Software Systems Development Monsoon - 2024

Lab-1 : Bash(Basic commands,grep,awk)

Basic Linux Terminal Commands

ls	Displays information about files in the current directory.(variations)
pwd	Displays the current working directory.
mkdir <directory_name>	Creates a directory.
cd <path>	To navigate between different folders.
touch <file_name>	Create empty files

Basic Linux Terminal Commands

clear	Clear terminal
cat <file_name>	Display file contents on terminal
chmod [options] [mode] [File_name]	chmod +x example.sh (most common)!
echo [option] [string]	Display active processes on the terminal
man [option] [command]	Access manual for all Linux commands man -k permissions **apropos** networking

grep

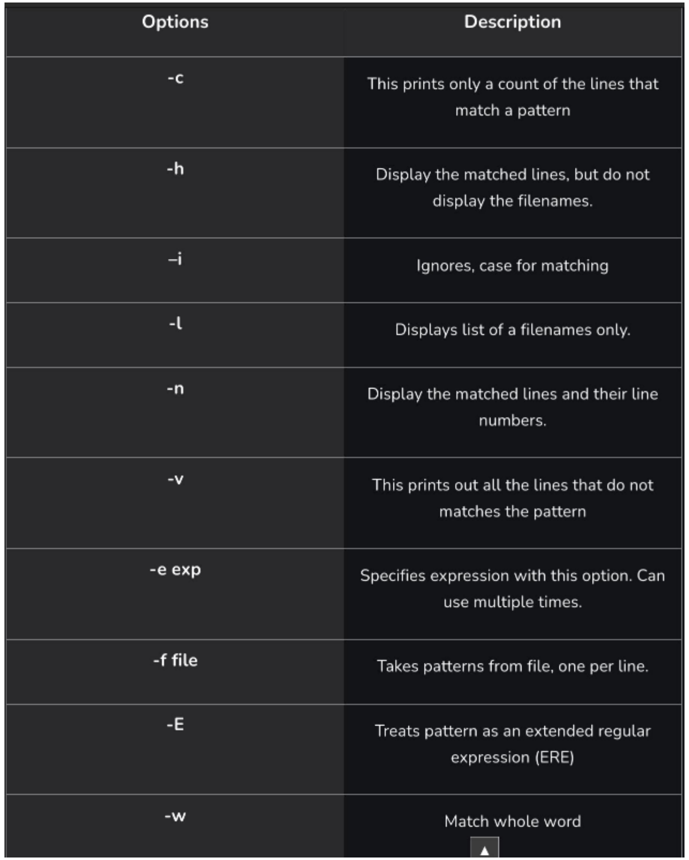
Used for **searching and manipulating text patterns** within files.

Syntax : **grep [options] [pattern] [files]**

[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

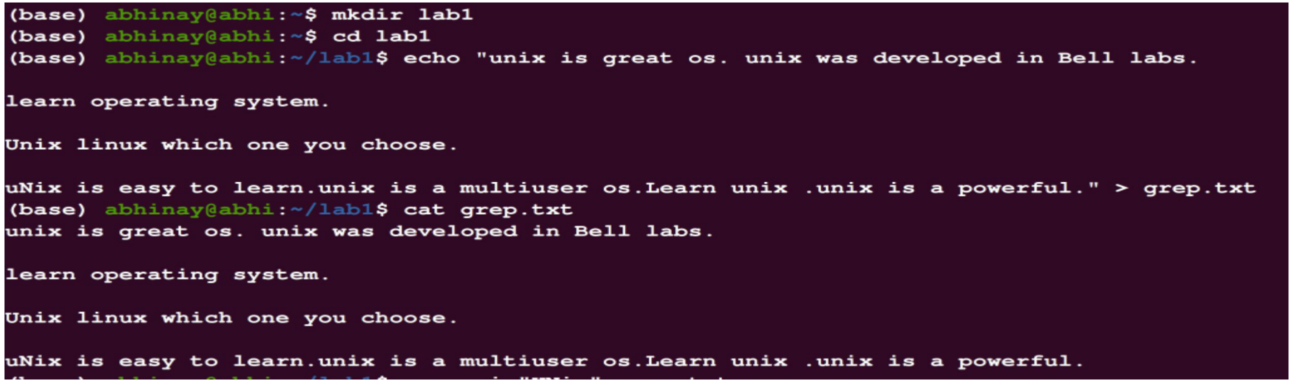


Options	Description
-c	This prints only a count of the lines that match a pattern
-h	Display the matched lines, but do not display the filenames.
–i	Ignores, case for matching
-l	Displays list of a filenames only.
-n	Display the matched lines and their line numbers.
-v	This prints out all the lines that do not matches the pattern
-e exp	Specifies expression with this option. Can use multiple times.
-f file	Takes patterns from file, one per line.
-E	Treats pattern as an extended regular expression (ERE)
-w	Match whole word

grep-Activity

unix is great os. unix was developed in Bell labs.
learn operating system.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.

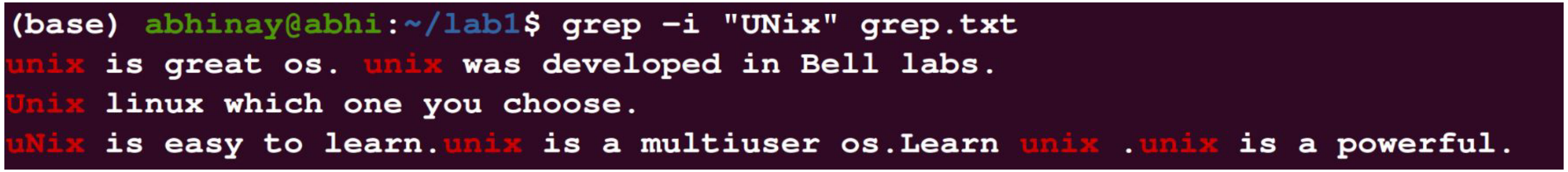
- Open terminal using command [Ctrl+Alt+T]
- Create directory **lab1** [mdir lab1]
- Create file **grep.txt** with text given in top right corner using terminal



```
(base) abhinay@abhi:~$ mkdir lab1
(base) abhinay@abhi:~$ cd lab1
(base) abhinay@abhi:~/lab1$ echo "unix is great os. unix was developed in Bell labs.

learn operating system.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful." > grep.txt
(base) abhinay@abhi:~/lab1$ cat grep.txt
unix is great os. unix was developed in Bell labs.

learn operating system.

Unix linux which one you choose.

uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

grep-Activity

❖ Case insensitive search

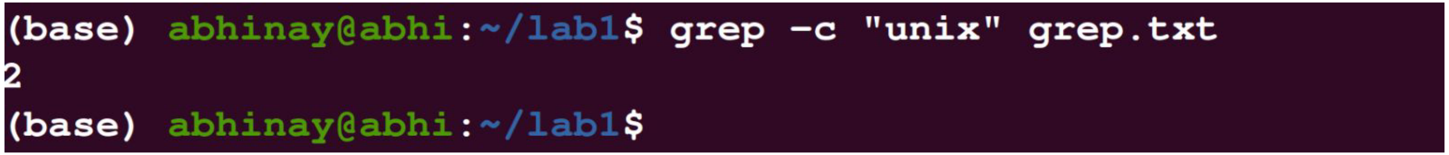


```
(base) abhinay@abhi:~/lab1$ grep -i "UNix" grep.txt
unix is great os. unix was developed in Bell labs.
Unix linux which one you choose.
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a powerful.
```

The -i option enables to search for a string case insensitively in the given file. It matches the words like “UNIX”, “Unix”, “unix”

grep-Activity

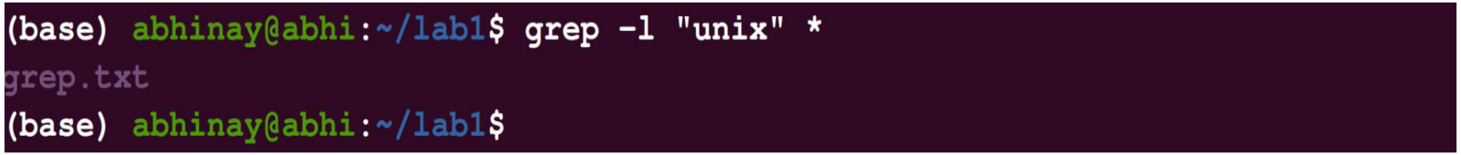
❖ Displaying the Count of Number of Matches Using grep



```
(base) abhinay@abhi:~/lab1$ grep -c "unix" grep.txt
2
(base) abhinay@abhi:~/lab1$
```

This prints only a count of the lines that match a pattern

❖ Display the File Names that Matches the Pattern Using grep



```
(base) abhinay@abhi:~/lab1$ grep -l "unix" *
grep.txt
(base) abhinay@abhi:~/lab1$
```

grep-Activity

- Checking for the Whole Words in a File Using grep
  - grep -w "unix" grep.txt
- Displaying only the matched pattern Using grep
  - grep -o "unix" grep.txt
- Show Line Number While Displaying the Output Using grep -n
  - grep -n "unix" grep.txt
- Inverting the Pattern Match Using grep
  - grep -v "unix" grep.txt

grep-Activity

- Matching the Lines that Start with a String Using grep
  - grep "^unix" grep.txt
- Inverting the Pattern Match Using grep
  - You can display the lines that are not matched with the specified search string pattern using the -v option.
  - grep -v "unix" grep.txt
- Matching the Lines that Start with a String Using grep
  - grep "^unix" grep.txt
- Matching the Lines that End with a String Using grep
  - grep "os$" grep.txt

grep-Activity

- Print n Specific Lines from a File Using grep

  -A prints the searched line and n lines after the result,

  -B prints the searched line and n lines before the result, and

  -C prints the searched line and n lines after and before the result.

  ```
  $ grep -A1 learn grep.txt
  ```

awk

Syntax : **awk [options] 'pattern { action }' input-file(s)**

[options]: Command line options, such as -f to specify a script file, or -F to set a field separator.

[pattern]: A pattern that awk looks for in the input data, which, when matched, triggers the action. If omitted, the action applies to all lines.

[action]: What awk does when it finds a match to the pattern. It’s a block of code enclosed in curly braces {}.

awk

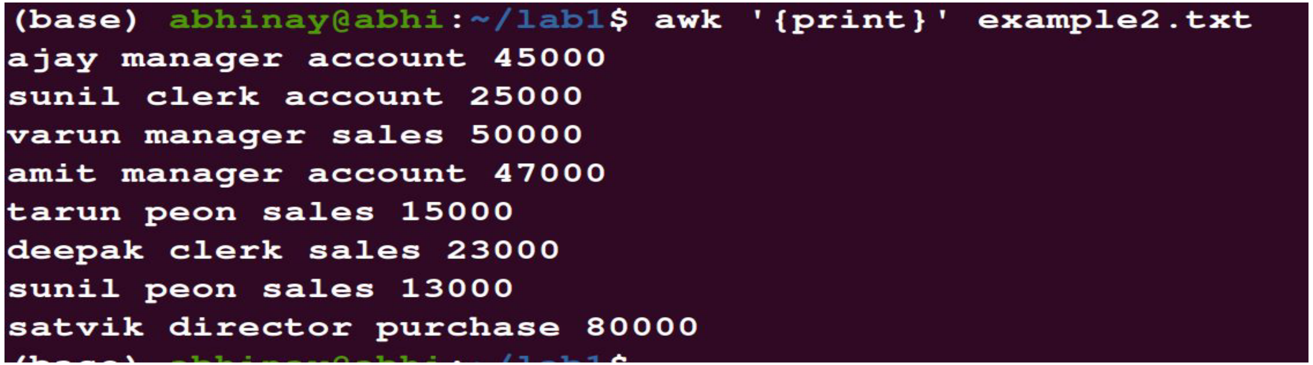
- AWK Operations:
  - Scans a file line by line
  - Splits each input line into fields
  - Compares input line/fields to pattern
  - Performs action(s) on matched lines
- Useful For:
  - Transform data files
  - Produce formatted reports

awk-Activity

- Create new file **example2.txt** with text given below
  -

```
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
```

awk-Activity

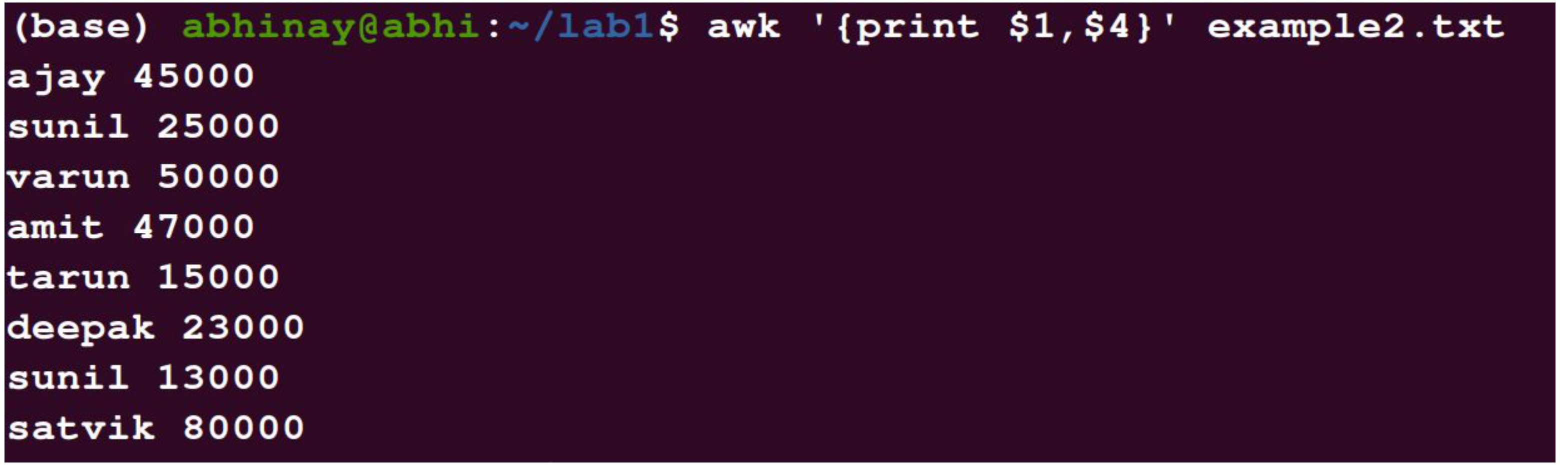
- Default behavior of awk : prints every line of data
- 

```
(base) abhinay@abhi:~/lab1$ awk '{print}' example2.txt
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
(base) abhinay@abhi:~/lab1$
```

awk-Activity

- Print the lines which match the given pattern
- For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables.
- If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively.
- $0 represents the whole line.
- -F option to split the fields using a comma (,) as a delimiter
  - awk -F<deliminator> ‘pattern{action} <file_name>
  - awk -F',' '{print $1}' people_emails.txt

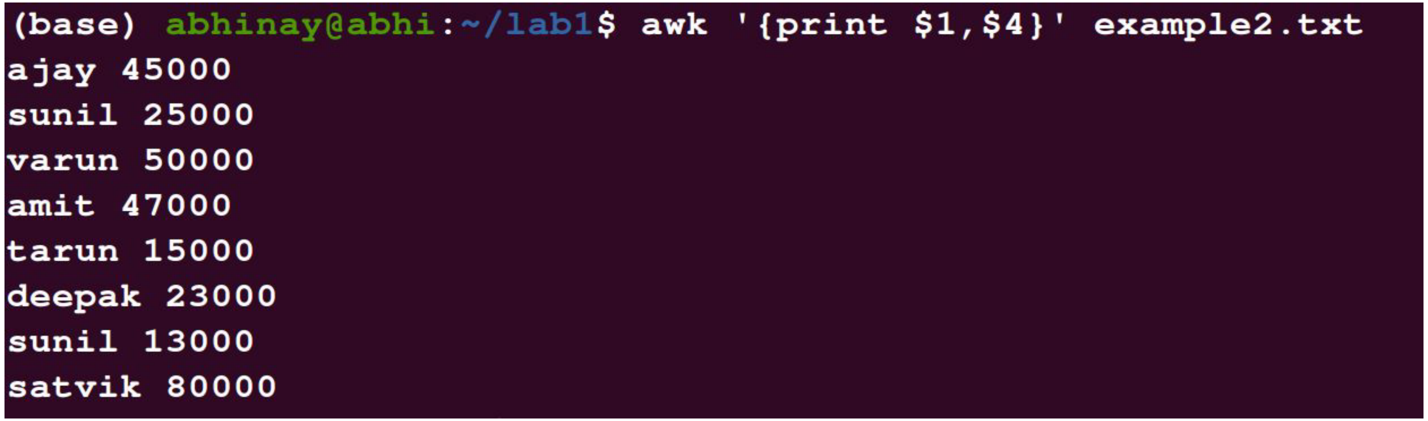
awk-Activity

- Print the lines which match the given pattern
- 

```
(base) abhinay@abhi:~/lab1$ awk '{print $1,$4}' example2.txt
ajay 45000
sunil 25000
varun 50000
amit 47000
tarun 15000
deepak 23000
sunil 13000
satvik 80000
```

awk-Activity

- ❖ Splitting a Line Into Fields
  - ➢ Default delimiter : whitespace character
  - ➢ By default and stores it in the $n variables
- ❖

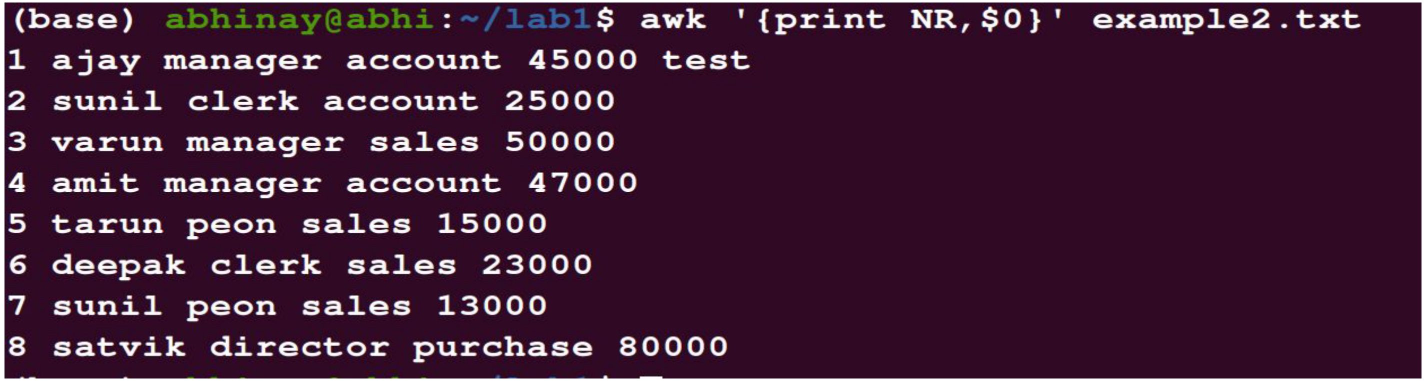


```
(base) abhinay@abhi:~/lab1$ awk '{print $1,$4}' example2.txt
ajay 45000
sunil 25000
varun 50000
amit 47000
tarun 15000
deepak 23000
sunil 13000
satvik 80000
```

awk-Built in variable

NR	Keeps a count of the number of input records processed.
NF	Contains the number of fields in the current input record.
FS	Stores the field separator used to split the input record into fields. The default is whitespace (space or tab).
RS	Stores the record separator character, which defines what constitutes a record. The default is a newline character.
OFS,ORS	output field separator, output record separator

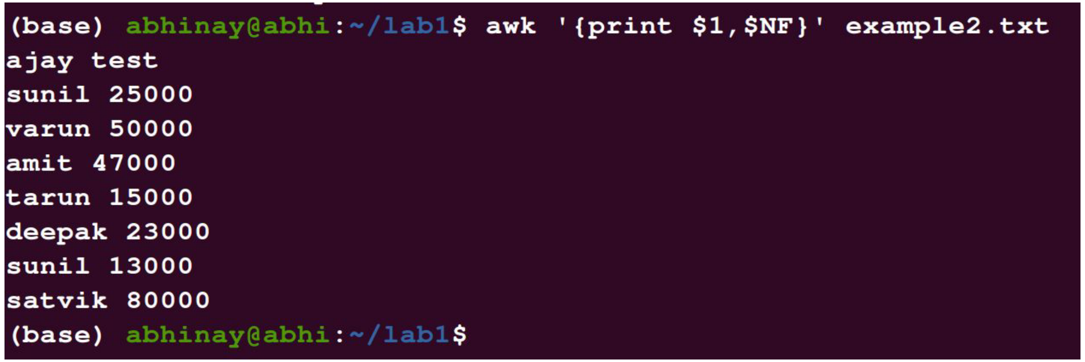
NR - Display Line Number



```
(base) abhinay@abhi:~/lab1$ awk '{print NR,$0}' example2.txt
1 ajay manager account 45000 test
2 sunil clerk account 25000
3 varun manager sales 50000
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
7 sunil peon sales 13000
8 satvik director purchase 80000
```

- the awk command with NR prints all the lines along with the line number
- awk '{print NR "- " $1 }' example2.txt

NF - Display Last Field



```
(base) abhinay@abhi:~/lab1$ awk '{print $1,$NF}' example2.txt
ajay test
sunil 25000
varun 50000
amit 47000
tarun 15000
deepak 23000
sunil 13000
satvik 80000
(base) abhinay@abhi:~/lab1$
```

❖ In the above example $1 represents Name and $NF represents Salary. We can get the Salary using $NF , where $NF represents last field.
❖ How to Print non empty lines? THINK!

References

- https://man7.org/linux/man-pages/man1/grep.1.html
- https://man7.org/linux/man-pages/man1/awk.1p.html
- https://man7.org/linux/man-pages/man1/ls.1.html
- https://man7.org/linux/man-pages/man1/echo.1p.html
- https://man7.org/linux/man-pages/man1/chmod.1p.html

Thankyou!