Experiment 7

Aim: Directory oriented commands: cd, pwd, mkdir, rmdir

cd command in Linux with Examples

cd command in linux known as change directory command. It is used to change current working directory.

Syntax:

```
$ cd [directory]
```

To move inside a subdirectory : to move inside a subdirectory in linux we use

```
$ cd [directory_name]
```

Different functionalities of cd command:

- **cd /:** this command is used to change directory to the root directory, The root directory is the first directory in your file system hierarchy.

```
$ cd /
```

Above, / represents the root directory.

- **cd dir_1/dir_2/dir_3:** This command is used to move inside a directory from a directory

```
$ cd dir_1/dir_2/dir_3
```

In above example, we have the document directory and inside the document directory we have a directory named geeksforgeeks and inside that directory we have example directory.

To navigate example directory we have used command cd

Documents/geeksforgeeks/example.

- **cd ~ :** this command is used to change directory to the home directory.

```
$ cd ~
```

or

```
$ cd
```

cd : this commad also work same as cd ~ command.

- **cd .. :** This command is used to move to the parent directory of current directory, or the directory one level up from the current directory. ".." represents parent directory.

```
$ cd ..
```

- **cd "dir name":** This command is used to navigate to a directory with white spaces. Instead of using double quotes we can use single quotes then also this command will work.

```
$ cd "dir name"
```

In above example, we have navigated the My songs directory by using cd "My songs" command.

or

```
$ cd dir\ name :
```

This command work same as cd "dir name" command.

pwd command in Linux with Examples

pwd stands for **P**rint **W**orking **D**irectory. It prints the path of the working directory, starting from the root.
pwd is shell built-in command(pwd) or an actual binary(/bin/pwd).
$PWD is an environment variable which stores the path of the current directory.

This command has two flags.

***pwd -L**: Prints the symbolic path.*
***pwd -P**: Prints the actual path.*

mkdir command in Linux with Examples

mkdir command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories. It is important to note that the user executing this command must have enough permission to create a directory in the parent directory, or he/she may recieve a 'permission denied' error.

Syntax:

```
mkdir [options...] [directories ...]
```

How to create directories using mkdir?

Creating directories is pretty simple; all you need to do is to pass the name of the directory you want to create to the mkdir command.

mkdir [dir-name]

Following is an example:

```
mkdir test-dir
```

Q2. How to make sure parent directories (if non-existent) are created in process?

Sometimes the requirement is to create a complete directory structure with a single mkdir command. This is possible, but you'll have to use the **-p** command line option.
For example, if you want to create dir1/dir2/dir2 when none of these directories are already existing, then you can do this in the following way:

```
mkdir -p dir1/dir2/dir3
```

```
himanshu@ansh:~/htf-daily/dir$ ls
himanshu@ansh:~/htf-daily/dir$ mkdir -p dir1/dir2/dir3
himanshu@ansh:~/htf-daily/dir$ ls -R
.:
dir1

./dir1:
dir2

./dir1/dir2:
dir3

./dir1/dir2/dir3:
```

Q3. How to set permissions for directory being created?

By default, the mkdir command sets rwx, rwx, and r-x permissions for the directories created through it.

```
himanshu@ansh:~/htf-daily$ mkdir testdir
himanshu@ansh:~/htf-daily$ stat testdir/
  File: 'testdir/'
  Size: 4096            Blocks: 8          IO Block: 4096   directory
Device: 807h/2055d      Inode: 5636223     Links: 2
Access: (0775/drwxrwxr-x)  Uid: ( 1000/himanshu)   Gid: ( 1000/himanshu)
Access: 2018-01-12 16:39:57.435890991 +0530
Modify: 2018-01-12 16:39:57.435890991 +0530
Change: 2018-01-12 16:39:57.435890991 +0530
 Birth: -
```

However, if you want, you can set custom permissions using the **-m** command line option.

```
himanshu@ansh:~/htf-daily$ mkdir -m 777 testdir
himanshu@ansh:~/htf-daily$ stat testdir/
  File: 'testdir/'
  Size: 4096            Blocks: 8          IO Block: 4096   directory
Device: 807h/2055d      Inode: 5636223     Links: 2
Access: (0777/drwxrwxrwx)  Uid: ( 1000/himanshu)   Gid: ( 1000/himanshu)
Access: 2018-01-12 16:40:49.153494367 +0530
Modify: 2018-01-12 16:40:49.153494367 +0530
Change: 2018-01-12 16:40:49.153494367 +0530
 Birth: -
```

rmdir command in Linux With Examples

rmdir command is used remove empty directories from the filesystem in Linux. The rmdir command removes each and every directory specified in the command line only if these directories are empty. So if the specified directory has some directories or files in it then this cannot be removed by *rmdir* command.

Example 1: This will first remove the child directory and then remove the parent directory.

```
rmdir -p mydir/mydir1
```

```
rahul@rahul-SVF15318SNB:~/Desktop/linux$ mkdir mydir
rahul@rahul-SVF15318SNB:~/Desktop/linux$ cd mydir/
rahul@rahul-SVF15318SNB:~/Desktop/linux/mydir$ mkdir mydir1
rahul@rahul-SVF15318SNB:~/Desktop/linux/mydir$ cd ..
rahul@rahul-SVF15318SNB:~/Desktop/linux$ rmdir -p mydir/mydir1/
rahul@rahul-SVF15318SNB:~/Desktop/linux$ ls
rahul@rahul-SVF15318SNB:~/Desktop/linux$
```

Example 2: Remove the directories *mydir1*, *mydir2*, and *mydir3*, if they are empty. If any of these directories are not empty, then an error message will be printed for that directory, and the other directories will be removed.

```
rmdir mydir1 mydir2 mydir3
```

Example 3: Remove the directory *mydir/mydir1* if it is empty. Then, remove directory mydir, if it is empty after *mydir/mydir1* was removed.

```
rmdir mydir/mydir1 mydir
```

Recursive delete

Remove all directories and subdirectories, use rm command with -r option: `rm -rf /home/data/2000`

What is the difference between rmdir and rm -r?

If you remember, you can also delete directories using the rm command by enabling the -r option it provides. So what's the difference between that and rmdir? Well, the answer is rmdir only works in the case of empty directories - there's no way whatsoever you can use to make rmdir delete non-empty directories.
So rmdir is a useful in tool in those situations where you otherwise need to check if a directory is empty before deleting it.