Linux interview questions

Linus [Torvalds quietly released the latest Linux 4.14 kernel](https://lkml.org/lkml/2017/11/12/123) on Nov. 12. It won't be a quiet release, though. The Linux developers had previously announced that 4.14 would be Linux's next long-term support (LTS) version of the Linux kernel. That's important because [Linux LTS version now has a six-year life span](https://www.zdnet.com/article/long-term-support-linux-gets-a-longer-lease-on-life/).

awk 'FNR == 1'

## What is Swap Space?

Swap space in Linux is used when the amount of physical memory (RAM) is full. If the system needs more memory resources and the RAM is full, inactive pages in memory are moved to the swap space. While swap space can help machines with a small amount of RAM, it should not be considered a replacement for more RAM. Swap space is located on hard drives, which have a slower access time than physical memory.

Swap space can be a dedicated swap partition (recommended), a swap file, or a combination of swap partitions and swap files

## What is difference between GRUB and LILO ?

All boot loaders work in a similar way to fulfill a common purpose. But LILO and GRUB do have a number of differences:

\* LILO has no interactive command interface, whereas GRUB does.

\* LILO does not support booting from a network, whereas GRUB does.

\* LILO stores information regarding the location of the operating systems it can to load physically on the MBR. If you change your LILO config file, you have to rewrite the LILO stage one boot loader to the MBR. Compared with GRUB, this is a much more risky option since a misconfigured MBR could leave the system unbootable. With GRUB, if the configuration file is configured incorrectly, it will simply default to the GRUB command-line interface.

\* LILO only loads **linux** and other boot loaders. and GRUB loads a large number of OS

## Booting process

### BIOS

* BIOS stands for Basic Input/Output System
* Performs some system integrity checks. POST-power on self test
* Searches, loads, and executes the boot loader program.
* It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12 of F2, but it depends on your system) during the BIOS startup to change the boot sequence.
* Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
* So, in simple terms BIOS loads and executes the MBR boot loader.

### 2. MBR

* MBR stands for Master Boot Record.
* It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
* MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes. this information pre-loads into ROM (Read Only Memory) by BIOS.
* It contains information about GRUB (or LILO in old systems).
* So, in simple terms MBR loads and executes the GRUB boot loader.

### 3. GRUB

* GRUB stands for Grand Unified Bootloader.
* If you have multiple kernel images installed on your system, you can choose which one to be executed.
* GRUB displays a splash screen, waits for few seconds, if you don’t enter anything, it loads the default kernel image as specified in the grub configuration file.
* GRUB has the knowledge of the filesystem (the older Linux loader LILO didn’t understand filesystem).
* Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

#boot=/dev/sda

default=0

timeout=5

splashimage=(hd0,0)/boot/grub/splash.xpm.gz

hiddenmenu

title CentOS (2.6.18-194.el5PAE)

root (hd0,0)

kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/

initrd /boot/initrd-2.6.18-194.el5PAE.img

* As you notice from the above info, it contains kernel and initrd image.
* So, in simple terms GRUB just loads and executes Kernel and initrd images.

### 4. Kernel

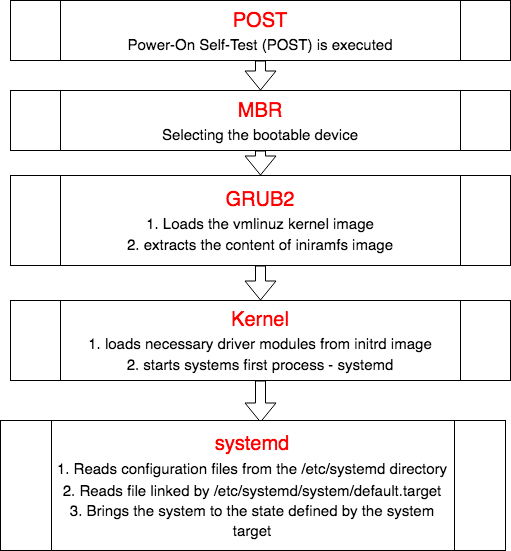
* Mounts the root file system as specified in the “root=” in grub.conf
* Kernel executes the /sbin/init program
* Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a ‘ps -ef | grep init’ and check the pid.
* initrd stands for Initial RAM Disk.
* initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

### 5. Init

* Looks at the /etc/inittab file to decide the Linux run level.
* Following are the available run levels
  + 0 – halt
  + 1 – Single user mode
  + 2 – Multiuser, without NFS
  + 3 – Full multiuser mode
  + 4 – unused
  + 5 – X11
  + 6 – reboot
* Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
* Execute ‘grep initdefault /etc/inittab’ on your system to identify the default run level
* If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
* Typically you would set the default run level to either 3 or 5.

### 6. Runlevel programs

### Update:

**nlevel 3** in SystemV. The **emergency.target** is similar to single user mode.

GRUB 2 menu-configuration settings are taken from /etc/default/grub when generating grub.cfg.

If changes are made to any of these parameters, you need to run grub2-mkconfig to re-generate the /boot/grub2/grub.cfg file.

# grub2-mkconfig –o /boot/grub2/grub.cfg

– GRUB2 searches the compressed kernel image file also called as vmlinuz in the /boot directory.

– GRUB2 loads the vmlinuz kernel image file into memory and extracts the contents of the initramfs image file into a temporary, memory-based file system (tmpfs).

– The initial RAM disk (initrd) is an initial root file system that is mounted before the real root file system.

**initramfs**  
– The job of the initial RAM file system is to preload the block device modules, such as for IDE, SCSI, or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted.  
– The initramfs is bound to the kernel and the kernel mounts this initramfs as part of a two-stage boot process.

systemctl get-default

ls -lrt /etc/systemd/system/default.target

systemctl set-default multi-user.target

Edits of grub.cfg will be lost any time **grub2-mkconfig** is used to regenerate the file, so care must be taken to reflect any manual changes in /etc/default/grub as well.

### systemd

systemd is the mother of all processes and it is responsible for bringing the Linux host up to a state in which productive work can be done. Some of its functions, which are far more extensive than the old init program, are to manage many aspects of a running Linux host, including mounting filesystems, and starting and managing system services required to have a productive Linux host. Any of systemd's tasks that are not related to the startup sequence are outside the scope of this article.

First, systemd mounts the filesystems as defined by **/etc/fstab**, including any swap files or partitions. At this point, it can access the configuration files located in /etc, including its own. It uses its configuration file, **/etc/systemd/system/default.target**, to determine which state or target, into which it should boot the host. The **default.target** file is only a symbolic link to the true target file. For a desktop workstation, this is typically going to be the graphical.target, which is equivalent to **runlevel 5** in the old SystemV init. For a server, the default is more likely to be the **multi-user.target** which is like **ru**

## Raid

Software RAID have low performance, because of consuming resource from hosts. Raid software need to load for read data from software raid volumes. Before loading raid software, OS need to get boot to load the raid software. No need of Physical hardware in software raids. Zero cost investment.

Hardware RAID have high performance. They are dedicated RAID Controller which is Physically built using PCI express cards. It won’t use the host resource. They have NVRAM for cache to read and write. Stores cache while rebuild even if there is power-failure, it will store the cache using battery power backups. Very costly investments needed for a large scale.

RAID0 = Striping

RAID1 = Mirroring

RAID5 = Single Disk Distributed Parity

RAID6 = Double Disk Distributed Parity

RAID10 = Combine of Mirror & Stripe.

Raid5:Assume we have 4 drives, if one drive fails and while we replace the failed drive we can rebuild the replaced drive from parity informations. Parity information’s are Stored in all 4 drives, if we have 4 numbers of 1TB hard-drive. The parity information will be stored in 256GB in each drivers and other 768GB in each drives will be defined for Users. RAID 5 can be survive from a single Drive failure, If drives fails more than 1 will cause loss of data’s.

Raid6:poor performance

Raid 10:If we have 6 numbers of 1TB hard-drives 4 drives will be used for data and 2 drives will be used for Parity. RAID 10 can be called as 1+0 or 0+1. This will do both works of Mirror & Striping. Mirror will be first and stripe will be the second in RAID 10. Stripe will be the first and mirror will be the second in RAID 01. RAID 10 is better comparing to 01.Assume, we have 4 Number of drives. While I’m writing some data to my logical volume it will be saved under All 4 drives using mirror and stripe methods.

### There are 3 types of modes in vi Editor:

* Regular mode or command mode
* Insertion mode or edit mode
* Replacement mode or Ex mode

### You wish to print a file ‘draft’ with 60 lines on a page. What command would you use?

**The command used : pr -l60 draft**

* The default page length when using pr is 66 lines.
* The -l option specifies a different length.

## What is LD\_LIBRARY\_PATH?

* LD\_LIBRARY\_PATH is an environment variable. It is used for debugging a new library or a non-standard library. It is also used to identify what directories need to be searched. Path to search for directories needs to given.

## **Finger** server

or (**finger** User Information Protocol)provides the user related information to the clients.**Finger** Information Protocol can be defined as a protocol which allows remote users to see information like login name,mail related information,login shell,login terminal for user,full name of users, login time etc

## xinetd

is a tcp wrapper super service. It can be use to control access to network services.

When a client host attempts to connect to a network service controlled by xinetd, the super service receives the request and checks for any TCP wrappers access control rules. If access is allowed, xinetd verifies that the connection is allowed under its own access rules for that service and that the service is not consuming more than its alloted amount of resources or in breach of any defined rules. It then starts an instance of the requested service and passes control of the connection to it. Once the connection is established, xinetd does not interfere further with communication between the client host and the server.

To determine if a client is allowed to connect to a service, TCP Wrappers reference the following two files, which are commonly referred to as hosts access files:

## What does Sar provide? Where are Sar logs stored?

Sar collects, reports, or saves system activity information, sar serves to log and evaluate a variety of information regarding system activity. With performance problems, sar also permits retroactive analysis of the load values for various sub-systems (CPUs, memory, disks, interrupts, network interfaces and so forth) and limitation of problems in this manner. If CPU utilization is near 100 % (user + nice + system), the workload sampled is CPU-bound.  
By default, log files of Sar command is located at /var/log/sa/sadd file, where dd parameter indicates the current day.

## How to check Memory stats and CPU stats as a Linux admin?

Using ‘free’ & ‘vmstat’ command, we can display the physical and virtual memory statistics, respectively. With the help of ‘sar’ command, we can see the CPU utilization & other stats.

## Hardening

Set BIOS password

Disable booting from alternative medias

Set user/group owner to root, and permissions to read and write for root only, on /boot/grub2/grub.cfg

Set bootloader password

Latest OS and patches.

Install the Red Hat GPG key and enable gpgcheck.

Create a separate partition with the nodev, nosuid, and noexec options set for /tmp

Create separate partitions for /var, /var/log, /var/log/audit, and /home

Set nodev option to /home

Set sticky bit on all world-writable directories.

Remove legacy services (e.g., telnet-server; rsh, rlogin, rcp; ypserv, ypbind; tftp, tftp-server; talk, talk-server).

Disable any services and applications started by xinetd or inetd that are not being utilized.Remove xinetd if possible

Disable or remove server services that are not going to be utilized (e.g., FTP, DNS, LDAP, SMB, DHCP, NFS, SNMP, etc.).

Set daemon umask.

Limit connections to services running on the host to authorized users of the service via firewalls and other access control technologies.

Set SSH loglevel to INFO.

Disable SSH root login.

S et SSH permitemptypasswords to no.

Configure selinux

Restrict root login to system console.

Install and enable anti-virus software. And update it regualerly

String password and age.

Acc lock after n failed attempts

Remove unwatned packages,users, groups

Regular log checking and auditing

Backup

## Rsyslog

**Rsyslog** daemon is the main log server preinstalled, followed by **Systemd Journal Daemon** (**journald**).

**Rsyslog** server in build as a client/server architecture service and can achieve both roles simultaneous. It can run as a server and collect all logs transmitted by other devices in the network or it can run as a client by sending all internal system events logged to a remote endpoint syslog server.

When rsyslog is configured as a client, the logs can be stored locally in files on the local filesystem or can be send remotely rather than write them in files stored on the machine or write events log files locally and send them to a remote syslog server at the same time.

Rsyslog server via **514** port. UDP is the standard protocol used for log transmission by Rsyslog.

## Selinux

SELinux users and roles do not have to be related to the actual system users and roles. For every current user or process, SELinux assigns a three string context consisting of a username, role, and domain (or type). This system is more flexible than normally required: as a rule, most of the real users share the same SELinux username, and all access control is managed through the third tag, the domain. T SELinux provides a flexible *Mandatory Access Control* (MAC) system built into the Linux kernel. Under standard Linux *Discretionary Access Control* (DAC), an application or process running as a user (UID or SUID) has the user's permissions to objects such as files, sockets, and other processes. Running a MAC kernel protects the system from malicious or flawed applications that can damage or destroy the system.

SELinux defines the access and transition rights of every user, application, process, and file on the system.

## A **sticky bit**

is a permission **bit** that is set on a directory that allows only the owner of the file within that directory or the root user to delete or rename the file. No other user has the needed privileges to delete the file created by some other user.

## SSL handshake

The following is a standard SSL handshake when RSA key exchange algorithm is used:

**1.  Client Hello**

Information that the server needs to communicate with the client using SSL. This includes the SSL version number, cipher settings, session-specific data.

**2.  Server Hello**

Information that the server needs to communicate with the client using SSL. This includes the SSL version number, cipher settings, session-specific data.

**3.  Authentication and Pre-Master Secret**

Client authenticates the server certificate. (e.g. Common Name / Date / Issuer) Client (depending on the cipher) creates the pre-master secret for the session, Encrypts with the server's public key and sends the encrypted pre-master secret to the server.

**4.  Decryption and Master Secret**

Server uses its private key to decrypt the pre-master secret. Both Server and Client perform steps to generate the master secret with the agreed cipher.

**5.  Encryption with Session Key**

Both client and server exchange messages to inform that future messages will be encrypted.

## Network Bonding

What are the different modes of Network bonding in Linux?

Below are list of modes used in Network Bonding :

* **balance-rr or 0 –** round-robin mode for fault tolerance and load balancing.
* **active-backup or 1 –** Sets active-backup mode for fault tolerance.
* **balance-xor or 2 –** Sets an XOR (exclusive-or) mode for fault tolerance and load balancing.
* **broadcast or 3 –** Sets a broadcast mode for fault tolerance. All transmissions are sent on all slave interfaces.
* **802.3ad or 4 –** Sets an IEEE 802.3ad dynamic link aggregation mode. Creates aggregation groups that share the same speed & duplex settings.
* **balance-tlb or 5 –** Sets a Transmit Load Balancing (TLB) mode for fault tolerance & load balancing.
* **balance-alb or 6 –** Sets an Active Load Balancing (ALB) mode for fault tolerance & load balancing.

**cat /proc/net/bonding/bond0,**

## **Ports**

|  |  |  |  |
| --- | --- | --- | --- |
| 20 | TCP | ftp-data | FTP data port |
| 21 | TCP | ftp | File Transfer Protocol (FTP) port; sometimes used by File Service Protocol (FSP) |
| 22 | TCP | Ssh | Secure Shell (SSH) service |
| 23 | TCP | telnet | The Telnet service |
| 25 | TCP | smtp | Simple Mail Transfer Protocol (SMTP) |

53 DNS  
68 DHCP

**110 – POP3**

**143 imap**

**389 – LDAP**

**636 ldaps**

**3306 – MySql**

## Kerberos

To start the Kerberos authentication process, the initiating client sends a request to an authentication server for access to a service. The initial request is sent as plaintext because no sensitive information is included in the request.

The authentication server retrieves the initiating client's private key, assuming the initiating client's username is in the KDC database. If the initiating client's username cannot be found in the KDC database, the client cannot be authenticated and the authentication process stops. If the client's username can be found in the KDC database, the authentication server generates a session key and a ticket granting ticket. The ticket granting ticket is timestamped and encrypted by the authentication server with the initiating client's password.

The initiating client is then prompted for a password; if what is entered matches the password in the KDC database, the encrypted ticket granting ticket sent from the authentication server is decrypted and used to request a credential from the ticket granting server for the desired service. The client sends the ticket granting ticket to the ticket granting server, which may be physically running on the same hardware as the authentication server, but performing a different role.

The ticket granting service carries out an authentication check similar to that performed by the authentication server, but this time sends credentials and a ticket to access the requested service. This transmission is encrypted with a session key specific to the user and service being accessed. This proof of identity can be used to access the requested "kerberized" service, which, once having validated the original request, will confirm its identity to the requesting system.

The timestamped ticket sent by the ticket granting service allows the requesting system to access the service using a single ticket for a specific time period without having to be re-authenticated. Making the ticket valid for a limited time period makes it less likely that someone else will be able to use it later; it is also possible to set the maximum lifetime to 0, in which case service tickets will not expire. Microsoft recommends a maximum lifetime of 600 minutes for service tickets; this is the default value in Windows Server implementations of Kerberos.

## Why should you avoid Telnet to administer a Linux system remotely?

Telnet uses most insecure method for communication. It sends data across the network in plain text format and anybody can easily find out the password using the network tool.

## What is the difference between Cron and Anacron?

1. Minimum granularity with Cron is minute while it is in days with Anacron.
2. Cron job can be scheduled by any normal user while Anacron can be scheduled only by the super user (the superuser is a special user account used for system administration. Depending on the operating system (OS), the actual name of this account might be root,administrator, admin or supervisor)
3. Cron expects the system to be up and running while the Anacron doesn’t expect the system to be up and running all the time. In case of Anacron, if a job is scheduled and the system is down that time, it will execute the job as soon as the system is up and running.
4. Cron is ideal for servers while Anacron is ideal for desktops and laptops.
5. Cron should be used when you want a job to be executed at a particular hour and minute while Anacron should be used in when the job can be executed irrespective of the hour and minute.

## If a volume group named VG0 already exists and we need to extend this volume group up to 4GB. How will you do it?

* Firstly create Physical volume (/dev/sda7) of size 4GB.
* Now run following command.  
  **vgextend VG0 /dev/sda7**

## Explain /proc filesystem?

* /proc is a virtual file system that provides detailed information about Linux kernel, hardware and running processes. Files under /proc directory named as Virtual files.  
  Since /proc contains virtual files, it is called **virtual file system**. These virtual files have unique qualities. Most of them are listed as zero bytes in size.  
  Virtual files such as /proc/interrupts, /proc/meminfo, /proc/mounts, and /proc/partitions provide an up-to-the-moment glimpse of the system’s hardware.

## Ext4

 Directory can contain a maximum of 64,000 subdirectories (as opposed to 32,000 in ext3)

*  You can also mount an existing ext3 fs as ext4 fs (without having to upgrade it).
*  Several other new features are introduced in ext4: multiblock allocation, delayed allocation, journal checksum. fast fsck, etc. All you need to know is that these new features have improved the performance and reliability of the filesystem when compared to ext3.
*  In ext4, you also have the option of turning the journaling feature “off”.

## Rescu mode

On CentOS 7 systems, you’re dropped into something called [rescue mode](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/sec-Terminal_Menu_Editing_During_Boot.html), which is a systemd unit that does almost everything you want, except:

In rescue mode, the system attempts to mount all local file systems and start some important system services, but it does not activate network interfaces or allow more users to be logged into the system at the same time.

his makes it hard to check them, especially if you have errors on /varou can force the system to fsck and repair by appending fsck.repair=yes to the kernel command line in grub

## Journaling Filesystem Definition

A journaling filesystem is a filesystem that maintains a special file called a journal that is used to repair any inconsistencies that occur as the result of an improper shutdown of a computer. Such shutdowns are usually due to an interruption of the power supply or to a software problem that cannot be resolved without a rebooting.

A filesystem is a way of storing information on a computer that usually consists of a hierarchy of directories (also referred to as the directory tree) that is used to organize files. Each hard disk drive (HDD) or other storage device as well as each partition (i.e., logically independent section of a HDD) can have a different type of filesystem if desired.

Journaling filesystems write metadata (i.e., data about files and directories) into the journal that is flushed to the HDD before each command returns. In the event of a system crash, a given set of updates may have either been fully committed to the filesystem (i.e., written to the HDD), in which case there is no problem, or the updates will have been marked as not yet fully committed, in which case the system will read the journal, which can be rolled up to the most recent point of data consistency.

## Which file would you examine to determine the levels of messages written to system log files?

- kernel.h

Minimum 2 partitions are needed for installing Linux. The one is "/ or root" which contains all the files and the other is swap.

## Which command is used to review boot messages?

dmesg command is used to review boot messages

## Which utility is used to make automate rotation of a log?

logrotate command is used to make automate rotation of log.  
  
**Syntax of the command is:**

logrotate [-dv] [-f|] [-s|] config\_file+

## Which commands are used to set a processor-intensive job to use less CPU time?

nice command is used for changing priority of the jobs.

## **How to increase the size of LVM partition ?**

Ans: Below are the Logical Steps :  
– Use the lvextend command (lvextend -L +100M /dev/<Name of the LVM Partition> , in this example we are extending the size by 100MB.  
– resize2fs /dev/<Name of the LVM Partition>  
– check the size of partition using ‘df -h’ command

## **Q:5 How to reduce or shrink the size of LVM partition ?**

Ans: Below are the logical Steps to reduce size of LVM partition :  
-Umount the filesystem using umount command,  
-use resize2fs command , e.g resiz2fs /dev/mapper/myvg-mylv 10G  
-Now use the lvreduce command , e.g lvreduce -L 10G /dev/mapper/myvg-mylv

Above Command will shrink the size & will make the filesystem size 10GB

**Where the kernel modules are located ?**

Ans: The ‘/lib/modules/kernel-version/’ directory stores all kernel modules or compiled drivers in Linux operating system. Also with ‘lsmod’ command we can see all the installed kernel modules.

**How to upgrade Kernel in Linux ?**

Ans: We should never upgrade Linux Kernel , always install the new New kernel using rpm command because upgrading a kenel can make your linux box in a unbootable state.

**How to add & change the Kernel parameters ?**

Ans: To Set the kernel parameters in linux , first edit the file ‘/etc/sysctl.conf’ after making the changes save the file and run the command ‘sysctl -p’ , this command will make the changes permanently without rebooting the machine.

**What is load average in Linux ?**

Ans: Load  Average is defined as the average sum of the number of process waiting in the run queue and number of process currently executing over the period of 1,5 and 15  minutes. Using the ‘top’ and ‘uptime’ command we find the load average of a Linux sever.

### What is fork bomb attack?

The fork bomb attack is a form of denial-of-service attack which uses the fork functionality to recursively create running processes by a single process. This attack works by creating a large number of processes very quickly in order to saturate the available space of the OS’s process table. Once the process table becomes saturated, no new programs may start until another process terminates.

### Check Top Processes sorted by RAM or CPU Usage in Linux

The following command will show the list of top processes ordered by RAM and CPU use in descendant form (remove the **pipeline** and **head** if you want to see the full list):

# ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head

## dns

### DNS Resource Records

Zone DNS database is a collection of resource records and each of the records provides information about a specific object. A list of most common records is provided below:

* [Address Mapping records](http://tools.ietf.org/html/rfc1035#page-12) (A)

The record A specifies IP address (IPv4) for given host. A records are used for conversion of domain names to corresponding IP addresses.

* [IP Version 6 Address records](http://tools.ietf.org/html/rfc3596#page-3) (AAAA)

The record AAAA (also quad-A record) specifies IPv6 address for given host. So it works the same way as the A record and the difference is the type of IP address.

* [Canonical Name records](http://tools.ietf.org/html/rfc1035#page-14) (CNAME)

The CNAME record specifies a domain name that has to be queried in order to resolve the original DNS query. Therefore CNAME records are used for creating aliases of domain names. CNAME records are truly useful when we want to alias our domain to an external domain. In other cases we can remove CNAME records and replace them with A records and even decrease performance overhead.

* [Host Information records](http://tools.ietf.org/html/rfc1035#page-14) (HINFO)

HINFO records are used to acquire general information about a host. The record specifies type of CPU and OS. The HINFO record data provides the possibility to use operating system specific protocols when two hosts want to communicate. For security reasons the HINFO records are not typically used on public servers.

**Note:** Standard values in [RFC 1010](http://tools.ietf.org/html/rfc1010)

* [Integrated Services Digital Network records](http://tools.ietf.org/html/rfc1183#section-3.2) (ISDN)

The ISDN resource record specifies ISDN address for a host. An ISDN address is a telephone number that consists of a country code, a national destination code, a ISDN Subscriber number and, optionally, a ISDN subaddress. The function of the record is only variation of the A resource record function.

* [Mail exchanger record](http://tools.ietf.org/html/rfc1035#section-3.3.9) (MX)

The MX resource record specifies a mail exchange server for a DNS domain name. The information is used by Simple Mail Transfer Protocol (SMTP) to route emails to proper hosts. Typically, there are more than one mail exchange server for a DNS domain and each of them have set priority.

 [Name Server records](http://tools.ietf.org/html/rfc1035#section-3.3.11) (NS)

The NS record specifies an authoritative name server for given host.

 [Reverse-lookup Pointer records](http://tools.ietf.org/html/rfc1035#section-3.3.12) (PTR)

As opposed to forward DNS resolution (A and AAAA DNS records), the PTR record is used to look up domain names based on an IP address.

 [Start of Authority records](http://tools.ietf.org/html/rfc1035#section-3.3.13) (SOA)

The record specifies core information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone.

## Main directories

The standard Ubuntu directory structure mostly follows the [Filesystem Hierarchy Standard](http://www.pathname.com/fhs), which can be referred to for more detailed information.

Here, only the most important directories in the system will be presented.

**/bin** is a place for most commonly used [terminal](https://help.ubuntu.com/community/UsingTheTerminal) commands, like ls, mount, rm, etc.

**/boot** contains files needed to start up the system, including the [Linux kernel](https://help.ubuntu.com/community/Kernel), a RAM disk image and [bootloader](https://help.ubuntu.com/community/GrubHowTo) configuration files.

**/dev** contains all device files, which are not regular files but instead refer to various hardware devices on the system, including hard drives.

**/etc** contains system-global configuration files, which affect the system's behavior for all users.

**/home** home sweet home, this is the place for users' home directories.

**/lib** contains very important dynamic libraries and kernel modules

**/media** is intended as a mount point for external devices, such as hard drives or removable media (floppies, CDs, DVDs).

**/mnt** is also a place for mount points, but dedicated specifically to "temporarily mounted" devices, such as network filesystems.

**/opt** can be used to store additional software for your system, which is not handled by the [package manager](https://help.ubuntu.com/community/InstallingSoftware).

**/proc** is a virtual filesystem that provides a mechanism for kernel to send information to processes.

**/root** is the [superuser](https://help.ubuntu.com/community/RootSudo)'s home directory, not in /home/ to allow for booting the system even if /home/ is not available.

**/run** is a tmpfs (temporary file system) available early in the boot process where ephemeral run-time data is stored. Files under this directory are removed or truncated at the beginning of the boot process.   
(It deprecates various legacy locations such as /var/run, /var/lock, /lib/init/rw in otherwise non-ephemeral directory trees as well as /dev/.\* and /dev/shm  which are not device files.)

**/sbin** contains important administrative commands that should generally only be employed by the [superuser](https://help.ubuntu.com/community/RootSudo).

**/srv** can contain data directories of services such as HTTP (/srv/www/) or FTP.

**/sys** is a virtual filesystem that can be accessed to set or obtain information about the kernel's view of the system.

**/tmp** is a place for temporary files used by applications.

**/usr** contains the majority of user utilities and applications, and partly replicates the root directory structure, containing for instance, among others, /usr/bin/ and /usr/lib.

**/var** is dedicated to variable data, such as logs, databases, websites, and temporary spool (e-mail etc.) files that persist from one boot to the next. A notable directory it contains is /var/log where system log files are kept.

## Describe a scenario when you get a "filesystem is full" error, but '**df**' shows there is free space

Answer: The filesystem can run out of inodes, '**df -i**' will show that.

## ncsd

The nscd is a daemon that provides a cache for the most common name service requests like passwd, group, hosts, service and netgroup. sssd primarily provides daemons to manage access to remote directories and authentication mechanisms like LDAP, Kerberos, NIS, etc.

## Strace

'strace' is a tool which enables **tracing** the **System calls** made by a process during its execution. It also reports the **signals** (or software interrupts) that are recieved by the process.

## Check shared library dependencies of a program executable

To find out what libraries a particular executable depends on, you can use ldd command. This command invokes dynamic linker to find out library dependencies of an executable.

$ ldd /path/to/program

Instead, a safer way to show library dependencies of an unknown application binary is to use the following command.

$ objdump -p /path/to/program | grep NEEDED

**Q. What is Ansible?**  
Ansible is developed in Python language.  
It is a software tool. It is useful while deploying any application using ssh without any downtime. Using this tool one can manage and configure software applications very easily.

**What are the Advantages of Ansible?**

1. Agentless
2. Very low overhead
3. Good performance

**How Ansible Works?**

* There are many similar automation tools available like [Puppet](https://tekslate.com/puppet-training), Capistrano, [Chef](https://tekslate.com/chef-devops-training/), [Salt](https://tekslate.com/saltstack-training/), Space Walk, etc, but Ansible categorizes into two types of server: controlling machines and nodes.
* The controlling machine, where Ansible is installed, and Nodes are managed by this controlling machine over SSH(Secure Shell). By controlling the machine through inventry, location of nodes is specified.
* Using SSH protocol, the controlling machine (Ansible) deploys modules to nodes and these modules are stored temporarily on remote nodes and through a [JSON](https://tekslate.com/json-data-in-jquery/) connection, they will communicate with the Ansible machine over the standard output.
* There is no need for any agent installation on remote nodes because Ansible is agent-less, so it means there are no any background daemons or programs are executing for Ansible when it’s not managing any nodes.
* Ansible can handle 100’s of nodes from a single system over SSH connection and the entire operation can be handled and executed by one single command ‘ansible’. But, in some cases, where you required to execute multiple commands for a deployment, here we can build playbooks.
* Playbooks are a bunch of commands which can perform multiple tasks, and each playbook are in YAML file format.
* **Q: How do you access Shell Environment Variables?**  
  If you are just looking to access the existing variables then you can use “env” lookup plugin.  
  For example:  
  Accessing the value of Home environment variable on management machine:
* local\_home:”{{lookup(‘env’,’HOME’)}}”
* Ansible can do the following for us:  
  1. Configuration management  
  2. Application deployment  
  3. Task automation  
  4. IT orchestration
* **Q: What does Fact mean in Ansible?**  
  The term “Facts” is commonly used in Ansible environment. They are described in the playbooks areas where it displays known and discovered variables about the system.  Facts are used to implement conditionals executions and also used for getting ad-hoc information of the information.
* You can see all the facts via:
* $ ansible all- m setup
* So if you want to extract only certain part of the information then you can use “setup” module where you will have an option to filter out the output and just get hold of the fact that you are in need of

## ITIL

[Service Strategy](https://wiki.en.it-processmaps.com/index.php/ITIL_Service_Strategy)

Process Objective: To decide on a strategy to serve customers. Starting from an assessment of customer needs and the market place, the Service Strategy process determines which services the IT organization is to offer and what capabilities need to be developed. Its ultimate goal is to make the IT organization think and act in a strategic manner.

[Service Design](https://wiki.en.it-processmaps.com/index.php/ITIL_Service_Design)

Process Objective: To design new IT services. The scope of the process includes the design of new services, as well as changes and improvements to existing ones.

[Service Transition](https://wiki.en.it-processmaps.com/index.php/ITIL_Service_Transition)

Process Objective: To build and deploy IT services. Service Transition also makes sure that changes to services and Service Management processes are carried out in a coordinated way.

[Service Operation](https://wiki.en.it-processmaps.com/index.php/ITIL_Service_Operation)

Process Objective: To make sure that IT services are delivered effectively and efficiently. The Service Operation process includes fulfilling user requests, resolving service failures, fixing problems, as well as carrying out routine operational tasks.

[Continual Service Improvement - CSI](https://wiki.en.it-processmaps.com/index.php/ITIL_CSI_-_Continual_Service_Improvement)

Process Objective: To use methods from quality management in order to learn from past successes and failures. The Continual Service Improvement process aims to continually improve the effectiveness and efficiency of IT processes and services, in line with the concept of continual improvement adopted in ISO 20000.

## ESX

# [Virtual Machine memory overhead](http://frankdenneman.nl/2010/05/31/virtual-machine-memory-overhead/)

[May 31, 2010](http://frankdenneman.nl/2010/05/31/virtual-machine-memory-overhead/) / [Frank Denneman](http://frankdenneman.nl/author/admin/) / [7 Comments](http://frankdenneman.nl/2010/05/31/virtual-machine-memory-overhead/#comments)

Every virtual machine running on an ESX host consumes some memory overhead additional to the current usage of its configured memory. This extra space is needed by ESX for the internal VMkernel datastructures like virtual machine frame buffer and mapping table for memory translation (mapping physical virtual machine memory to machine memory). Two kinds of virtual machine overhead exists:

**Static overhead**  
Static overhead is the minimum overhead that is required for the virtual machine startup. DRS and the VMkernel uses this metric for admission control and VMotion calculations. The destination host must be able to back the virtual machine reservation and the static overhead otherwise the VMotion will fail.

**Dynamic overhead**  
Once the virtual machine has started up, the virtual machine monitor (VMM) can request additional memory space. The VMM will request the space, but the VMkernel is not required to supply it. If the VMM does not obtain the extra memory space, the virtual machine will continue to function but this can lead to performance degradation. The VMkernel treats virtual machine overhead reservation the same as [VM-level memory reservation](http://frankdenneman.nl/2009/12/impact-of-memory-reservation/) and it will not reclaim this memory once it used.

**Overhead memory used in admission control**  
As mentioned before, DRS and the VMkernel will not allow the virtual machine to be powered up if reservations cannot be guaranteed, this means that the effective memory reservation for a virtual machine is the user configured memory reservation (VM-level reservation) plus the overhead reservation.

**Resource pool memory reservations**  
This means that during the design phase of a resource pool, the memory overhead of a virtual machine must be included in the calculation of the memory reservation specified on the resource pool. The behavior of dynamic overhead must also be taken into account.  
Table 3.2 of the vSphere resource management guide list the overhead memory on virtual machines. [VMware vSphere Online Library – Table 3.2 overhead memory](http://pubs.vmware.com/vsp40_i/wwhelp/wwhimpl/common/html/wwhelp.htm#href=resmgmt/r_overhead_memory_on_virtual_machines.html#1_7_9_9_10_1&single=true)

Please be aware of the fact that memory overheads are growing with each new release of ESX, so keep this in mind when upgrading to a new version. Verify the documentation of the virtual machine memory overhead and check the specified memory reservation on the resource pool.

Top of Form

Bottom of Form