



# Linux Exercise 1

**Author:** Himanshu Chhetri  
**Date:** 30-Apr-2013 16:44  
**URL:** <https://nebula.addteq.com/display/cm/Linux+Exercise+1>



# Table of Contents

---



1.

- a. Use the `df` command to display the amount of used and available space on your hard drive.

```
df | awk '{print $2 " " $3}'
```

- b. Check the man page for `df`, and use it to find an option to the command which will display the free space in a more human-friendly form. Try both the single-letter and long-style options.

```
df -h  
df --human-readable
```

2.

- a. Use the `find` command to list all the files and directories under your home directory. Try the `-type d` and `-type f` criteria to show just files and just directories.

```
find ~/   
find ~/ -type f  
find ~/ -type d
```

- b. Use `locate` to find files whose name contains the string 'bashbug'. Try the same search with `find`, looking over all files on the system. You'll need to use the `*` wildcard at the end of the pattern to match files with extensions.

```
find ~/ -type f -exec grep "bashbug" {} \;  
locate bashrc | xargs grep -i "bashbug"
```

- c. Find out what the `find` option `-iname` does.

```
Like -name, but the match is case insensitive
```

3.

- a. Use the `hostname` command, with no options, to print the hostname of the machine you are using.

```
hostname
```

- b. Use `man` to display some documentation on the `hostname` command. Find out how to make it print the IP address of the machine instead of the hostname. You will need to scroll down the manpage to the 'Options' section.

```
hostname -i
```



c. Use the locate command to find files whose name contains the text 'hostname'. Which of the filenames printed contain the actual hostname program itself? Try running it by entering the program's absolute path to check that you really have found it.

```
locate "*hostname*"
file /bin/hostname
```

4.

a. The \* wildcard on its own is expanded by the shell to a list of all the files in the current directory. Use the echo command to see the result (but make sure you are in a directory with a few files or directories first)

```
echo *
```

b. Use quoting to make echo print out an actual \* symbol.

```
echo "*"
or
echo \*
```