**Linux Commands**

**alias**

Create an alias, aliases allow a string to be substituted for a word when it is used as the first word of a simple command.

**Syntax**

alias [-p] [name[=value] ...]

unalias [-a] [name ... ]

**Options**

-p Print the current values

-a Remove All aliases

If arguments are supplied, an alias is defined for each name whose value is given.

If no value is given, alias will print the current value of the alias.  
  
Without arguments or with the -p option, alias prints the list of aliases on the standard output in a form that allows them to be reused as input.

The value cannot contain any positional parameters ($1 etc), if you need to do that use a shell [function](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\function.html) instead.

**apt-get**

Search for and install software packages (Debian/Ubuntu).

**Syntax**

apt-get [-sqdyfmubV] [-o= config\_string ] [-c= config\_file ]

[-t= target] [options…]

**Options**

apt-get and [aptitude](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\aptitude.html) now share the same list of installed packages and so can be used interchangeably.

All command line options may be set using the configuration file, the descriptions indicate the configuration option to set. For boolean options you can override the config file by using something like -f-,--no-f, -f=no or several other variations.

--no-install-recommends

Do not consider recommended packages as a dependency for

installing. Configuration Item: APT::Install-Recommends.

-d, --download-only

Download only; package files are only retrieved, not unpacked or

installed. Configuration Item: APT::Get::Download-Only.

-f, --fix-broken

Fix; attempt to correct a system with broken dependencies in place.

--no-download

Disables downloading of packages. This is best used with

--ignore-missing to force APT to use only the .debs it has already

downloaded. Configuration Item: APT::Get::Download.

-u, --show-upgraded

Show upgraded packages; Print out a list of all packages that are

to be upgraded. Configuration Item: APT::Get::Show-Upgraded.

**aspell**

Spellcheck a file.

**Syntax**

aspell check [options] filename

**Options**

-mode=mode

The mode to use when checking files.

The available modes are none, url, email, sgml, tex, texinfo, nroff and any

others available on your system. [more..](http://aspell.net/man-html/Notes-on-Various-Filters-and-Filter-Modes.html)

–dont-backup

Don't create a backup file. Normally, if there are any corrections the Aspell utility

will append .bak to the existing file name and then create a new file with corrections

made during spell checking.

-l name

The language the document is written in. The default depends on the current locale.

**awk or gawk (gnu awk)**

Find and Replace text, database sort/validate/index

**Syntax**

awk <options> 'Program' Input-File1 Input-File2 ...

awk -f PROGRAM-FILE <options> Input-File1 Input-File2 ...

**Basic functions**  
  
The basic function of awk is to search files for lines (or other units of text) that contain a pattern. When a line matches, awk performs a specific action on that line.

The Program statement that tells awk what to do; consists of a series of "rules". Each rule specifies one pattern to search for, and one action to perform when that pattern is found.

**basename**

Strip directory and suffix from filenames

**Syntax**

basename NAME [SUFFIX]

basename OPTION

**Options**

--help

Display help

--version

Output version information and exit

basename will print NAME with any leading directory components removed. If specified, it will also remove a trailing SUFFIX (typically a file extention).

**bc**

An arbitrary precision calculator language

**Syntax**

bc options file...

**Options**:

-h, --help

Print the usage and exit.

file A file containing the calculations/functions to perform.

May be piped from standard input

-i, --interactive

Force interactive mode.

-l, --mathlib

Define the standard math library.

-w, --warn

Give warnings for extensions to POSIX bc.

-s, --standard

Process exactly the POSIX bc language.

-q, --quiet

Do not print the normal GNU bc welcome.

-v, --version

Print the version number and copyright and quit.

**bg**

Send job to background

**Syntax**

bg [PID...]

**Options**:

If PID is specified, the jobs with the specified group ids are put in the background.

Send the specified jobs to the background. A background job is executed simultaneously with fish, and does not have access to the keyboard. If no job is specified, the last job to be used is put in the background.

**break**

Exit from a for, while, until, or select loop

**Syntax**

break [n]

If n is supplied, the nth enclosing loop is exited. n must be greater than or equal to 1.   
  
The return status is zero unless n is not greater than or equal to 1.

**cal**

Display a calendar

**Syntax**

cal [-mjy] [[month] year]

**Options**:

-m Display monday as the first day of the week.

-j Display julian dates (days one-based, numbered from January 1).

-y Display a calendar for the current year.

A single parameter specifies the 4 digit year (1 - 9999) to be displayed.

Two parameters denote the Month (1 - 12) and Year (1 - 9999).

If arguments are not specified, the current month is displayed.

A year starts on 01 Jan.

**cat**

Concatenate and print (display) the content of files.

**Syntax**

cat [Options] [File]...

Concatenate FILE(s), or standard input, to standard output.

**Options:**

-A, --show-all equivalent to -vET

-b, --number-nonblank number nonblank output lines

-e equivalent to -vE

-E, --show-ends display $ at end of each line

-n, --number number all output lines

-s, --squeeze-blank never more than one single blank line

-t equivalent to -vT

-T, --show-tabs display TAB characters as ^I

-u (ignored)

-v, --show-nonprinting use ^ and M- notation, except for LFD and TAB

--help display this help and exit

--version output version information and exit

With no FILE, or when FILE is -, read standard input.

**cd**

Change Directory - change the current working directory to a specific Folder.

**Syntax**

cd [Options] [Directory]

**Options**

-P Do not follow symbolic links

-L Follow symbolic links (default)

If directory is not given, the value of the HOME shell variable is used.   
  
If the shell variable CDPATH exists, it is used as a search path.   
If directory begins with a slash, CDPATH is not used.   
  
If directory is `-', this will change to the previous directory location (equivalent to $OLDPWD ).   
  
The return status is zero if the directory is successfully changed, non-zero otherwise.

**clear**

clear the terminal screen

**Syntax**

clear

**Description**

clear clears your screen if this is possible. It looks in the environment for the terminal type and then in the terminfo database to figure out how to clear the screen. clear ignores any command-line parameters that may be present.

**chmod**

Change access permissions, change mode.

**Syntax**

chmod [Options]... Mode [,Mode]... file...

chmod [Options]... Numeric\_Mode file...

chmod [Options]... --reference=RFile file...

**Options**

-f, --silent, --quiet suppress most error messages

-v, --verbose output a diagnostic for every file processed

-c, --changes like verbose but report only when a change is made

--reference=RFile use RFile's mode instead of MODE values

-R, --recursive change files and directories recursively

--help display help and exit

--version output version information and exit

|  |  |
| --- | --- |
| Permission | letter |
| Read | r |
| Write | w |
| Execute (or access for directories) | x |
| Execute only if the file is a directory  (or already has execute permission for some user) | X |
| Set user or group ID on execution | s |
| Save program text on swap device | t |
| The permissions that the User who owns the file currently has for it | u |
| The permissions that other users in the file's Group have for it | g |
| Permissions that Other users not in the file's group have for it | o |

**cmp**

Compare two files, and if they differ, tells the first byte and line number where they differ.  
  
You can use the `cmp' command to show the offsets and line numbers where two files differ. `cmp' can also show all the characters that differ between the two files, side by side.

**Syntax**

cmp options... FromFile [ToFile]

**Options**

-c

Print the differing characters.

--ignore-initial=BYTES

Ignore any differences in the the first BYTES bytes of the input

files. Treat files with fewer than BYTES bytes as if they are

empty.

--print-chars

Print the differing characters. Display control characters as a

`^' followed by a letter of the alphabet and precede characters

that have the high bit set with `M-' (which stands for "meta").

--silent

Do not print anything; only return an exit status indicating

whether the files differ.

--verbose

Print the (decimal) offsets and (octal) values of all differing

bytes.

--version

Output the version number of `cmp'.

**comm**

Common - compare two sorted files line by line and write to standard output:   
the lines that are common, plus the lines that are unique.

**Syntax**

comm [options]... File1 File2

**Options**

-1 suppress lines unique to file1

-2 suppress lines unique to file2

-3 suppress lines that appear in both files

With no options, `comm' produces three column output. Column one contains lines unique to FILE1, column two contains lines unique to FILE2, and column three contains lines common to both files. Columns are separated by a single TAB character.  
  
Unlike some other comparison utilities, `comm' has an exit status that does not depend on the result of the comparison. Upon normal completion `comm' produces an exit code of zero. If there is an error it exits with nonzero status.

**cp**

Copy one or more files to another location  
  
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

**Syntax**

cp [options]... Source Dest

cp [options]... Source... Directory

**Options**

-a, --archive same as -dpR

-b, --backup make backup before removal

-d, --no-dereference preserve links

-f, --force remove existing destinations, never prompt

-i, --interactive prompt before overwrite

-l, --link link files instead of copying

-p, --preserve preserve file attributes if possible

-P, --parents append source path to DIRECTORY

-r copy recursively, non-directories as files

--sparse=WHEN control creation of sparse files

-R, --recursive copy directories recursively

-s, --symbolic-link make symbolic links instead of copying

-S, --suffix=SUFFIX override the usual backup suffix

-u, --update copy only when the SOURCE file is newer

than the destination file or when the

destination file is missing

-v, --verbose explain what is being done

-V, --version-control=WORD override the usual version control

-x, --one-file-system stay on this file system

--help display this help and exit

--version output version information and exit.

**cut**

Divide a file into several parts (columns)  
Writes to standard output selected parts of each line of each input file, or standard input if no files are given or for a file name of `-'.

**Syntax**

cut [OPTION]... [FILE]...

**Options**

--bytes=BYTE-LIST

Print only the bytes in positions listed in BYTE-LIST. Tabs and

backspaces are treated like any other character; they take up 1

byte.

--characters=CHARACTER-LIST

Print only characters in positions listed in CHARACTER-LIST. The

same as `-b' for now, but internationalization will change that.

Tabs and backspaces are treated like any other character; they

take up 1 character.

-n

Do not split multi-byte characters (no-op for now).

-s

--only-delimited

For `-f', do not print lines that do not contain the field

separator character.

--output-delimiter=OUTPUT\_DELIM\_STRING

For `-f', output fields are separated by OUTPUT\_DELIM\_STRING The

default is to use the input delimiter.

**date**

Display or change the date.

**Syntax**

date [option]... [+Format]

date [option] [MMDDhhmm[[CC]YY][.ss]]

'date' with no arguments prints the current time and date, in the format of the %c directive (described below).

**Options**

-d, --date=String Display time described by String, instead of 'now'

this can be in almost any common format.

It can contain month names, timezones, 'am' and 'pm',

'yesterday', 'ago', 'next', etc.

-r, --reference=File Display the last modification time of File

-s, --set=String Set time described by String (see -d above)

-u, --utc, --universal Print or set Coordinated Universal Time

--help Display this help and exit

--version output version information and exit

Here are the same format codes in alphabetical order:

%% a literal %

%a locale's abbreviated weekday name (Sun..Sat)

%A locale's full weekday name, variable length (Sunday..Saturday)

%b locale's abbreviated month name (Jan..Dec)

%B locale's full month name, variable length (January..December)

%c locale's date and time (Sat Nov 04 12:02:33 EST 1989)

%d day of month (01..31)

%D date (mm/dd/yy)

%e day of month, blank padded ( 1..31)

%h same as %b, locale's abbreviated month name (Jan..Dec)

%H hour :24 hour(00..23)

%I hour :12 hour(01..12)

%j day of year (001..366)

%k hour :24 hour(00..23)

%l hour :12 hour(01..12)

%m month (01..12)

%M minute (00..59)

%n a newline

%p locale's AM or PM

%r Time, 12-hour (hh:mm:ss [AP]M)

%s Seconds since 1970-01-01 00:00:00, (a GNU extension)

Note that this value is defined by the localtime system

call. It isn't changed by the '--date' option.

%S second (00..60)

%t a horizontal tab

%T Time, 24-hour (hh:mm:ss)

%U Week number of year with Sunday as first day of week (00..53)

%V Week number of year with Monday as first day of week (01..53)

If the week containing January 1 has four or

more days in the new year, then it is considered week 1;

otherwise, it is week 53 of the previous year, and the next week

is week 1. Similar to [ISO 8601](file:///C:\Users\Harshvardhan\Desktop\ \dates.html) (but not 100% compliant.)

%w day of week (0..6); 0 represents Sunday

%W week number of year with Monday as first day of week (00..53)

%x locale's date representation (mm/dd/yy)

%X locale's time representation (%H:%M:%S)

%y last two digits of year (00..99)

%Y year (1970...)

**dc**

Desk calculator

**Syntax**

dc <options>

**Options**

-e EXPR

--expression=EXPR

Evaluate EXPR as DC commands.

-f FILE

--file=FILE

Read and evaluate DC commands from FILE.

-h

--help

Print a usage message summarizing the command-line options, then

exit.

-V

--version

Print the version information for this program, then exit.

To exit, use `q'.

**df**

Disk Free - display free disk space.  
With no arguments, `df' reports the space used and available on all currently mounted filesystems (of all types). Otherwise, `df' reports on the filesystem containing each argument file.

**Syntax**

df [option]... [file]...

Normally the disk space is printed in units of 1024 bytes, but this

can be overridden.

**Options**

-a

--all

Include in the listing filesystems that have a size of 0 blocks,

which are omitted by default. Such filesystems are typically

special-purpose pseudo-filesystems, such as automounter entries.

Also, filesystems of type "ignore" or "auto", supported by some

operating systems, are only included if this option is specified.

-h

--human-readable

Append a size letter such as `M' for megabytes to each size.

Powers of 1024 are used, not 1000; `M' stands for 1,048,576 bytes.

Use the `-H' or `--si' option if you prefer powers of 1000.

-H

--si

Append a size letter such as `M' for megabytes to each size. (SI

is the International System of Units, which defines these letters

as prefixes.) Powers of 1000 are used, not 1024; `M' stands for

1,000,000 bytes. Use the `-h' or `--human-readable' option if you

prefer powers of 1024.

**diff**

Display the differences between two files, or each corresponding file in two directories.   
Each set of differences is called a "diff" or "patch". For files that are identical, diff normally produces no output; for binary (non-text) files, diff normally reports only that they are different.

**Syntax**

diff [options] FILES

**Options**

-a

--text

Treat all files as text.

-b

--ignore-space-change

Ignore changes in the amount of white space.

-B

--ignore-blank-lines

Ignore changes whose lines are all blank.

--help

Output this help.

-i

--ignore-case

Ignore case differences in file contents.

-r

--recursive

Recursively compare any subdirectories found.

**dir**

Briefly list directory contents

**Syntax**

`dir' (also installed as `d')

Equivalent to `ls -C -b'; that is,by default files are listed in columns, sorted vertically, and special characters are represented by backslash escape sequences.

**du**

Disk Usage - report the amount of disk space used by the specified files and for each subdirectory.

**Syntax**

du [options]... [file]...

With no arguments, `du' reports the disk space for the current directory. Normally the disk space is printed in units of 1024 bytes, but this can be overridden

**Options**

-a

--all

Show counts for all files, not just directories.

-b

--bytes

Print sizes in bytes, overriding the default block size (\*note

Block size::).

-c

--total

Print a grand total of all arguments after all arguments have been

processed. This can be used to find out the total disk usage of a

given set of files or directories.

-D

--dereference-args

Dereference symbolic links that are command line arguments. Does

not affect other symbolic links. This is helpful for finding out

the disk usage of directories, such as `/usr/tmp', which are often

symbolic links.

-h

--human-readable

Append a size letter such as `M' for megabytes to each size.

Powers of 1024 are used, not 1000; `M' stands for 1,048,576 bytes.

Use the `-H' or `--si' option if you prefer powers of 1000.

**echo**

Display message on screen, writes each given STRING to standard output, with a space between each and a newline after the last one.

**Syntax**

echo [options]... [string]...

**Options**

-n

Do not output the trailing newline.

-E

Disable the interpretation of the following backslash-escaped characters

-e

Enable interpretation of the following backslash-escaped

characters in each STRING:

\a alert (bell)

\b backspace

\c suppress trailing newline

\e escape   
   
 \f form feed

\n new line

\r carriage return

\t horizontal tab

\v vertical tab

\\ backslash

\NNN

the character whose ASCII code is NNN (octal); if NNN is not

a valid octal number, it is printed literally.

\xnnn  
 the character whose ASCII code is the hexadecimal value   
 nnn (one to three digits)

**eject**

Eject removable media

**Syntax**

eject -h

eject [-vnrsfmqp] [<name>]

**Options**

-h --help  
 -v --verbose  
 -d --default  
 -a --auto  
 -x --cdspeed  
 -X --listspeed  
 -n --noop  
 -s --scsi  
 -V --version  
 -m --no-unmount

**exec**

Execute a command

**Syntax**

exec [-cl] [-a name] [command [arguments]]

**Options**

-l Place a dash at the beginning of the zeroth arg passed to command.

(This is what the login program does.)

-c Causes command to be executed with an empty environment.

-a The shell passes name as the zeroth argument to command.

**exit**

Exit from a program, shell or log out of a Unix network.

**Syntax**

exit

If supported will exit you from the program, shell or log you out of network.

If exit does not log you out you can also do logout, lo, bye, quit, also Ctrl-D may work.

**expr**

Evaluate expressions, evaluates an expression and writes the result on standard output.

**Syntax**

expr expression...

usual manner (you must quote parentheses to avoid the shell evaluating them, however).

**Examples**

Here are a few examples, including quoting for shell metacharacters.

To add 1 to the shell variable `foo', in Bourne-compatible shells:

foo=`expr $foo + 1`

To print the non-directory part of the file name stored in `$fname', which need not contain a `/'.

expr $fname : '.\*/\(^.\*\)' '^|' $fname

An example showing that `\+' is an operator:

expr aaa : 'a\+'

=> 3

expr abc : 'a\(.\)c'

=> b

expr index abcdef cz

=> 3

expr index index a

error--> expr: Syntax error

expr index quote index a

=>

**fold**

Wrap input lines to fit in specified width, writes each FILE (`-' means standard input), or standard input if none are given, to standard output, breaking long lines.

**Syntax**

fold [options]... [FILE]...

By default, `fold' breaks lines wider than 80 columns. The output is split into as many lines as necessary.  
  
`fold' counts screen columns by default; thus, a tab may count more than one column, backspace decreases the column count, and carriage return sets the column to zero.

**Options**

-b

--bytes

Count bytes rather than columns, so that tabs, backspaces, and

carriage returns are each counted as taking up one column, just

like other characters.

-s

--spaces

Break at word boundaries: the line is broken after the last blank

before the maximum line length. If the line contains no such

blanks, the line is broken at the maximum line length as usual.

-w WIDTH

--width=WIDTH

Use a maximum line length of WIDTH columns instead of 80.

**Examples**:

$ function longlist() { ls -l $1; }

**free**

Display amount of free and used memory in the system

**Syntax**

free [-b|-k|-m|-g] [-c count] [-l] [-o] [-t] [-s delay] [-V]

**Options**

-b

Display the amount of memory in bytes.

-c count

Display the result count times. Requires the -s option.

-g

Display the amount of memory in gigabytes.

-k

Display the amount of memory in kilobytes. This is the default.

-l

Show detailed low and high memory statistics.

-m

Display the amount of memory in megabytes.

-o

Display the output in old format, the only difference being this option will disable the display of the "buffer adjusted" line.

-t

Display a line showing the column totals.

**format**

Format a string in the style of sprintf

**Syntax**

format formatString ?arg arg ...?

Introduction

This command generates a formatted string in a fashion similar to the ANSI C sprintf procedure. FormatString indicates how to format the result, using % conversion specifiers as in sprintf, and the additional arguments, if any, provide values to be substituted into the result.

**Options**

0

Specifies that the number should be padded on the left with zeroes instead of spaces.

#

Requests an alternate output form

d

Convert integer to signed decimal string.

i

Convert integer to signed decimal string (equivalent to d).

o

Convert integer to unsigned octal string.

x or X

Convert integer to unsigned hexadecimal string, using digits “0123456789abcdef” for x and “0123456789ABCDEF” for X).

c

Convert integer to the Unicode character it represents.

s

No conversion; just insert string.

%

No conversion: just insert %.

**fuser**

Identify processes using files or sockets, optionally: Kill the process that is accessing the file.

**Syntax**

fuser [-a|-s|-c] [-4|-6] [-n space ] [-k [-i] [-signal ] ] [-muvf] name

fuser -l

fuser -V

**Options**:

-a Show all files specified on the command line.

By default, only files that are accessed by at least one process are shown.

-c Same as -m option, used for POSIX compatibility.

-f Silently ignored, used for POSIX compatibility.

-k Kill processes accessing the file.

Unless changed with -signal, SIGKILL is sent.

An fuser process never kills itself, but may kill other fuser processes.

The effective user ID of the process executing fuser is set to its real

user ID before attempting to kill.

-i Ask the user for confirmation before killing a process.

This option is silently ignored if -k (kill) is not present too.

-l List all known signal names.

**grep**

Search file(s) for specific text.

**Syntax**

grep [options] PATTERN [FILE...]

grep [options] [-e PATTERN | -f FILE] [FILE...]

A simple example:

$ grep "Needle in a Haystack" /etc/\*

**Options**

-A NUM

--after-context=NUM

Print NUM lines of trailing context after matching lines.

Places a line containing -- between contiguous groups of matches.

-B NUM

--before-context=NUM

Print NUM lines of leading context before matching lines.

Places a line containing -- between contiguous groups of matches.

-C NUM

-b

--byte-offset

Print the byte offset within the input file before each line of output.

-c

--count

Suppress normal output; instead print a count of matching lines for each input file.

With the -v, --invert-match option (see below), count non-matching lines.

-f FILE

--file=FILE

Obtain patterns from FILE, one per line. The empty file contains zero

patterns, and therefore matches nothing.

-i

--ignore-case

Ignore case distinctions in both the PATTERN and the input files.

-o

--only-matching

Show only the part of a matching line that matches PATTERN.

**groupadd**

Create a new group.

**Syntax**

groupadd [options] group

Creates a new group account using the values specified on the command line plus the default values from the system. The new group will be entered into the system files as needed.

**Options**

-g, --gid GID

The numerical value of the group's ID.

-h, --help

Display help message and exit.

-o, --non-unique

This option permits to add a group with a non-unique GID.

-p, --password PASSWORD

The encrypted password, as returned by crypt(3).

**groupdel**

Delete a user security group.

**Syntax**

groupdel group

groupdel modifies the system account files, deleting all entries that refer to group. The named group must exist.

**Exit Values**

The groupdel command exits with the following values:

0 success

2 invalid command syntax

3 invalid argument to option

6 specified group doesn't exist

8 can't remove user's primary group

10 can't update group file

**groupmod**

Modify a user security group.

**Syntax**

groupmod [options] GROUP

groupmod modifies the definition of the specified GROUP by modifying the appropriate entry in the group database.

**Options**

-g, --gid GID

The group ID of the given GROUP will be changed to GID.

-h, --help

Display help message and exit.

-n, --new-name NEW\_GROUP

The name of the group will be changed from GROUP to NEW\_GROUP.

-o, --non-unique

When used with the -g option, allow to change the

group GID to a non-unique value.

-p, --password PASSWORD

The encrypted password, as returned by crypt(3).

**Exit Values**

The groupmod command exits with the following values:

0 Success

2 Invalid command syntax

3 Invalid argument to option

4 Specified group doesn't exist

6 Specified group doesn't exist

9 Group name already in use

10 Can't update group file

**groups**

Print group names a user is in.

**Syntax**

groups [username]...

Prints the names of the primary and any supplementary groups for each given username, or the current process if no names are given.

If names are given, the name of each user is printed before the list of that user's groups.

**gzip**

Compress or decompress named file(s)

**Syntax**

gzip options ...

**Options**

-c

Write output on standard output; keep original files unchanged.

If there are several input files, the output consists of a

sequence of independently compressed members. To obtain better

compression, concatenate all input files before compressing them.

-d

Decompress.

-f

Force compression or decompression even if the file has multiple

links or the corresponding file already exists.

-h

Print a help message describing the options, then quit.

-l

For each compressed file, list the following fields:

compressed size: size of the compressed file

uncompressed size: size of the uncompressed file

ratio: compression ratio (0.0% if unknown)

uncompressed\_name: name of the uncompressed file

-t

Test. Check the compressed file integrity.

**head**

Output the first part of files, prints the first part (10 lines by default) of each file.

**Syntax**

head [options]... [file]...

**Options**:

-NUMBER

Return the first NUMBER of lines from the file. (must be the first option specified)

-CountOptions

This option is only recognized if it is specified first. Count is a decimal number optionally followed by a size letter ('b', 'k',  
'm' for bytes, Kilobytes or Megabytes) , or 'l' to mean count by lines, or other option letters ('cqv').

-c BYTES

--bytes=BYTES

Print the first BYTES bytes, instead of initial lines. Appending

'b' multiplies BYTES by 512, 'k' by 1024, and 'm' by 1048576.

-n N

--lines=N

Output the first N lines.

-q

--quiet

--silent

Never print file name headers.

-v

--verbose

If no files are given (or if given a FILE of '-') head will read from standard input.

If more than one FILE is specified, 'head' will print a one-line header consisting of ==> FILE NAME <== before the output for each FILE.

**history**

Command Line history

**Syntax**

history

history [n]

history -c

history -d offset

history [-anrw] [filename]

history -ps arg

**Options**

-c Clear the history list. This may be combined with

the other options to replace the history list completely.

-d offset

Delete the history entry at position offset.

offset should be specified as it appears when the history is displayed.

-a Append the new history lines (history lines entered since

the beginning of the current Bash session) to the history file.

-n Append the history lines not already read from the history file

to the current history list. These are lines appended to the

history file since the beginning of the current Bash session.

-r Read the current history file and append its contents to the history list.

-w Write out the current history to the history file.

With no options, display the history list with line numbers. Lines prefixed with with a `\*' have been modified. An argument of n lists only the last n lines.

**hostname**

Print or set system name

**Syntax**

hostname [name]

With no arguments, `hostname' prints the name of the current host system. With one argument, it sets the current host name to the specified string. You must have appropriate privileges to set the host name.

**ifconfig**

Interface configurator - display your ip address, network interfaces, transferred and received data information, configure a network interface.

**Syntax**

ifconfig [interface]

ifconfig interface [aftype] options | address ...

**Options**

up This flag causes the interface to be activated.

It is implicitly specified if an address is assigned to the interface.

down This flag causes the driver for this interface to be shut down.

[-]arp Enable or disable the use of the ARP protocol on this interface.

[-]allmulti Enable or disable all-multicast mode.

If selected, all multicast packets on the network will be received by the interface.

mtu N Set the Maximum Transfer Unit (MTU) of an interface.

dstaddr addr Set the remote IP address for a point-to-point (PPP)link (obsolete; use pointopoint instead)

netmask addr Set the IP network mask for this interface.

This value defaults to the usual class A, B or C network mask

(as derived from the interface IP address), but it can be set to any value.

**ifup / ifdown**

Bring a network interface up or down

**Syntax**

ifup [options] -a | IFACE...

ifdown [options] -a|IFACE...

**Options**

-a, --all If given to ifup, affect all interfaces marked auto.

Interfaces are brought up in the order in which they are defined in /etc/network/interfaces.

If given to ifdown, affect all defined interfaces.

Interfaces are brought down in the order in which they are currently listed in the state file.

Only interfaces defined in /etc/network/interfaces will be brought down.

--force Force configuration or deconfiguration of the interface.

-i FILE

Read interface definitions from FILE instead of from /etc/network/interfaces.

-n, --no-act Don't configure any interfaces or run any "up" or "down" commands.

-v, --verbose Show commands as they are executed.

-h, --help Display help for ifup

**install**

Copy files and set attributes, copies files while setting their permission modes and, if possible, their owner and group.

**Syntax**

install [options]... SOURCE DEST

install [options]... SOURCE... DIRECTORY

install -d [options]... DIRECTORY...

**Description**

The 3 variants above install either a single SOURCE file to DEST target

or copy multiple SOURCE files to the destination. In the last variant,

each DIRECTORY (and any missing parent directories) is created.

`install' is similar to `cp', but allows you to control the

attributes of destination files. It is typically used in Makefiles to

copy programs into their destination directories. It refuses to copy

files onto themselves.

**Options**

-b

--backup

Make a backup of each file that would otherwise be overwritten or

removed. \*Note Backup options::.

-C

Install file, unless target already exists and is the same file,

in which case the modification time is not changed.

-g GROUP

Set the group ownership of installed files or directories to

GROUP. The default is the process's current group. GROUP may be

either a group name or a numeric group id.

-m MODE

Set the permissions for the installed file or directory to MODE,

which can be either an octal number, or a symbolic mode as in

`chmod', with 0 as the point of departure (\*note File

permissions::). The default mode is 0755--read, write, and execute

for the owner, and read and execute for group and other.

-s

Strip the symbol tables from installed binary executables.

-V METHOD

Change the type of backups made with `-b'. The METHOD argument

can be `numbered' (or `t'), `existing' (or `nil'), or `never' (or

`simple').

**jobs**

Print currently running jobs and their status.

**Syntax**

jobs [OPTIONS] [PID]

**Options**:

-c

--command Print the command name for each process in jobs

-g

--group Only print the group id of each job

-h

--help Display a help message and exit

-l

--last Only the last job to be started is printed

-p

--pid Print the process id for each process in all jobs

On systems that supports this feature, jobs will print the CPU usage of each job since the last command was executed. The CPU usage is expressed as a percentage of full CPU activity. Note that on multiprocessor systems, the total activity may be more than 100%.

**kill**

Stop a process from running, either via a signal or forced termination.

**Syntax**

kill [-s sigspec] [-n signum] [-sigspec] jobspec or pid

kill -l [exit\_status]

kill -l [sigspec]

**Options**

-l List the signal names

-s Send a specific signal

-n Send a specific signal number

Send a signal specified by sigspec or signum to the process named by job specification jobspec or process ID pid.   
  
sigspec is either a case-insensitive signal name such as SIGINT (with or without the SIG prefix) or a signal number; signum is a signal number.  
  
If sigspec is not present, SIGTERM is used (Terminate).   
  
If any arguments are supplied when `-l' is given, the names of the signals corresponding to the arguments are listed, and the return status is zero. exit\_status is a number specifying a signal number or the exit status of a process terminated by a signal.  
  
The return status is true if at least one signal was successfully sent, or false if an error occurs or an invalid option is encountered.

**killall**

kill processes by name .

**Syntax**

killall [option(s)] [--] name ...

killall -l

killall -V,--version

**Options**

-e

Require an exact match for very long names.

-g

Kill the process group to which the process belongs.

-I Do case insensitive process name match.

-i Interactively ask for confirmation before killing.

-l List all known signal names.

-q Do not complain if no processes were killed.

-w

--wait

Wait for all killed processes to die.

**less**

Display output one screen at a time, Search through output, Edit the command line.

**Syntax**

less [options]

<command> | less [options]

**Moving Commands:**

Commands marked with \* may be preceded by a number, N.

Notes in parentheses indicate the behavior if N is given.

e ^E j ^N CR \* Forward one line (or N lines).

z \* Forward one window (and set window to N).

ESC-( RightArrow \* Left 8 character positions (or N positions).

**Searching Commands:**

/pattern \* Search forward for (N-th) matching line.

?pattern \* Search backward for (N-th) matching line.

**Jumping commands:**

g < ESC-< \* Go to first line in file (or line N).

G > ESC-> \* Go to last line in file (or line N).

p % \* Go to beginning of file (or N percent into file).

'' Go to the previous position.

**Changing files:**

:e [file] Examine a new file.

:p \* Examine the (N-th) previous file from the command line.

:d Delete the current file from the command line list.

**Miscellaneous Commands:**

- Toggle a command line option [see OPTIONS below].

-- Toggle a command line option, by name.

\_ Display the setting of a command line option.

\_\_ Display the setting of an option, by name.

**Options**

**Finding**:

-a ........ --search-skip-screen

Forward search, skips current screen.

-i ........ --ignore-case

Ignore case in searches.

-p [pattern] --pattern=[pattern]

Start at pattern (from command line).

**Display** **Options**:

-c -C .... --clear-screen --CLEAR-SCREEN

Repaint by scrolling/clearing.

-P [prompt] --prompt=[prompt]

Define new prompt.

-n -N .... --line-numbers --LINE-NUMBERS

Use line numbers.

-z [N] .... --window=[N]

Set size of window.

**File** **Options**:

-b [N] .... --buffers=[N]

Number of buffers.

-B ........ --auto-buffers

Don't automatically allocate buffers for pipes.

-f ........ --force

Force open non-regular files.

Copy to log file (unconditionally overwrite).

**Misc** **Options**:

-d ........ --dumb

Dumb terminal.

-e -E .... --quit-at-eof --QUIT-AT-EOF

Quit at end of file.

-q -Q .... --quiet --QUIET --silent --SILENT

Quiet the terminal bell.

**locate**

Find files.

**Syntax**

locate [options] pattern

**Options**

-d path, --database=path

Search databases in path.

path must be a colon- separated list.

-h, --help Print a help message and exit.

--version Print version information and then exit.

Search database(s) of filenames and print matches. \*, ?, [, and ] are treated specially; / and . are not.

**logname**

Print current login name

**Syntax**

logname

Prints the calling user's name, as found in the file`/var/run/utmp', and exits with a status of 0.  
  
If there is no `/var/run/utmp' entry for the calling process, `logname' prints an error message and exits with a status of 1.  
  
The only options are `--help' and `--version'.

**lpc**

line printer control program

**Syntax**

lpc [command [argument ...]]

**Description**

Lpc is used by the system administrator to control the operation of the

line printer system. For each line printer configured in /etc/printcap,

lpc may be used to:

· Disable or enable a printer,

· Disable or enable a printer's spooling queue,

· Rearrange the order of jobs in a spooling queue,

· Find the status of printers, and their associated spooling

queues and printer dameons.

Without any arguments, lpc will prompt for commands from the standard input. If arguments are supplied, lpc interprets the first argument as a command and the remaining arguments as parameters to the command. The standard input may be redirected causing lpc to read commands from file.  
Commands may be abreviated;

**Commands**

? [command ...]

help [command ...]

clean { all | printer }

Remove any temporary files, data files, and control files that

cannot be printed (i.e., do not form a complete printer job) from

the specified printer queue(s) on the local machine.

disable { all | printer }

Turn the specified printer queues off. This prevents new printer

jobs from being entered into the queue by lpr.

quit Exit from lpc.

start { all | printer }

Enable printing and start a spooling daemon for the listed printers.

status { all | printer }

Display the status of daemons and queues on the local machine.

stop { all | printer }

Stop a spooling daemon after the current job completes and disable printing.

**lpr**

off line print - sends a print job to the default system queue.

**Syntax**

lpr [-Pprinter] [-#num] [-C class] [-J job] [-T title] [-U user] [-i [numcols]]

[-1234 font] [-wnum] [-cdfghlnmprstv] [name ...]

**Description**

Lpr uses a spooling daemon to print the named files when facilities become

available. If no names appear, the standard input is assumed.

The following single letter options are used to notify the line printer

spooler that the files are not standard text files. The spooling daemon

will use the appropriate filters to print the data accordingly.

-c The files are assumed to contain data produced by cifplot(1)

-d The files are assumed to contain data from tex (DVI format from

Stanford).

-f Use a filter which interprets the first character of each line as

a standard FORTRAN carriage control character.

-g The files are assumed to contain standard plot data as produced

by the plot routines (see also plot for the filters used by the

printer spooler).

-l Use a filter which allows control characters to be printed and

suppresses page breaks.

-r Remove the file upon completion of spooling. Can not be used with

the -s option, due to security concerns.

**ls**

List information about files.

**Syntax**

ls [Options]... [File]...

**Options**

Sort entries alphabetically if none of -cftuSUX nor --sort.

-a, --all Do not hide entries starting with .

-b, --escape Print octal escapes for nongraphic characters

-C List entries by columns

-d, --directory List directory entries instead of contents

-i, --inode Print index number of each file

-u sort by last access time; with -l: show atime

-U do not sort; list entries in directory order

-v sort by version

-X sort alphabetically by entry extension

-1 list one file per line

By default, colour is not used to distinguish types of files. That is equivalent to using --color=none.   
Using the --color option without the optional WHEN argument is equivalent to using --color=always.   
With --color=auto, color codes are output only if standard output is connected to a terminal (tty).

**lsof**

List open files.

**Syntax**

lsof [ -?abChlnNOPRstUvVX ] [ -A A ] [ -c c ] [ +c c ] [ +|-d d ]

[ +|-D D ] [ +|-f [cfgGn] ] [ -F [f] ] [ -g [s] ] [ -i[i] ]

[ -k k ] [ +|-L [l] ] [ +|-m m ] [ +|-M ] [ -o [o] ] [ -p s ]

[ +|-r [t] ] [ -S [t] ] [ -T [t] ] [ -u s ] [ +|-w ]

[ -x [fl] ] [ -z [z] ] [ -Z [Z] ] [ -- ] [filenames]

**Options**

-? –h These two equivalent options select a usage (help) output list

-a This option causes list selection options to be ANDed, as described below.

-b

This option causes lsof to avoid kernel functions that might

-C

This option disables the reporting of any path name components from the kernel's name cache.

? - report device cache file paths

b - build the device cache file

i - ignore the device cache file

r - read the device cache file

u - read and update the device cache file

The b, r, and u functions, accompanied by a path name, are sometimes restricted.

**man / info / help**

Format and display help pages.

**Syntax**

man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config\_file]

[-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S section\_list]

[section] name ...

**Options**

-M path The list of directories to search for man pages. Separate the directories with colons.

-B Which browser to use on HTML files. This option overrides the BROWSER environment variable.

By default, man uses /usr/bin/less-is,

-a By default, man will exit after displaying the first manual page it finds.

-c Reformat the source man page, even when an up-to-date cat page exists.

-d Don't actually display the man pages, but do print jobs of debugging information.

-D Both display and print debugging info.

-h Print a help message and exit.

-K Search for the specified string in \*all\* man pages.

Warning: this can be very slow and may match multiple pages!

**make**

GNU make utility to maintain groups of programs

**Syntax**

make [ -f makefile ] [ options ] ... [ targets ] ...

**Options**

-b, -m

These options are ignored for compatibility with other versions of make.

-C dir, --directory=dir

Change to directory dir before reading the makefiles or doing anything else.

-d

Print debugging information in addition to normal processing.

-f file, --file=file, --makefile=FILE

Use file as a makefile.

-i, --ignore-errors

Ignore all errors in commands executed to remake files.

-n, --just-print, --dry-run, --recon

Print the commands that would be executed, but do not execute them (except in certain circumstances).

-s, --silent, --quiet

Silent operation; do not print the commands as they are executed.

--warn-undefined-variables

Warn when an undefined variable is referenced.

**mkdir**

Create new folder(s), if they do not already exist.

**Syntax**

mkdir [Options] folder...

mkdir "Name with spaces"

**Options**

-m, --mode=MODE set permission mode (as in [chmod](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\chmod.html)), not rwxrwxrwx - umask

-p, --parents no error if existing, make parent directories as needed

--verbose print a message for each created directory

mkdir creates the standard entries . (dot) for the current folder and .. (dot dot) for its parent

**mmv**

Mass Move and rename - Move, copy, append or link Multiple files using wildcard patterns.

**Syntax**

mmv [Source\_Option] [-h] [-d|p] [-g|t] [-v|n] [--] [from to]

**Options**:

-h help

-d Delete

-p Protect (don't delete or overwrite)

-g Go

-t Terminate

-v verbose mode

-n no-execute mode (display messages about what would have been done)

Source\_Options:

-m Move source file to target name.

-r Rename source file or directory to target name.

-c Copy source file to target name.

-o Overwrite target name with source file.

-a Append contents of source file to target name.

-l Link target name to source file.

command\_name default task

mmv -x (Move except cross device)

mcp -c (Copy)

mad -a (Append)

mln -l (Link)

**mv**

Move or rename files or directories.

**Syntax**

mv [options]... Source Dest

mv [options]... Source... Directory

If the last argument names an existing directory, `mv' moves each other given file into a file with the same name in that directory. Otherwise, if only two files are given, it renames the first as the second. It is an error if the last argument is not a directory and more than two files are given.

**Options**

-b

Make a backup of each file that would otherwise be overwritten or

removed.

-f

Remove existing destination files and never prompt the user.

-i

Prompt whether to overwrite each existing destination file,

regardless of its permissions. If the response does not begin

with `y' or `Y', the file is skipped.

-S SUFFIX

Append SUFFIX to each backup file made with `-b'.

The backup suffix is ~, unless set with SIMPLE\_BACKUP\_SUFFIX.

-u

Do not move a nondirectory that has an existing destination with

the same or newer modification time.

-v

Print the name of each file before moving it.

-V METHOD

Change the type of backups made with `-b'. METHOD can be:

t, numbered make numbered backups

nil, existing numbered if numbered backups exist, simple otherwise

never, simple always make simple backups

--help display help and exit

--version output version information and exit

**netstat**

Print network connections, routing tables, interface statistics, masquerade connections, and multicast memberships

**Syntax**

netstat [address\_family\_options] [--tcp|-t] [--udp|-u] [--udplite|-U] [--raw|-w] [--listening|-l] [--all|-a] [--numeric|-n] [--numeric-hosts] [--numeric-ports] [--numeric-users] [--symbolic|-N] [--extend|-e[--extend|-e]] [--timers|-o] [-- program|-p] [--verbose|-v] [--continuous|-c] [--wide|-W] [delay]

Note

This program is obsolete. Replacement for netstat is ss. Replacement for netstat -r is ip route. Replacement for netstat -i is ip -s link. Replacement for netstat -g is ip

**Options**

-v

Tell the user what is going on by being verbose. Especially print some useful information about unconfigured address families

.

-W

Do not truncate IP addresses by using output as wide as needed. This is optional for now to not break existing scripts.

-n

Show numerical addresses instead of trying to determine symbolic host, port or user names.

-hosts

shows numerical host addresses but does not affect the resolution of port or user names.

-ports

shows numerical port numbers but does not affect the resolution of host or user names.

-users

shows numerical user IDs but does not affect the resolution of host or port names.

-C

Print routing information from the route cache.

delay

Netstat will cycle printing through statistics every delay seconds.

**open**

Open a file in its default application, using virtual terminal (VT).

**Syntax**

open Files...

**Options**:

files The filename(s) to open, wildcards are accepted.

Example

Open all the text files in the current directory using your default text editor:

open \*.txt

**passwd**

Modify a user password.

**Syntax**

passwd [options...]

**Options**

-d, --delete delete the password for the named account (root only)

-f, --force force operation (effectively calls `chfn'?)

-k, --keep-tokens keep non-expired authentication tokens

-l, --lock lock the named account (root only)

-S, --status report password status on the named account (root only)

--stdin read new tokens from stdin (root only)

-u, --unlock unlock the named account (root only)

-?, --help Show this help message

--usage Display brief usage message

If no options are specified - passwd will change the password of the currently logged in user - will prompt for the old and new passwords.

**paste**

Merge lines of files, write to standard output lines consisting of sequentially corresponding lines of each given file, separated by a TAB character.

**Syntax**

paste [options]... [file]...

**Options**

-s

--serial

Paste the lines of one file at a time rather than one line from

each file.

-d DELIM-LIST

--delimiters DELIM-LIST

Consecutively use the characters in DELIM-LIST instead of TAB to

separate merged lines. When DELIM-LIST is exhausted, start again

at its beginning.

Standard input is used for a file name of - or if no input files are given.

**ping**

Test a network connection. When using ping for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be `pinged'.

**Syntax**

ping [options] destination\_host

**Options**

-a Audible ping.

-b Allow pinging a broadcast address.

-c count Stop after sending (and receiving) count ECHO\_RESPONSE packets.

-f Flood ping, output packets as fast as they come back or 100 times per second.

-i wait Set an interval of wait seconds between sending each packet. default=one second.

-L Suppress loopback of multicast packets.

only applies if the ping destination is a multicast address.

-n Numeric output only. No attempt will be made to lookup symbolic

names for host addresses.

-q Quiet output. Only display the summary lines at startup time and when finished.

-U Print full user-to-user latency (the old behaviour).

Normally ping prints network round trip time, which can be different f.e. due to DNS failures.

-v Verbose output. ICMP packets other than ECHO\_RESPONSE that are received are listed.

**ps**

Process status, information about processes running in memory. If you want a repetitive update of this status, use top.

**Syntax**

ps option(s)

ps [-L]

**Options**

-L List all the [keyword](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\ps_keywords.html) options

**Simple Process Selection:**

-A a select all processes (including those of other users)

-a select all with a tty except session leaders

-d select all, but omit session leaders

r restrict output to running processes

T select all processes on this terminal

x select processes without controlling ttys

**Process Selection by List:**

-C select by command name

-G select by RGID (supports names)

-g select by session leader OR by group name

**Output Format Control:**

-c Different scheduler info for -l option

-f Full listing

-j j Jobs format

-l l Long format

-O O Add the information associated with the space or comma separated

list of [keywords](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\ps_keywords.html) specified, after the process ID, in the default

information display.

-o o Display information associated with the space or comma separated

list of [keywords](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\ps_keywords.html) specified.

--format [user-defined format](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\ps_keywords.html)

s display signal format

u display user-oriented format

v display virtual memory format

X old Linux i386 register format

-y do not show flags; show rss in place of addr

**pwd**

Print Working Directory (shell builtin)

**Syntax**

pwd [-LP]

**Options**

-P : The pathname printed will not contain symbolic links.

-L : The pathname printed may contain symbolic links

The default action is to show the current folder as an absolute path.  
All components of the path will be actual folder names - none will be symbolic links.

**quota**

Display disk usage and limits, by default only the user quotas are printed.

**Syntax**

quota [ -guv | q ]

quota [ -uv | q ] user

quota [ -gv | q ] group

**Options**

-g Print group quotas for the group of which the user

is a member.

-u Print user quotas (this is the default)

-v Verbose, will display quotas on filesystems where no

storage is allocated.

-q Print a more terse message, containing only information

on filesystems where usage is over quota.

**read**

Read a line from standard input

**Syntax**

read [-ers] [-a aname] [-p prompt] [-t timeout]

[-n nchars] [-d delim] [name...]

**Options**

-a aname

The words are assigned to sequential indices of the array variable aname,

starting at 0

-d delim

The first character of delim is used to terminate the input line,

rather than newline.

-e

If the standard input is coming from a terminal

-n nchars

read returns after reading nchars characters rather

than waiting for a complete line of input.

-p prompt

Display prompt, without a trailing newline, before attempting

to read any input.

-r

If this option is given, backslash does not act as an escape character.

-s

Silent mode. If input is coming from a terminal, characters are not echoed.

**rename (linux only, not ubuntu)**

Rename files.

**Syntax**

rename from to file...

rename will rename the specified files by replacing the first occurrence of from in their name by to.

**Examples**

Given the files foo1, ..., foo9, foo10, ..., foo278, the commands

$ rename foo foo0 foo?  
$ rename foo foo0 foo??

will turn them into foo001, ..., foo009, foo010, ..., foo278.

**rev**

Reverse lines of a file.

**Syntax**:

rev [file]

The rev utility copies the specified files to the standard output, reversing the order of characters in every line.

If no file is specified, the standard input is read (type CTRL-D to end).

The most common use of rev is to reverse the lines, apply a [sort](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\sort.html) to the result and then pipe through rev a second time to restore the original.

**Example**

Given the input of:

The quick brown fox   
jumps over the lazy dog

Rev will output:

xof mworb kciuq ehT   
god yzal eht revo spmuj

**rm**

Remove files (delete/unlink)

**Syntax**

rm [options]... file...

**Options**

-d, --directory unlink directory, even if non-empty (super-user only)

-f, --force ignore nonexistent files, never prompt

-i, --interactive prompt before any removal

-r, -R, --recursive remove the contents of directories recursively

-v, --verbose explain what is being done

--help display this help and exit

--version output version information and exit

**rmdir**

Remove directory, this command will only work if the folders are empty.

**Syntax**

rmdir [options]... folder(s)...

**Options**

--ignore-fail-on-non-empty

Ignore each failure that is solely because the

directory is non-empty.

-p, --parents Remove explicit parent directories if being emptied

--verbose Output a diagnostic for every directory processed

--help Display help and exit

--version Output version information and exit

**shutdown**

Shutdown or restart linux

**Syntax**

shutdown [options] when [message]

**Options**

-c Cancel a shutdown that is in progress.

-f Reboot fast, by suppressing the normal call to fsck

when rebooting.

-h Halt the system when shutdown is complete.

-k Print the warning message, but suppress actual shutdown.

-n Perform shutdown without a call to init.

-r Reboot the system when shutdown is complete.

-t sec

Ensure a sec-second delay between killing processes

and changing the runlevel.

**sort**

Sort text files.  
Sort, merge, or compare all the lines from the files given (or standard input.)

**Syntax**

sort [options] [file...]

sort --help

sort --version

**Options**

sort has three modes of operation:

Sort (the default), Merge (-m), and Check(-c)

-c Check whether the given files are already sorted: if they are

not all sorted, print an error message and exit with a status of 1.

-m Merge the given files by sorting them as a group. Each input

file should already be individually sorted. It always works to

sort instead of merge; merging is provided because it is faster,

in the case where it works.

`-b'

Ignore leading blanks when finding sort keys in each line.

`-d'

Sort in "phone directory" order: ignore all characters except

letters, digits and blanks when sorting.

`-f'

Fold lowercase characters into the equivalent uppercase characters

when sorting so that, for example, `b' and `B' sort as equal.

`-i'

Ignore unprintable characters.

`-n'

Sort numerically: the number begins each line.

`-r'

Reverse the result of comparison, so that lines with greater key

values appear earlier in the output instead of later.

**su**

Substitute user identity  
Run a command with substitute user and group id, allow one user to temporarily become another user. It runs a command (often an interactive shell) with the real and effective user id, group id, and supplemental groups of a given user.

**Syntax**

su [options]... [user [arg]...]

**Options**

-c COMMAND

Pass COMMAND, a single command line to run, to the shell with a

-c option instead of starting an interactive shell.

-f

Pass the `-f' option to the shell.

-l

Make the shell a login shell.

-p

Do not change the environment variables `HOME', `USER', `LOGNAME',

or `SHELL'

-s SHELL

Run SHELL instead of the shell from USER's passwd entry, unless

the user running `su' is not the superuser and USER's shell is

restricted (see `-m' just above).

If no user is given, the default is 'root', the super-user.

**sudo**

sudo allows a permitted user to execute a command as the superuser or another user, as specified in the sudoers file.

**Syntax**

sudo -K | -L | -V | -h | -k | -l | -v

sudo [-HPSb] [-a auth\_type] [-c class|-] [-p prompt]

[-u username|#uid] {-e file [...] | -i | -s | command}

sudoedit [-S] [-a auth\_type] [-p prompt] [-u username|#uid] file [...]

**Options**

-H Set the HOME environment variable to the homedir of the target user

(root by default) as specified in passwd.

-K (sure kill) like kill (-k) except that it removes the user's timestamp

entirely.

-L list defaults, list out the parameters that may be set in a Defaults

line along with a short description for each.

-S stdin, read the password from the standard input instead of the terminal.

-V Version, print the version number and exit.

-b Background, run the given command in the background.

-c Class, run the specified command with resources limited by

the specified login class.

-e Edit, instead of running a command.

-h Help, print a usage message and exit.

**tail**

Output the last part of files, print the last part (10 lines by default) of each FILE;   
tail reads from standard input if no files are given or when given a FILE of `-'.

**Syntax**

tail [options]... [file]...

tail -Number [options]... [file]...

tail +Number [options]... [file]...

**Options**

-COUNT

+COUNT

This option is only recognized if it is specified first. COUNT is

a decimal number optionally followed by a size letter (`b', `k',

`m') as in `-c', or `l' to mean count by lines, or other option

letters (`cfqv').

-c BYTES

Output the last BYTES bytes, instead of final lines

-f

Loop forever trying to read more characters at the end of the file.

--retry

This option is meaningful only when following by name

--sleep-interval=N

Change the number of seconds to wait between iterations.

--pid=PID

When following by name or by descriptor, you may specify the

process ID, PID, of the sole writer of all FILE arguments. Then,

shortly after that process terminates, tail will also terminate.

-n N

Output the last N lines.

-q

Never print file name headers.

-v

Always print file name headers.

If more than one FILE is specified, `tail' prints a one-line header consisting of ==> FILE NAME <== before the output for each FILE.

**tee**

Redirect output to multiple files, copies standard input to standard output and also to any files given as arguments. This is useful when you want not only to send some data down a pipe, but also to save a copy.

**Syntax**

tee [options]... [file]...

**Options**

-a

--append

Append standard input to the given files rather than overwriting

them.

-i

--ignore-interrupts'

Ignore interrupt signals.

**Example**:

ps -ax | tee processes.txt | more

If a file being written to does not already exist, it is created.   
If a file being written to already exists, the data it previously contained is overwritten unless the `-a' option is used.

**time**

Measure the running time of a program.

The `time' command will run another program, and record the elapsed time or CPU Resource Used time used by that program.   
The information mmay be displayed on screen or saved in a file.

**Syntax**

time [option...] command [arg...]

**Options**

-o FILE

Write the resource use statistics to FILE.

-f FORMAT

Use FORMAT as the format string, see below for formatting options.

--help

Summary of the command line options.

**The Format String**  
  
The "format string" controls the contents of the `time' output. It consists of "resource specifiers" and "escapes", interspersed with  
plain text.

Time Resources

%e Elapsed real (wall clock) time used by the process, in seconds.

%S Total number of CPU-seconds used by the system on behalf of the

process (in kernel mode), in seconds.

%U Total number of CPU-seconds that the process used directly (in user

mode), in seconds.

%P Percentage of the CPU that this job got. This is just user +

system times divied by the total running time.

**Memory Resources**

%t Average resident set size of the process, in Kilobytes.

%D Average size of the process's unshared data area, in Kilobytes.

%p Average size of the process's unshared stack, in Kilobytes.

%X Average size of the process's shared text, in Kilobytes.

**top**

Process viewer, find the CPU-intensive programs currently running. See [ps](file:///C:\Users\Harshvardhan\Desktop\ \System%20Software\ps.html) for explanations of the field descriptors.

**Syntax**

top options

**Options**

-b Run in batch mode; don't accept command-line input.

Useful for sending output to another command or to a file.

-c Show command line in display instead of just command name.

-d delay

Specify delay between refreshes.

-i Suppress display of idle and zombie processes.

-n num

Update display num times, then exit.

-p pid

Monitor only processes with the specified process ID.

-q Refresh without any delay.

If user is privileged, run with highest priority.

-s Secure mode. Disable some (dangerous) interactive commands.

-S Cumulative mode. Print total CPU time of each process,

including dead child processes.

**touch**

Change file timestamps, change the access and/or modification times of the specified files.

**Syntax**

touch [options]... File...

**Options**

-a

--time=atime

--time=access

--time=use

Change the access time only.

-c

--no-create

Do not create files that do not exist.

-d

--date=time

Use time instead of the current time. It can contain month names,

timezones, `am' and `pm', etc.

-f Ignored; for compatibility with BSD versions of `touch'.

-m

--time=mtime

--time=modify

Change the modification time only.

-r FILE

--reference=FILE

Use the times of the reference FILE instead of the current time.

-t [[CC]YY]MMDDhhmm[.ss]

Set to a specific time.

**type**

Describe a command, for each name, indicate how it would be interpreted if used as   
a command name.

**Syntax**

type [-atp] [name ...]

**Options**

If the `-t' option is used, type prints a single word which is one of:

`alias' (shell alias)

`function' (shell function)

`builtin' (shell builtin)

`file' (disk file)

`keyword' (shell reserved word)

If the `-p' option is used, type either returns the name of the disk file that

would be executed, or nothing if `-t' would not return `file'.

If the `-a' option is used, type returns all of the places that contain an

executable named file. This includes aliases and functions, if and only if the

`-p' option is not also used.

The return status is zero if any of the names are found, non-zero otherwise.

`type' is a BASH builtin command.

**ulimit**

User limits - limit the use of system-wide resources.

**Syntax**

ulimit [-acdfHlmnpsStuv] [limit]

**Options**

-S Change and report the soft limit associated with a resource.

-H Change and report the hard limit associated with a resource.

-a All current limits are reported.

-c The maximum size of core files created.

-d The maximum size of a process's data segment.

-f The maximum size of files created by the shell(default option)

-l The maximum size that may be locked into memory.

-m The maximum resident set size.

-n The maximum number of open file descriptors.

-p The pipe buffer size.

-s The maximum stack size.

-t The maximum amount of cpu time in seconds.

-u The maximum number of processes available to a single user.

-v The maximum amount of virtual memory available to the process.

**uniq**

Uniquify files, write out the unique lines from the given InputFile  
uniq only removes adjacent matching lines.  
If an InputFile of - (or nothing) is given, then uniq will read from standard input.

**Syntax**

uniq [options]... [InputFile [OutputFile]]

**Options**

-f N

Skip N fields on each line before checking for uniqueness. Fields

are sequences of non-space non-tab characters that are separated

from each other by at least one spaces or tabs.

-s N

Skip N characters before checking for uniqueness

-c

Print the number of times each line occurred along with the line.

-i

Ignore differences in case when comparing lines.

-d

Print only duplicate lines.

-D

Print all duplicate lines and only duplicate lines

-u

Print only unique lines.

**useradd**

Create a new user or update default new user information .

**Syntax**

useradd [options] LOGIN

useradd -D

useradd -D [options]

**Options**

-b, --base-dir BASE\_DIR

The default base directory for the system if -d HOME\_DIR is not specified.

-c, --comment COMMENT

Any text string.

-d, --home HOME\_DIR

The new user will be created using HOME\_DIR as the value for the user's

login directory.

-e, --expiredate EXPIRE\_DATE

The date on which the user account will be disabled.

-f, --inactive INACTIVE

The number of days after a password expires until the account is

permanently disabled.

-g, --gid GROUP

The group name or number of the user's initial login group.

-p, --password PASSWORD

The encrypted password, as returned by crypt(3).

The default is to disable the password.

-r, --system

Create a system account.

**userdel**

Delete a user account and related files.

**Syntax**

userdel [options] LOGIN

userdel modifies the system account files, deleting all entries that refer to the user name LOGIN. The named user must exist.

**Options**

-f, --force

This option forces the removal of the user account, even if

the user is still logged in.

-h, --help

Display help message and exit.

-r, --remove

Files in the user's home directory will be removed along with

the home directory itself and the user's mail spool.

The mail spool is defined by the MAIL\_DIR variable in the login.defs file.

**usermod**

Modify user account information.

**Syntax**

usermod [options] [user]

**Options**

-c comment Comment field.

-d dir Home directory.

-e date Account expiration date.

date is in the format MM/DD/YYYY.

-f days Permanently disable account this many days after the

password has expired.

-g group Initial group name or ID number.

-G groups Supplementary groups given by name or number in a comma-separated

list with no whitespace.

-l name Login name. This cannot be changed while the user is logged in.

-o Override. Accept a nonunique uid with the -u option. (Probably a bad idea.)

**users**

Print login names of users currently logged in, print on a single line a blank-separated list of user names of users currently logged in to the current host.

**Syntax**

users [file]

With no file argument, `users' extracts its information from the file `/var/run/utmp'.

If a file argument is given, `users' uses that file instead. A common choice is `/var/run/wtmp'.

**Options**

The only options are `--help' and `--version'.

Each user name corresponds to a login session, so if a user has more than one login session, that user's name will appear the same number of times in the output.

**which**

Locate a program file in the user's path.

For each of its arguments which prints to stdout the full path of the executable(s). It does this by searching the directories listed in the environment variable PATH.

**Syntax**

which [options] [--] program\_name [...]

**Options**

--all, -a

Print all matching executables in PATH, not just the

first.

--read-alias, -i

Read aliases from stdin, reporting matching ones on

stdout.

--skip-alias

Ignore option --read-alias, if any.

--skip-dot

Skip directories in PATH that start with a dot.

--skip-tilde

Skip directories in PATH that start with a tilde and

executables which reside in the HOME directory.

--tty-only

Stop processing options on the right if not on tty.

**RETURN VALUE**

Which returns the number of failed arguments, or -1 when

no programname was given.

**who**

Print who is currently logged in

**Syntax**

who [options] [file] [am i]

**Options**

-m

Print the current user id, name and domain

(Same as `who am i')

-q

--count

Print only the login names and the number of users logged on.

-s

Ignored; for compatibility with other versions of `who'.

-i

-u

After the login time, print the number of hours and minutes that

the user has been idle. `.' means the user was active in last

minute. `old' means the user was idle for more than 24 hours.

-l

Attempt to canonicalize hostnames found in utmp through a DNS lookup.

-H

Print a line of column headings.

-w

--writable

After each login name print a character indicating the user's

message status:

+ allowing `write' messages

- disallowing `write' messages

? cannot find terminal device

**whoami**

Print the current user id and name.

**Syntax**

whoami [options]

**Options**

--help Display Help

--version Display program version info

whoami produces the same result as the id -un command (the id command by default provides more detailed information)

**write**

Send a message to another user

**Syntax**:

write user [ttyname]

Write allows you to communicate with other users, by copying lines from your terminal to theirs.

When you run the write command, the user you are writing to gets a message of the form:

Message from yourname@yourhost on yourtty at hh:mm ...

Any further lines you enter will be copied to the specified user's terminal. If the other user wants to reply, they must run write as well.

When you are done, type an end-of-file or interrupt character. The other user will see the message EOF indicating that the conversation is over.

You can prevent people (other than the super-user) from writing to you with the mesg(1) command. Some commands, for example nroff(1) and pr(1), may disallow writing automatically, so that your output isn't overwritten.

If the user you want to write to is logged in on more than one terminal, you can specify which terminal to write to by specifying the terminal name as the second operand to the write command. Alternatively, you can let write select one of the terminals - it will pick the one with the shortest idle time. This is so that if the user is logged in at work and also dialed up from home, the message will go to the right place.

The traditional protocol for writing to someone is that the string '-o', either at the end of a line or on a line by itself, means that it's the other person's turn to talk. The string 'oo' means that the person believes the conversation to be over.

**Example**

$ Write maude   
This is a test message <CTRL-D>

**wall**

send a message to everybody's terminal.

**Syntax**

wall [-n] [ message ]

**Options**

-n

Suppresses the normal banner printed by wall, changing it to "Remote broadcast message".

**wc**

print newline, word, and byte counts for each file

**Syntax**

wc [OPTION]... [FILE]... wc [OPTION]... --files0-from=F

**Options**

-c, --bytes

print the byte counts

-m, --chars

print the character counts

-l, --lines

print the newline counts

--files0-from=F

read input from the files specified by NUL-terminated names in file F; If F is - then

read names from standard input

-L, --max-line-length

print the length of the longest line

-w, --words

print the word counts

--help

display this help and exit

--version

output version information and exit

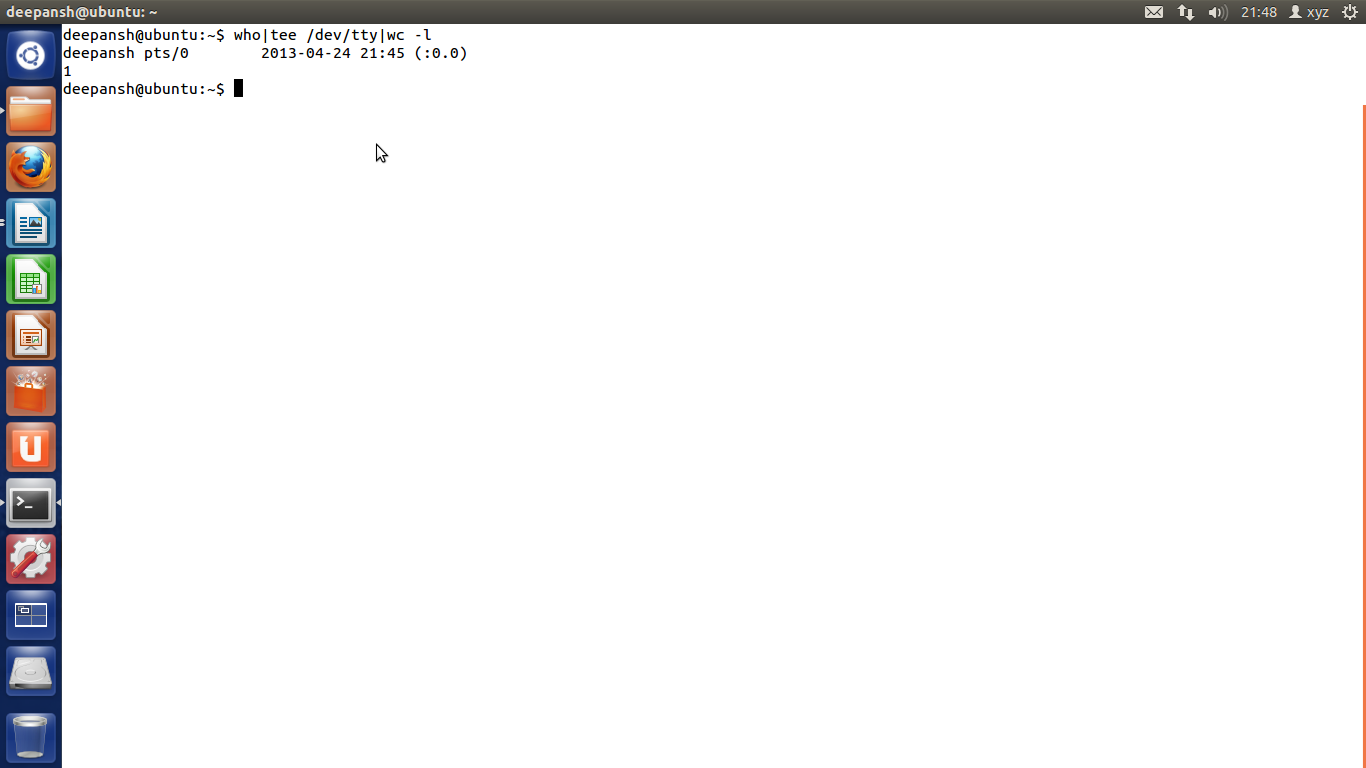
**Pipelines**

**&**

**Shell Scripts**

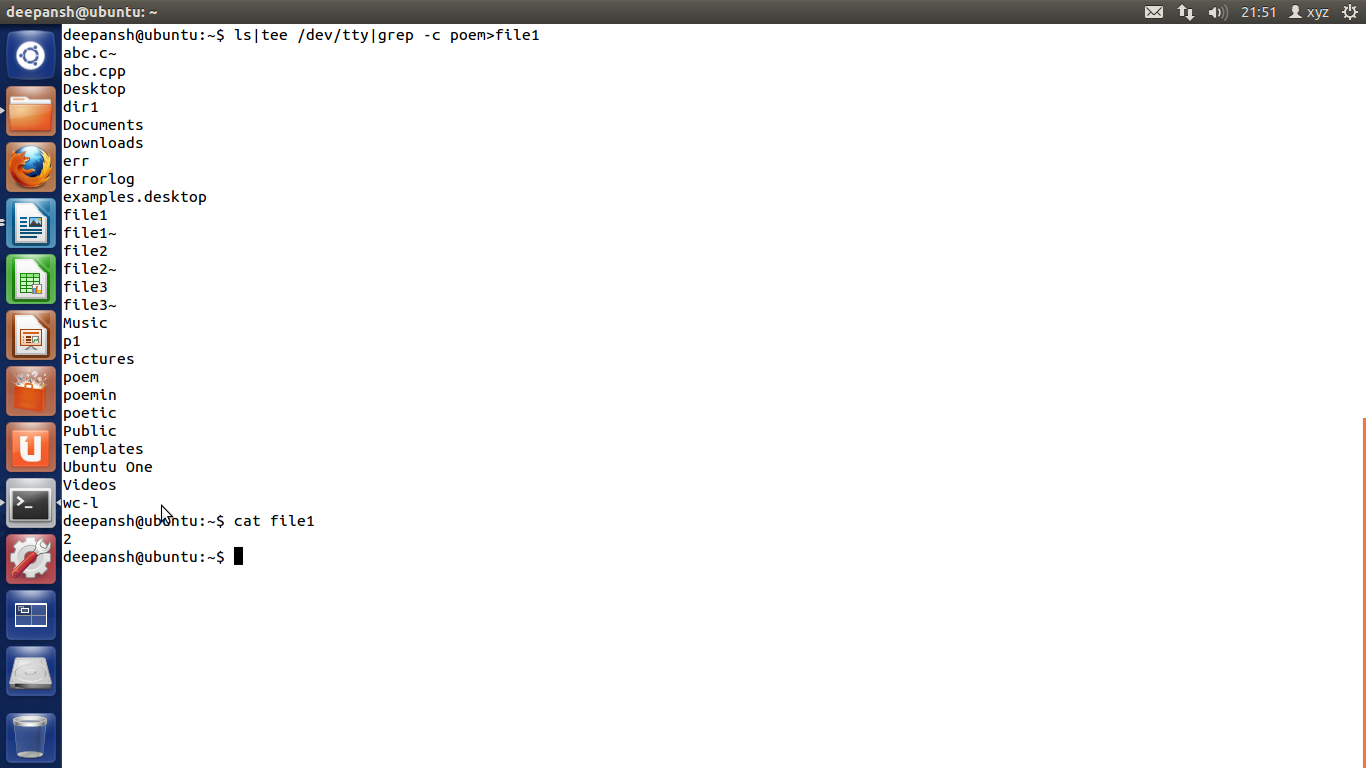
1. Output of who should be displayed on the screen with the value of total number of users who have logged in displayed at the bottom of the list.

$who|tee /dev/tty|wc -l



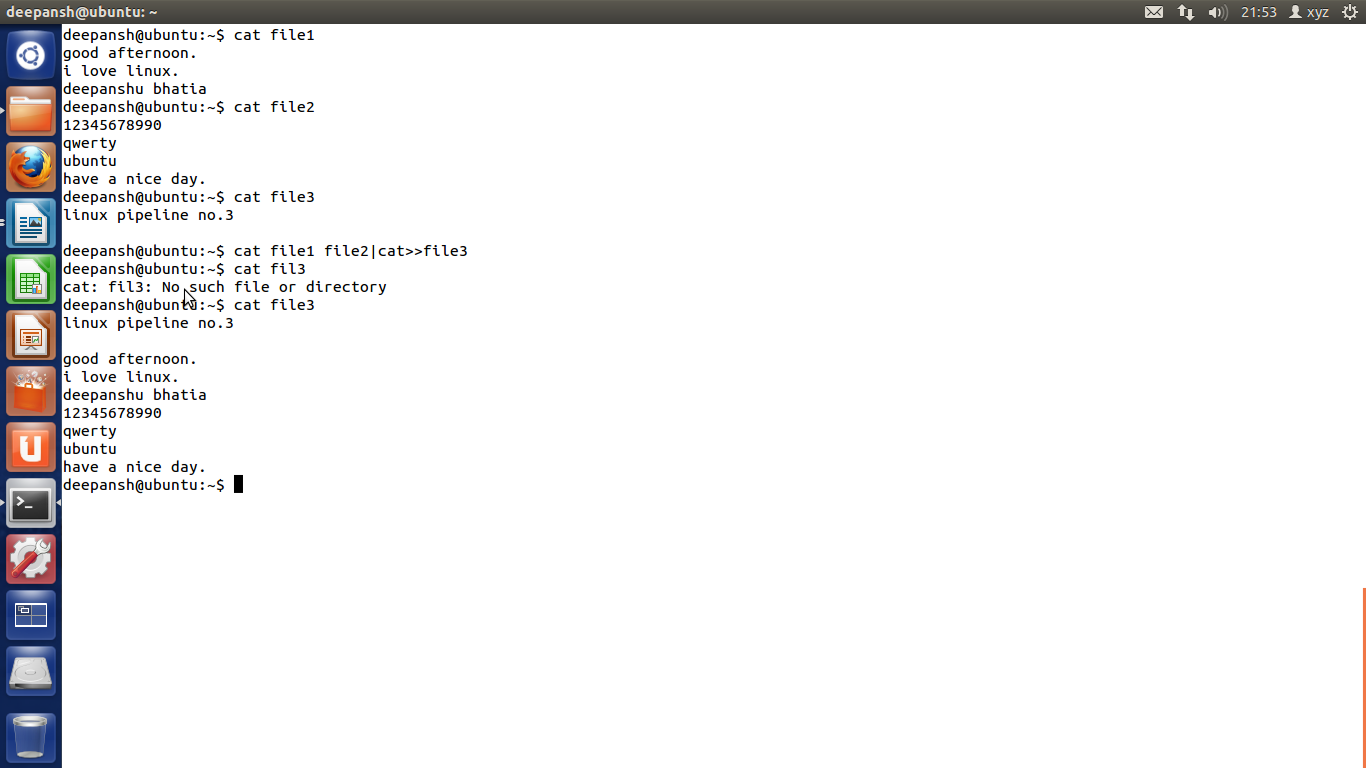
1. Output of ls should be displayed on the screen and from this output the lines containing word poem should be counted and the count should be stored in a file called file1.

$ ls|tee /dev/tty|grep –c poem>file1



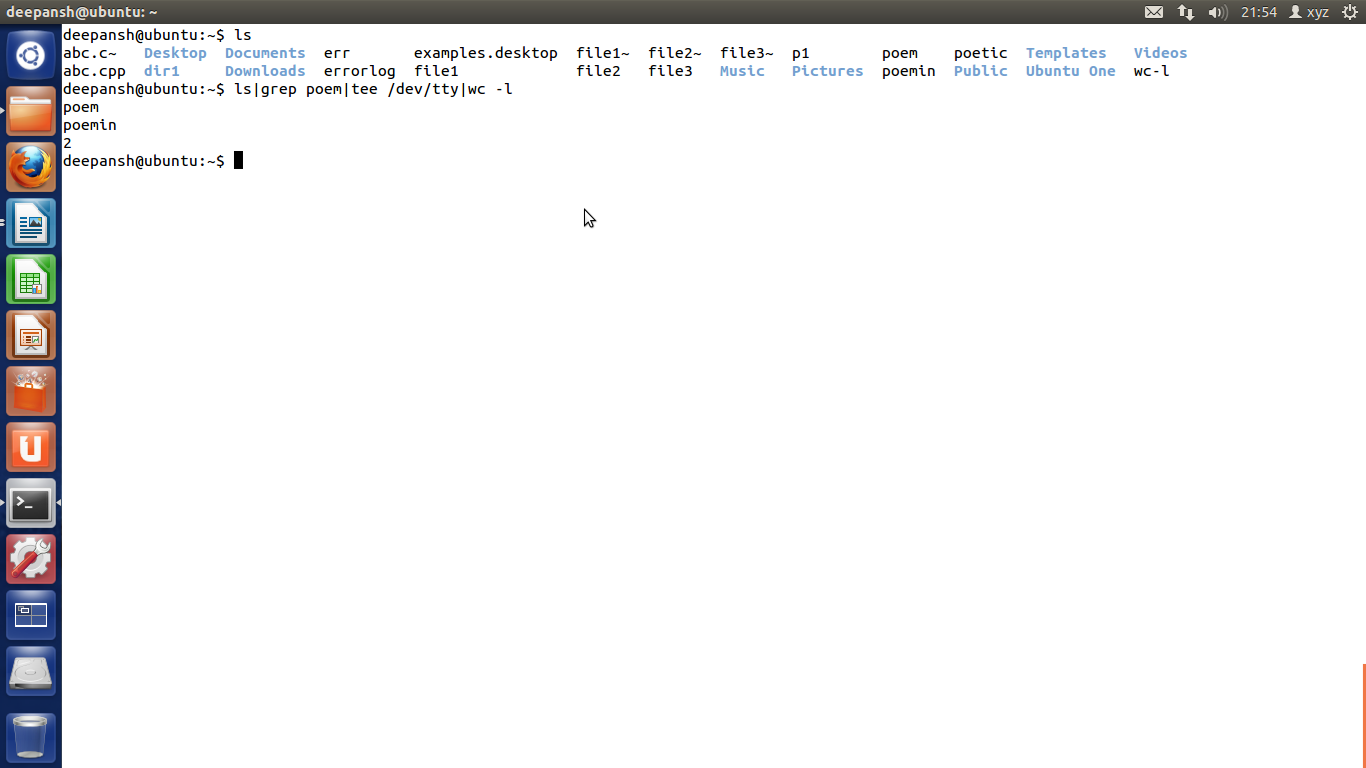
1. Contents of file1 and file2 should be displayed on the screen and this output should be appended to the file3.

$ cat file1 file2|cat>>file3



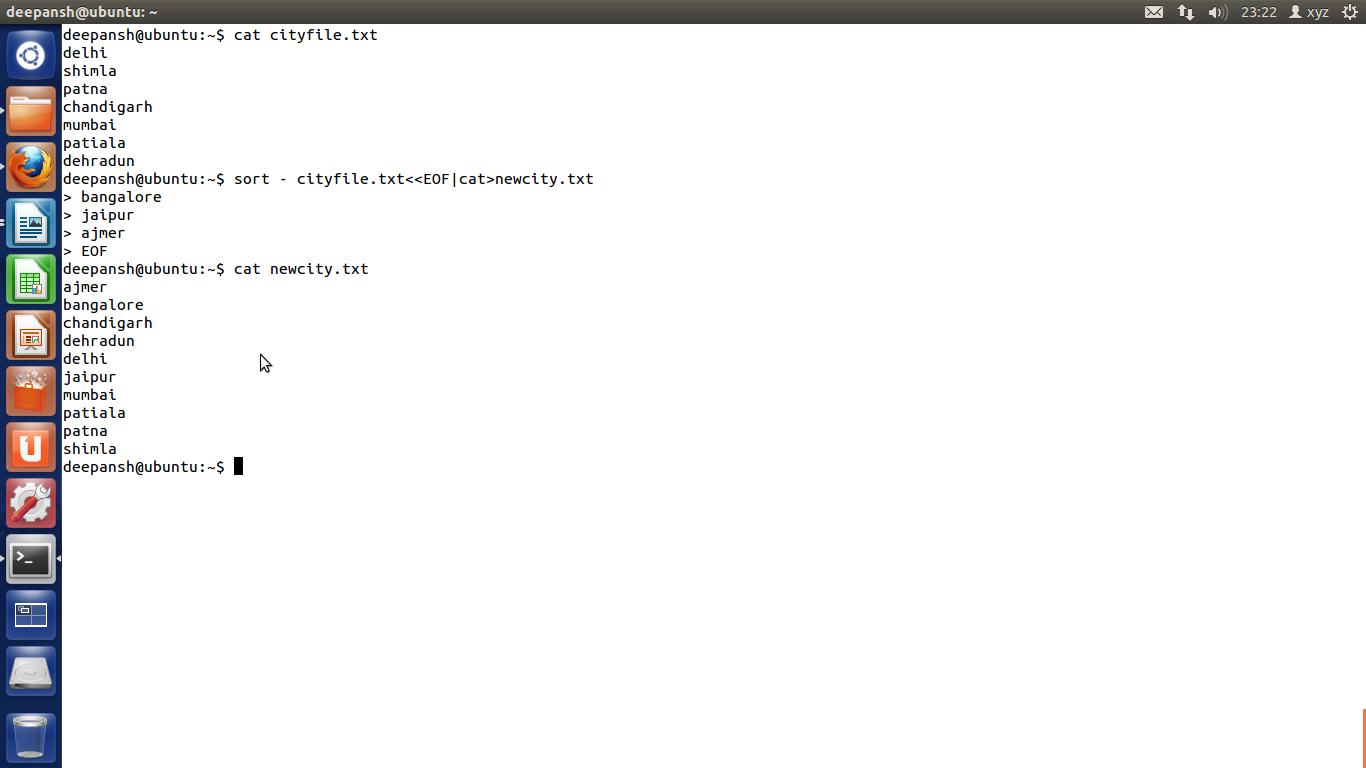
1. From output of ls the lines containing poem should be displayed on the screen along with the count.

$ ls|grep poem|tee /dev/tty|wc -l



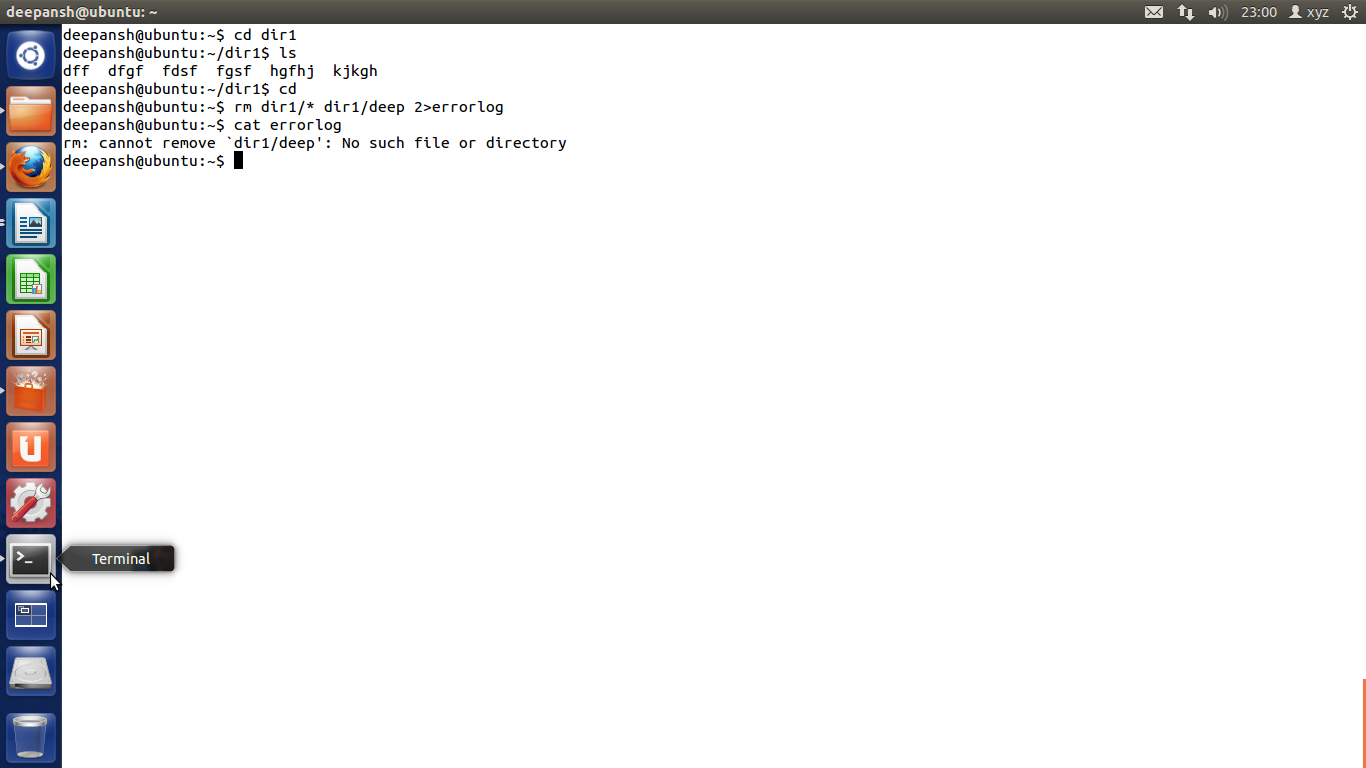
1. Name of cities should be accepted from the keyboard. This list of cities should be combined with the list of cities present in the file cityfile. This combined list should be sorted and the sorted output should be stored in a file newcity.

$ sort – cityfile.txt<<EOF|cat>newcity.txt



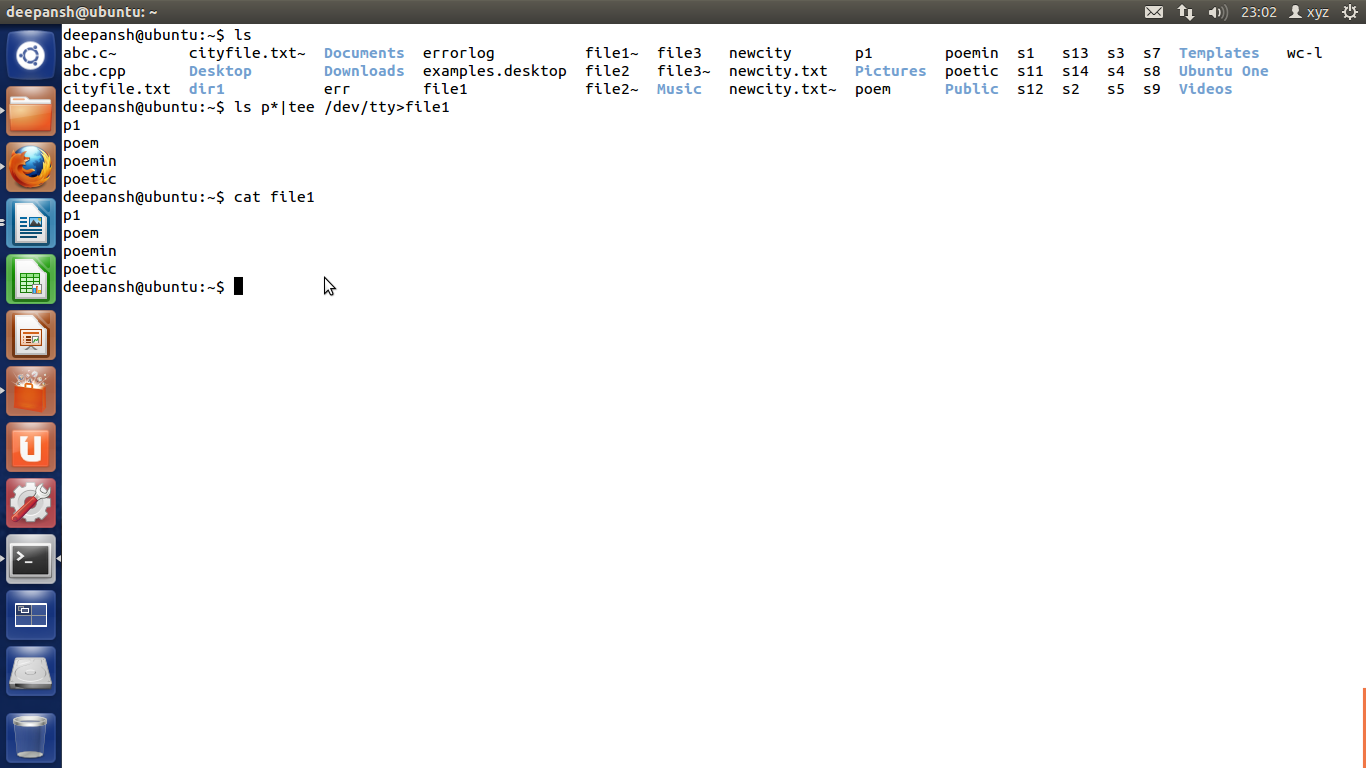
1. All files present in a directory dir1 should be deleted. Any error, if it occurs while carrying out this operation should be stored in a file errorlog.

$ rm dir1/\* 2>errorlog



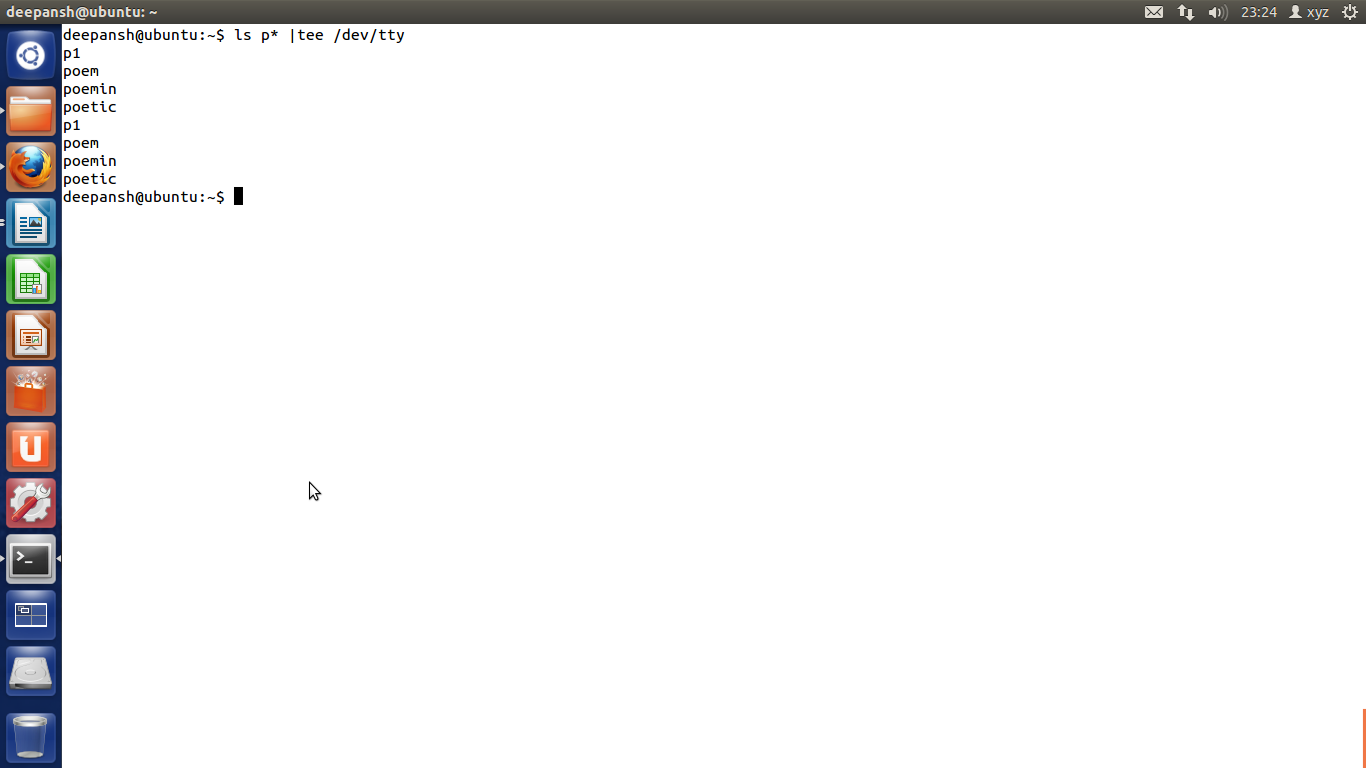
1. List of all files beginning with character P on the screen and also store them in a file called file1.

$ ls p\*|tee /dev/tty>file1



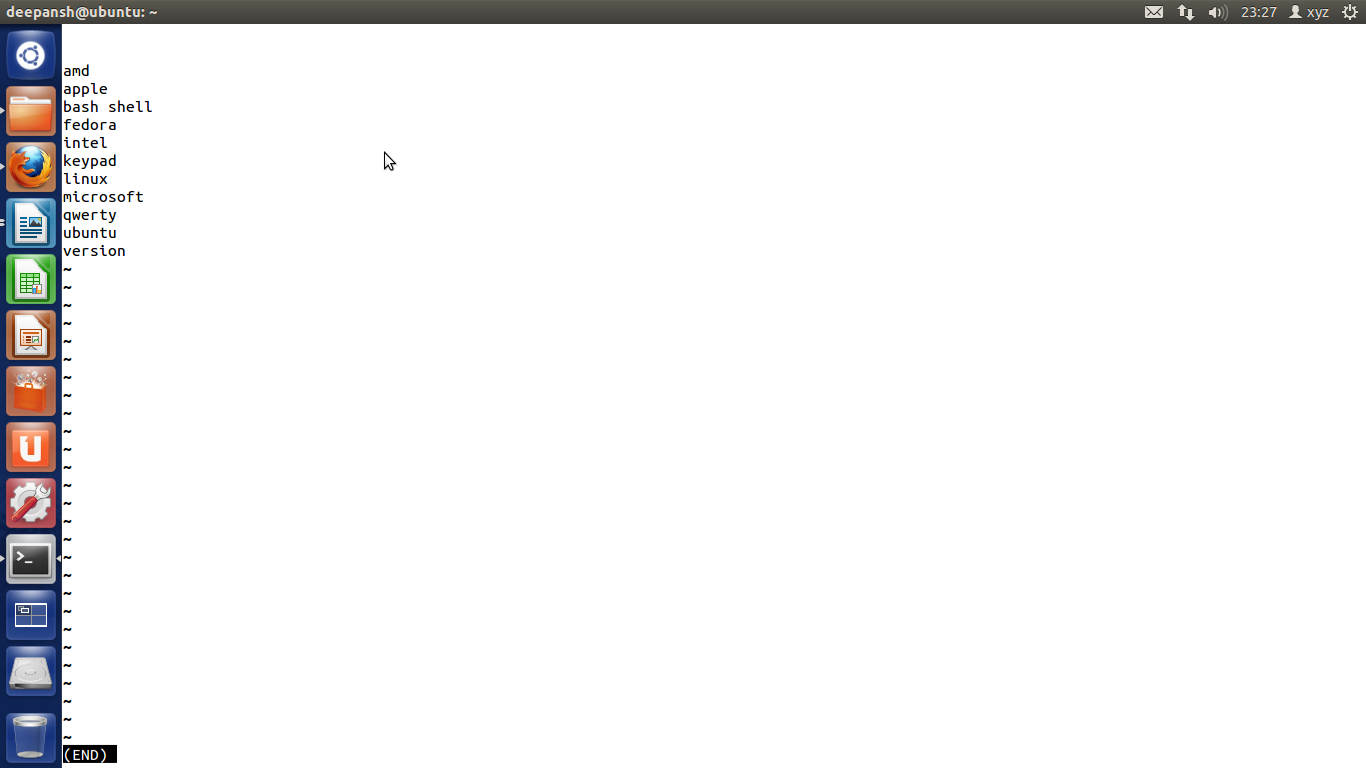
1. List of files beginning with character P on the screen twice in succession.

$ ls p\* |tee /dev/tty



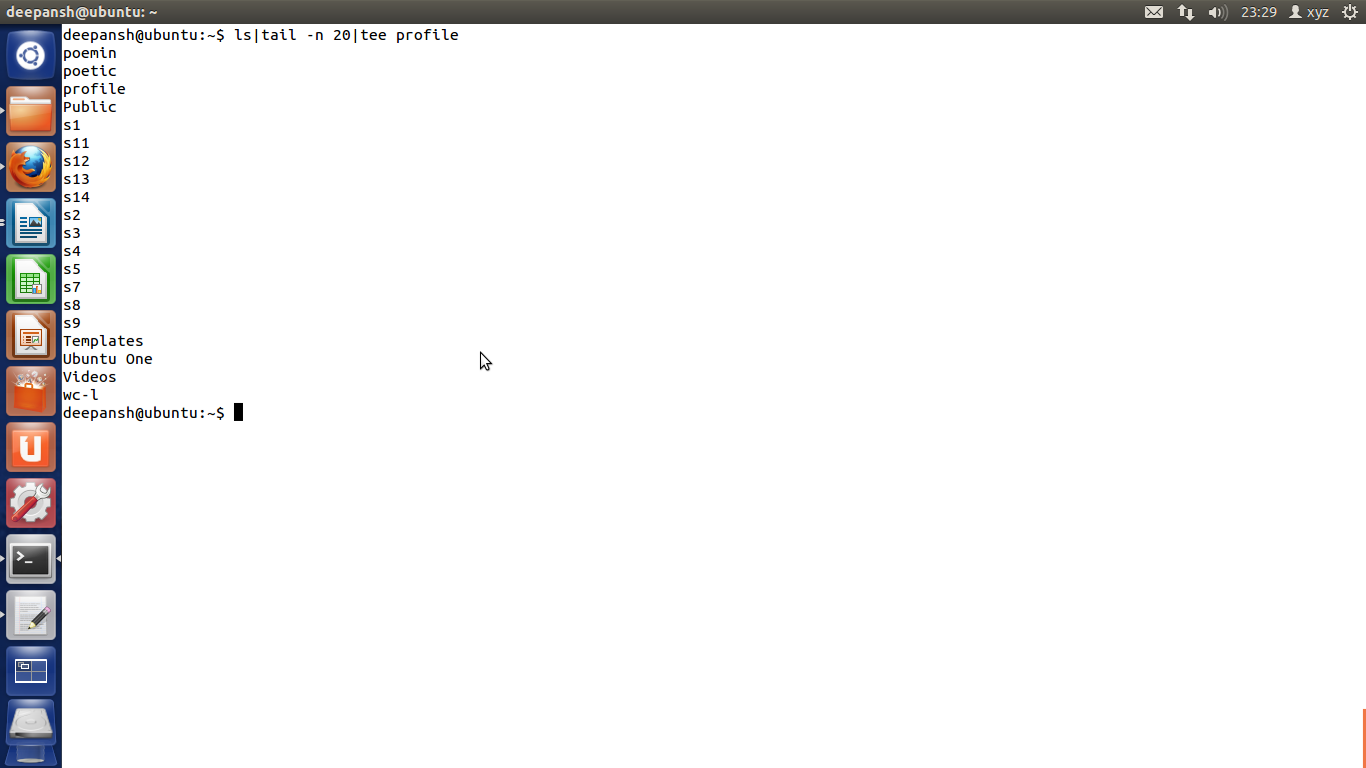
1. Merge the contents of the files a.txt, b.txt and c.txt, sort them and display the sorted output on the screen page by page.

$ sort a.txt b.txt c.txt|less



1. Display the list of last 20 files present in current directory. Also store them in a file called profile.

$ ls|tail –n 20|tee profile



1. Write a shell script which receives any year from the keyboard and determine whether the year is a leap year or not. If no argument is supplied the current year should be assumed.

echo Enter year:

read year

if [ -z $year ]

then

year=`date +%Y`

fi

num1=`echo $year % 4 | bc`

num2=`echo $year % 100 | bc`

if [ $num1 -eq 0 -a $num2 -ne 0 ]

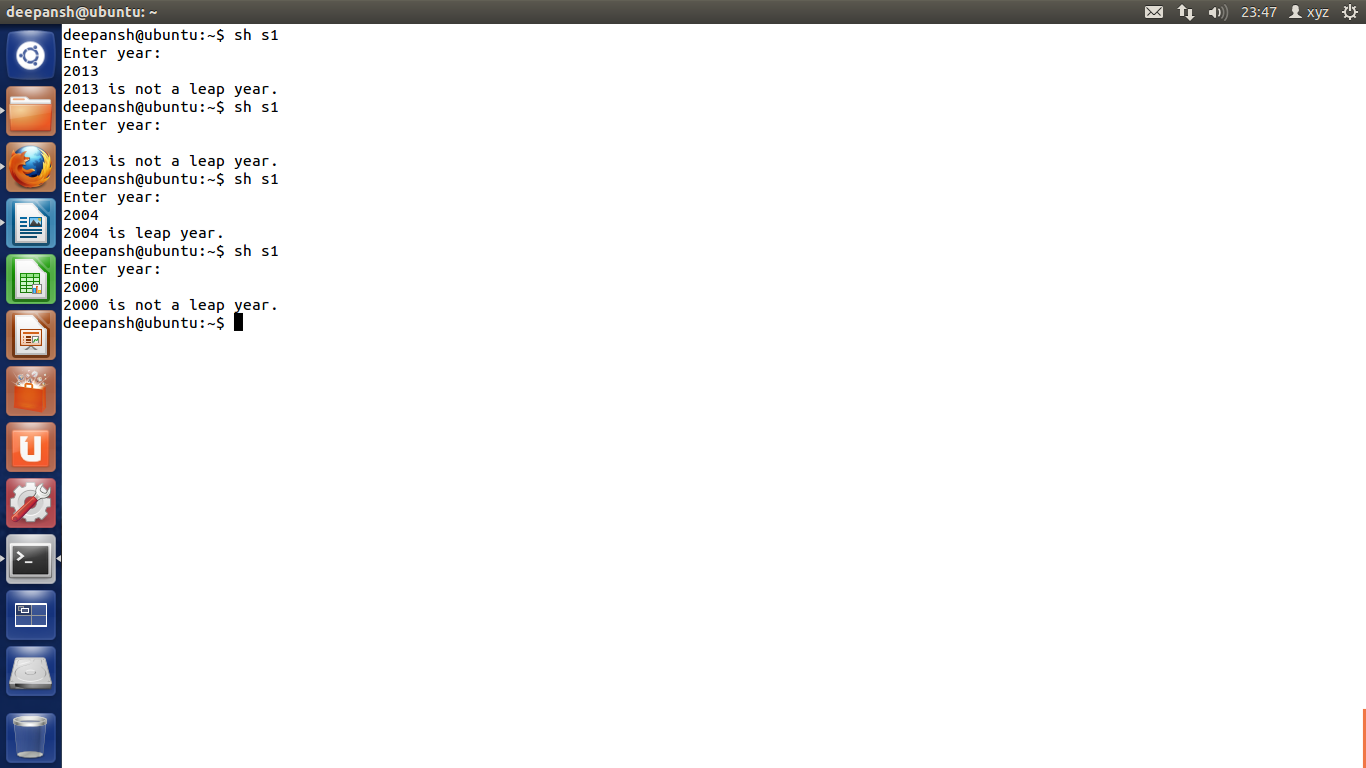
then

echo $year is leap year.

else

echo $year is not a leap year.

fi



1. Write a shell script which can receive an argument 'one', 'two', 'three'. If the argument supplied is 'one' display it in bold, if it is 'two' display it in reverse video and if it is 'three' make it blink on the screen.

echo one|cat>one

echo two|cat>two

echo three|cat>three

if [ `grep $1 one` ]

then

tput bold

echo one

elif [ `grep $1 two` ]

then

echo $1|rev

elif [ `grep $1 three` ]

then

tput blink

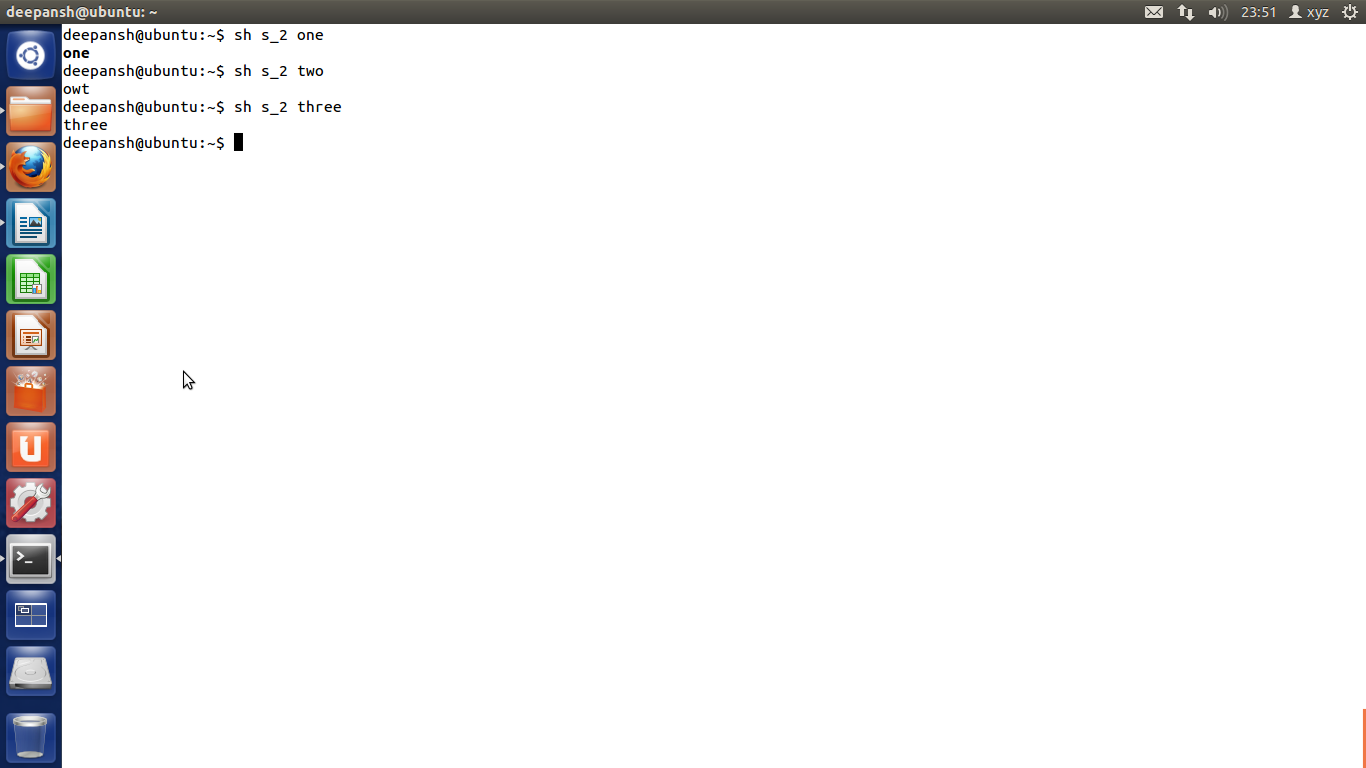
echo $1

else

echo wrong arguement

fi

tput sgr0



1. Write a shell script which get executed the moment the user logs in. it should display the message “Good Morning”/”Good Afternoon”/”Good Evening” depending upon the time at which user logs in.

check=`date +%H`

time=`date +%T`

echo $time

if [ $check -ge 06 -a $check -le 12 ]

then

echo "Good morning"

elif [ $check -ge 12 -a $check -le 16 ]

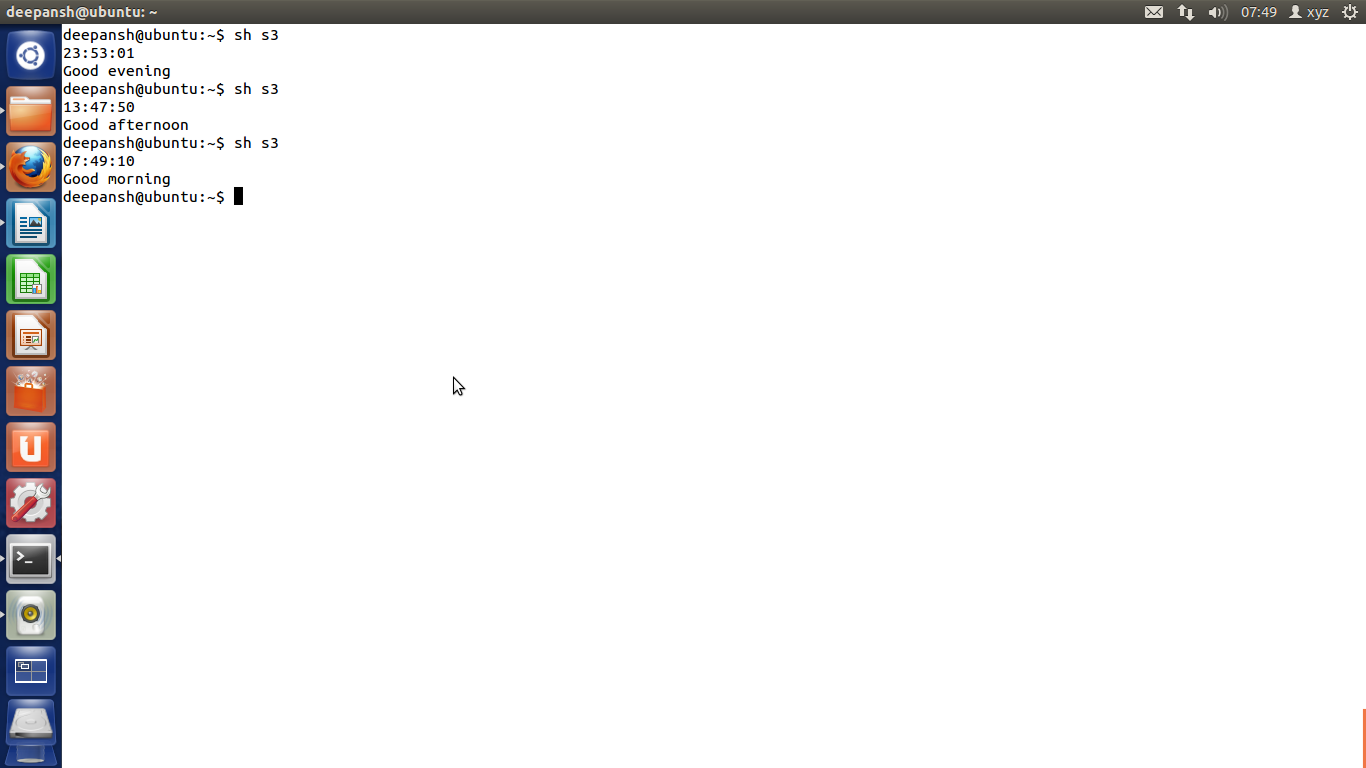
then

echo "Good afternoon"

else

echo "Good evening"

fi



1. Write a shell script to count and report the number of entries present in each sub-directory mentioned in the path which is supplied as a command-line argument.

sum=0

total=0

if [ $# -eq 0 ]

then

echo "Path value not passed"

else

cd $1

ls >fil.txt

t=`cat fil.txt`

for a in $t

do

if [ -d $a ]

then

total=`ls -l $a | wc -l`

total=`expr $total - 1`

sum=`expr $sum + $total`

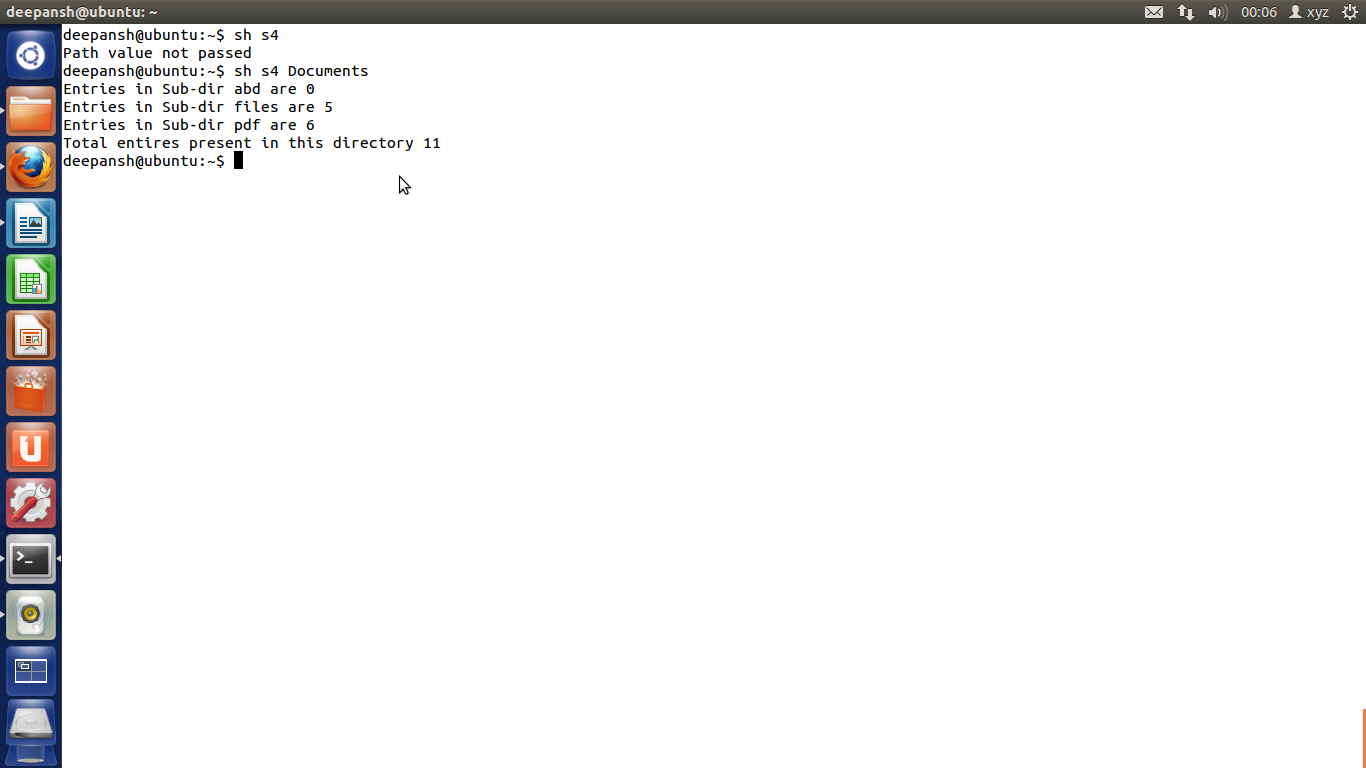
echo Entries in Sub-dir $a are $total

fi

done

echo Total entires present in this directory $sum

fi



1. Write a shell script to find the value of one number raised to the power of another number.

echo Input number

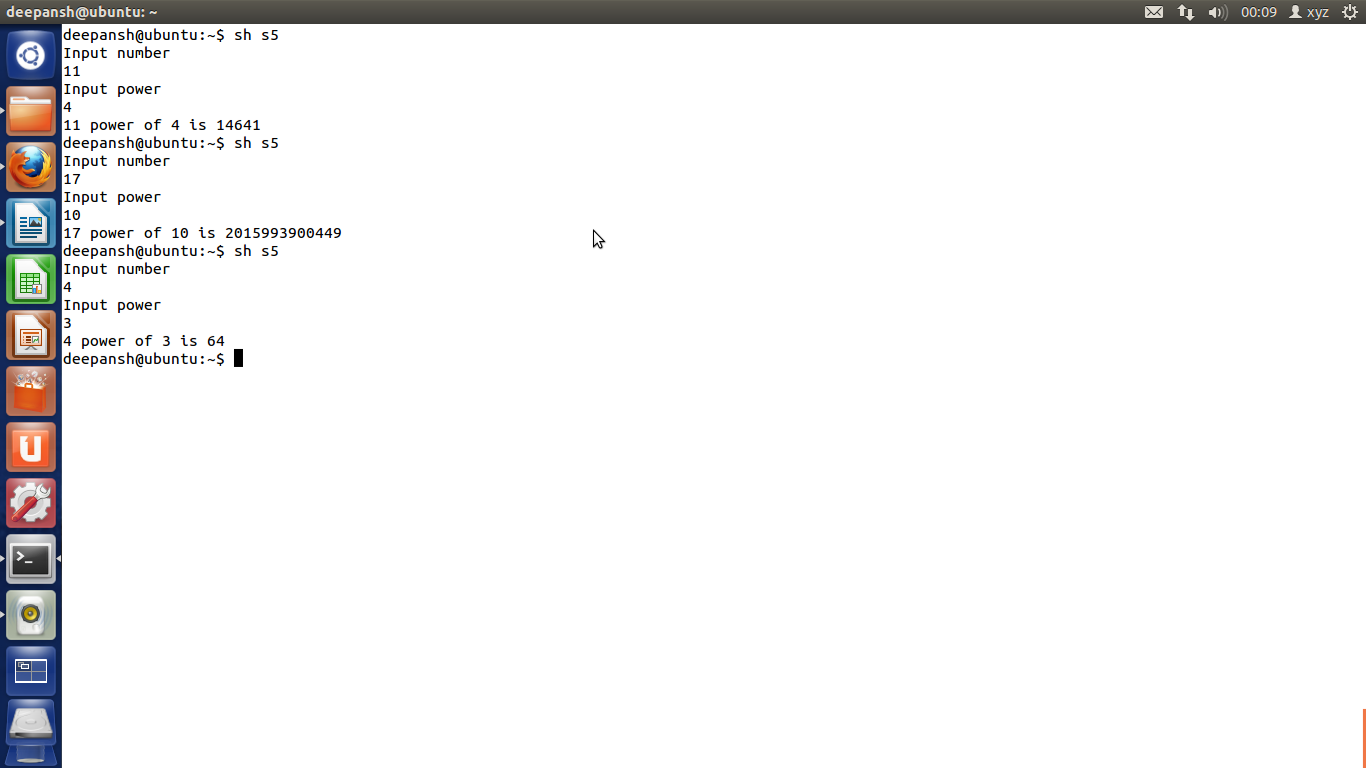
read no

echo Input power

read power

ans=`echo $no^$power|bc`

echo $no power of $power is $ans



1. Write a shell script which reports names and sizes of all files in a directory (directory would be supplied as an argument to the shell script) whose size is exceeding 1000 bytes. The file names should be printed in descending order of their sizes. The total number of such files should also be reported.

if [ $# -gt 0 ]

then

cd $1

mkdir Temp

for a in \*

do

size=`stat --format=%s "$a"`

if [ $size -gt 1000 ]

then

cp "$a" ./Temp/ 2>error

fi

done

ls -1S Temp/|tee /dev/tty|wc -l

rm -r Temp

rm error

fi



1. Write a shell script which checks whether your friend has logged in or not. If your friend is logged in then send a message to your friend else check the status of your friend after every one minute up to five minutes. The log name should be supplied to the shell script at command prompt.

echo Enter user name

read name

count=5

who >temp\_file

res=`grep -c $name <temp\_file`

while test $count -gt 0

do

if [ $res -ne 0 ]

then

echo User is logged in

echo Write your message:

read msg

echo Msg sent..!!

echo Your msg is:"\n"$msg

break

else

echo User is not logged in

sleep 60

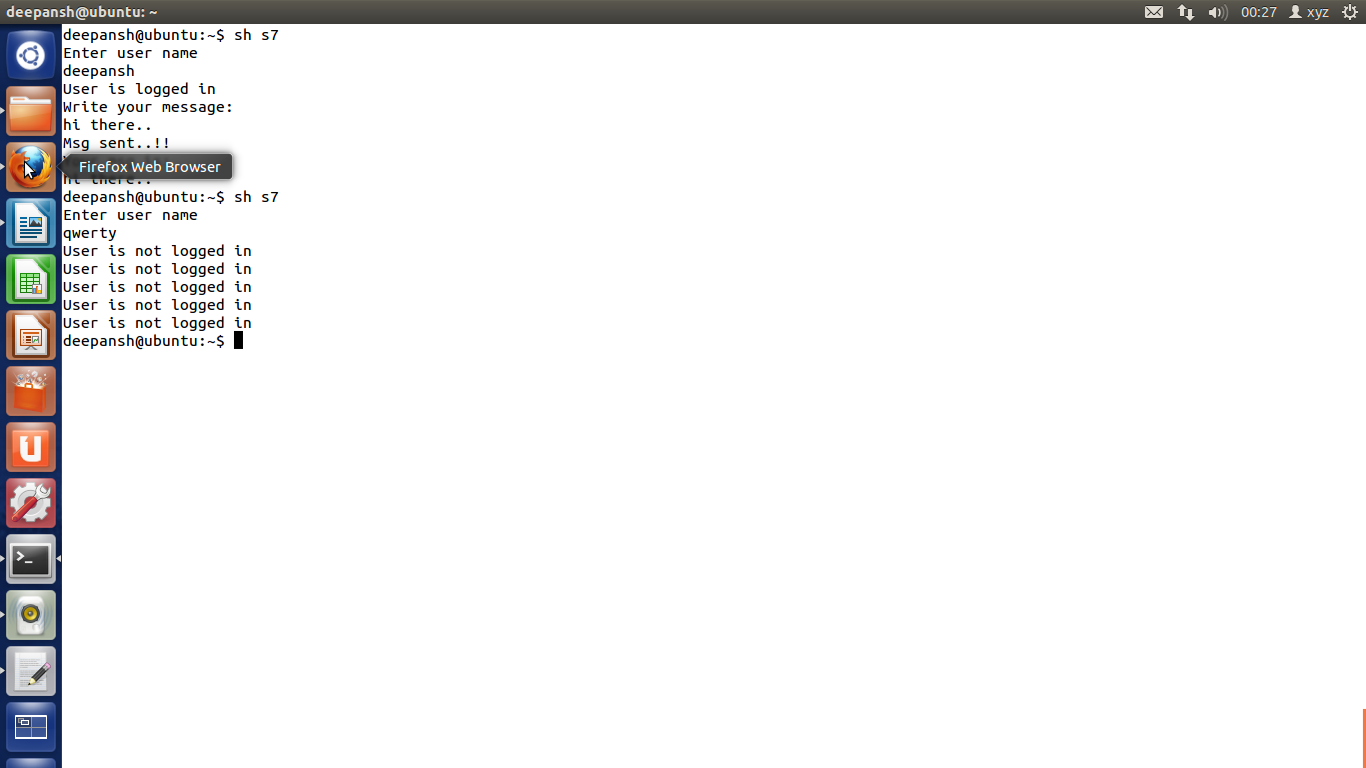
who >temp

res=`grep -c $name <temp`

count=`expr $count - 1`

fi

done



1. Write a shell script to print all prime numbers from 1 to 1000.

echo PRIME NUMBERS from 1-1000 are:

rng=1000

echo 2" \c"

j=3

while test $j -le $rng

do

i=2

x=`expr $j - 1`

while test $i -le $x

do

if [ `expr $j % $i` -ne 0 ]

then

i=`expr $i + 1`

else

break

fi

done

if [ $i -eq $j ]

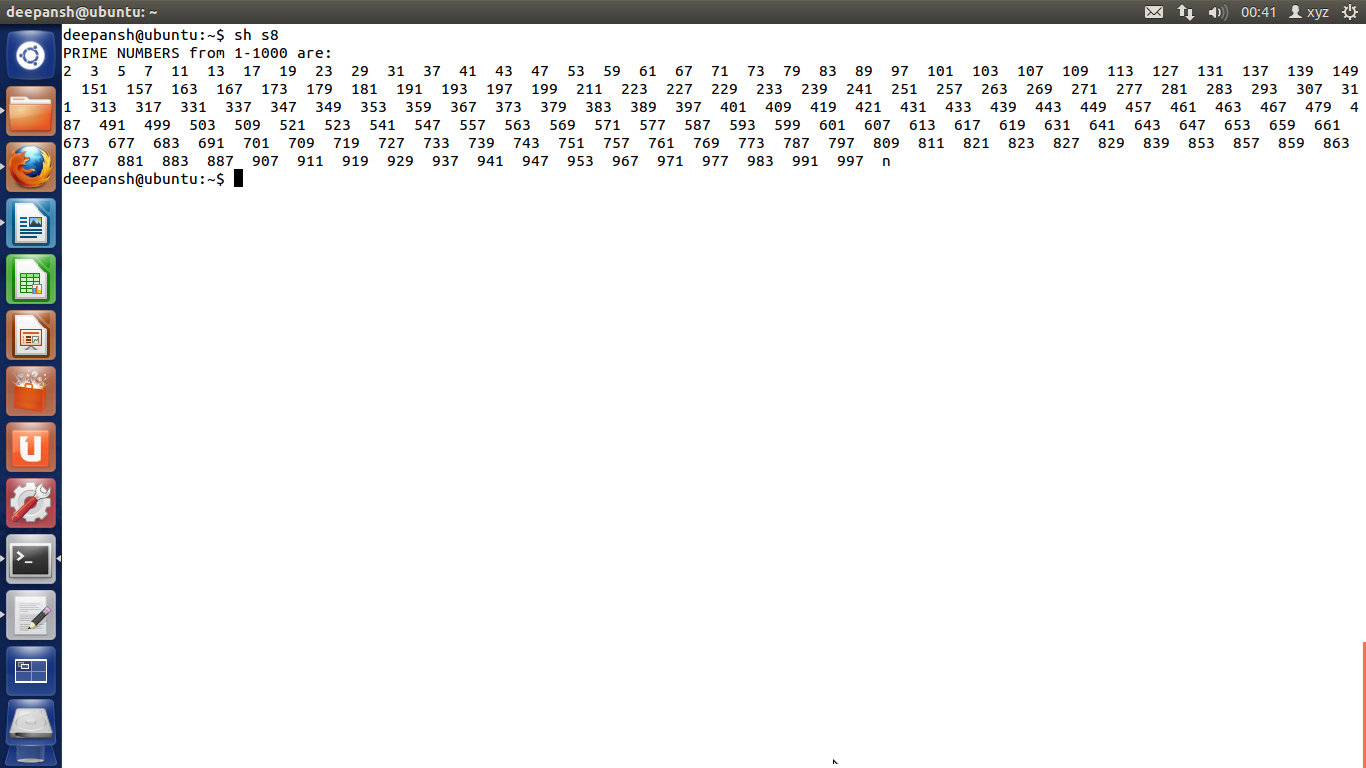
then

echo -n $j" "

fi

j=`expr $j + 1`

done



1. Write a shell script to generate all the combinations of 1,2 and 3.

echo "All combinations of 1 2 & 3 are:"

for i in 1 2 3

do

for j in 1 2 3

do

for k in 1 2 3

do

if test $i -ne $j -a $i -ne $k -a $j -ne $k

then

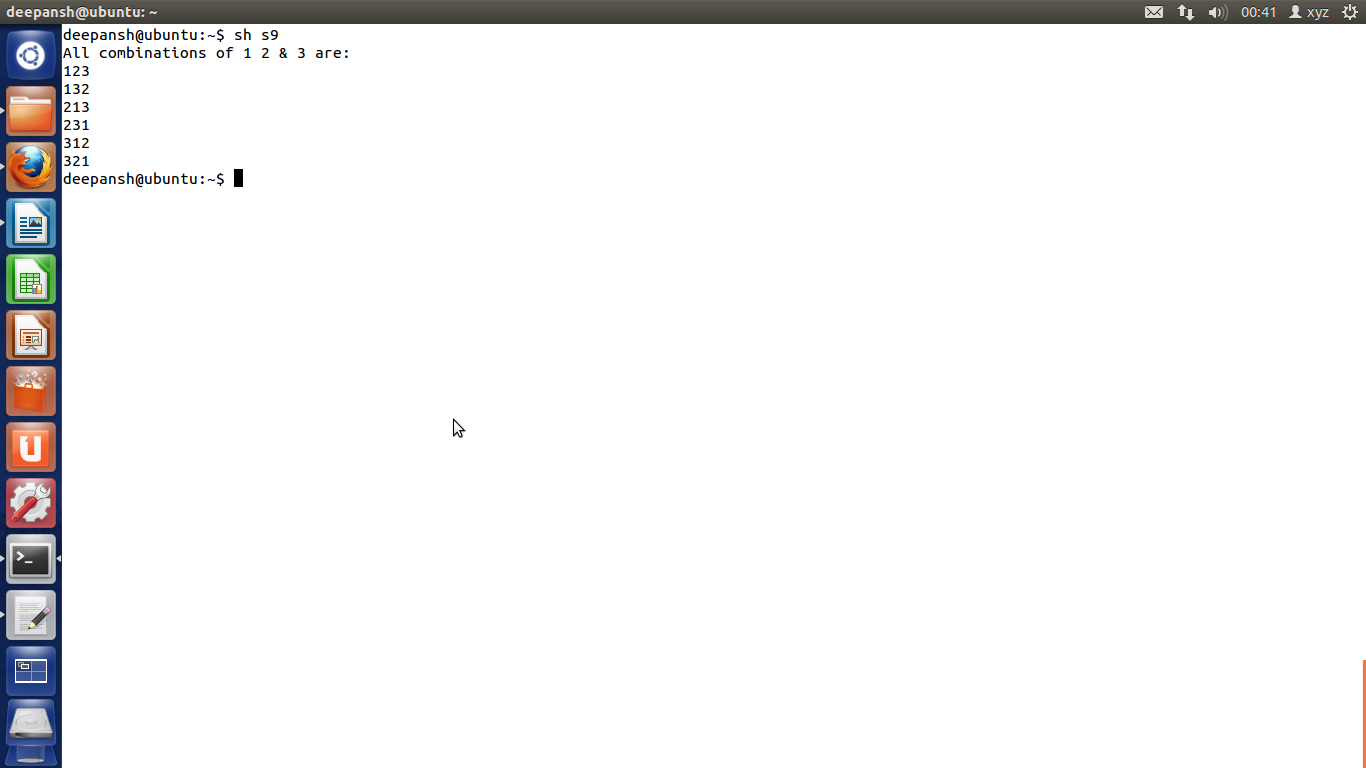
echo $i$j$k

fi

done

done

done



1. Write a shell script which will receive a list of file names, the first of which would be wordfile (wordfile is a file that consists of several words). The shell script should report all occurrence of each word in word file in the rest of the files supplied as arguments.

cp $1 temp

shift 1

for token in `cat temp`

do

echo Occurances of $token:"\c"

i=2

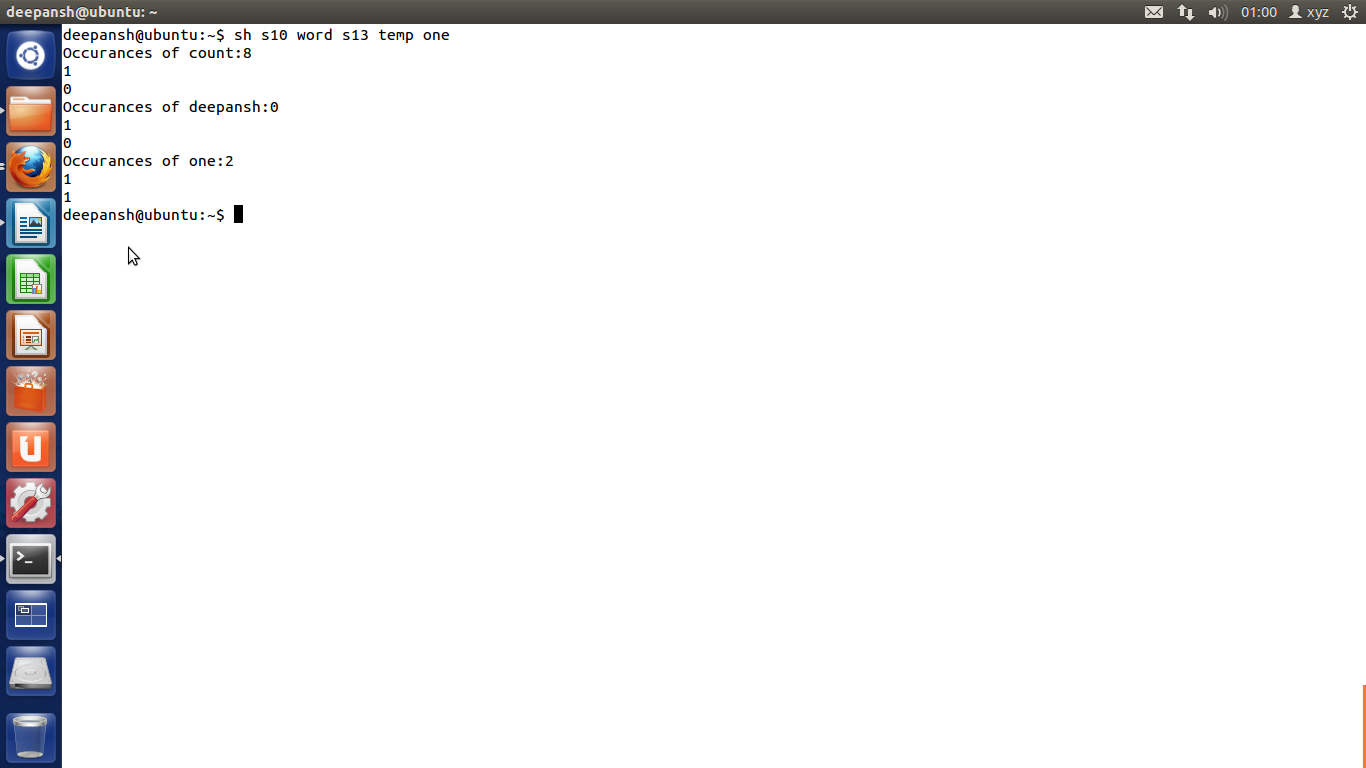
for i in $@

do

grep -c $token $i

done

done



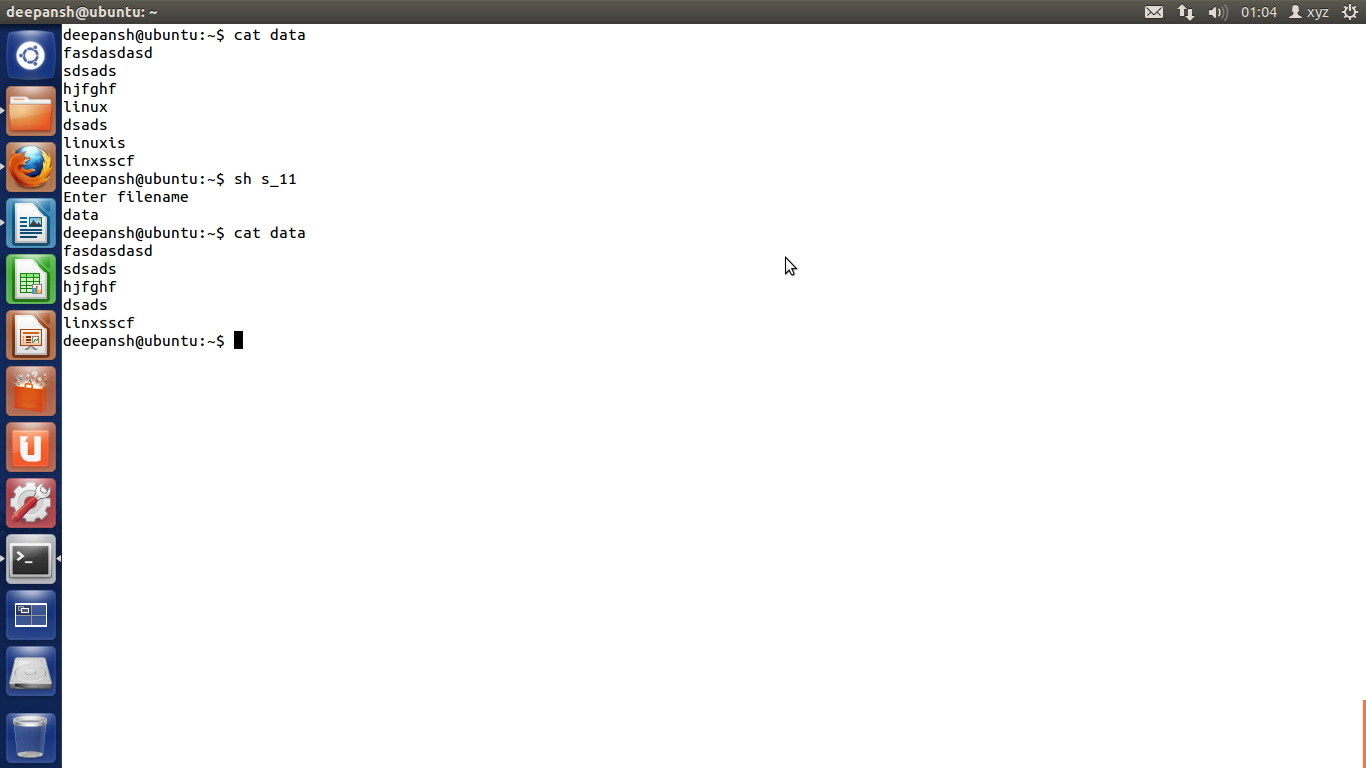
1. Write a shell script which deletes all lines containing the word linux in the files supplied as arguments to this shell script.

echo Enter filename

read name

grep -v linux $name|cat>tempname

mv tempname $name



1. Write a shell script which receives even number of file names. Suppose four file names are supplied then the first should get copied into second file, third should get copied into fourth file, and so on. If odd number of filenames are supplies then no copying should take place and an error message should be displayed.

file\_num=`expr $# % 2`

if [ $file\_num -ne 0 ]

then

echo "ERROR: Even Number Of Arguments required.!!"

else

cnt=1

while [ $cnt -lt $# ]

do

cp $1 $2

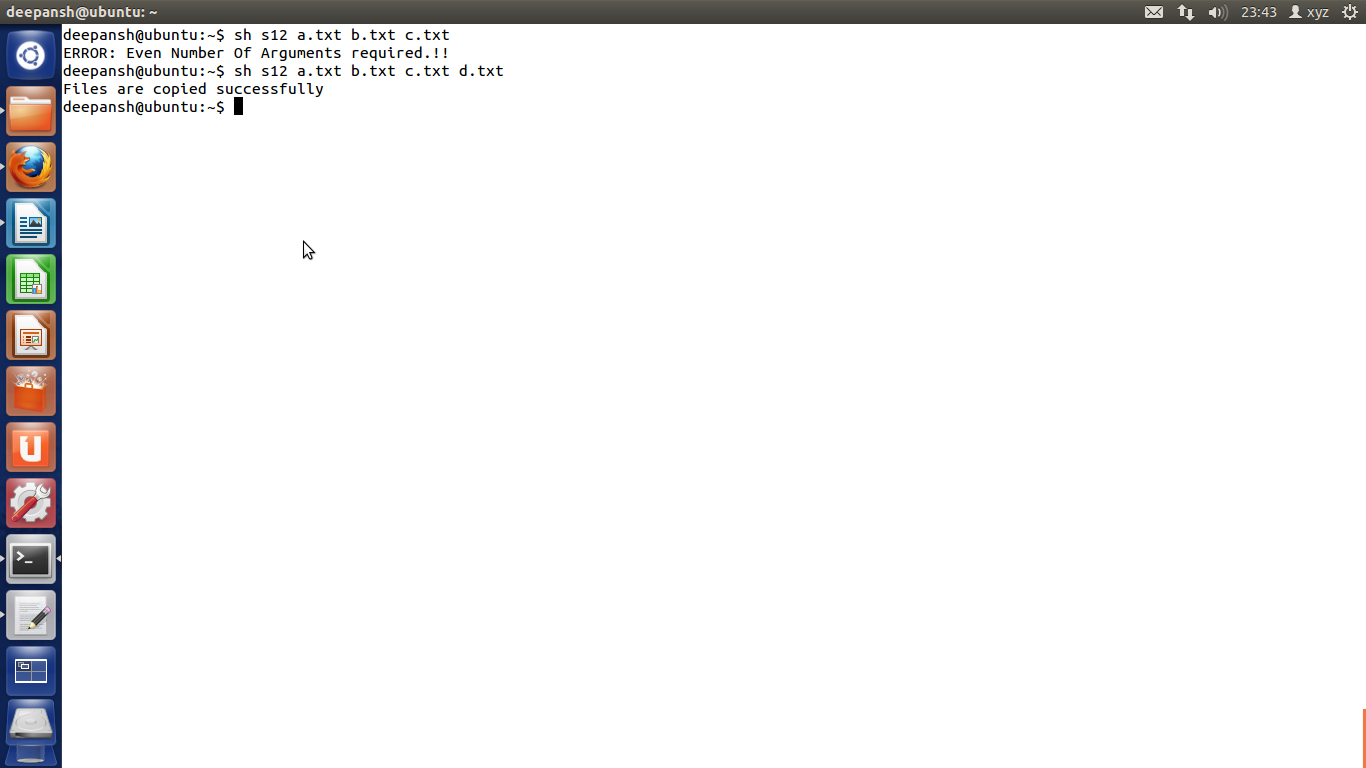
shift 2

cnt=`expr $cnt + 2`

done

echo Files are copied successfully

fi



1. Write a shell script which will receive any number of filenames as arguments. The shell script should check whether such files already exist. If they do, then it should be reported. If these files do not exist then check if a sub-directory called mydir exists in the current directory. If it doesn't exist then it should be created and in it the files supplied as arguments should get created. If mydir already exists then it should be reported alonf with the number of files that are currently present in mydir.

count=0

for i in $\*

do

if [ -f $i ]

then

echo "File $i is already exist."

else

if [ -d mydir -a $count -lt 1 ]

then

count=0

else

if [ $count -eq 0 ]

then

mkdir mydir

count=1

echo DIRECTORY "mydir" has been made

fi

cd mydir

> $i

echo file $i made

cd ..

fi

fi

done

if [ $count -eq 0 ]

then

cd mydir

for j in `ls`

do

if [ -f $j ]

then

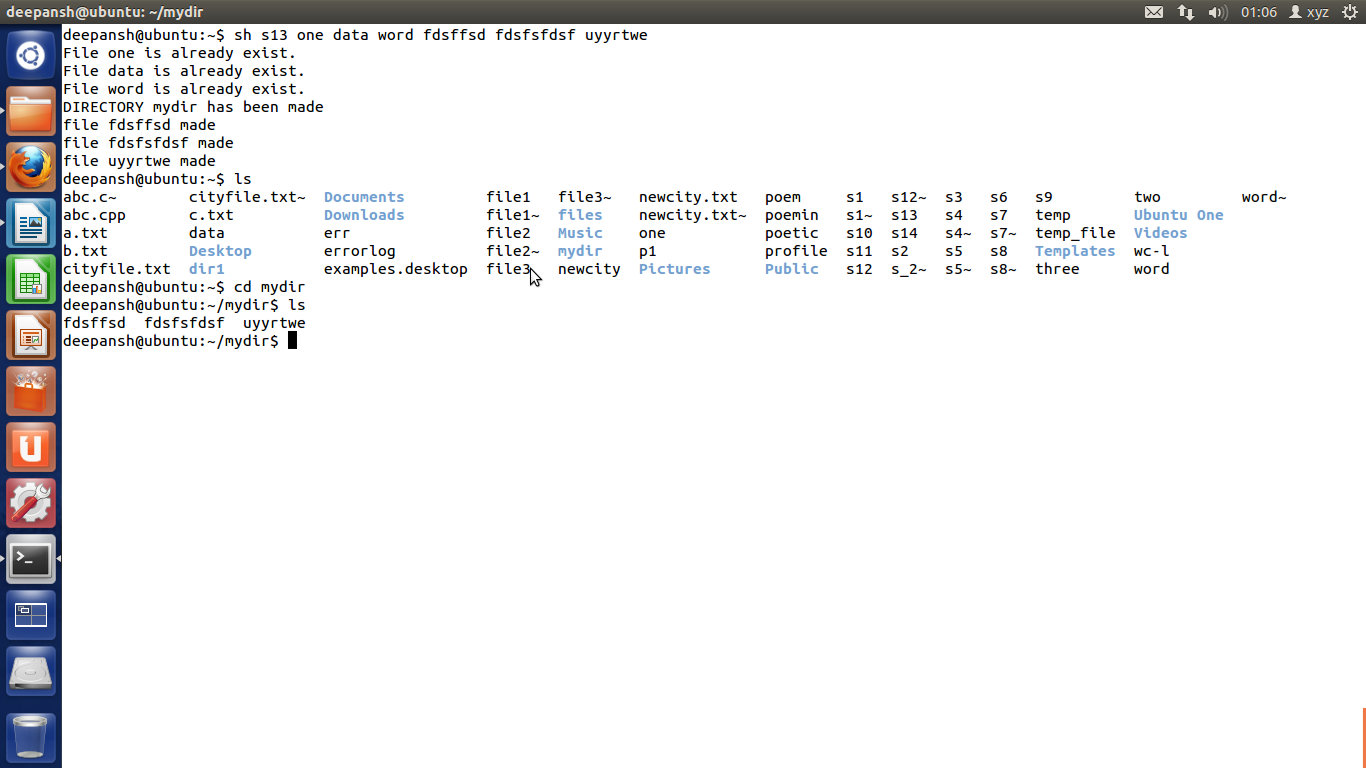
count=`expr $count + 1`

fi

done

echo "Mydir already exist...contains $count files"

fi



1. Write a shell script which works similar to the wc command. This script can receive the option -l, -w, -c to indicate whether number of lines, number of words, or number of characters from the input stream are to be counted. The user may use any or all of these options. Your script should be intelligent to identify invalid options and reject them.

l=0

w=0

c=0

if [ $# -eq 0 ]

then

echo arguement required

exit

fi

echo Enter the Text. To Stop TYPE end

k=1

ch=1

while [ $ch -gt 0 ]

do

read data

if [ "$data" = "end" ]

then

break

fi

if [ $k -eq 1 ]

then

echo $data>file2

k=`expr $k + 1`

else

echo $data>>file2

fi

l=`expr $l + 1`

for a in $data

do

if [ "$a" != "end" ]

then

w=`expr $w + 1`

else

echo yes

ch=0

break

fi

done

done

c=`ls -l file2|cut -f 5 -d ' '`

c=`expr $c - $l`

if [ $ch -eq 0 ]

then

c=`expr $c - 3`

fi

for b in $@

do

case $b in

-l) echo number of lines is $l

;;

-w) echo number of words is $w

;;

-c) echo number of characters is $c

;;

\*) echo WRONG CHOICE..!!

esac

done



1. Write a function mkcd() which would create all the directories present in the path supplied to it as argument and change over to the last directory in this path. Thus,

$ mkcd a1/a2/a3/a4/a5

should create the five nested directories and change the present working directory to a5.

#run the shell script as . script path

#it will run the script in current shell

mkcd()

{

dir=$1

mkdir -p $dir

cd $dir

pwd

}

if [ $# -gt 0 ]

then

mkcd $1

else

echo Path required

fi

