Name:Prathamesh Sudhir Paraswar

class:TE9

Batch:L9

Roll No:33155.

1)Echo.

NAME

echo - display a line of text

SYNOPSIS

echo [SHORT-OPTION]... [STRING]...

echo LONG-OPTION

DESCRIPTION

Echo the STRING(s) to standard output.

-n do not output the trailing newline

-e enable interpretation of backslash escapes

-E disable interpretation of backslash escapes (default)

--help display this help and exit

--version

output version information and exit

If -e is in effect, the following sequences are recognized:

\\ backslash

\a alert (BEL)

\b backspace

\c produce no further output

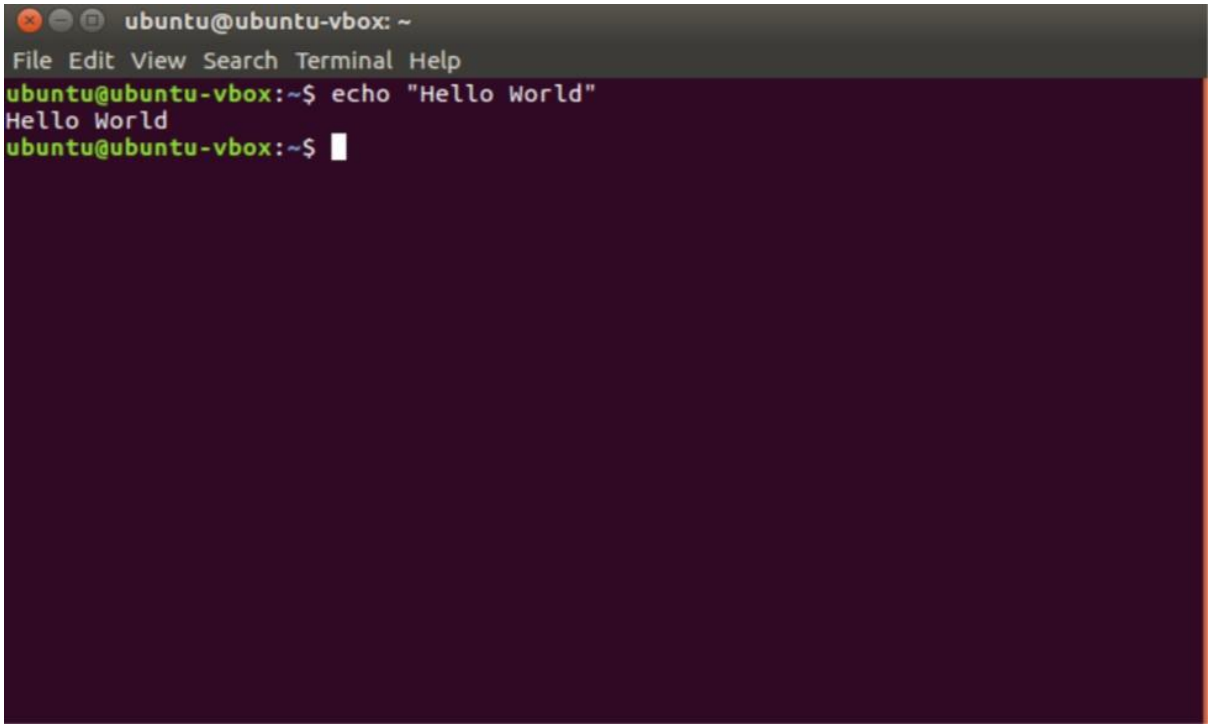
\e escape

\f form feed

\n new line

\r carriage return

\t horizontal tab



2)ls.

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-

fied.

Mandatory arguments to long options are mandatory for short options too.

-a, --all

do not ignore entries starting with .

-A, --almost-all

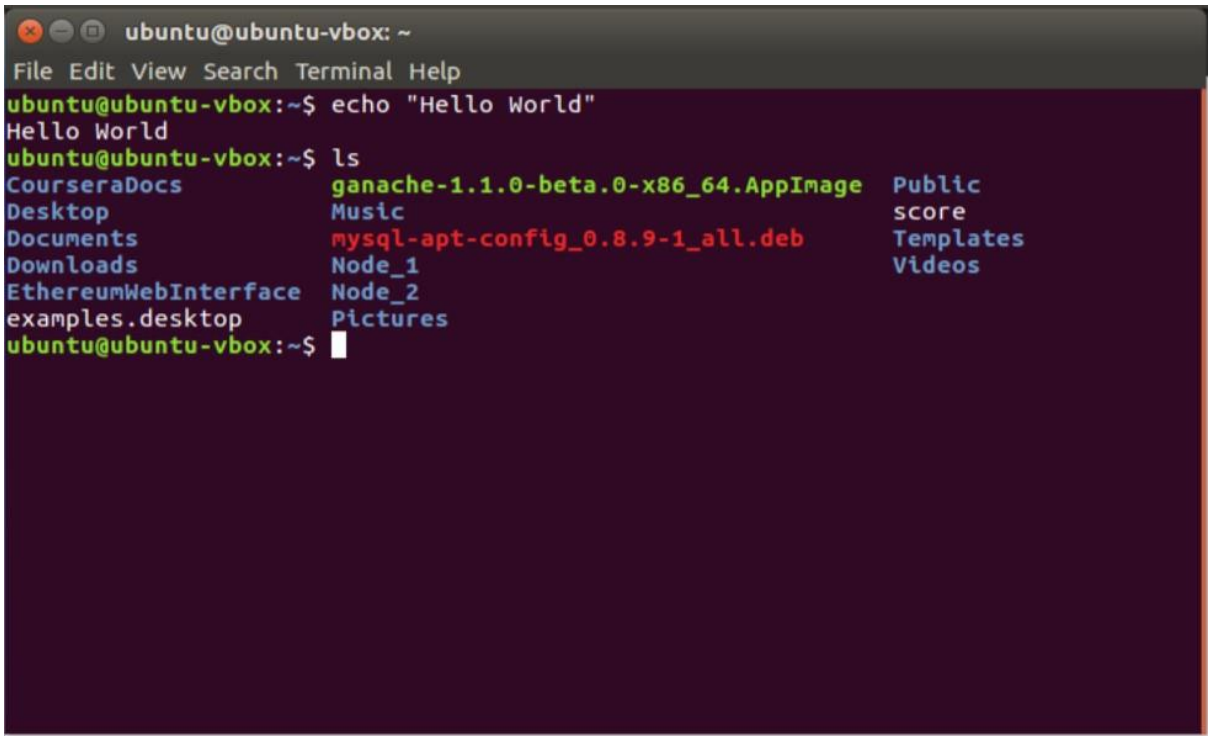
do not list implied . and ..

--author

with -l, print the author of each file

-b, --escape

print C-style escapes for nongraphic characters



3) read.

NAME

read - read from a file descriptor

SYNOPSIS

#include <unistd.h>

ssize_t read(int fd, void *buf, size_t count);

DESCRIPTION

read() attempts to read up to count bytes from file descriptor fd into the buffer starting at buf.

On files that support seeking, the read operation commences at the current file offset, and the file offset is incremented by the number of

bytes read. If the current file offset is at or past the end of file, no bytes are read, and read() returns zero.

If count is zero, read() may detect the errors described below. In the absence of any errors, or if read() does not check for errors, a

read() with a count of 0 returns zero and has no other effects.

If count is greater than SSIZE_MAX, the result is unspecified.

```
ubuntu@ubuntu-vbox:~$ read -p 'name'
namePrathamesh paraswar
ubuntu@ubuntu-vbox:~$ echo name
name
ubuntu@ubuntu-vbox:~$ echo $name
Prathamesh Paraswar
ubuntu@ubuntu-vbox:~$
```

4)cat.

NAME

cat - concatenate files and print on the standard output

SYNOPSIS

cat [OPTION]... [FILE]...

DESCRIPTION

Concatenate FILE(s) to standard output.

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

-A, --show-all

equivalent to -vET

-b, --number-nonblank

number nonempty output lines, overrides -n

-e equivalent to -vE

-E, --show-ends

display $ at end of each line

-n, --number

number all output lines

-s, --squeeze-blank

suppress repeated empty output lines

-t equivalent to -vT

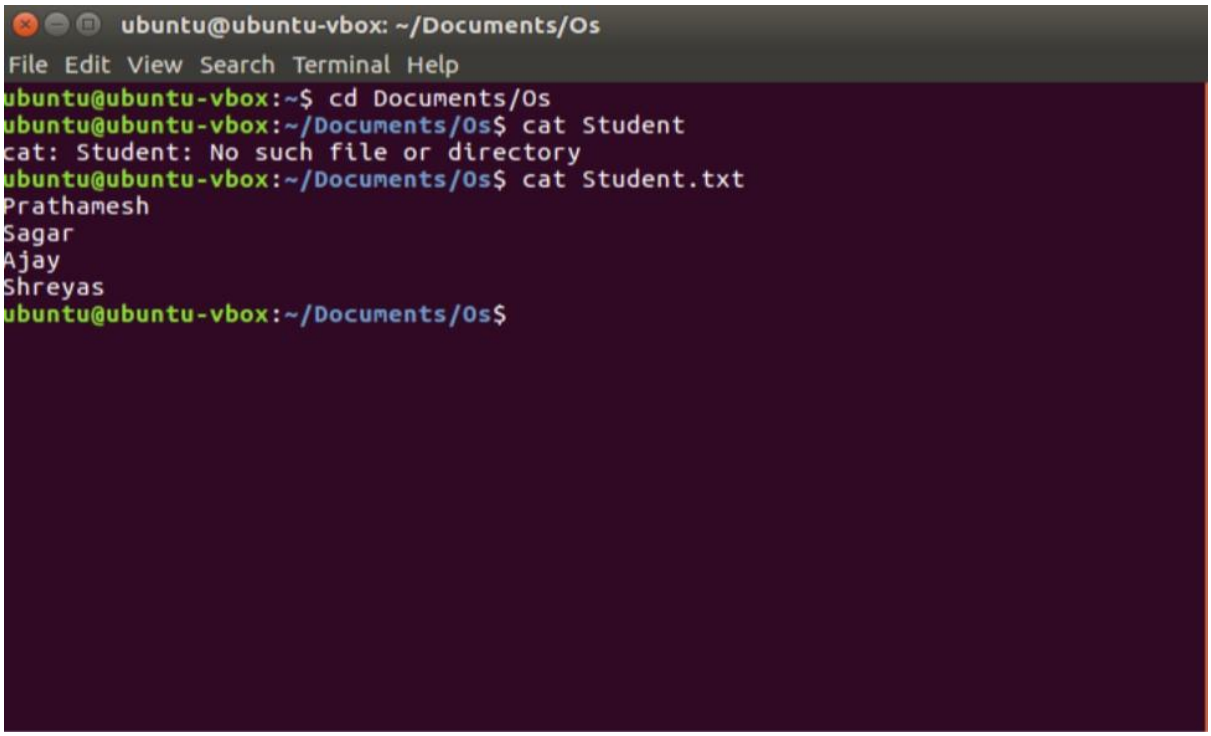
-T, --show-tabs

display TAB characters as ^I

-u (ignored)

-v, --show-nonprinting

use ^ and M- notation, except for LFD and TAB



5)touch.

NAME

touch - change file timestamps

SYNOPSIS

touch [OPTION]... FILE...

DESCRIPTION

Update the access and modification times of each FILE to the current time.

A FILE argument that does not exist is created empty, unless -c or -h is supplied.

A FILE argument string of - is handled specially and causes touch to change the times of the file associated with standard output.

Mandatory arguments to long options are mandatory for short options too.

-a change only the access time

-c, --no-create

do not create any files

-d, --date=STRING

parse STRING and use it instead of current time

-f (ignored)

-h, --no-dereference

affect each symbolic link instead of any referenced file (useful only on systems that can change the timestamps of a symlink)

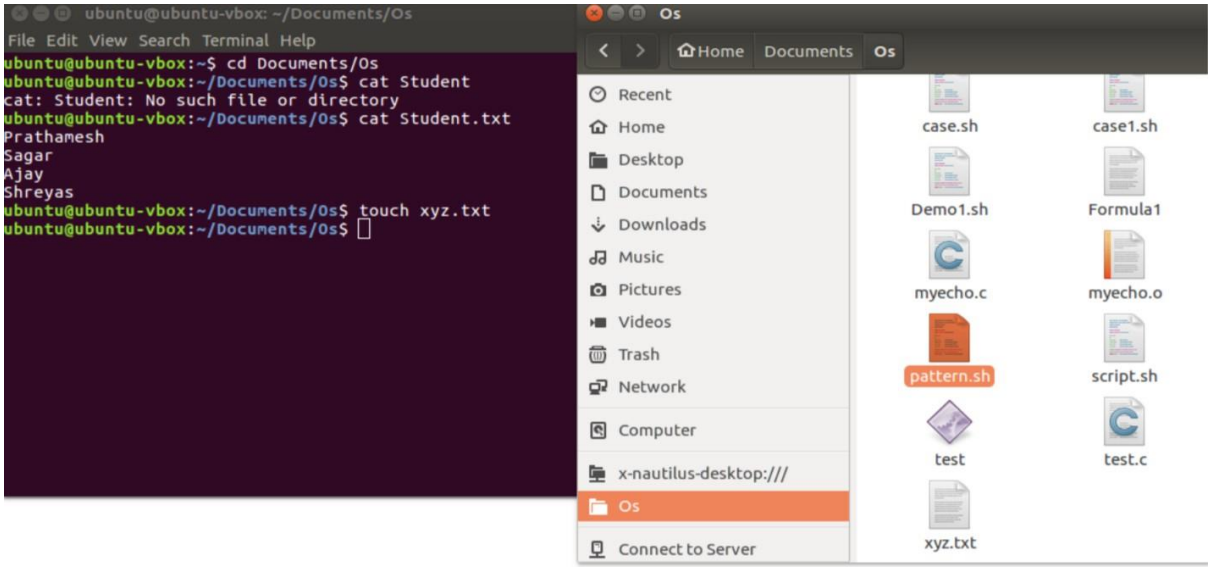
-m change only the modification time

-r, --reference=FILE

use this file's times instead of current time

-t STAMP

use [[CC]YY]MMDDhhmm[.ss] instead of current time



6)test.

NAME

test - check file types and compare values

SYNOPSIS

test EXPRESSION

test

[EXPRESSION]

[ ]

[OPTION

DESCRIPTION

Exit with the status determined by EXPRESSION.

--help display this help and exit

--version

output version information and exit

An omitted EXPRESSION defaults to false. Otherwise, EXPRESSION is true or false and sets exit status. It is one of:

(EXPRESSION)

EXPRESSION is true

! EXPRESSION

EXPRESSION is false

EXPRESSION1 -a EXPRESSION2

both EXPRESSION1 and EXPRESSION2 are true

EXPRESSION1 -o EXPRESSION2

either EXPRESSION1 or EXPRESSION2 is true

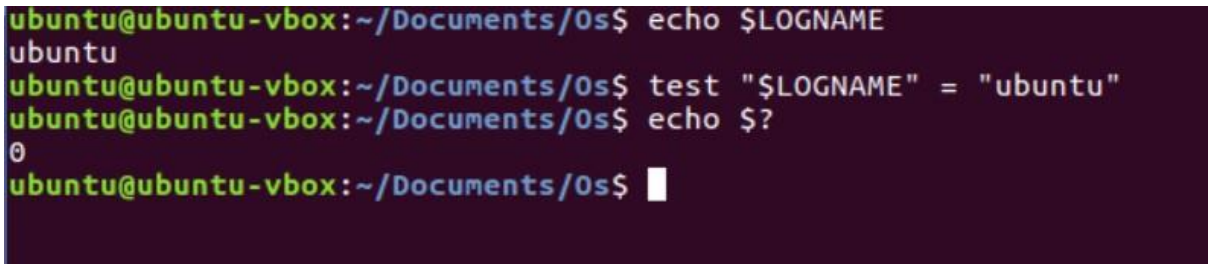
-n STRING

the length of STRING is nonzero

STRING equivalent to -n STRING

-z STRING

the length of STRING is zero



7)grep.

NAME

grep, egrep, fgrep, rgrep - print lines matching a pattern

SYNOPSIS

grep [OPTIONS] PATTERN [FILE...]

grep [OPTIONS] [-e PATTERN]... [-f FILE]... [FILE...]

DESCRIPTION

grep searches the named input FILEs for lines containing a match to the given PATTERN. If no files are specified, or if the file "-" is

given, grep searches standard input. By default, grep prints the matching lines.

In addition, the variant programs egrep, fgrep and rgrep are the same as grep -E, grep -F, and grep -r, respectively. These variants are

deprecated, but are provided for backward compatibility.

OPTIONS

Generic Program Information

--help Output a usage message and exit.

-V, --version

Output the version number of grep and exit.

Matcher Selection

-E, --extended-regexp

Interpret PATTERN as an extended regular expression (ERE, see below).

-F, --fixed-strings

Interpret PATTERN as a list of fixed strings (instead of regular expressions), separated by newlines, any of which is to be matched.

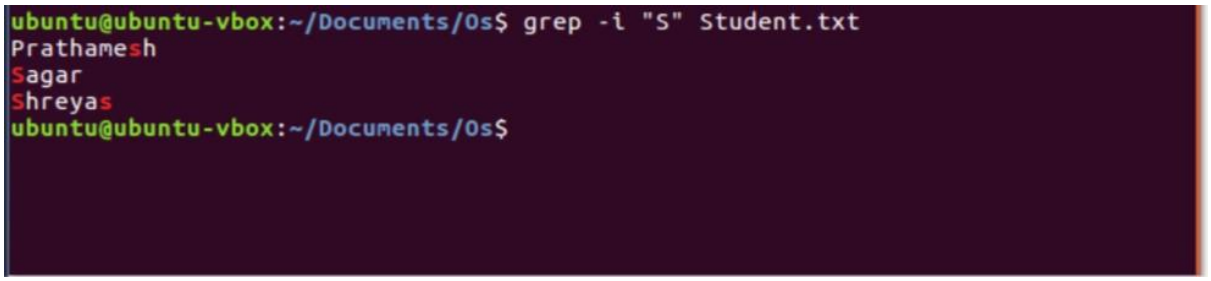
-G, --basic-regexp

Interpret PATTERN as a basic regular expression (BRE, see below). This is the default.

-P, --perl-regexp

Interpret the pattern as a Perl-compatible regular expression (PCRE). This is highly experimental and grep -P may warn of

unimplemented features.



8)sed

NAME

sed - stream editor for filtering and transforming text

SYNOPSIS

sed [OPTION]... {script-only-if-no-other-script} [input-file]...

DESCRIPTION

Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline).

While in some ways similar to an editor which permits scripted edits (such as ed), sed works by making only one pass over the input(s), and

is consequently more efficient. But it is sed's ability to filter text in a pipeline which particularly distinguishes it from other types

of editors.

-n, --quiet, --silent

suppress automatic printing of pattern space

-e script, --expression=script

add the script to the commands to be executed

-f script-file, --file=script-file

add the contents of script-file to the commands to be executed

--follow-symlinks

follow symlinks when processing in place

-i[SUFFIX], --in-place[=SUFFIX]

edit files in place (makes backup if SUFFIX supplied)

-l N, --line-length=N

specify the desired line-wrap length for the `l' command

--posix

disable all GNU extensions.