DAY 1

10 October 2024 14:33

1. Linux history
2. Commands
    1. Cat /etc/os-release
    2. Whoami - to fing user
    3. Clear - to clear the terminal
    4. Useradd username
    5. Cat /etc/passwd
    6. Passwd username
    7. Su username
    8. Time,date,timedatectl,cal
    9. Exit,sudo su(to go to root)
    10. Su -(used when we know root password /during installation of package)
    11. Sudo id username
    12. sudo adduser username
    13. Sudo passwd username
    14. Userdel username
    15. Create files
        a. Echo "text to write in file">filename
        b. Touch
        c. cat>filename
        d. Vi/nano filename
        e. -x create encrypted file
    16. Ls command
        a. Ls -l(list all with permissions)
        b. Ls -a(hidden files)
        c. Ls -il (inode number) filename
        d. Ls -R(gives all paths and files in subdirectories)
        e. Ls *a*/*a/a*/????/?????/[ac]*
    17. Pattern
        a. [ac]*
        b. {s,m,t,w}.txt
        c. File{1..3}.txt
        d. File{a,b}{1,2].txt
        e. File{{a,b}b,c}.txt
        f. Varibale=value --> echo $Variable
        g. Ls -l|wc -l
        h. Ls -1|tee a.txt|less (gives output on terminal and saves the same in a.txt)
    18. COMMANDS
        a. Cp file1 file2 (overwrirts)
        b. Cp -r/-i/-v/-a/-n/-f/-u/--backup
        c. Cat file1>>destfile
        d. Mv file1 file2 (rename,inode number will be same)
        e. Rm -ir(removes all files and subdir files)
        f. Rm/rmdir -r/-i/--backup(make backup file before deleting)
        g. Whatis command_name(gives use of command)
        h. Yum update(updates the package)
    19. Helping commands
        a. Command --help
        b. Man command_name
        c. Command_name -h
        d. Info command_name
        e. Help help /command_name
    20. Edittors

a. Vim/vi editor(gg,dd,yy,p, -x(to encrypt))
b. Gedit filename
c. Nano filename(crl+o--->enter-->ctrl+x)

DAY 2

11 October 2024 10:00

1. Sed (substitute,add,delete,print) --- sed 's/old/new/g' filenname
    a. -I (chnages the original file)
    b. Sed '/pattern(word)/d' filename (deletes the line which has that word)
    c. sed '2d/3d/' filename deletes that line number
    d. Sed -n '1,3p' kavya.txt (to print specific lines)
    e. Sed '2i/word' kavya.txt (inserts word in 2nd line starting)
    f. Sed '/pattern/or2a/word' kavya.txt (appends word to line edn)
    g. Sed 's/a.b/c/g' filename (replaces with c where a_b pattern macthes)
    h. Sed '/^$/d' filename (deletes all spaces)
    i. -e(when we want to execute more than one command)
    j. -s (chnages in multiple files)

2. Awk
    a. Awk {' print $1'} filename (gives first words of every line)
    b. awk 'pattern { action }' file
    c. awk -F, '{ print $1 }' file.csv # Uses comma as the field separator
    d. awk '{ sum += $1 } END { print sum }' file # Sums the first column
    e. awk '{ if ($1 > 10) print $1 }' file # Prints values greater than 10 in the first column
    f. awk '/pattern/ { print }' file # Prints lines matching 'pattern'
    g. awk '{ printf "Name: %s, Age: %d\n", $1, $2 }' file # Custom formatted output
    h. NR- number of records
    i. NFv-number of fields
3. Tar (compress the file) - gzip,unzip and xz (tar -cf archive.tar file1 file2 file3)
    a. -c
    b. -v
    c. -f
    d. -x
    e. -Z
    f. -t
    g. -j
    h. -J
    i. /var/log/msg (stores log msgs)
    j. -czf, -cjf, cJf
    k. -xvf (to extract)
    l. -rf (add files or directories to already existing)
    m. -cvf (back up file using tar)

- Process management & performance monitoring
    1. s,d,k,l,r,t(process state flags)
    2. Zombie(z,x)
    3. Ctrl+x to kill
    4. Ctrl+z background
    5. Pgrep -l -u
    6. Pstree -p
    7. Uptime
- Monitoring Process Activity
    - Top
- COMMANDS
    - Wc -l,-c,-w(gives no.of lines,characters,words in file)
    - Head -n filename(gives top n lines,by default 10)
    - Tail -f -n filename(gives last n lines ,by default 10)

- Ln flename hardlink(backup file,on regular files)
- Ln -l filename soflink
- whereis (man files also)
- Which(executable files in path)

learnings

15 October 2024 23:09

1. Directories
    a. /bin - binary files
    b. /boot - boot loader files
    c. /dev - device files
    d. /etc -system configuration files
    e. /home - home directories
    f. /lib - system libraries
    g. /media - mount point for removable media
    h. /mnt - mount point for temporary files
    i. /opt - optional add-on applications
    j. /sbin - system binaries
    k. /srv - service data
    l. /tmp - temporary files
    m. /usr - user programs
    n. /var - variable files
    o. /root- root user directory
    p. /proc - process info
    q. /lost_found - misplaced data
2. Ls
    a. -i inode number
    b. -l long format
    c. -R all files with sub directories
    d. -r reverse order
    e. -t time (newest first)
    f. -a hidden files
    g. -h(human readable format)
    h. -p appending slash to directory names
    i. -S sort by size
    j. -d list directories without contents
    k. --color
    l. -1
3. Creating files
    a. Cat>file1
        i. Cat file1
        ii. Cat file1 file2
        iii. Cat "djhfeirufr">> file1
        iv. Cat -n,-s(supress blanks),-v(non-printing characters),-T(tab-^I),-E(end of line -$)
    b. Touch file2
4. Copy/move
    a. Mv - rename/move
        i. --backup - makes a backup file for destination file
    b. Cp same file copying
        i. -r -recursive
        ii. -I - interactive
        iii. -u - update(copies when source file is new or destination is not found)
        iv. -a - archive (preserves file attributes)
        v. -v verbose
        vi. -f - forces the copy
5. Wc - word count
    a. -l
    b. -c(bytes)
    c. -w
    d. -m(chars)
    e. -L (length of longest line)
6. Remove -rm
    a. -f force
    b. -I interactive
    c. -r recursive
    d. -d empty_directories
7. Standard linux streams
    a. 0-stdin
    b. 1-stdout
    c. Stderr-2
8. File permissions
    a. Owner/grp/other
    b. Read/write/execute
    c. Symbolic permission
        i. Chmod u/g/o(+/-/=)r/w/x file/directory name(we can also use combinations like rw,wx,rwx)
        ii. For all at once a=rwx(ameans all)

    d. Using octal
        i. Read -4
        ii. Write- 2
        iii. Execute -1
        iv. 0 - no permission
        v. Umask is used for default permissions (0002), we can change the umask value to change default permissions for files/directories
    e. SUID AND GUID
        i. Chmod u/g+s file/directory(in ls result if 's' then there is execute permission previously , no execute permission results 'S')
9. Variables
    a. Etc/profile -contains mandatory thigs to be done during start of system , if not found any then it'll move to home/profile and does the same
    b. Environt ment variables (uses $ symbol and all in caps)
    c. .bash_profile/history all files in root directory
10. Basic utilities
    a. Pr option filesname - to print (pr -2 filename ,prints in two columns)
    b. $mail [-s subject] [-c cc-addr] [-b bcc-addr] to-addr]
    c. Lpr filename - it will send file to printer
    d. Export SEPARMDB ="mydb"(setting/exporting environment varible)
    e. Unset env_name - to unset environmental variable
11. Pies and filters
    a. |
    b. Sort filename - sorts the content of the file
    c. Sort -f filename - capitals are given priority
    d. Sort numbers - takes ascii values , use -n to sort accroding numbers(like 1,2,3,4..), -r for reverse
    e. Pg
12. File editors
    a. Gedit -save n exit
    b. Nano editors -ctrl+o (then name then enter) to save
    c. Vim editors -
        i. Vi -R (read only mode)
        ii. View filename (command mode)
        iii. Yy - copy a line
        iv. P - paste the line
        v. Nyy - n number of lines copied
        vi. Dd -delete single line
        vii. Ndd - n number of lines deleted
        viii. Cc/ncc - to cut the lines
13. Shell scripting
    a. Types of shells
        i. Sh -secured shell
        ii. Ksh - korn shell
        iii. Csh - supports c program
        iv. Restricted shell - limits several functonalities to users
        v. Bash - bourne again shell
    b. Basic shell scripting
        i. Commands should be in order of execution
        ii. #deyfiuhf - to comment the line
        iii. Echo to print lines on console
        iv. Read var_name --> reads from user
        v. To display the variable that has read from the user - $a
    c. Set variables
        i. Name="Kavya" ($Name ->in nextline **readonly** Name , we will be unable to modify the Name)
        ii. Unset Name - will unset the varibale
    d. Special variables
        i. $0 - holds the script/command name
        ii. $1,$2 - holds the arguments passed in command line arguments or if we wnt to run only particular command then $n
        iii. $# - holds the number of arguments passed in command line/it'll check for the all the instructions supplied to script file
        iv. $* - stores the passed varibales to script as strings ,with quites also saves as single string
        v. $@ - without quotes behvaes similar to $* ,with quotes treats as individual (\"$@\")
        vi. $?- exit status of last command executed
        vii. $$ - process id of current shell
        viii. $!- process number of last background command
    e. Shell arrays
        i. Scalar variable - each varibale contains single value
        ii. Array - single varibale contains multiple values
        iii. Name=kavya,Name=divya - it'll store only divya .in order to save both in single varibale array is used
        iv. name[0]=kavya ,name[1]= divya , name[2]=arush - echo "hello,${name[0]/[1]/[2]}"/$name[*/@](to print all)
        v. Array=(1,2,3,4)/{"kavya","divya"..} - echo"total elements :${#array[@]}" -gives total elemnts in an array
    f. Shell operators
        i. Arithmetic operators
            1) '+' - 'exp $a +$b' (-, *, /, %,, =, ==, !=)
        ii. Relational operators
            1) -eq : checks if vaalue of two operands is equal or not. If yes, then condition becomes truw
            2) -ne : not equal then condition true
            3) -gt : greaterthan

4) -lt : lesser than
5) -ge : greater than or equl
6) -le : lesser than or equal

iii. Boolean operators
1) ! : inverts the value
2) -o : logical or
3) -a : logical and

iv. String operators
1) =,!=,
2) -z - checks if given string length is zero -if yes true
3) -n - true if non-zero length
4) Str - returns false if empty string

v. Files test operators
1) -f : tests whether given string is file or not
2) -d : for directory
3) -g filename : checks if file has set groupID bit ,if yes then true
4) -e : file exists or not
5) -r,-w,-x : checks does file has read ,write , execution permission

g. Shell decision making
i. If-else
1) If..then…….fi
2) If…..then….else…fi
3) If…then…elif…..then….else….fi

ii. Case_easc
1) Case varibale in list …..block …another case ….block ……esac
2) ? - checks for special character
3) * - checks if user have given more than required input

iii. Control shell loop
1) Break
2) Continue

h. Shell substitution
i. Commnad-name or $(command-name)
ii. Running commands in script using with (`ls -l`) or directly

i. Shell quouting mechanism - to avoid special meaning of characters(exception : `,$,',",\\)
i. Single quote
ii. Double quote ((exception : `,$,',",\\)
iii. Backslash
iv. Back quote

j. Shell function

```
hello()
{
    echo "hello,welocme $1 $2"
    return 10
}

hello technical trainer

ret=$?

echo "value returned by function is $ret"


func_1()
{
    echo "this is the first function"
    func_2
}

func_2()
{
    echo "this is function 2"
}

func_1
```

14. Manual pages
a. Man commandname
b. Info commanname
c. Help commandname