## 1) Inter-process Communication in LINUX:

processes Communicate with each other and with kernal to their activities. Linux supports a number of Inter-process communication Mechanisms. They are

signals:-

signals are one of the oldest interprocess Communication methods used by Linux system. They are used to signal asynchroonous events to one or more process. A signal could be generated by a keyboard Interrupt or an Error condition such as the process attempting to access a non Existed location in its virtual memory there are set of defind signals that the kernal can generated in the system by using the kill command

1> SIGHUP
2> SIGTRAP
3> SIGPIPE
4> SIG KILL
5> SIGCONT
6> SIGVALRM
7> SIGWR
8> SIGINT
9> SIG
11> SIGUSRI
12> SIGALRM
13> SIG STOP
14> SIG URG
15> SIG PROF
16> SIGGDET
17> SIG BOS
18> SIGSEGV
19> SIGTERM
20> SIGTSTP
21> SIG XCPO
22> SIGWINCH
23> SIG ILL
24> SIG FPE
25> SIGCHLB
26> SIG CHLB
27> SIGTIIN
28> SIG XFSZ
29> SIGIO
30> SIG UL

process can choose to ignore mast of the signals that are generated with two notable Exceptions, neither the SIGSTOP signal which causes a process to halt its Exections nor the SIGKILL signal which causes a process to Exit can be ignored.

Signals have no inheritent relative priorities. signals are generated for a process at the same time then they may be presented to the process in any orde Linux Implemente the signals using information stored the task struck for the process about. Every process in the system can send signals to every other process, the kernal can and super user can.

pipes :-

It provides a mechanism for one process to stream data to another. A pipe has two ends associated with a pair of file descriptrs making it a one-to-one massaging or commuption mechanism. one end of the pipe is the write end which is associated with a file descriptors that can only be written. Anonymous pipes can be setup and used only b/w process that share parent child relationship Generally the parent process creates a pipe and then flocks child processes. Each child process gets access to the pipe created by the parent via the file descriptors that gets duplicated into their Address space this allows the parent to communicate with its children or the children to communicate with each other using shared pipe named pipes or (FIFO) are variant of pipe that allow communication b/w process that are not related to each other The processes communicate using named pipes by opening a special file known as a FIFO file. Thus one process opens the FIFO file for reading. the FIFO file on disk acts as the contract b/w the two processes that wish to communicate using named pipes by opening a special file known as a FIFO file. Thus One process opens the FIFO file for reading. The FIFO file on disk acts as the contract b/w the two process that wish to ② Communicate.

Sockets :-

1> SysV Message queues :-

The AT & T sysV message queues supports message channlin Each message packet sent by sender carry a message Number. The recievers can either choose to relieve all messages Excluding a paticular message no. or all messages

2> POSIX message queues :-

Supports message priorities. Each message packet set by the senders carry a priority number along with the message payload. The message get ordered based on the priority to read a message queue. when the reciver tries to read. The message with higher priority number get delivered first POSIX message queues also supports asynchonous message delivery using threads of signals based notification.

3> shared memory :-

This allows two or more process to Communicate data or more efficiently among themselves with minimal kernal intervention. There are two standard specifications for stardard memory.

<i> SysV shared memory :

Many applications even today use this mehanisms for historical reverse. It follows some of artifat of sysv IPC semantics.

(ii) POST x shared Memory:

The POSIx specification provides a more elegant shared memory interface on Linux POSIx shared memory is actually implement by using files backed by RAM based file system.

Semaphores :-

Semaphores are locking and synchronization Mechanism used most widely when process share resources linux supports both sysv semaphaoos and POSIX semaphores POSIX semaphores provide a more simple and elegant implimentation and this is most widely used when compared to sysv semaphores on Linux.

Futexs :-

Futexes are high performance low overhead locking mechanisms provide by the kernal. Direct use of futexes highly discouraged in system programs. Futexes are used

2) Write about process management in Linux?

like most Linux is general is a fairly stable system. OS's it a multitasking operating system. this mean that many process can be running at the same time what is currently running?

Ps - program used to look at process is called ps which stands for process. In its normal usages it will show you just the processes running in your ps(Aux) - If we add the argument access with ps then it will show a complete system view which is a bit more helpful.

grep - ps(Aux) does give quite bit of of output so people usually pipe the output to group to filters out just the data they are after Killing a crashed processes when a program crashes it can be quite annoying you try and close the window but nothing happend, it has become completly unresponsive we can easily kill firefox and then reopen it. To start off we need to identity the process id. Kill [signa] <PID> - It is the number next to the owner of the processes that is PID. we will use this to identity which process to kill. To do so we use a kill program. Normal users may only kill processes which they are the owners for the root users or the system may kill only ones processes.

Change priority of process with nice. and since As we have seen before, each process have a priority assigned to it, which indicates how much the process have to wait for other proceser to indicates and to free resources before it can access them. this priority can be specified before it can access them. this priority can specified with a values which ready to run. since used to set the priority which is already running.

3) Write about the file system of Linux.

On a linux system, everything is a file, it Somet is not a file, it is a process.

A linux system makes no difference b/w a file and a directory since a directory is Just a file containing names of other files. programme services, texts, images and so forth, are all filer

Sort of filer:-

Most files are Just files, called regular files. They contain normal data, for Example text files, Executable files or programs input for or output from a program

Directories; files that are lists of other files

specified files:- The mechanism used for used and o/p

Ex:- /dev

Links:- A system to make a file or directory visible in multiple parts of the system file tree.

Domain (Sockets):- A special file type similar to TCP/IP sockets, providing interprocess networking protected by the file systems access control.

Named pipes:- Act or more or less sockets and form away for process communications.

| symbol | meaning |
|---|---|
| - | regular file |
| d | Diractory |
| L | link |
| c | special files |
| s | sockets |
| p | Narmed pipe |
| b | Block device |

Is displays the files types, using the first charecters of input type.

partitioning :-

Linux uses more than one partition on the same disk, even when using the Standard Installgition procedure so some Explanation is called for one of the goals of having different partitions is to achive higher data security in case of disaster By dividing the hard disk in partitions data can be grouped and separat -ed . when accident occures. only the data in the parti -tion that got the hit will be damaged, while the data on the other partition will most likely data on the other partition will most likely servive data on the

Survive.

partition types:-

There are two kinds of major partitions on linux system

<i> data partition:

Normal linux system data, including the root partition containing all the data to start up and run the system.

<ii> swap partition:

Expansion of the computers physical memory. Extra memory on hard disk.

Most linux system use fdisk at installation time to set the partition type. The standard linux partitions have number 82 for swap, 83 for data.