

LINUX – DAY 1

GNU:

GNU stands for Gnu's Not Unix. GNU is a free and open-source operating system that was started in 1984 by Richard Stallman. GNU is based on the Unix operating system, but it has been greatly modified over the years.

The GNU Project initially created most of the components and services used in GNU/Linux and later added the Linux kernel to create the GNU/Linux OS. The Linux kernel is the core component of GNU/Linux, as it provides basic services and allocates OS resources.

The GNU Core Utilities or coreutils is a package of GNU software containing implementations for many of the basic tools, such as cat, ls, and rm, which are used on Unix-like operating systems.

BASH:

Bash, or Bourne Again Shell, is a free, command-line interface (CLI) shell program that is a default part of most Linux distributions:

What it is

Bash is a shell program that allows users to control a computer's operating system (OS) using a CLI. It's a versatile tool that can be used for a variety of tasks, including system administration, software development, and network engineering.

How it works

Bash operates within a text window, where users input commands to perform tasks. It also supports the execution of commands from files, known as shell scripts, which can automate tasks.

Features

Bash includes features from other shells, such as the C shell and the Korn shell. It also has a powerful scripting language that allows users to automate tasks and perform complex operations.

Availability

Bash is available on nearly all modern operating systems, including Linux and macOS.

SH:

sh command in Linux is a command language interpreter that executes commands read from the terminal standard input, a .sh file in Linux, or some string from the command line. In most modern systems, the sh command is symlinked to /bin/bash.

TCSH:

Tcsh is a command-line interpreter and interactive shell for Linux that provides an interface between the user and the operating system:

Features

Tcsh is an enhanced version of the Berkeley UNIX C shell (csh) that includes additional utilities like command line editing and filename/command completion.

Use cases

Tcsh is a good choice for users who prefer C syntax to bash, or for those who have trouble remembering Unix commands. Some programs also require the C shell to install.

Functions

Tcsh is a high-level programming language and interactive command interpreter. However, unlike other shells, functions cannot be defined in a tcsh script. Instead, users must use aliases.

CAT:

The cat command in Linux is a frequently used command that allows users to read, display, and concatenate text files. It's a simple yet powerful way to manage file content from the command line.

Here are some things you can do with the cat command:

Preview file contents: Quickly preview the contents of a text file without opening it in a large application.

Combine files: Concatenate multiple files together.

Create new files: Create new files.

Investigate files: Use the -A switch to see where lines end with a \$, tab characters with a ^I, and other non-printing characters.

Here are some options you can use with the cat command:

-n: Adds line numbers to the output

-s: Squeezes multiple adjacent blank lines into a single blank line

-e: Displays end-of-line characters as \$

-t: Displays tabs as ^I

-u: Uses unbuffered I/O for stdout

-v: Displays nonprinting characters, except for tabs and the end of line character

WC:

wc stands for word count. As the name implies, it is mainly used for counting purpose.

It is used to find out number of lines, word count, byte and characters count in the files specified in the file arguments.

By default it displays four-columnar output.

First column shows number of lines present in a file specified, second column shows number of words present in the file, third column shows number of characters present in file and fourth column itself is the file name which are given as argument.

Syntax: `wc [OPTION]... [FILE]...`

GCC:

GCC stands for GNU Compiler Collections which is used to compile mainly C and C++ language. It can also be used to compile Objective C and Objective C++. The most important option required while compiling a source code file is the name of the source program, rest every argument is optional like a warning, debugging, linking libraries, object file etc. The different options of gcc command allow the user to stop the compilation process at different stages.

Syntax: `gcc [-c|-S|-E] [-std=standard]`

Example: This will compile the source.c file and give the output file as a.out file which is default name of output file given by gcc compiler, which can be executed using `./a.out`

EMACS:

The Emacs is referred to a family of editors, which means it has many versions or flavors or iterations. The most used version of Emacs editor is GNU Emacs and was created by Richard Stallman.

To use emacs editor, use command – `emacs [-option] [file name]`

GREP:

The grep command in Unix/Linux is a powerful tool used for searching and manipulating text patterns within files. Its name is derived from the ed (editor) command `g/re/p` (globally search for a regular expression and print matching lines), which reflects its core functionality. grep is widely used by programmers, system administrators, and users alike for its efficiency and versatility in handling text data.

The basic syntax of the ``grep`` command is as follows:

`grep [options] pattern [files]`

Here,

[options]: These are command-line flags that modify the behavior of grep.

[pattern]: This is the regular expression you want to search for.

[file]: This is the name of the file(s) you want to search within. You can specify multiple files for simultaneous searching.

AWK:

1. AWK Operations:

- (a) Scans a file line by line
- (b) Splits each input line into fields
- (c) Compares input line/fields to pattern
- (d) Performs action(s) on matched lines

2. Useful For:

- (a) Transform data files

(b) Produce formatted reports

3. Programming Constructs:

(a) Format output lines

(b) Arithmetic and string operations

(c) Conditionals and loops

Syntax: `awk options 'selection _criteria {action }' input-file > output-file`

```
$ whoami  
webmaster
```

```
$ hostname  
export "PS1=$ "  
hostname  
  
cc052c1a545b  
export "PS1=$ "
```

```
$ echo "hello, world"  
export "PS1=$ "  
echo "hello, world"  
  
hello, world  
export "PS1=$ "
```

```
$ echo $HOME  
export "PS1=$ "  
echo $HOME
```

```
/home/webmaster  
export "PS1=$ "
```

```
$ date  
export "PS1=$ "  
date
```

```
Thu Nov 14 21:14:25 IST 2024  
export "PS1=$ "
```

```
$ echo $HOME  
/home/webmaster
```

```
$ echo i am $(whoami)  
export "PS1=$ "  
echo i am $(whoami)
```

```
i am webmaster  
export "PS1=$ "
```

```
terminal@terminal-temple ~ $ cal
```

```
November 2024
```

```
Su Mo Tu We Th Fr Sa
```

```
1 2
```

```
3 4 5 6 7 8 9
```

```
10 11 12 13 14 15 16
```

```
17 18 19 20 21 22 23
```

```
24 25 26 27 28 29 30
```

```
export "PS1=$ "
```

```
$ touch fruits.txt  
touch fruits.txt
```

```
export "PS1=$ "
```

```
$ echo "apple,grapes" > fruits.txt  
echo "apple,grapes" > fruits.txt
```

```
export "PS1=$ "
```

```
$ cat fruits.txt  
apple,grapes  
$ |
```

```
$ cat fruits.txt  
apple,grapes  
$ cut -b 1,2,3 fruits.txt  
cut -b 1,2,3 fruits.txt
```

```
app  
export "PS1=$ "
```

```
$ cut -b 1,2 fruits.txt  
cut -b 1,2 fruits.txt
```

```
ap
```

```
$ tail test.txt
export "PS1=$ "
tail test.txt

Serial No. : 44

Name : dubey

Phone No. : 1234567890

Address : butiburi

export "PS1=$ "
```

```
$ diff friend.txt friend2.txt
export "PS1=$ "
diff friend.txt friend2.txt

1c1
< Serial No. : 01
---
> Serial No. : 11
11c11
< Serial No. : 02
---
> Serial No. : 22
21c21
< Serial No. : 03
---
> Serial No. : 33
31c31
< Serial No. : 01
---
> Serial No. : 44
export "PS1=$ "
```

```
$ less friend.txt
export "PS1=$ "
less friend.txt
```

Serial No. : 01

Name : Aditya

Phone No. : 1234567890

Address : Nagpur

Serial No. : 02

Name : kittu

Phone No. : 1234567890

Address : gondia

Serial No. : 03

Name : jay

Phone No. : 1234567890

Address : amravati

friend.txt

:-

:

Serial No. : 01

:-

:

...skipping...

Serial No. : 01

Name : Aditya


```
:  
No next tag (press RETURN)  
:-  
:  
:-  
:-  
:  
:  
:1.1  
  
friend.txt lines 1-29/39 byte 225/302 75% (press RETURN).  
:  
Serial No. : 01  
  
Name : dubey  
  
Phone No. : 1234567890  
  
Address : butiburi
```

```
$ sed 's/unix/linux/' geekfile.txt  
export "PS1=$ "  
sed 's/unix/linux/' geekfile.txt  
  
linux is great os. unix is opensource. unix is free os.  
  
learn operating system.  
  
linux linux which one you choose.  
  
linux is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.  
export "PS1=$ "
```

```
$ sort test.txt
sort test.txt
Address : Nagpur
Address : amravati
Address : butiburi
Address : gondia
Name : dubey
Name : james
Name : jay
Name : kittu
Phone No. : 1234567890
Phone No. : 1234567890
Phone No. : 1234567890
Phone No. : 1234567890
Serial No. : 11
Serial No. : 22
Serial No. : 33
Serial No. : 44export "PS1=$ "
```

```
$ split friend.txt
export "PS1=$ "
split friend.txt

export "PS1=$ "

$ ls
fontlist-v330.json  friend.txt  friend2.txt  geekfile.txt  xaa
$ wc xaa
39  56 302 xaa
# |
```

```
$ uniq test.txt  
export "PS1=$ "  
uniq test.txt
```

Serial No. : 11

Name : james

Phone No. : 1234567890

Address : Nagpur

Serial No. : 22

Name : kittu

Phone No. : 1234567890

Address : gondia

Serial No. : 33

Name : jay

Phone No. : 1234567890

Address : amravati

Serial No. : 44

Name : dubey

```
$ wc test.txt
```

```
export "PS1=$ "
```

```
wc test.txt
```

```
39  56 301 test.txt
```

```
export "PS1=$ "
```