LINUX BASIC COMMANDS

1) pwd command

‘pwd’ command prints the absolute path to current working directory.

$ pwd  
/home/raghu

2) cal command

Displays the calendar of the current month.

$ cal  
July 2012  
Su Mo Tu We Th Fr Sa  
1 2 3 4 5 6 7  
8 9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30 31

‘cal ’ will display calendar for the specified month and year.

$ cal 08 1991  
August 1991  
Su Mo Tu We Th Fr Sa  
1 2 3  
4 5 6 7 8 9 10  
11 12 13 14 15 16 17  
18 19 20 21 22 23 24  
25 26 27 28 29 30 31

3) echo command

This command will echo whatever you provide it.

$ echo "linoxide.com"  
 linoxide.com

#### 4) date command

Displays current time and date.

$ date  
Fri Jul 6 01:07:09 IST 2012

If you are interested only in time, you can use 'date +%T' (in hh:mm:ss):

$ date +%T  
 01:13:14

#### 5) whoami command

This command reveals the user who is currently logged in.

$ whoami  
raghu

#### 6) id command

This command prints user and groups (UID and GID) of the current user.

$ id  
uid=1000(raghu) gid=1000(raghu) groups=1000(raghu),4(adm),20(dialout),24(cdrom),46(plugdev),112(lpadmin),120(admin),122(sambashare)

By default, information about the current user is displayed. If another username is provided as an argument, information about that user will be printed:

$ id root  
uid=0(root) gid=0(root) groups=0(root)

#### 7) clear command

This command clears the screen.

#### 8) Changing Directories Command

$ cd [path-to-directory]

Change the current working directory to the directory provided as argument. If no argument is given to ‘cd’, it changes the directory to the user's home directory. The directory path can be an absolute path or relative to current directory. The absolute path always starts with /. The current directory can be checked with ‘pwd’ command (remember?):

$ pwd  
/home/raghu  
$ cd /usr/share/  
$ pwd  
/usr/share  
$ cd doc  
$ pwd  
/usr/share/doc

In the first ‘cd’ command, absolute path (/usr/share) is used, and with second command, relative path (doc) is used.

#### 9) Listing File And Directories Command

$ ls [files-or-directories]

List files and/or directories. If no argument is given, the contents of current directory are shown.

$ ls  
example file1.txt file2.txt file3.txt

If a directory is given as an argument, files and directories in that directory are shown.

$ ls /usr  
bin games include lib lib64 local sbin share src

‘ls -l’ displays a long listing of the files.

$ ls -l  
total 4  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 12:52 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file1.txt

-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file2.txt  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file3.txt

In this long listing, the first character is 'd' or '-'. It distinguishes between file types. The entries with a '-' (dash) are regular files, and ones with 'd' are directories. The next 9 characters are permissions ('rwxr-xr-x' in first listing). The number following the permissions is the link count. Link count follows user and group owner. In the above example, the file owner is 'raghu' and group owner is 'raghu' as well. Next is the size of the file. And then time stamp before the name of file (or directory).  
By default, hidden files or directories are not shown, to see hidden files as well, -a option is used. Hidden files in Linux start with a period sign (.). Any file that starts with a period is hidden. So, to hide a file, you just need to rename it (and put a period before it).

$ ls -la odesk  
total 16  
drwxr-xr-x 4 raghu raghu 4096 2012-07-06 13:46 .  
drwxr-xr-x 11 raghu raghu 4096 2012-07-06 13:15 ..  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 12:52 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file1.txt  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file2.txt  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 12:52 file3.txt  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 13:46 .hiddendir  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 13:46 .hiddenfile1.txt  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 13:46 .hiddenfile2.txt

If you want to see the properties of a directory instead of the files contained in it, use -d (with -l) option:

$ ls -ld odesk/  
drwxr-xr-x 4 raghu raghu 4096 2012-07-06 13:46 odesk/

### Creating files and directories Command

#### 10) mkdir command

To create a directory, the ‘mkdir’ command is used.

$ mkdir example  
$ ls -l  
total 4  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example

#### 11) touch command

For creating an empty and multiple files, use the touch command.

$ touch file1 file2 file3  
$ ls -l  
total 4  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3

If a file already exists, touch will update its time stamp. There are a lot of other methods to create a new file, e.g. using a text editor like vi or gedit, or using redirection. Here is an example of creating a file using redirection:

$ ls -l /usr > usrlisting  
$ ls -l  
total 8  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 14:23 usrlisting

A file named usrlisting is created in this example.

## Copy, move and remove commands

#### 12) copy command

$cp source destination

Copy files and directories. If the source is a file, and the destination (file) name does not exit, then source is copied with new name i.e. with the name provided as the destination.

$ cp usrlisting listing\_copy.txt  
$ ls -l  
total 12  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 14:09 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 16:02 listing\_copy.txt  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 14:23 usrlisting

If the destination is a directory, then the file is copied with its original name in that directory.

$ cp listing\_copy.txt example/  
$ ls -l example/  
total 4  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 16:07 listing\_copy.txt

Multiple files can also be copied, but in that case, the last argument will be expected to be a directory where all the files are to be copied. And the rest of the arguments will be treated as file names.

$ cp file1 file2 example/  
$ ls -l example/  
total 4  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 16:10 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 16:10 file2  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 16:07 listing\_copy.txt

If a directory is to be copied, then it must be copied recursively with the files contained in it. To copy a directory recursively, use -r option with ‘cp’ command:

$ cp -r example /tmp/expertslogin/  
$ ls -l /tmp/expertslogin  
total 4  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 16:12 example

#### 13) move command

$ mv source destination

Move files or directories. The 'mv' command works like 'cp' command, except that the original file is removed. But, the mv command can be used to rename the files (or directories).

$ mv listing\_copy.txt usrcopy  
$ ls -l  
total 12  
drwxr-xr-x 2 raghu raghu 4096 2012-07-06 16:10 example  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file2  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 16:02 usrcopy  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 14:23 usrlisting

Here, 'listing\_copy.txt' is moved with the name 'usrcopy' in the same directory (or you can say that it has been renamed).

#### 14) To remove or Delete

$ rmdir

'rmdir' command removes any empty directories, but cannot delete a directory if a file is present in it. To use ‘rmdir’ command, you must first remove all the files present in the directory you wish to remove (and possibly directories if any).

#### To remove files and directories

$ rm files|directories

A directory must be removed recursively with -r option.

$ rm file2  
$ rm -r example/  
$ ls -l  
total 8  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file1  
-rw-r--r-- 1 raghu raghu 0 2012-07-06 14:20 file3  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 16:02 usrcopy  
-rw-r--r-- 1 raghu raghu 491 2012-07-06 14:23 usrlisting

Here, the file named 'file2' is removed first, and then the directory 'example' is removed recursively. This can be seen in the output of ‘ls -l’ command where these two are no longer present.

#### cat command

The 'cat' command is actually a concatenator but can be used to view the contents of a file.

$ cat /etc/passwd  
root:x:0:0:root:/root:/bin/bash  
daemon:x:1:1:daemon:/usr/sbin:/bin/sh  
bin:x:2:2:bin:/bin:/bin/sh  
sys:x:3:3:sys:/dev:/bin/sh  
sync:x:4:65534:sync:/bin:/bin/sync  
games:x:5:60:games:/usr/games:/bin/sh