0). Your second port is eth1.

Static IP address assignment:

Command Line:

```
/sbin/ifconfig eth0 192.168.10.12 netmask 255.255.255.0 broadcast 192.168.10.255
```

- Network address by convention would be the lowest: 192.168.10.0

Broadcast address by convention would be the highest: 192.168.10.255

The gateway can be anything, but following convention: 192.168.10.1

- Note: the highest and lowest addresses are based on the netmask. The previous example is based on a netmask of 255.255.255.0

Ubuntu / Debian IP Configuration Files:

File: /etc/network/interfaces

Static IP example:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 208.88.34.106
    netmask 255.255.255.248
    broadcast 208.88.34.111
    network 208.88.34.104
    gateway 208.88.34.110
```

Dynamic IP (DHCP) example:

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet dhcp

auto eth2
iface eth2 inet dhcp

auto ath0
iface ath0 inet dhcp

auto wlan0
iface wlan0 inet dhcp
```

Interfaces:

- lo: Loopback interface (network within your system without slowing down for the real ethernet based network)
- eth0: First ethernet interface card
- wlan0: First wireless network interface

Network IP aliasing:

Assign more than one IP address to one ethernet card:

```
ifconfig eth0 XXX.XXX.XXX.XXX netmask 255.255.255.0 broadcast XXX.XXX.XXX.255
```

```
ifconfig eth0:0 192.168.10.12 netmask 255.255.255.0 broadcast 192.168.10.255
ifconfig eth0:1 192.168.10.14 netmask 255.255.255.0 broadcast 192.168.10.255
```

route add -host XXX.XXX.XXX.XXX dev eth0

```
route add -host 192.168.10.12 dev eth0
route add -host 192.168.10.14 dev eth0
```

In this example 0 and 1 are aliases in addition to the regular eth0. The result of the ifconfig command:

```
eth0      Link encap:Ethernet  HWaddr 00:10:4C:25:7A:3F
inet addr:XXX.XXX.XXX.XXX  Bcast:XXX.XXX.XXX.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:14218 errors:0 dropped:0 overruns:0 frame:0
TX packets:1362 errors:0 dropped:0 overruns:0 carrier:0
collisions:1 txqueuelen:100
Interrupt:5 Base address:0xe400

eth0:0    Link encap:Ethernet  HWaddr 00:10:4C:25:7A:3F
inet addr:192.168.10.12  Bcast:192.168.10.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
Interrupt:5 Base address:0xe400

eth0:1    Link encap:Ethernet  HWaddr 00:10:4C:25:7A:3F
inet addr:192.168.10.14  Bcast:192.168.10.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
Interrupt:5 Base address:0xe400
```

Config file: /etc/sysconfig/network-scripts/ifcfg-eth0:0

```
DEVICE=eth0:0
ONBOOT=yes
BOOTPROTO=static
BROADCAST=192.168.10.255
IPADDR=192.168.10.12
NETMASK=255.255.255.0
NETWORK=192.168.10.0
ONBOOT=yes
```

Aliases can also be shut down independently. i.e.: `ifdown eth0:0`

Changing the host name:

This is a three step process:

1. Issue the command:

`hostname new-host-name`

2. Change network configuration file: /etc/sysconfig/network

Edit entry:

`HOSTNAME=new-host-name`

3. Restart systems which relied on the hostname (or reboot):

- Restart network services: `service network restart`
(or: `/etc/init.d/network restart`)

Network Classes:

The concept of network classes is a little obsolete as subnets are now used to define smaller networks using CIDR (Classless Inter-Domain Routing) as detailed above. These subnets may be part of a class A, B, C, etc network. For historical reference the network classes are defined as follows:

- Class A:** Defined by the first 8 bits with a range of 0 - 127.
First number (8 bits) is defined by Internic i.e. 77.XXX.XXX.XXX
One class A network can define 1.6 million hosts per network.
Range: 0.0.0.0 - 127.255.255.255
- Class B:** Defined by the first 16 bits with a range from 128 - 191
First two numbers (16 bits) are defined by Internic i.e. 182.56.XXX.XXX
One class B network can define 65,534 hosts.
Range: 128.0.0.0 - 191.255.255.255
- Class C:** Defined by the first 24 bits with a range from 192 - 223
First three numbers (24 bits) are defined by Internic i.e. 220.56.222.XXX
This class allows for nearly 2 million networks with up to 254 hosts
Range: 192.0.0.0 - 223.255.255.255
- Class D:** Defined by the first 8 bits with a range from 224 - 239
This is reserved for multicast networks (RFC988)
Range: 224.0.0.0 – 239.255.255.255
- This class allows for nearly 2 million networks with up to 254 hosts
- Class E:** Defined by the first 8 bits with a range from 240 - 255
This is reserved for experimental use.
Range: 240.0.0.0 – 255.255.255.255

Address Resolution:

A mechanism is needed to map IP addresses onto the addresses of the underlying network. The mechanism used is the *Address Resolution Protocol* (ARP).

When ARP wants to find the Ethernet address corresponding to a given IP address, it uses an Ethernet feature called *broadcasting*, in which a datagram is addressed to all stations on the network simultaneously. The broadcast datagram sent by ARP contains a query for the IP address. Each receiving host compares this query to its own IP address and if it matches, returns an ARP reply to the inquiring host. The inquiring host can now extract the sender's Ethernet address from the reply.

By default Class C subnet mask is 255.255.255.0 = 24 bits for network id and 8 bits for host id.
in Binary

1111 1111. 1111 1111. 1111 1111. 0000 0000

Here all 1s are Network bits and all 0s are host bits. For this subnet mask you can have 256 hosts. And you can use 254 host and assign IP address to them.

```
-----
Host Address    192.168.110.23
Network Mask    255.255.255.0
Network Portion 192.168.110.
Host portion    .23
-----
Network Address 192.168.110.0
Broadcast Address 192.168.110.255
```

IP Forwarding

Forwarding allows the network packets on one network interface (i.e. eth0) to be forwarded to another network interface (i.e. eth1). This will allow the Linux computer to connect ("ethernet bridge") or route network traffic.

Method:

alter the Linux kernel config file:

/etc/sysctl.conf

Set the following value

```
net.ipv4.ip_forward = 1
```

See file /etc/sysconfig/network for storing this configuration.

```
FORWARD_IPV4=true
```

All methods will result in a proc file value of "1". Test: `cat /proc/sys/net/ipv4/ip_forward`

Case study:



OneHost network configuration:

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.1.100
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
# dns-* options are implemented by the resolvconf package, if installed
dns-search example.com
# Set IP address for eth1 post installation
```



```
# Private n/w interface
auto eth1
iface eth1 inet static
address 192.168.2.1
netmask 255.255.255.0
sudo /etc/init.d/networking restart
```

Cross check the contents of /etc/resolve.conf of OneHost

```
search example.com
nameserver 192.168.1.1
```

Setting up OneHost as a “Router” for VMHost

edit /etc/sysctl.conf and UN-comment the below line

```
net.ipv4.ip_forward = 1
```

Edit /etc/rc.local and add the following lines just above the “exit 0” line and restart OneHost machine

```
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
iptables -t nat -I POSTROUTING -s 192.168.2.0/24 -d 192.168.1.0/8 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```

```
sudo reboot
```

VMHost Network Configuration:

```
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet manual
auto br0
iface br0 inet static
address 192.168.2.2
netmask 255.255.255.0
network 192.168.2.0
broadcast 192.168.2.255
gateway 192.168.2.1
# dns-* options are implemented by the resolvconf package, if installed
dns-search example.com
bridge_ports eth0
bridge_fd 9
bridge_hello 2
bridge_maxage 12
bridge_stp off

sudo /etc/init.d/networking restart
```

Edit /etc/resolv.conf of VMHost

```
search example.com
nameserver 192.168.1.1
```

Now try to ping 192.168.1.1 from vmhost. It should ping. Also try connecting to internet from Vmhost