S a n j e e v i

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID CONCEPTS AND LEVELS:

RAID

•**What is RAID?**

- Stands for **Redundant Array of Independent Disks**.
- Developed to meet the growing demands for data reliability and performance.
- Multiple hard drives are grouped together to form a single logical drive.

•**Why RAID?**

- Mass storage is successful only with the benefits of this data protecting scheme.
- Increases the performance and reliability of data storage by spreading data across multiple disks.

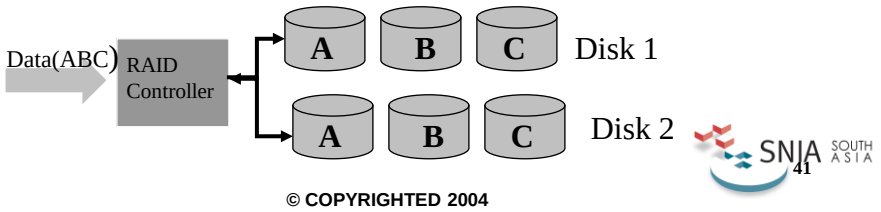
40

RAID: Information is increasing tremendously and to protect it from being lost is mission critical.For eg. Any down time affecting popular websites can mean loss of business.So RAID was developed by researches at UC- Berkeley, to increase the performance and reliability of data storage.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID Concepts

- RAID uses Mirroring, Parity and Striping.
- Mirroring
- Increases fault tolerance by having two copies of the same data on separate hard drives.
- Downtime is minimal and data recovery is simple.
- Increased cost and twice as much as storage.



Raid Controller:Manages how the data is stored and accessed across the disk arrays.It ensures that the OS sees only the logical drives and does not need to worry about the managing the physical disks.RAID controller can be either hardware or software.

Mirroring:

The system writes data simultaneously to both hard drives.when one hard drive fail, the system still continues to operate. The data is rebuilt from the existing copy.To set up mirroring the number of drives will have to be in the power of two.

The drawback here is both drives are tied up during writing process, but has a performance increase while dealing with reads.While one of the drive is used for reading, the free drive is used for other requests.

Mirroring is a good solution where data in a company is mission critical.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID Concepts (cont'd)

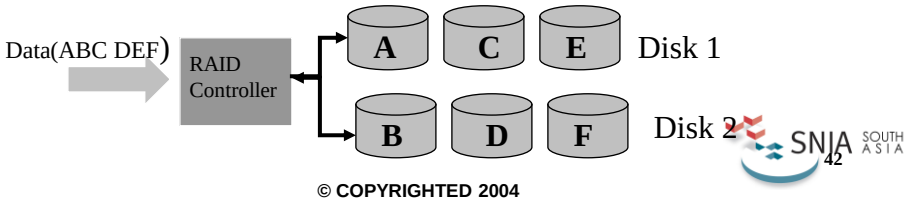
•Striping

•Improves performance by distributing data across all drives.

• The transfer rates for read and write operations are greatly increased.

• There are two levels of striping:

- Byte level striping: breaking up of data into bytes.
- Block level striping:breaking up of data into specific block sizes.



The main principle in striping is parallelism. Every piece of data that comes into the RAID controller is divided into smaller pieces. Depending on different techniques, either block or byte level striping is used.

Byte Level Striping:

For example, if data is broken into 16 bytes and there are 4 hard drives, the 1st byte is stored in the 1st hard drive and the 2nd in the 2nd hard drive and so on.So during reading, one has to wait only as long as it takes to read each piece since the drives are working in parallel.

Block Level Striping: Data is broken into block sizes.The size of the block depends on how large the application is.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID Concepts (cont'd)

- **Parity**
  - Data redundancy technique used in RAID.
  - Parity data is created using the logical operation called XOR on the data elements.
  - If any of the data elements is lost, it is recreated from the parity element and vice versa.
  - As in mirroring there is no need to keep two copies of data.
  - The parity can be either **distributed** across the multiple disks or be **dedicated** to a single disk.

43

For example, there are X number of data elements and it is used to create the parity element. This results in a total of X+1 data elements.

Mirroring and striping can be combined with parity to improve the performance.

RAID Levels

- **Combinations of mirroring, parity and striping results in various raid levels.**

Commonly adopted RAID Levels

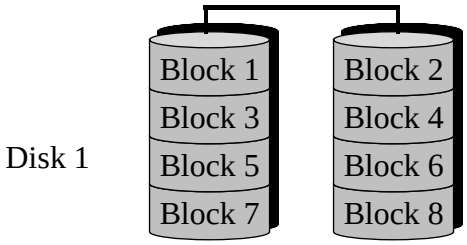
- RAID 0 – Striping (no parity)
- RAID 1 – Disk mirroring
- RAID 0+1 – Striping, each stripe then mirrored
- RAID 2 – Bit-level Striping, ECC Disk
- RAID 3 – Byte-level Striping, fixed parity
- RAID 4 – Block-level Striping, fixed parity
- RAID 5 – Striping, distributed parity
- RAID 6 – is two parity over all drives, Handles two disk failures.

44

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID 0: Striping

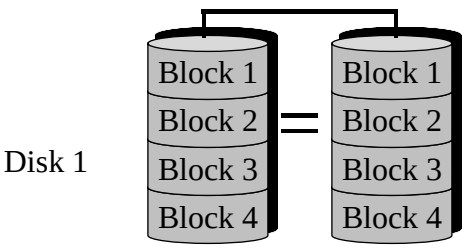
- Implements a **striped disk array**, the data is broken into
- blocks – each block written to separate disk drive
- Not fault-tolerant – not a "true" RAID
- Lower cost & higher access rate
- Applications in Image Editing & Video Production



Data redundancy is not present in this level. RAID 0 offers highest level of performance out of any single RAID level.It also offers the lower cost since no extra storage is involved. At least 2 hard drives are required, preferably identical.It is important to note that if any of the hard drives fails, all the data is lost.

RAID 1 : Mirroring

- Consists exactly 2 disk modules bound together as mirrored pair.
- Controller must perform 2 concurrent Reads or 2 duplicate Writes, per mirrored pair.
- If both disks fail, the RAID 1 mirrored pair becomes inaccessible
- Recommended Application-Accounting & Payroll.

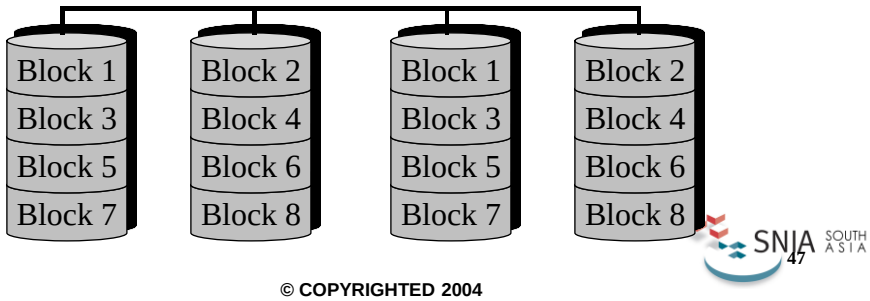


Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Two identical copies of data are stored on two drives. When one drive fails, the other drive still has the data to keep the system going.This adds data redundancy to the system.Even though the performance benefits are not great, some might just be concerned with preserving their data. It is ideal for applications that use critical data.

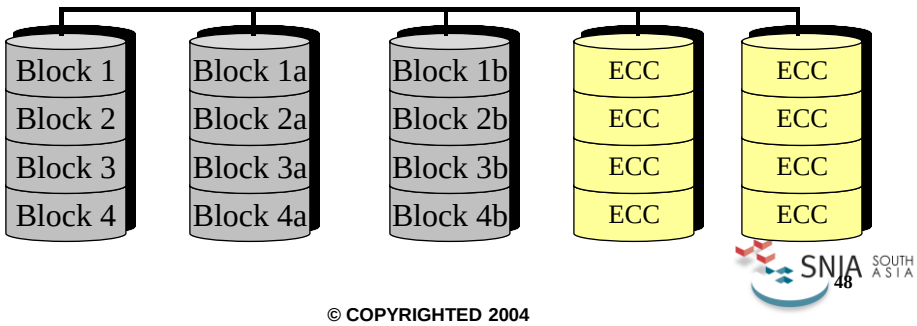
RAID 0/1 : Striping & Mirroring

- Even number of 4-16 disk modules
- Half are data disks and the other half are disk mirrors
- Uses block striping for performance & mirroring for redundancy
- - so mirrored RAID 0 group



RAID 2 : Bit-level Striping, Fixed parity

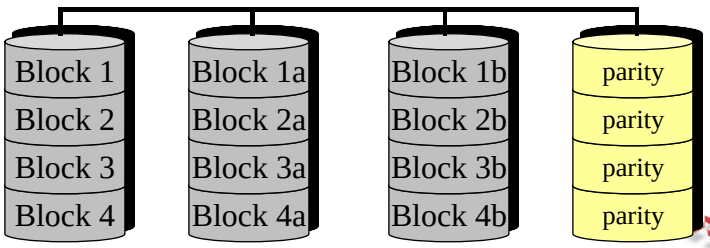
- Uses bit level striping with dedicated Error Correction Code (ECC).
- Use multiple drive dedicated ECC disks.
- Need high number of drivers for ECC generation.



Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID 3 : Byte-level Striping, Fixed parity

- Uses byte level striping with dedicated parity
- An additional drive dedicated to parity
- Uses Error Correction Code to detect errors
- Added parity slow down writes



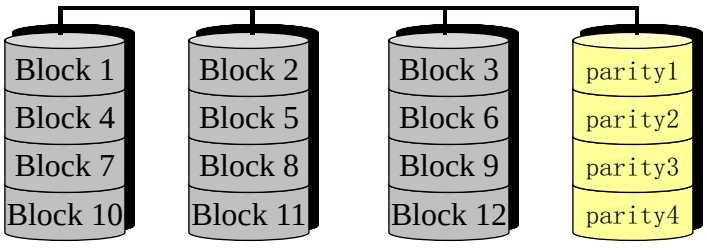
SNIA SOUTH ASIA

49

In this RAID level, Striping the data increases performance and using dedicated parity takes care of redundancy.At least 3 hard drives are required, 2 for striping and 1 for parity.The parity information has to be written to the parity drive whenever a write occurs. This increases the computation.

RAID 4 : Block-level Striping,Fixed parity

- Uses block level striping with dedicated parity
- Multi user read, single user writes .
- Uses Error Correction Code (ECC) to detect error.
- Added parity slow down writes.



SNIA SOUTH ASIA

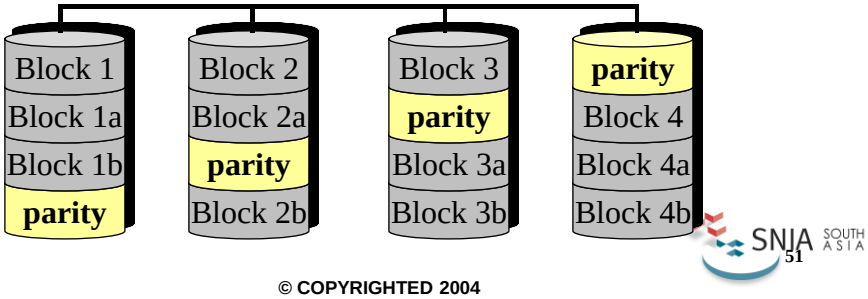
50

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

In this RAID level, Striping the data increases performance and using dedicated parity takes care of redundancy.At least 3 hard drives are required, 2 for striping and 1 for parity.The parity information has to be written to the parity drive whenever a write occurs. This increases the computation.

RAID 5 Striping and Distributed Parity

- Uses block level striping with distributed parity
- Data transferred to disk by independent read and write operations
- Combination of redundancy, cost effectiveness and storage efficiency
- Good for multitasking environment.



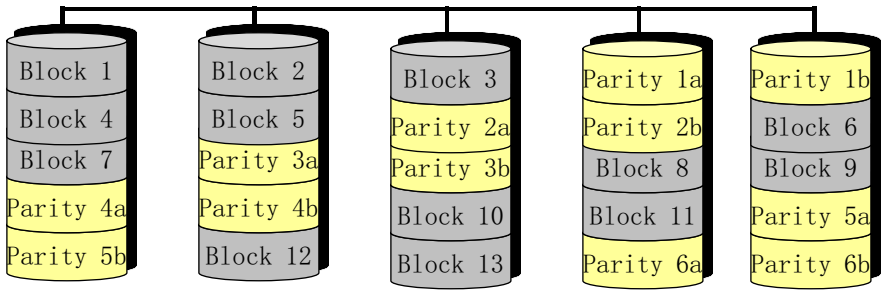
This level tries to remove the bottleneck of dedicated parity drive.With the use of a distributed parity algorithm, this level writes the data and the parity data across all the drives.Basically, the blocks of data are used to create the parity blocks which are then stored across the array.This removes the bottleneck of writing to just one parity drive.

However, the parity information still has to be calculated and written whenever a write occurs, so the slow down involved with that still applies.The fault tolerance is maintained by separating the parity information for a block from the actual data block.so when one drive goes, all the data on that drive can be rebuilt from the data on other drives.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

RAID 6 Two Parity Over All Drives

- 2 parity (P,Q) Redundancy, over all drives.
- Handles two disk failures



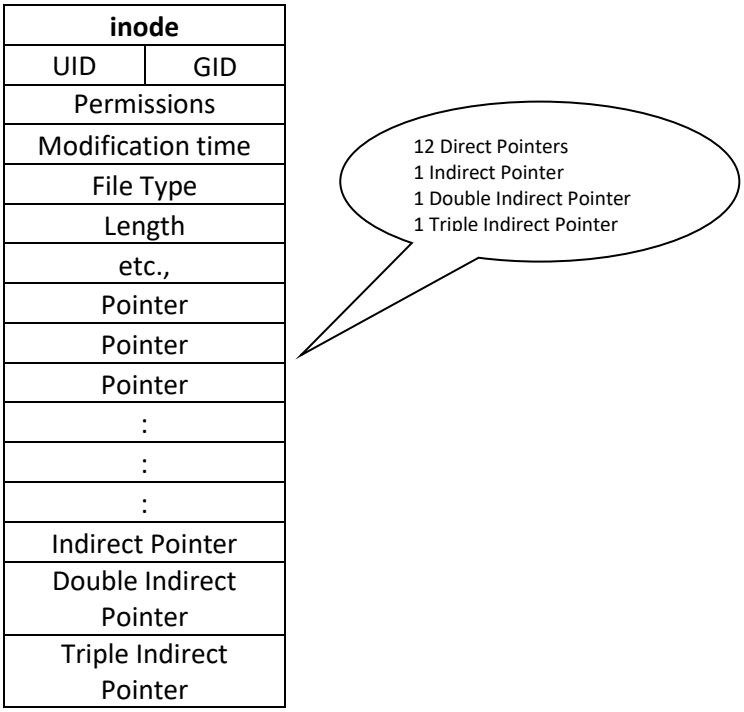
Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

The I-node based Filesystem:-

Inode:-

The inode contains information such as file permissions, owner, group, modification time, file length, pointer to datablocks that contain data which makes the filesystem.

It is a datastructure that contains important information pertaining to a file. When filesystem is created, a set of inodes will get created and usually 1% of diskspace is allocated for inodes.



A file inode is quite small, typically 32 or 64 bytes (jfs is 128 and jfs2 is 512 bytes). Use istat command to know inode information of file or directory. We can also use 'ls –li' command to list inode numbers with long list information. Inode number is unique within a filesystem.
The first block of the filesystem is super block and contain control information of filesystem such as

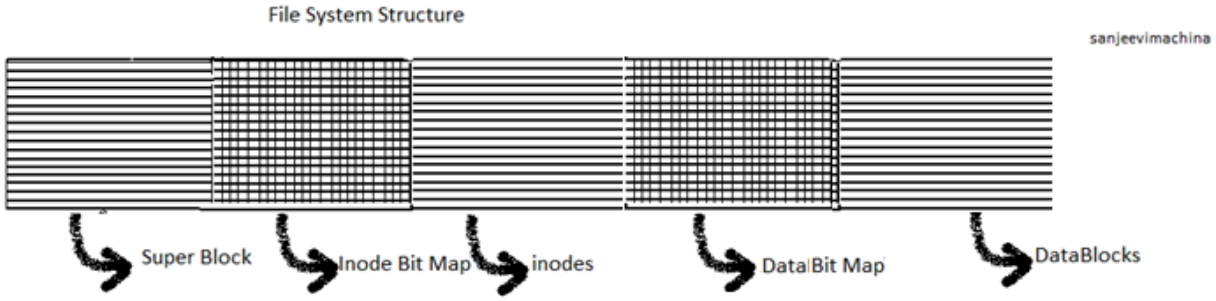
I. Overall size of fileystem in 512 byte block
II. Filesystem name
III. Filesytem version number (jfs or jfs2)
IV. Number of inodes
V. List of free inodes
VI. List of free datablocks etc.,

All the above information is stored in first block of filesystem i.e. Super Block. Corruption of this data may render the filesystem inaccessible. This is why the Operating System keeps second copy of

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Superblock at logical block number 31, so that we can restore 31st block with 0th block, in case of disaster (any failure).
Syntax for the same: dd count=1 skip=31 seek=1 if=/dev/lvname of=/dev/lvname

Superblock is created when the filesystem is created. It contains the information about inode, inode bitmap, data block bitmap and datablocks.



The first block of filesystem is Superblock, and this gives information regarding tunable parameters of filesystem such as number of inodes, number of datablocks, size of datablocks etc.,. It may include information such as volume name to identify the partition.

The datablock refers to 4KB of disk space. Data bitmap will show the allocation, whether the dtablock is free or allocated.

The inode bitmap is used keep track of allocated and free inodes in the filesystem. Every inode has corresponding bitmap in the inode bitmap table. The state of the bit describes the state of the object.

A file is represented by an inode and a bunch of datablocks. In a standard UNIX file system, files are made up of two different types of objects. Every file has an index node (inode for short) associated with it that contains the metadata about that file: permissions, ownerships, timestamps, etc. The contents of the file are stored in a collection of data blocks.

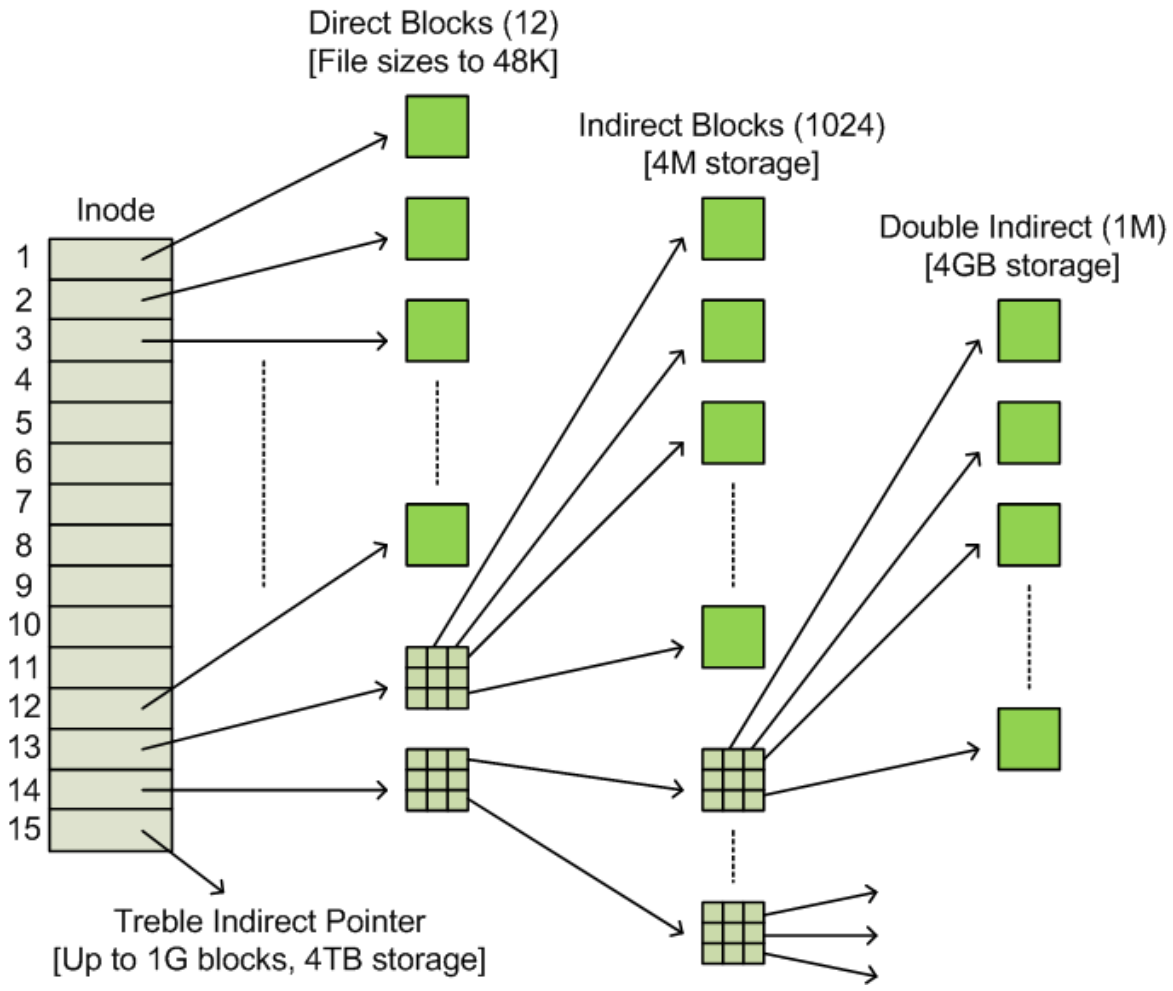
There are only fifteen block pointers in the inode. Assuming standard 4K data blocks, that means that the largest possible file that could be addressed directly would be 60K-- obviously not nearly large enough. In fact, only the first 12 block pointers in the inode are reserved for direct block pointers. This means you can address files of up to 48K just using the direct pointers in the inode.
Beyond that, you start getting into indirect blocks:

- The thirteenth pointer is the indirect block pointer. Once the file grows beyond 48K, the file system grabs a data block and starts using it to store additional block pointers, setting the thirteenth block pointer in the inode to the address of this block. Block pointers are 4-byte quantities, so the indirect block can store 1024 of them. That means that the total file size that can be addressed via the indirect block is 4MB (plus the 48K of storage addressed by the direct blocks in the inode).
- Once the file size grows beyond 4MB + 48KB, the file system starts using doubly indirect blocks. The fourteenth block pointer points to a data block that contains the addresses of other indirect blocks, which in turn contain the addresses of the actual data blocks that make up the file's

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

contents. That means we have up to 1024 indirect blocks that in turn point to up to 1024 data blocks-- in other words up to 1M total 4K blocks, or up to 4GB of storage.

- At this point, you've probably figured out that the fifteenth inode pointer is the trebly indirect block pointer. With three levels of indirect blocks, you can address up to 4TB (+4GB from the doubly indirect pointer, +4M from the indirect block pointer, +48K from the direct block pointers) for a single file.



Filesystem Check (to check integrity of filesystem):-

e2fsck: check a Linux ext2/ext3/ext4 file system. Command e2fsck is used to check the filesystem consistency. Below is the syntax:-

e2fsck /mountPoint
Ex:- e2fsck –y /mntpoint
Where –y : repair any errors

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

e2fsck –n /MountPoint

Where –n : Don't repair (reporting mode). Open the filesystem read-only, and assume an answer of `no' to all questions

e2fsck is used to repair any filesystem if there is any inconsistency. Running fsck on mounted filesystem (opened filesystem) is not recommended and doing so, cause data corruption and is always recommended to run fsck after unmounting filesystem.

Phases of e2fsck :

e2fsck -fy /dev/sde

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

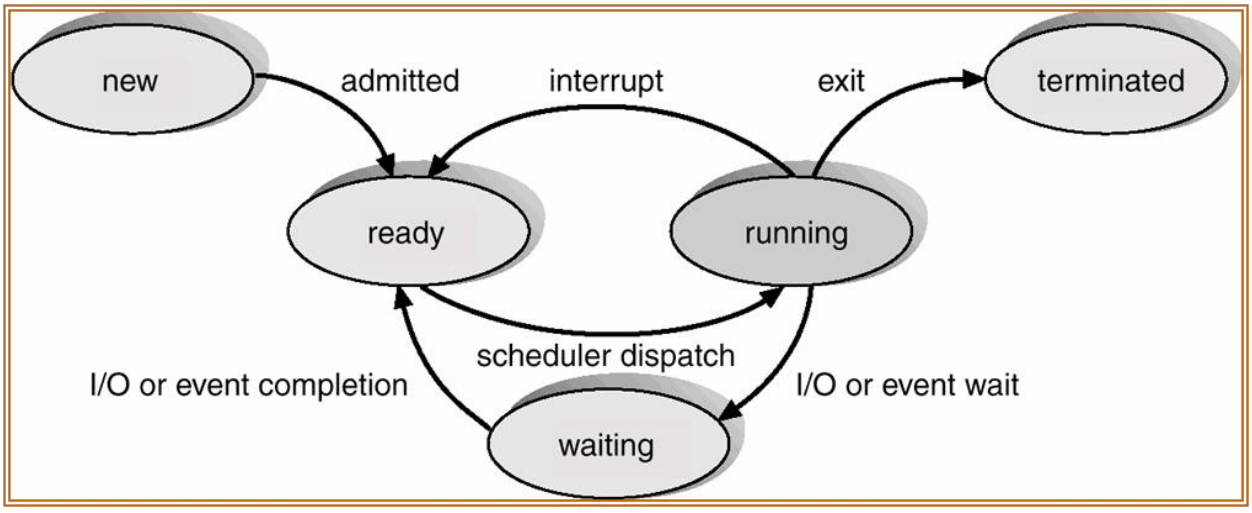
/dev/sde: 3161/6553600 files (1.1% non-contiguous), 8276142/26214400 blocksfsck command can be used in read only mode

Ex:- fsck –n /mntPoint

Using –y flag, in e2fsck command attempts to repair the filesystem. Repairing the inconsistency filesystem may cause loss of information related to datablock and inodes and hence there may be loss of data. However, in such cases (when data is loss), the removed datails kept in lost+found directory (this directory is created by default at the time of creation of the filesystem)

Process State Diagram:-

A process is program in running state. The following is process state diagram.



When you divided real memory in to 4KB units, is called frames (or page frames) and when you divide virtual memory in 4KB units, is called pages.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Virtual memory (or swap memory or paging space) is illusion to operating system as if it is having more memory. The virtual (or swap or paging space or logical memory) is brought from hard disk space.

In case of memory scarcity, least recently used programs are brought from real memory to virtual memory, so that, RAM space (real memory or physical memory or primary memory or simple memory) is cleared to load new job or a process.

When CPU is executing a job, tries to load the jobs from memory to CPU. There is a chance that the referenced page is not found in the RAM, and need to bring it from paging space, and this is called a pagemiss.

When the system need to access the page, and the same page is already exists in the memory and so that the data can be accessed without any swapping operations and this situation is called pagehit.

Pagefaults can cause the swapping (pageIn and PageOut). The thumb rule is that to have swaping space double to RAM size.

Fork:- Fork is a system call which is used to allocate memory dynamically to the process. Form system call is used to create a process. When there is no real memory to execute job (or process), we will get “fork failed” error, which means that there is no enough real memory.

Swap Space:-

For a process to be actively running, it must be loaded into memory. When it is loaded into memory, it is assigned a number of 4 KB areas called page frames. As more processes are loaded into memory, memory may become full. Not everything that resides in memory is active. When memory is full, memory is scanned to locate those page frames that are least-recently used. When one is located, a 4 KB block or "page" of disk space is allocated and the data from the page frame is moved to disk. This area on disk is called paging space.
The swap space is a reserved area on disk that can contain data that resided in memory but was inactive and was moved to make room for processes that are active. If a "paged-out" process is needed in memory again, the page is retrieved and brought back into memory or "paged-in".

Swap space is disk storage information that is resident in virtual memory, but is not currently being accessed. As memory fills, inactive pages are moved to the paging area on disk. It is very important to remember that paging is a temporary holding area for inactive pages; it is not a substitute for real memory.

If your machine has many active processes, it will require more real memory. You must make sure the machine has enough memory to maintain all the active processes. If you run out of memory, your machine will reach a constant state of paging called "thrashing".

As it attempts to make room in memory, it completes **a swap-out**; as soon as the page reaches the disk, it is needed again because it is still active. Your machine's resources will be wasted performing only paging activity and no real work will get done. Increasing the amount of swap space when your machine is thrashing will not solve the problem. Thrashing is result of not enough real memory.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Paging space is created during installation. The initial size is dependent on the amount of RAM in your system. If RAM is greater than or equal to 64 MB, paging space is RAM + 16 MB. If RAM is less than 64 MB, paging space is twice the size of RAM.

This is just a starting point. This is not necessarily the amount of the paging space that is right for your machine. The number and types of applications will dictate the amount of paging space needed. Many sizing "rules of thumb" have been published, but the only way to correctly size your machine's paging space is to monitor the amount of paging activity.

Monitoring the activity is done with the command swapon -s. This command and its output will be covered shortly. If your system runs low on paging space, a message will be sent to the console and sometimes to users as well. At this point the system will be unable to start any new processes until some running processes are terminated or release allocated memory. This situation should obviously be avoided. If any of the following messages appear on the console or in response to a command on any terminal, it indicates a low paging space.

Placement of swap:-
Placement and size of your paging space will impact its performance. The following are tips for paging space. Do not have more that one paging space per disk. The paging space is allocated in a round robin manner and will use all paging areas equally. If you have two paging areas on one disk, then you are no longer spreading the activity across several disks.

Paging space will perform best when it is not competing with other activity on the disk. Use disks that do not have much activity. Paging spaces should be roughly the same size. Because of the round robin technique that is used, if they are not the same size, then the paging space usage will not be balanced. Smaller paging areas will fill faster.

Link Files:-

Syntax:-

ln sourceFile TargetFile

where source file is a file which exists already and target file is a new file which we need to create. There are two type of link files

i) Hard Link
ii) Soft Link

- ➔ UNIX file consists of two parts i)data part ii)Filename part.
- ➔ Filename part carries a name and associated inode number
- ➔ The data part associated with data and inode carries the map where the data is, the file permissions etc., for that data.

Hardlink:-
More than one file can be referenced by a same inode number and these files said to be hard linked each other.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Directory as a file is just an array of filenames. When a directory is created, it initially populates filenames (. and .. special files). The . is a file that maps with the current directory and .. maps to the parent directory.

Softlink:-

Syntax:-

ln –s sourceFile TargetFile

where targetFile is newly created softlink

A softlink is a file which contains the name of the other file. It also called Symbolic link or Sym link.

Differences between Hard and Soft link:-

*)These files share the same inode number	*)These files have different inode number
*)Hard link file exists withing the same file system	*)Softlink can exists in different filesystem and also can exists in the same filesystem as well
*)When the original file is deleted, the hardlink file willnot be deleted and can access the contents by using link file	*)If the original file is deleted, the softlink can not be accessible. However, the softlink will not be deleted
*)Hardlink files can be identified by looking at link count on ls -li command or istat filename command	*)Softlink files can be identified by --> Symbol

Note:-You can not create hardlink referring to different filesystem. Incase of hardlink both source and target must be in the same filesystem and they can be in different directories provided those directories are in the same

Devices:-

A number of pieces of hardware and software must interact correctly for the device to function correctly.

Physical Devices:-

Actual hardware that is connected in some way to the system.

Ports:-

The physical connectors/adapters in the system where physical devices are attached. Most ports are programmable by the system software to allow attachment of many different types of devices.

Device Drivers:-

Software in the kernel that controls the activity on a port and the format of the data that is sent to the device.

Logical Devices:-

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Software interfaces (special files) that present a means of accessing a physical device to the users and application programs. Data appended to logical devices will be sent to the appropriate device driver. Data read from logical devices will be read from the appropriate device driver.

/dev:-

The directory which contains all of the logical devices that can be directly accessed by the user.

Device Types:-

There are a large number of devices that can be configured in the system. Devices can be one of two types:

Block device is a structured random access device. Buffering is used to provide a block-at-a-time method of access. Usually only disk file systems.

Character (raw) device is a sequential, stream-oriented device which provides no buffering.

The ls -l command allows you to see the type of a file. A special file (in the /dev directory) will be indicated by a b in the first column for a block device or a c for a character device.

Network

Network - A network is a group of computers connected together in a way that allows information to be exchanged between the computers.

Node - A node is anything that is connected to the network. While a node is typically a computer, it can also be something like a printer.

Segment - A segment is any portion of a network that is separated, by a switch, bridge or router, from other parts of the network.

Backbone - The backbone is the main cabling of a network that all of the segments connect to.

Topology -Topology is the way that each node is physically connected to the network.

Local Area Network (LAN) - A LAN is a network of computers that are in the same general physical location, usually within a building or a campus. If the computers are far apart (such as across town or in different cities), then a Wide Area Network (WAN) is typically used.

Network Interface Card (NIC)- Every computer (and most other devices) is connected to a network through an NIC. In most desktop computers, this is an Ethernet card (normally 10 or 100 Mbps or 1 Gig) that is plugged into a slot on the computer's motherboard.

Media Access Control (MAC) address - This is the physical address of any device -- such as the NIC in a computer -- on the network. The MAC address, which is made up of two equal parts, is 6 bytes long. The first 3 bytes identify the company that made the NIC. The second 3 bytes are the serial number of the NIC itself.

Unicast –A unicast is a transmission from one node addressed specifically to another node.

Multicast - In a multicast, a node sends a packet addressed to a special group address. Devices that are interested in this group register to receive packets addressed to the group. An example might be a Cisco router sending out an update to all of the other Cisco routers.

Broadcast - In a broadcast, a node sends out a packet that is intended for transmission to all other nodes on the network

Media speed of an Ethernet adapter

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Speed is any one of below

a) Auto_Negotiation
b) 10_Full_Duplex
c) 10_Half_Duplex
d) 100_Full_Duplex
e) 100_Half_Duplex
f) 1000_Full_Duplex
g) 1000_Half_Duplex

Note that devices in the network that communicate each other must be communicate at same speed. If two devices at different speed, connection will not get established. Communication can happened one sided at a time (half duplex) or two sided (full duplex)

Half Duplex:- It is single direction, two devices in network can not communicate simultaneously. When one device start communication, the other devices will be in receive mode and vice-versa. This makes communication delay.

Full Duplex:- Communication happens simultaneously by both the devices. This makes data transfer fast.

Maximum Transmission Unit: A piece of data traveling on a network is known a datagram. A datagram travels in network frames. Physical networks can place an upper limit, that can fit into a physical frame. This limit is known as the MTU (Maximum Transmission Unit)

Ethernet terminologies:-

eth0: The notation eth0 is used to specify the hardware adapter.

Protocol:-

Protocol is a set of rules and conventions to be followed in communication. Protocols are implemented with Software. Protocols are operating system independent.

Ex:- telnet, ssh, ftp, nfs etc.,

Port:-

A port is a communication channel. Each port is associated with particular protocol of particular service. All services and respective port numbers are defined in /etc/services file.

Protocols for computer networking generally use packet transmission technique to send and receive data. A port or logical port is a communication channel through which a connection established. Each logical port is associated with a protocol and a unique protocol number is assigned.

TCP:- Transmission Control protocol, which is connection oriented protocol, where is an acknowledgement for each packet that is been transmitted to another host in the network. As there is an acknowledgement, the time taken and bandwidth used is more compare to UDP.

UDP:- User datagram protocol, which is connection less. There is no acknowledgement for the packet been sent. Hence there is no guarantee if the packet been reached the other end. However, UDP is faster compare to TCP as there is no acknowledgement happens.

Physical Port:-

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

A physical port is a physical connection of an adapter within the system where physical devices are attached. Each protocol is associated with a service name also called protocol name and is defined in /etc/service file. Each line in /etc/services file defines the protocol name and associated port number. To comment a line # symbols are used. The protocol definitions in /etc/services look like below

PROTOCOLNAME PORTNUMBER/TCP Or UDP # COMMENT

TCP/IP (transmission control Protocol/Internet Protocol):-

It is a set of protocols which defines various types of aspects where two computers on a network may communicate with each other.

A protocol is a set of rules which describes the mechanism and data structure involved**.**

Ping command:-

The ping command is used for basic networking troubleshooting. It got ability to communicate with

a) Ping sends ICMP echo requests
b) Ping expects to receive ICMP echo replies

Syntax:- *ping ipaddress*
Ping hostname
Ping –c Number IP_or_Hostname
Ex:- *ping -5 192.168.1.1*

Ping command works with ICMP (Internet control messaging protocol). Ping command is used to check whether the other system is alive or not.

Firewalls may cause ICMP traffic to get blocked and hence ping may fail. So an administrator should not conclude that the server is down when ping fails. You may need to check with network team to see if your ICMP packets are getting blocked at firewall. Usually if two machines are at the same subnet, and hence the packets will not travel through firewall (usually), and ping should work.

Name Resolution :-

Name (hostname) should be converted (resolute) to IP, when host name is used in the communication. UNIX server keeps a local /etc/hosts file in order to keep mapping host to IP static lookups. When hostname is used, /etc/hosts file is visited and if there is a successful lookup (first successful search), the IP will be picked up and with that IP communication will get established.

Name resolution will also happens with DNS, which is central repository to convert name to IP and vice-versa. The DNS ip address is given in /etc/resolv.conf file, so that the UNIX server query the given DNS server to look up name to IP and vice-versa.

You can have /etc/nsswitch.conf file to have name resolution order. The systax is

hosts: files dns

Address Resolution Protocol:-

Ethernet Adapter consists of 48 bit long Physical Address assigned at the time manufacture. It is also called Media Access Control Address (or MAC address). MAC address operates at data link layer. The IP address is logical address and there should be a provision to convert the IP to MAC and vice-versa. This is achieved with Address Resolution Protocol.

ARP resolves layer 2 (OSI Model) MAC addresses to layer 3 ip addresses. You can examine the MAC address by using arp command. Arp table is dynamically updated by the Kernel. When two tcp/ip hosts

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

communicate, arp is performed to translate the IP address to the MAC address. If more and more routers separate the communicating hosts, then MAC address of the default router's (gateways) interface is stored by each client
To see MAC to IP Conversions:

arp -a

The MAC address is unique for an NIC and will not be reused, though the device got destroyed. To see a)you can ifconfig (in RHEL7, user ip a command)command

b)To list all interfaces and its link status

netstat –i (lowercase i)

c)To list all interfaces (do not resolute to names)

netstat –in

d)To list all network protocol information

netstat –an

Uptime Command:-
Uptime command prints the current time, length of the time the system has been up, the number of users online and the load average. The load average is the number of processes over the preceding 1, 5, 15 minutes intervals. The output of uptime is the heading line provided by w command.
SCSI Devices:-

1. HVD (High Voltage Devices)
2. LVD (Low Voltage Devices)

The default SCSI id is 7.The SCSI adapter can address maximum of 16 targets. The maximum SCSI ID is 15. The SCSI ID 0 is reserved for adapter itself. Many devices will take default SCSI id as 7. Hence it is not recommended to use SCSI-ID 7 for any new device.

Device addressing location codes are used for device addressing. Location codes for a device is a path from the adapter in the CPU drawer or system unit, through the signal cables and the asynchronous distribution box (if there is one) to the device. Location codes consists of up to four field of information depending on the type of the device.

Traceroute command:-
The traceroute command prints the route that IP packets takes to network host. The traceroute command attempts to trance the route a max of 30 hops (default value).

Network File systems (NFS)

NFS is a distributed filesystem protocol originally developed by SUN Microsystems in 1984. NFS allows users on client computers to access files over a network in a manner similar how local storage is accessed. NFS is built on remote procedure call (RPC).
Remote Procedure Call:-
RPC is an inter process communication technique that allows client and server software to communicate.
Cleint:-
A program such as process or task that request a server provided by another program.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Server :-
A program such as process or task that respond to a request from the client.

Portmap Daemon:-
The portmap maps RPC services to the ports where they are listening on. RPC processes notify portmap when they start, registering the ports they are listing on and RPC program numbers they expect to serve. The portmap daemon converts RPC program numbers into port numbers. Its port number is 111.

Kernel Extension:-
The kernel extension is a machanism as a means of dynamic loading of pieces of code in the kernel without need of recompilation of the kernel. These pieces of code is generally known as plug-Ins or Kernel extensions.

Find command:-
To search a given file exists in the server or not
syntax :-
find /path/where/SearchTobePerformed -name NameOfFile -print
where there is a successful search, an absolute path of file or dirname will be printed on the screen.
ex:- find /path -print
The above command will print all filenames with absolute path under directory /path.

- To print all files in the server:
  find / -print
- To print files based on modification time
  find /Fsname -ctime 2 -print
  where -ctime 2 : last modified since last two days
- Find command can be used to see files
  a) based on modified since x hours
  b) size > x mb
  c) size < y mb
  d) size=z mb
  e) modified time etc., [see man pages for more options :-)]
- Find command to search files only in the particular device (such as particular LV)
  syntax:- find /path -xdev -name finamePattern -print
  ex:- to search hosts file only in /dev/hd4 (its mount point is /)
  find / -xdev -name hosts -print

Taking backup of individual files or filesystems data:-
Operating System provides native command to backup and restore files or directories, filesystems
To backup files:
Syntax:- backup -i -q -v -f /dev/DevName filenameTobeBackedUP
where -i: read filename from standard input device (keyboard)
-q: quite (do not prompt)

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

-v : verbose (output printed on the screen)
-f : devicename (such as rmt0, rmt1 etc.,)

ex:-

backup -iqvf /dev/rmt0

and then provide filenames one in a line and press enter. Once done as end of the input, press control D (ctrl + d)

To list table of contents on the tape which is been backed up using backup command and below is the syntax:-

restore -TVqf /dev/rmt0

where -T : List table of contents only
-v : Verbose
-q : Quite
-f : DeviceName

You can make a file with list of file names, one per line, and redirect the contents of that file to backup command, so that you no need to enter many files with keyboard interactively.

ex:- cat /tmp/list | backup -ivqf /d/ev/rmt0

Where /tmp/list is a text file that consists of filenames one per line

To take backup of / (ie /dev/hd4 contents only)

find / -xdev -print | backup ivqf /dev/rmt0

To restore file(s) which has been backed up by backup command

restore -qvf /dev/DevName /File/Name/to/Restore1 /File/Name/Torestore2 ..

To know a particular file is backed up or not:

restore -Tvqf /dev/rmt0 | grep /path/to/FileToSearch

To restore everything on tape (which was backed up by backup command)

resteore -qvf /dev/rmt0

To backup of entire server

find / -print | backup -ivqf /dev/rmt0

Tape Archive Command (tar):-

tar command manipulates archives by writing files or retrieving files from an archive storage medium.

To Archive:

tar -cvf archiveName /Dir_Or_FileToArchive

ex:-

tar -cvrf /dev/rmt0 /etc

In the above example, /etc contents are archived into tape device rmt0

1. You can also archive into a single file and store it on local disk

   tar -cvf /tmp/etc.tar /etc

   The above command archives /etc/ into a single file /tmp/etc.tar

2. You can list tar archive contents using

   tar -tvf ArchiveName

   where -t : table of contents
   -v : Verbose
   -f : archiveName (can be tape /dev/rmt0)

3. You can extract an archive

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

tar -xvf ArchiveName
-x : extract
Examples:-

*) To archive /var contents to tape, /dev/rmt0
*tar -cvf /dev/rmt0 /var/**
*) To archive /var contents to local Filesystem /backup
*tar -cvf /backup/myarchive.tar /var/**
*) To see contents of tar'ed file
tar -tvf /backup/myarchive.tar
*) To see tar'ed contents on tape
tar -tvf /dev/rmt0
*) To extract archived contents to their original location
tar -xvf /backup/myarchive.tar
*) To extract specific files from archive
tar -xvf /backup/myarchive.tar /fileName/to/Restore1 /FileName2

Note:-
tar commond will not compress the data. It only archives into to single file. You can zip (gzip) and an archive file to compress. gzip and gunzip commands are using to compress and uncompress respectively.

To commpress a file
gzip FileName
You will get out FileName.gz file in compressed form
To uncompress(gunzip) the compressed file:
gunzip FileName.gz
FileName.gz will be uncompressed and resulting file will be 'FileName'

SUDO

Sudo (su "do") allows a system administrator to delegate authority to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while providing an audit trail of the commands and their arguments.

Sudo (su "do") allows a system administrator to give certain users (or groups of users) the ability to run some (or all) commands as root while logging all commands and arguments. Sudo operates on a per-command basis, it is not a replacement for the shell. Its features include:

- ➔ The ability to restrict what commands a user may run on a per-host basis.
- ➔ Sudo does copious logging of each command, providing a clear audit trail of who did what.
- ➔ Sudo uses timestamp files to implement a "ticketing" system. When a user invokes *sudo* and enters their password, they are granted a ticket for 5 minutes (this timeout is configurable at compile-time). Each subsequent *sudo* command updates the ticket for another 5 minutes. This

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

avoids the problem of leaving a root shell where others can physically get to your keyboard. There is also an easy way for a user to remove their ticket file, useful for placing in a .logout file.

➔ Sudo's configuration file, the **sudoers** file, is setup in such a way that the same sudoers file may be used on many machines. This allows for central administration while keeping the flexibility to define a user's privileges on a per-host basis. Please see the samples sudoers file below for a real-world example.

➔ The /etc/sudoers file should be edited with **visudo** command. There are two reasons for the same.

- It prevents two users from editing the file at the same time.
- It also provides limited syntax checking.

➔ /etc/sudoers files composed of two types of entries:

a) Aliases (basically variables)
b) User Specification (Which specify who may run what)

a)Aliases:- There are four kind of aliases
1)User_Alias
2)Runas_Alias
3)Host_Alias
4)Cmnd_Alias
Each alias definition is in the form of

Alias_type NAME_OF_ALIAS=item1, item2, item3……

Where Alias_type is one of liases discussed above, and name of the alias string is uppercase letters and underscores characters.

It is possible to put several aliases in a single line by joining with a colon(:)

```
User_Alias      BACKUPADMINS=user1,user2,user3:OSADMINS=user4,user5,user6
```

When multiple entries match, they are applied in an order and the last match is used.

Special Characters and reserved words used in sudoers file:-
1 .Hash symbol indicates a comment
2. A reserved word ALL is a built-in alias and always cause a match to be success.
3. The following characters must be escaped with a backslash '\' when used as a part of words

@, !, =, :, (,), \, |

User list: A user list is made up of one or more user names, user ids, system group names and ids, net groups, etc., Each list item may be predefined with an exclamatory symbol. An odd number of exclamatory (!) symbols negates the value of an item and even number just cancels each other out.

Eg:- ! (! true)
! (false)
True

Commad List:- It is a list of one or more command names or directories and other aliases. A command name is fully qualified filename which may include shell style wild cards.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Tags:- A command may have one or more tags, associated with it. There are 8 possible tag values: NOPASSWD, PASSWD, NOEXEC, EXEC, SETENV, NOSETENV, LOG_INPUT, LOG_OUTPUT, NOLOG_INPUT, NOLOG_OUTPUT

By default sudo requires that a user authenticate him or herself before running a command. This behavior can be modified by NOPASSWD tag.

Ex:- Sanjeevi Linux12=NOPASSWD:/usr/sbin/bootinfo,/usr/bin/startsrc

The above syntax would allow the user Sanjeevi to run /usr/sbin/bootinfo and /usr/bin/startsrc command as roon on machines Linux12 without authenticating him self.
The generic Syntax is

User host = commands

ALL ALL = ALL This means, all users, on all hosts and all commands

Previlege Specification Examples:-

Syntax:-

1) Users host = commands
   ALL ALL = ALL
2) UserAlias HostAlias = (RunasAlias) CommandAliase

Assume that you want all users (fin1, fin2, dbadmin) to run mksysb, savevg and backup commands, here is the syntax:-

Define user Alias:

User_Alias ADMINS=fin1,fin2,dbadmin

Define Command Alias

Cmnd_Alias BACKUPS=/usr/bin/mksysb, usr/bin/savevg,/usr/bin/restvg

Now the privilege specification is :

ADMINS ALL = BACKUPS

You can set, in such as way that no self authentication is required

ADMINS ALL-NOPASSWD:BACKUPS

By default, self authentication is enabled.

Default Specification:-

Used to define certain default veriables. Log file specification can eb specified in this section:
Defaults logfile=/var/adm/ras/sudo.log

Secure Shell Utility:-

ssh Port Number: 22

Syntax :- ssh -l username IpAddressOrServerName
or
ssh ServerNameOrIPAddress
ex:- ssh -l testuser myserver
or

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

ssh myserver
or
ssh username@ServerNameOrIpaddress

ssh Server:-

- Daemon is : sshd
- Configuration file is : /etc/ssh/sshd_config
- You can use startsrc and stopsrc to start and stop daemon.
- When you install ssh software a user sshd is created automatically.

sshd::202:201::/var/empty:/usr/bin/ksh*

scp:-

scp command is used for secure copy. ie. you can copy files from one machine to another machine in secured way (usually rcp or ftp are not secure and hence scp is used)

Syntax:-

scp Sourceservername:/path/of/fileName RemoteServerName:/path/of/fileCopied

Or

scp filename1 RemoteServerOrIP:FileName
where filename1 is local file

Or

scp Filename1 username@RemoteSErver:FileName

- Secure socket layer provides a secure layer and is pre-requisite for ssh
- ssh application is used to deal with confidentiality, authentication, authorization areas of security.
- ssh is a protocol for secure access, and used to login to remote host or secure copy of data from one host to another.
- ssh is a replacement for telnet, rlogin, rsh and ftp.
- ssh uses encryption and decryption mechanism to protect the data from unauthorized access.
- telnet sends all the data in a clear text and any hacker host can see the traffic. This is a disadvantage with telnet and can overcome with ssh which uses encryption and a hacker can not understand the encrypted data until unless hacker is having a private key.
- ssh provide secure ftp (sftp) and is a subsystem of ssh. Separate protocol layer over ssh will handle sftp.

Components of ssh:-

1.sshd server daemon:-

sshd is a daemon tha allows incoming ssh connection to a machine, and handles authentication.

2.ssh client:-

ssh is a client program(/usr/bin/ssh) and is used to connect ssh servers.

3.Session:

A session is an ongoing connection between a client and server.

User imitates the ssh connection which attempts to connect port No. 22 (default) on remote host. If this is successful, sshd on remote host creates a child process to handle ssh connection between two hosts. Ssh client and ssh server can encrypt and decrypt messages.

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

sshd configuration file:-

- The configuration file is : /etc/ssh/sshd_config
- The sshd daemon reads /etc/ssh/sshd_config file. This files contains keword-Argument Pair.
- sshd_config is used to configure the login controls of ssh
- Lines that are preceded with # symbol mean that they are commented out.
- Arguments are case sensitive and keyword are case insensitive. Below are some of keyworkd and arguments:

AllowUsers, AllowGroups, DenyUsers, DenyGroups, Banner, MaxSessions, PermitRootLogin, PidFile, Port

SSH Key Generation:-

Two type of keyes:
RSA (Row Rivest, Adi Shamir and Leonard Adlemon)
DSA (Digital signature Authentication)

The public key can be used to encrypt a message and only a holder of private key in the other side can decrypt the data.
The private key is used to decrypt and the public key is used to encrypt. We can generate ssh keys either by RSA or DSA using the below command:

```
ssh-keygen -t TypeOfKeys
        ex:- ssh-keygen -t rsa
or
    ssh-keygen -t dsa
```

Password less authentication between two hosts using ssh keys:-

You can have secured password less connectivity between two hosts by using ssh keys. Here is the procedure.
Assumptions:-

1. Assume that two hosts Linux48 and Linux50 installed with ssh software
2. Create user (ex:- ora_adm) in both the servers
3. Generate RSA or DSA key pair (This creates .ssh directory in users home directory)

To make Linux48 as trusted to Linux50, you need to copy the public key content of Linux48 server to Linux50 server's authorized_keys file which is located in $HOME/.ssh/ directory (ex:- /home/ora_adm/.ssh/authorized_keys. Here is the procedure for the same:

In Linux48 Server:

1. Generate ssh keys (can be of type RSA or DSA or both)
2. $ ssh-keygen -t rsa
   Generating public/private rsa key pair
   Enter File in which to save the key (/home/ora_adm/.ssh/id_rsa) : Hit Enter key
   Enter Passphrase (Empty for no passphrase):
   Enter the same passphrase again:
   Your identification has been saved in /home/ora_adm/.ssh/id_rsa

Sanjeevi Machina
sanjeevim@yahoo.com
Ph: +919845311569/9611131569

Your public has been saved in /home/ora_adm/.ssh/id_rsa.pub
The key finger print is : xxxxxxxxxxxxx

Note: In the above passphrase given as empty, so that you no need to enter any pass phrase (id password less)

3. Copy or ftp or scp the public key (/home/ora_adm/.ssh/id_rsa.pub) to remote machine (Linux50) and copy the content of id_rsa.pub of Linux48 to Linux50's /home/ora_adm/.ssh/authorized_keys file
ex:- in Linux48, scp /home/ora_adm/.ssh/id_rsa.pub ora_adm@192.168.1.50:/home/ora_adm/.ssh/id_rsa.pub

In Linux50 Server:-

Login as ora_adm user and in its home directory create .ssh directory (if it does not exists)
cd /home/ora_adm/.ssh and then cat id_rsa.pub >> authorized_keys

4.Now you made ora_adm of Linux48 as trusted user to Linux50's ora_adm (you can make another user too)

5.Now from Linux48, try to ssh to Linux50 as ora_adm, and you will not be prompted for password.
ex:- ssh 192.168.1.50 or ssh -l ora_adm 192.168.1.50