UNIT - II
Secondary Storage Management

Dr.T.Logeswari

Disk Space Management

- User keep unwanted files on the disk due to which files tend to accumulate therefore making the system slow
- The allotted space for a file system may not be utilized fully or someone else may be in need of more space
- Whenever a new user is created, disk space has to be allocated to him
- Whenever new applications are created they need more space

- There are various issues associated with managing disk space such as
    - Formatting a disk
    - Making file system
    - Checking disk space
    - Creation of partition on a disk
    - Mounting a file system

Formatting a disk

- Before you use a disk for backup purposes first you format it first
- A disk can be formatted in two ways
  - Low level format
  - High level format
- Low level format (or) Physical format
  - It is the first step in formatting
  - It is dividing the disk track into specified number of sector
  - Filling the data area of each sector with dummy bytes value
  - This result in destruction of any existing data on the disk

- High level format (or) Logical format
  - It is the creation of a file system including a table of content for the disk
  - It does not destroy data already on the disk
- A disk can be formatted in Unix using format command or fd format command
- The fdformat command is used to perform the lower level formatting of floppy disk

- Syntax:

fdformat[-n]device

Device – it is the location of the floppy disk

$fdformat/dev/fd0

o/p: it will perform low level format on floppy disk press enter key it will start formatting

Making file system

- Once partition on the hard disk is created by system administrator, he has to create a file system on this partition to make it usable
- Mkfs command is the universal file system creation tool
- Mkfs command is mostly used to create a new file system
- newfs command provides friendlier interface and call mkfs internally
- To create a file system in Linux with their own tree structure and root directory

- Syntax

  mkfs[-t type][fs –option]filesys[blocks]

- The item in square bracket is optional and filesys is mandatory
- Filesys is the name of the device
- It is mount point for new file system such as root directory, /usr or /home
- Option
  - -t fstype(specifies the type of the file system to built)
  - fs –option(file system specific option to be passed)
  - Squrt(x) – square root

    $ mkfs /dev /fd0

Checking disk Space

- In unix number of command that can aid you in this task of checking free space
- df and du command can be used by any user to report disk usage or free space in term of blocks

dfcommand – Display free space

$df[-option(s)] [device(s)]

- It is used to find the amount of space used by mounted file system
- It can report the amount of free space available for each file system
- It is commonly used to show not only space available and used but also what file system are mounted, the number of inodes still available or to install a new program

- When used with no option and no argument, df generate the following information
  - First column display device name
  - Second column shows the number of block available
  - Third column display the number of used blocks
  - Fourth column shows the number of inodes available
  - Fifth column shows the percentage of blocks used
  - Last column represent the directory on the file system mounted

- Option

  -h (it provides easy to read output by reporting in large unit like GB,MB etc

  -k (display block available in 1k block)

  -m(display block available in 1m block)

  -i(display inode usage)

  $df/dev/hda3

ducommand – disk usage

- Syntax

  ducommand[-option]directories

  - It report the size of directory tree inclusive of all their content and size of the individual files
  - It report the amount of space taken by each sub directory as well as current directory
  - This command generate report in terms of block used

- Option

  -s (report on each user home directory)

  -a(display the space that each file taking up)

  -k(report the file size in units)

  -ch(display file size as well as total capacity of file combined

  $du-s*.txt

  $du-ch*.txt

Unlimit command – user limit

- faulty program or corrupted file may occupy huge amount of space
- It may run into several mega bytes of disk space, ultimately harming the file system
- Creation of such file can be avoided using ulimit command. This command is built in all shell
- This command imposes a restriction on the maximum size of the file that a user is permitted to create
- When used by itself, display the current setting

Mounting a file system

- Data store in device like
  - Floppy disk
  - CD ROMS
  - Hard disk
  - Migrate to unix from window
    - You can access all file easily
- However is not in case of linux
  - You can attach to some existing directory on your system before they can accessed

- The process of attaching a device to a directory is called mounting
- The device where the directory is attached is called mount point
- A device is mounted to a directory using the mount command
- After the device is mounted you can access the file on that device
- By accessing the directory where the device is attached
- Mounting a file system means that you are presenting the file system to the system to the end user

Mount Command

- Synatx

  $mount[-option] device_name mount_point

- It takes two argument
  - The name of the file system
  - The directory under which it is mounted
  - Before mounting an empty directory must be created in the file system
  - The root directory of new system need to be mounted on this directory
  - Two directories are created as default mount point

- /mnt - it is directory that use to mount occasionally it is going to test whether some device is really mountable
- /media – mount devices that are connected on a regular basics

    $mount/dev/cdrom/media

* If you are using mount command without argument it will display a list of all mounted devices

umount command

- It is unmounting file system
- All mounted file system are unmounted automatically when a computer is shut down(USB)
- The reverse process of detaching a file system from another file system is known as unmounting
- Syntax

$unmount[option]file system

Disk Partitioning – fdisk command

- It is the act of dividing a hard disk drive into multiple logical storage unit referred to as partition (Slices)
- It is created after the disk is formatted
- Linux uses fdisk command to create partition
- It is easy to name the partition device
- The first is represented by /dev/hda with partition hda1,hda2 etc
- The second hard disk will have the name /dev/hdb with simillar extension

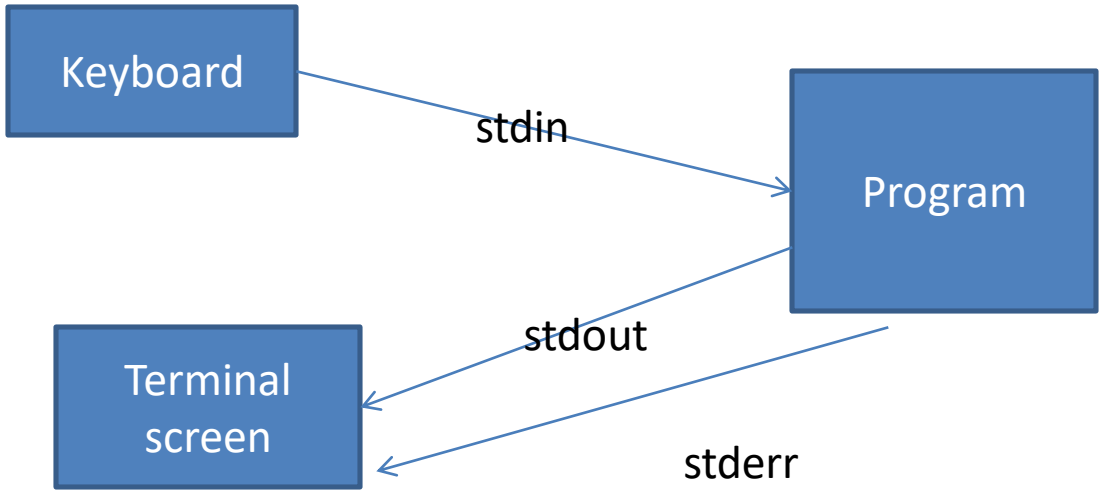
Special Tools and Utilities

Dr.T.Logeswari

Standard Streams in Unix

- Most of the commands in unix take input from keyboard and send the output to terminal screen
- The command have been programmed to accept input from standard input file(keyboard) and produce output on a standard output file(terminal screen)
- The input and output files are stream of character which are given as input or sent as output

- In unix OS the standard streams are pre connected input and output channel between a computer program and its I/O devices
- The three I/O streams files are called standard input(stdin), standard output(stdout) and standard error(stderr)



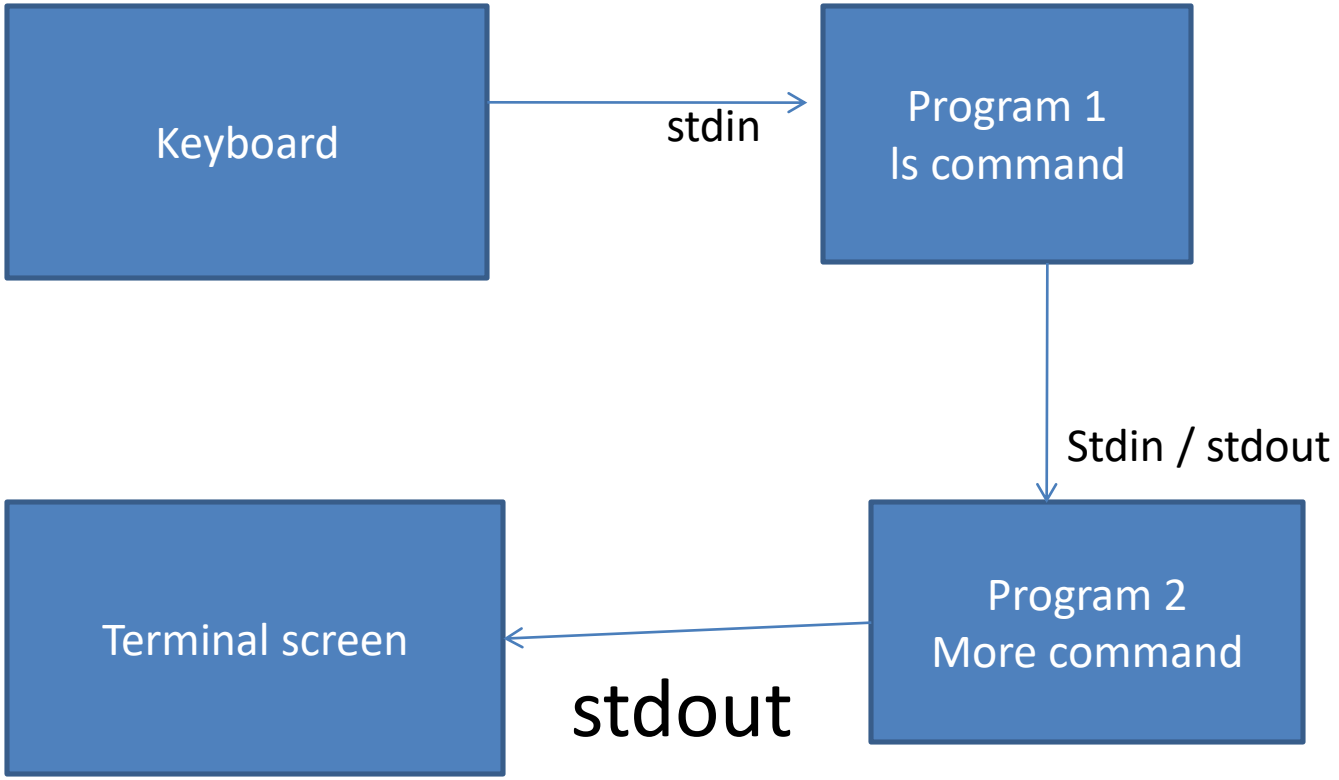
- Standard input is textual data going into a program
  - The redirection operator < ,input is expected from keyboard which started the program
  - The input also be given from output of another command using pipe (|) symbol
- Standard output is the stream where a program writes its output data
  - The redirection operator > , output is the text terminal which initiated the program
  - The output also be sent as input to another command using pipe (|) symbol

- Standard error is output stream used by program to output error message or diagnostics

Pipes in Unix

- You can connect two program together so that the output from one program become the input of the next program (pipe)
- The pipe is represented by vertical line character (|)
- The mechanism of redirecting the output of one command as input to another command directly without using intermediate files is called piping
- The sequence of commands using one or more pipes is called a pipeline

Pipeline of two program sent to standard output



Filter

- A filter is a computer program to process a data stream
- Unix is rich with filter programs
- Filter commands accept some data as input perform some manipulation on it and produce output
- They perform action on the data they are appropriately called filters
- A common use of filter is to modify output

Filter command – head command

- To display the beginning of a file
- Syntax

  $head[-option][-number]file
- It display the contents at the top of the file
- It display first 10 lines of the file
- **HEAD**: This command is used to display the lines from top of the file.
- Options:-
- a) Head-n filename - To display n lines from top of the file.
- b) Head-nc filename – To display the number of character.

If you want first 50 lines you can use head -50 filename or for 37 lines head -37 filename and so forth.

Tail command

- This command is used to display the lines from bottom of the file.
- Options:-

a) Tail-n filename - To display n lines from bottom of the file.

b) Tail+n filename – To display from the nth line to end of the file.

- tail filename by default will display the last 10 lines of a file.
  If you want last 50 lines then you can use tail -50 filename.

- It does not perform any filtering action on its input. It gives out exactly what it takes
- It can be used with any command that sends its output to standard output
- When it is used with –a option, it append the redirected output to the specified file rather than overwriting

Cut command

- cut command selects a list of columns or fields from one or more files.
- Option -c is for columns and -f for fields.
- It is entered as cut *options* [files]
  for example if a file named testfile contains
- this is firstline
- this is secondline
- this is thirdline
- Examples:
  cut -c1,4 testfile will print this to standard output (screen)

  ts

  ts

  ts
- It is printing columns 1 and 4 of this file which contains t and s (part of *this*).

- **Options:**

-c *list* cut the column positions identified in list.

-f *list* will cut the fields identified in list.

-s could be used with -f to suppress lines without delimiters.

a) cut-c 1, 4, 7 filename – To cut and display fields 1, 4, and 7 only if fields are separated by space or tab.

b) cut-c 1-20 filename - To cut and display 1 to 20 characters in a file.

Paste Command

- paste command merge the lines of one or more files into vertical columns separated by a tab.
  for example if a file named testfile contains
- this is firstline
- and a file named testfile2 contains
- this is testfile2
- then running this command
  paste testfile testfile2 > outputfile
  will put this into outputfile
- this is firstline this is testfile2
- it contains contents of both files in columns.

- Options: -
- a) paste file1 file2 – To paste and display from file1 to file2.
- -d'char' separate columns with char instead of a tab.
- -s merge subsequent lines from one file.

Sort command

- sort command sort the lines of a file or files, in alphabetical order. for example if you have a file named testfile with these contents
- zzz
- aaa
- 1234
- yuer
- wer
- qww
- wwe
- Then running
  sort testfile
  will give us output of
- 1234
- aaa
- qww
- wer
- wwe
- yuer
- zzz

- **Options:**

-b ignores leading spaces and tabs.

-c checks whether files are already sorted.

-d ignores punctuation.

-i ignores non-printing characters.

-n sorts in arithmetic order.

-o *file* put output in a file.

+m[-m] skips n fields before sorting, and sort up to field position m.

-r reverse the order of sort.

-u identical lines in input file appear only one time in output.

Uniq command

- uniq command –when duplicate entries are found in file they can removed from file by using this command.
  - It takes only one sorted files as its argument
  - When the command is used without any option, the output is displayed without the duplicate entries

Options:

- -c print each line once and counting number of times
- -d print duplicate lines once.
- -u print only unique lines.

tr command – translating character

- Syntax

  tr[-option]expression1 expression2 standard_input

  - It copies the standard input to the standard output with substitution or deletion of selected character
  - It can also squeeze repeating character into a single character using certain option
  - Option

    -d delete the specified character in expression1

    -s replace the repeated character in the expression

    -

Tee command

- Display output and redirect output
- Syntax

  tee[-option][file]

  - It is used when you want to redirect the output to another file and also see the output on the screen
  - It uses standard output and standard output, which means that it can be placed anywhere in a pipeline
  - It break the input into two component
    - One component is saved in a file
    - The other is connected to the standard output

Grep Command

- grep command is the most useful search command.
- You can use it to find processes running on system, to find a pattern in a file, etc.
- It can be used to search one or more files to match an expression.
- It can also be used in conjunction with other commands as in this following example, output of ps command is passed to grep command, here it means search all processes in system and find the pattern **sleep**.

- **Options:**

-b option will precede each line with its block number.

-c option will only print the count of matched lines.

-i ignores uppercase and lowercase distinctions.

-l lists filenames but not matched lines.

- Options:-

a) grep “billu” filename - To search a word “billu” in the file and display that line.

b) grep-v “rock” | filename - To search except the word rock in the file.

c) grep “e$” | filename - To search for a line ending with e.

d) grep”lf” | filename - To search for a line beginning with f.

SED – Stream Editor

- Sed stand for stream Editor
- It allows editing of files non interactively
- It can do search-replace operation and insert & delete lines into text files
- Synatx

  $sed<option><address –action list><file list>

  Option – option such as –e, -f and –n

  Address - single line or multi line perform action to take place

  Actionlist – specifies the action or list of action

  Filelist – it hold zero or more filename from where the lines are searched

- Option
  - e this option give multiple instruction in a single command
  - f this option is for script file which contain the sed instruction to be executed
  - n by default sed display or print all lines selected

Working mechanism of sed Command

- Each line or record is read from the input files
- Which is held in the memory area called pattern space
- All the command are applied on this space
- Since sed work on one line at a time, large files can be altered without invoking an editor, or worry about disk space
- The processing does not affect the original content of the input file

Addressing

- There are three methods of selecting the required line for processing
  - Line addressing
  - Context addressing
  - Range addressing
- Line addressing
  - By using the line number or group of line number
- Context addressing
  - By specifying a search pattern which occur in line
- Range addressing
  - Range of line number. comma between starting and ending range is compulsory

Sed commands

- Sed consist of different types of command, different types of action on a single line or all selected line
- **q command** – it accept all the line from line addressing and then quit
- **d command** – it is used to delete a line or range of line
- **p command** – the selected line can be printed

Using search pattern

- To print all the lines containing a given pattern, the search pattern must be enclosed within two slashes
- **s command** it stand for substitution command the old pattern followed by the replacement pattern is given within slashes
- It substitute the old pattern with the new pattern
- **a command** is used to append one or more record / line to an existing file. The command must be given with $sign. The last line must be terminated with backslash
- **i command** – text can be inserted into a file at a specific line number. 1i insert the text at the beginning of the input file

- **c command** – one or more files/records can be changed or replaced in the input file
- **w command** – the output of the sed command can be written into separate file using write command w
- **r command** – the content of an input file can be read into another specified file by using the read command r
- **The = command** – the line number can be printed using the = command

awk

- It is a general purpose programming language that is designed for processing data, either in files or data streams
- The awk is derived from family names of its author Alfred Aho, Peter Weinberger, and Brain Kernighan
- It is a language for processing files of text
- A file is treated as a sequence of record
- A awk program is a sequence of pattern action statement.
- It read the input one line at a time

- A line is scanned for each pattern in the program and for each pattern that matches, the association action is executed
- Structure of awk program

Format

$awk option program filemanes

Where

Option – f and F

-f it inform the program is on a separate file

-F specifies thee input field separator

filename - contain the list of zero or more input filename

program - general format

pattren{action}

- The structure of an awk program consist of 3 section

Begin

•Statement starting with BEGIN are executed before reading the input line

•It is used for variable initialization

Fs is the input field separator

- END statement are executed once after all the input are processed
- These instruction are generally used to generate summary report

  $awk END

  NR stand for number of records
- Body section contain one or more program statement

variables

- Awk uses two types of variable
  - User desined variable
  - Built-in variable
- Built in variable
  - It is predefined by awk
  - It is written in uppercase letter
  - It has some default value

Variables	Description
FILENAME	Name of the current input file
FS	Input field separator
NF	Number of field in input record
NR	Number of current Record
OFS	Output field separator
ORS	Output record separator
RS	Input Record separator

Built-in Variable

User defined Variables

- Variable names can use any of the character[A-Z, a-z,0-9]
- You cant place two variable adjacent to each other without having space in between
- String constant are delimited by double quotes
- Statement need not end with semicolon
- Comment can added to program by using #
- When variable are defined in awk they get initialized to zero or null string automatically

Pattern

- Awk support different types of patterns.
- When the given search pattern matches a record or line in the awk statement, the action part of the statement is executed.
- A statement may not have a pattern at all
- If present a pattern is made up of expression
- The expression may be arithmetic, relational, logical or regular

operator

- Awk expression include the tilde operator~ called the match and no match !~

awk command

- awk command are the statement, which is substituted for the action part
- It include function call, variable assignment, calculation or any combination or these
- Print command
  - It is used for output text.
  - It is terminated by a predefined string called the output record separator(ORS)
  - The default value is newline

    {print}

  This display the content of the current line

    {print $1} – display first field of current line

awk control structure

- Similar to any programming language, awk contain decision structure and looping structure

If else

```
If(expression)
{
    Statement 1
}
Else
{
    Statement 2
}
```

Conditional operator

Expression ? Action1 : action 2

While statement

```
While (expression)
{
        statement
}
```

do statement

do

Statement

While(expression)

for

```
{
for (expr1; condition; expr2)
Statement
}
```

Unix System call

- System call is the interface between the process and the operating system
- It is the direct entries to the kernel
- It can only interact with commands, like shell , text editor, and other application program
- It is the only way to access kernel such as the file system, the multitasking mechanism and inter process communication

System call for low level I/O

- The lowest level of the I/O is actually a direct entry into the operating system
- The system call are the programs that make a request to the 0's for the service (look function call in c)
- The system call used for low level file I/O are

  creat(name, permission)

  Open(name, mode)

  close(fd)

  unlink(fd)

  read(fd, buffer, n)

  write(fd, buffer, n)

  Iseek(fd, offset, whence)

creat

- No e to create a file in system call
- If the file does not exist, the creat call create the specified filename(name) with the permission specified by second argument permission
- The permission are specified by an octal number with leading 0(zero)

fd= creat (name, permission)

open

- It is a system call to open a file
- Open return a file descriptor(fd)
- It is an integer specifying the position of this open file in the table of open files for current process

fd = open(name, mode)

- name argument contain the filename
- Mode specifies access mode

0 – read

1-write

2 – both

-1 return error

close

- It is a system call to close the file
- It break the file name and descriptor(fd)
- Termination of the program with an exit statement or return statement in the main program, closes all open file

Close(fd)

unlink

- Remove the file from the file system itself

  Unlink(fd)

read

- Read file from the file opened for reading

  read = read(fd, buffer,n)

- Where fd is the file descriptor
- Buffer is an array(acting as data source or destination
- N is the number of bytes to be transformed

write

- Write data from the file opened for writing

  nwritten = write(fd, buffer, n)

- Where fd is the file descriptor
- Buffer is an array
- N is the number of bytes to be transformed

lseek

- Seek a specified position in a file
- It is used for random access while reading or writing a file

pos=seek(fd,offset, whence)

- Offset – is the position to move
- Whence is the position where the offset position is measured 0 – beginning 1- current 2 – end of file
- Pos – value return

System call for process control

- The unix system provide several system call to create and send program, to send and receive software interrupt, to allocate memory and do other job for a process
- Four system call for creating, ending, waiting for a process to complete.
- They are

  fork()

  Wait()

  execl(), exev(), execlp(), execvp()

  Exit()

fork()

- It is a system call that create a new process under UNIX OS
- Eg if a program contain fork() the execution of the program result in the execution of two processes (the parent process another child process).
- It is a time shared operating system the two process run concurrently
- Syntax

proc_id=fork()

The value return by fork() is stored in proc_id (integer)

Each process will return their pid

To find out id of a process getid() system call is used

Wait()

- The parent wait for the child to terminate before continuing itself
- Syntax

  wait(&status)
- Status is the pointer to an integer where unix stores the value returned by the child
- The process to wait for a signal
- It is zero for normal termination and nonzero to indicate different kind of error

execl & execv

- The unix system call that transform an executable binary file into a process are the exec family of system call
- Execl (execute and leave) system call load a new executable into memory and associated it with the current process
- Execl takes the path name of an executable program(binary file) as its first arguments
- The execl and excev() the filename must be fully qualified path name of the executable binary file

execlp and execvp

- The system call execlp is used to execute another program without returning
- It halt the currently running program, execute the new program and then exit
- Execvp a variant of execlp is used when the number of argument are not known in advance

execvp(filename, argp)

argp is an array of pointer to the argument, the last pointer is null to indicate end of list

- Letter added to the end of exec indicate the type of argument

  l – argn is specified as a list of argument

  v – argv is specified as a vector

  p- user path is searched for command and command can be shell program

exit()

- The exit() system call end a process and return a value to its parents
- The prototype that exit() system call is

exit(status)

Status is the integer between 0 to 255

This number is returned to the parent via wait(), as the exit status of the process

If zero – success

If non zero - failure

System call for IPC

- The system call used for inter process communication

  Pipe(files)

  dup(fd)

- Pipes
  - The pipe is the connection between two process
  - One process cannot read from buffer until the another has written to it
  - The Unix command line interpreter provide a pipe facility

    $ prog |more

Pipe system call

- The pipe is a system call that facilities inter process communication
- It open a pipe which is an area of main memory that is treated as a virtual file
- The pipe can be used by creating parent process as well as child process for reading and writing
- One process can write to this virtual file or pipe and another related process can read from it