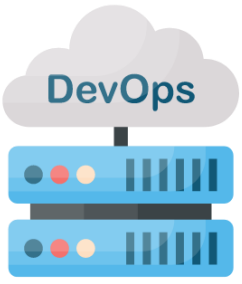
iNeuron

DevOps

Interview Questions

PART -4: Linux Hardware and Architecture



Question 27: The /proc file system

What exactly is /proc file system in LINUX? How does it work?

A: Under Linux, *proc provides information on any running process at* proc/PID. It also includes the ff:

A symbolic link to the current (traversing) process at *proc* self-
Information on hardware, kernel, and module configuration Access to dynamically-configurable kernel options under *proc* sys
Information about the system, such as *proc* meminfo, which provides memory statistics.

The basic utilities that use *proc can be in the procps package. It is required that* proc is mounted to function.

The procfs plays an important role in moving functionality from kernel mode to user mode. For

example, the GNU version of ps operates entirely in user mode, using the procfs to obtain its data.

The *proc filesystem contains an illusionary filesystem. It does not exist on a disk. Instead, the kernel creates it in memory. It is used to provide information about the system (originally about processes, hence the name). Some of the more important files and directories are explained below. The* proc filesystem is described in more detail in the proc manual page.

*proc*1
A directory with information about process number 1. Each process has a directory below /proc with the name being its process identification number.

Proc cpuinfo
Information about the processor, such as its type, make, model, and performance.
Proc devices
List of device drivers configured into the currently running kernel.

*proc*dma
Shows which DMA channels are being used now.

*proc*filesystems
Filesystems configured into the kernel.

*proc*interrupts
Shows which interrupts are in use, and how many of each there has been.

*proc*kcore
This represents an image of the physical memory of the system. This is exactly the same size as your physical memory but does not really take up that much memory; it is generated on the fly as programs access it. (Remember: unless you copy it elsewhere, nothing under /proc takes up any disk space at all.)

*proc*kmsg
Messages output by the kernel. These are also routed to syslog.

*proc*ksyms
Symbol table for the kernel.

*proc*loadavg
The `load average' of the system; three meaningless indicators of how much work the system has to do at the moment.

*proc*meminfo
Information about memory usage, both physical and swap.

*proc*modules
Which kernel modules are loaded now.

*proc*net
Status information about network protocols.

*proc*self
A symbolic link to the process directory of the program that is looking at /proc. When two processes look at /proc, they get different links. This is mainly a convenience to make it easier for programs to get at their process directory.

*proc*stat
Various statistics about the system, such as the number of page faults since the system was booted.

*proc*uptime
The time the system has been up.

*proc*version
The kernel version.

Please take note that while the above files tend to be easily readable text files, they can sometimes be formatted in a way that is not easily digestible. There are many commands that do little more than read the above files and format them for easier understanding.

Question 28: OpenGL - hardware acceleration

I have installed all drivers and I have a problem with Open GL, Quake 3 fps is too low but in Windows 2K it runs fine.

What is the proper procedure to accelerate my hardware?

A: Try to check if 3D hardware acceleration is working for openGL by using the following command:

glxinfo |grep rendering

This should output "direct rendering: Yes".

If it gives the same but then with "No", hardware 3d acceleration is disabled and all rendering will be done on the CPU.

If it's a 'No', check the following:

Rerun glxinfo with debugging output: In a bash shell this can be done by running:

LIBGL_DEBUG=verbose glxinfo

This should give some extra info at the top of the output. The common error shown is “permission denied” messages on the drm device. This is often caused by not having loaded the drm kernel module for your video card or there **could** be some error message in the kernel log when you tried loading it. Another cause can be the permissions set on the drm device. This can be configured in the 'DRI' section of the XF86Config.

Another common error is having multiple libGL library files installed on the system (often caused by installing different drivers and XFree86 versions on top of a current XFree86 install). Check to which libGL glxinfo is linked with:

ldd `which glxinfo`

This will output a list of libraries then

look for a line similar to this one:

libGL.so.1 => *usr*X11R6/lib/libGL.so.1

(0x400a2000)

Check the file that it's pointing to (this of often a symlink itself, keep following it until you reach the real file). Make sure this libGL library matches your XFree86 version or your card driver. Also check your system of more (non-symlink) libGL files. It often happens that there is one in *usr*lib and another in *usr*X11R6/lib. There should only be one real file and the rest should all be symlinks pointing the correct one.

Question 29: Scan ide devices

I installed a brand new hard drive and issuing the fdisk -l command shows the drive as *dev*hdb.

Is there a way to list the file system?

A: You have to create a partition table and then create a file system in the partition/s.

Enter the cfdisk *dev*hdb command either in your cfdisk (if you have one) or enter the fdisk *dev*hdb command.

Now create a partition and make it type 83. Write the partition table back onto the disk. Exit the cfdisk or fdisk utility.

Next you have to create a file system. This is the same as when you format a partition in Windows. In Linux we use the mkfs command. Do this to create an ext2 file system on *dev*hdb1 if you want to use ext2 file system.

root> mkfs -t ext2 -cv *dev*hdb1

This command will take some time to end because it also checks the partition for bad blocks as it creates the file system.

Once it's done, then you can mount the file system. Assume you want to mount this file system at *mnt*hdb1. Then you have to create the directory where you want to mount it after which you mount the file system. Note that you only have to create the directory/mount point once.

root> mkdir *mnt*hdb1 root> mount *dev*hdb1 *mnt*hdb1

Now you can enter the following line in your *etc*fstab file to mount the file system automatically when you restart Linux.

If you don’t want people from running programs that are on this partition, do this using the noexec option. This would go next to the word defaults in the fstab entry as in:

*dev*hdb1 *mnt*hdb1 ext2 defaults 0 0

*dev*hdb1 *mnt*hdb1 ext2 defaults,noexec 0 0

You can look up all of the mount options by looking at the man page for the mount command and the man page for the fstab file.

Question 30: Partition Table

What does a partition table look like?

A: One may have an arbitrary number of partitions on a disk. However, the Master Boot Record (MBR, sector 0 of the disk) only holds descriptors for 4 partitions, called the primary partitions. Usually the BIOS can boot only from a primary partition. (Of course it can boot a boot loader that itself is able to access non-primary partitions or other disks.) The descriptors for the remaining partitions, called *logical* partitions, are scattered along the disk in a linked list of partition table sectors, starting with the MBR.

Each partition table sector contains 4 partition descriptors. A partition descriptor may be of type 05 (DOS extended partition), 0f (W95 extended partition), 85 (Linux extended partition), or c5 (DRDOS/secured extended partition), in which case it points to another partition table sector. In this way, we obtain a quaternary tree of

partitions. Linux accepts 85 as a synonym for 05 - this is useful if one wants to have extended partitions past the 1024-cylinder limit (to prevent DOS fdisk from crashing or hanging). Windows 95 uses 0f for LBA mapped extended partitions. Thus, an extended partition is not a partition containing data, but is a box containing other partitions. Nevertheless, the partition table sector that starts an extended partition has enough room left to contain a boot loader like LILO, so that it is possible to boot an extended partition.

Most operating systems severely restrict the accepted trees. Usually branching is not allowed, and one gets a linear chain of partition table sectors. Linux will accept several extended primary partitions.

Question 31: Advantage of partition

What is the advantage of partitioning my disk?

A: The partition table of a disk cuts it into 'logical disks'. There are several reasons for wanting to do this. DOS does not support filesystems larger than 2 GB, so partitioning is required to break this '2 GB barrier'. Different partitions may carry different operating systems or different filesystems (FAT, HPFS, NTFS, ext2, ...) to be used by one operating system. Sometimes small partitions are used for special purposes (OS/2 Boot Manager uses a small partition for itself; various laptops have a 'hibernation' partition where the state of the system is stored when it goes asleep). Some 'reliable' systems have backup partitions. For backup purposes, say to tape, it is often convenient to have partitions of a size such that the entire partition can be written to a single tape.

It is a good idea to keep your own things (say

under *home) and privately installed packages (say under* usr/local) separate from the software installed from a distribution. In case these are on a different partition, it is easier to do a complete reinstall (or switch to a different distribution) without losing your own stuff.

For well-designed systems it is often possible to have all basic system software on a read-only partition, thus diminishing the probability of corruption and saving backup time. There is also a security aspect; for example on a Unix system one might mount all file systems other than the root file system 'nosuid,nodev', and have *tmp*, home, /var not on the root file system, to minimize the possibility that some suid program is tricked into overwriting a vital system file via a hard link to it.

Finally there is the old BIOS problem that can make it impossible to boot a system that lives past cylinder 1024. This may mean that one has to have a partition that ends before the 1024 cylinder limit where the stuff needed at boot time

is stored.

Question 32: FDISK/MBR

What do FDISK /MBR do?

A: People often recommend the undocumented DOS command FDISK /MBR to solve problems with the MBR. This command however does not rewrite the entire MBR - it just rewrites the boot code, the first 446 bytes of the MBR, but leaves the 64-byte partition information alone. Thus, it won't help when the partition table has problems. Moreover, it can be dangerous to restore the boot code to its original state: if the cause of the problems was a boot sector virus, then vital information may have been stored elsewhere by the virus and killing the virus may mean killing access to this information. (For example, the stoned.empire.monkey virus encrypts the original MBR to sector 0/0/3.) However, people who want to uninstall LILO, and do not know that LILO has a
-u option, can use FDISK /MBR for this purpose.

In a Linux environment, one can wipe all of the

MBR with a command like "dd if=/dev/zero of=/dev/hda count=1 bs=512". If only the boot code must be removed, but not the partition table, then "dd if=/dev/zero of=/dev/hda count=1 bs=446" will do. Be very careful with such commands. Usually one regrets them later.

Question 33: Hard drive partitioning

What is hard drive partitioning?

A: Hard disk drive partitioning is the creation of logical divisions upon a hard disk that allows one to apply operating system-specific logical formatting. In simple terms, partitioning a hard drive makes it appear to be more than one hard drive, especially in how each partition is formatted for different operating systems and in how files are copied from one partition to another.

Partitioning allows the creation of several file systems on a single hard disk. This has many benefits, including:

allowing for dual boot setups (for example, to boot Microsoft Windows and Linux), which means the user can have more than one operating system on his/her computer, although only one can be used at a time; So, you will need at least one partition for Windows and one

(actually two) for Linux. In Linux terminology, your entire first hard drive is called *dev*hda and partitions within it are called *dev*hda1, *dev*hda2 and so on. Your original Windows single partition is *dev*hda . The notation for your second hard drive, if you have one, will be *dev*hdb, with partitions within it named *dev*hdb1, *dev*hdb2, *etc*. Your first SCSI disk, if any, is *dev*sda, *etc*.

sharing swap partitions between multiple Linux distributions, which means less hard drive space is wasted on Linux swap partitions, and

Protection or isolation of files, which means if the operating system stops working, it can just be reinstalled without, hopefully, deleting the user's personal files and settings.

Partitions may be customized to different requirements. One of these is to allow for read-only partitions to protect data. Should one partition be damaged, none of the other file

systems are affected and the drive's data may still be salvageable.

There are two types of partition. One of them is primary partition. A primary partition contains one file system. In MS-DOS and earlier versions of Microsoft Windows systems, the first partition (C:) must be a primary partition. Other operating systems may not share this limitation; however, this can depend on other factors such as a PC's BIOS.

Another type is the extended partition, which is able to contain several file systems, known as logical disks (terminology may vary slightly with operating systems).

Question 34: Partition with fdisk

How can I partition my hard drive with the fdisk utility?

A: The following answer shows you how to actually partition your hard drive with the fdisk utility. Linux allows only 4 primary partitions. You can have a much larger number of logical partitions by sub-dividing one of the primary partitions. Only one of the primary partitions can be sub-divided.

The “fdisk” is started by typing (as root) fdisk device at the command prompt. The device might be something like *dev*hda or *dev*sda. The basic fdisk commands you need are:

p-print the

partition table

n-create a new

partition d-

delete a

partition

q-quit without saving changes

w-write the new partition table and exit

Changes you make to the partition table do not take effect until you issue the write (w) command. Here is a sample partition table:

Disk *dev*hdb: 64 heads, 63
sectors, 621 cylinders Units =
cylinders of 4032 * 512 bytes

Device Boot Start End Blocks Id System
*dev*hdb1*1184 370912+ 83 Linux
*dev*hdb2 185 368 370944 83 Linux
*dev*hdb3 369 552 370944 83 Linux
*dev*hdb4 553 621139104 82 Linux swap

The first line shows the geometry of your hard drive. It may not be physically accurate, but you can accept it as though it were. The hard drive in this example is made of 32 double-sided platters with one head on each side (probably not true). Each platter has 621 concentric tracks. A 3-dimensional track (the same track on all disks) is called a cylinder. Each track is divided into 63 sectors. Each sector contains 512 bytes of data. Therefore the block size in the partition table is 64 heads *63 sectors* 512 bytes...divided by 1024. The start and end values are cylinders.

Decide on the size of your swap space and where it ought to go. Divide up the remaining space for the three other partitions.

Example:
To start fdisk from the shell prompt:

fdisk *dev*hdb

That indicates that you are using the second drive on IDE controller. When you print the (empty)

partition table, you'll get configuration information.

Command (m for help): p

Disk *dev*hdb: 64 heads, 63
sectors, 621 cylinders Units =
cylinders of 4032 * 512 bytes

You had a 1.2 GB drive, but now you really know: 64 * 63 *512* 621 = 1281982464 bytes. Reserve 128Mb of that space for swap, leaving 1153982464. If you use one of the primary partitions for swap, which means you have three left for ext2 partitions. Divided equally, that makes for 384Mb per partition.

Command (m for help): n

Command action

eextended
pprimary partition (1-4)

Partition number (1-4): 1
First cylinder (1-621,
default 1):<RETURN>
Using default value 1
Last cylinder or +size or +sizeM or

+sizeK (1-621, default 621): +384M Next,

set up the partition you want to use for

swap:

Command (m for help): n

Command action

eextended
pprimary partition (1-4)

Partition number (1-4): 2
First cylinder (197-621, default
197):<RETURN> Using
default value 197
Last cylinder or +size or +sizeM or +sizeK

(197-621, default 621): +128M Now the

partition table will look similar to this:

Device Boot Start EndBlocks Id System
*dev*hdb1 1 196 39510483 Linux
*dev*hdb2 197

26213305683 Linux

Finally, make the first

partition bootable:

Command (m
for help): a
Partition
number (1-4):
1

And make the second partition

of type swap: Command (m for

help): t
Partition number (1-4): 2
Hex code (type L to list codes): 82
Changed system type of partition
2 to 82 (Linux swap) Command
(m for help): p

The end result:
Disk *dev*hdb: 64 heads, 63
sectors, 621 cylinders Units =
cylinders of 4032 * 512 bytes

Device Boot Start End BlocksId System
*dev*hdb1*1 196 395104+ 83 Linux
*dev*hdb2 197 262 13305682 Linux swap
*dev*hdb3 263 458 39513683 Linux
*dev*hdb4 459 621 32860883 Linux

Finally, issue the write command (w) to write the table on the disk.

Question 35: Dual booting

I recently installed Fedora Core - dual booting with Windows 2000. I plan to buy a new hard drive and install Fedora on it and then just connect the cables to boot into Linux every time I need to use it.

Is this feasible?

A: You can easily achieve dual booting by setting your current drive to master and the new one to slave. To be safe, disconnect the power from the master. Install Linux to the slave drive. Once you're done installing, reboot, test, then re-connect the power to the master drive. Change the setting in the BIOS of your computer rather than switching cables every time you want to change to Linux.

Change the initial boot drive from HDD0 to HDD1 (look at how your BIOS lists them). Windows has to be the first hard drive but Linux will know that it is not "hda"; it will be "hdb" in

the case. Also by using this method, you can mount the windows drive into the running Linux install and share files.

Another option is to create a boot disk or cd and only boot from it when you want to load Linux. Or, you can set the bootloader you're using to default to Windows instead of Linux.

Question 36: An easy way to dual boot Windows and Linux

Is there an easy way to dual boot Windows and Linux?

A: The following procedure is applicable if you have a newer pc. This method will not work with older BIOS due to the 1024 cylinder boot limit.

It is assumed that you have an Intel-compatible desktop or notebook with a bootable CD-ROM or DVD drive. You should have at least 128M of RAM. I recommend that you have at least 10G of disk space available for Linux, though 5G would probably be enough.

The latter condition should hold for almost any machine bought in the last five or six years. One other point, though, is that the boot priority should be set so that the machine tries to boot from CD-ROM or DVD before trying to boot from the hard drive. Your machine probably already does this but if not, you can reset the

BIOS to do so. Consult your manual on this, or ask at any computer store.

This example will use a 20G HDD and will be installing Windows and Linux Red Hat for concreteness of discussion but the principles should be similar for most other distributions.

Here is our planned partition layout:

1. Windows

2. /boot
3. /swap
4. /

Step I: You will need three things:

1. A Windows 98SE boot disk. This can be created on Windows 98 (control panel > add/remove programs > startup > create boot disk) or alternatively you

can search on the internet for one.

2. Microsoft Fdisk. This is again available from Windows 98. Just do a Find Files and folders for 'fdisk' (I think it is in c:\windows\com) and copy the file to floppy

3. A bootable Windows disk and Linux installation disksStep II: Partitioning

Before you do the partitioning, I recommend that you run Windows' chkdsk command first in case you have any bad sectors on your hard drive.

Insert your boot disk and boot the PC. Go into your BIOS and set the first boot to Floppy. Save changes and exit.

Your PC should boot and eventually prompt you to make a selection of three options. Choose 2 - Start without the CD ROM support. You can choose 1, but you don't need CD ROM support.

It will take a minute or so and you will end

up at the command prompt (A:\). Now insert your disk with 'fdisk' on it and type: fdisk. You will then be prompted whether you want large disk support. Select 'Yes'

Now you will be presented with some options.

Select option 1: Create a DOS

partition of logical dos drive. The

next screen will give you three

options.

Select option 1 again: Create Primary DOS partition. Now you're going to create a Primary partition. If asked to use *all* space, answer 'No' and enter the amount you wish for the C: drive. If you want 12Gigs, enter: 12000

You can also go back and delete it if you made

the wrong choice.

There are two things you can do from here. You can create the other partitions or continue and create the other partitions when installing Linux. If you want to create the partitions now, read the next step otherwise skip to Step V.

Step III: Create extended and logical partitions:

After creating the first partition, we now go back to the main menu and select option 1 again: Create a DOS partition of logical dos drive.

Now we select option 2: Create Extended DOS partition. This partition will hold the swap, /boot and root.

When prompted to enter the amount, use *all* the space left. If you don't, you'll wind up with unused space on your hard drive.

After doing so, FDISK should automatically advance to the next step -> creating Logical DOS drives. It should give you a message that says something like “This drive has no Logical DOS

drives. Would you like to create some now?”

The logical DOS drives are just all the partitions within the extended partition.

Select 'Yes'. Now enter the first amount. If you want 70MB for your boot, then type 70. Now create another one and enter 512 for your swap. Again, enter the remaining space which will be your root (/).

Step IV: Formatting the drive for Windows:

Now, exit from the fids menu, put your boot-disk back in and reboot. Select option 2 and when arriving at the prompt (A:\) type: format c:

This will format the drive for installation. When finished, reboot the computer (make sure you take out the floppy). Go into the BIOS and set the first boot to your CD ROM.

Now put your Windows CD in and save the changes you have made to the BIOS. Windows

will boot and install from the CD. After installation, don't worry

about installing all your drivers,

you can do that later. Step V:

Installing Linux:

Now reboot the computer with your Linux disk 1 in the CD ROM. Make sure that your first boot device is set to CD.

Run the installation program. When prompted to partition, you can just use the existing partitions and assign them to swap, boot and /. Otherwise you can create the partitions from within the program by manual partitioning. Make the partitions to suit you preferred sizes and assign them to their mount points. Continue with the installation.

When prompted to install a bootloaded, choose GRUB if it is there. LILO will also work fine but GRUB is better. Make sure you install the bootloader on the Master Boot Record (MBR),

otherwise you will not have a boot option screen to boot Linux and Windows.

When prompted to make a boot disk, insert a

floppy and let it create a boot disk. When

finished, reboot and you are done.

Question 37: Confirm PSU status

I had a Redhat installed in my pc. During BIOS boot a battery missing can be seen.

How can I confirm the battery or psu status in the OS environment?

A: The information you are after are in the *proc directory and under* proc/acpi. You can do cat *proc*cpuinfo to see what is in the file.

I also suggest writing down all the information that you see in the BIOS and get a new battery. To check the voltage of the CMOS battery, use lm_sensors. The voltage will not tell you that it is dead. You need a button battery checker to test if it is still good or bad.

Question 38: Disable driver verifier

I'm having an error that is related to drivers but I am unable to find any conflicts. I have tried disabling all new hardware, but to no avail.

How can I disable the driver verifier?

A: To set file signature verification options, you can do the ff:

1. Open System in Control Panel.
2. On the Hardware tab, click Driver Signing.
3. Under File signature verification, click one of the following:

Ignore to allow all device drivers to be installed on this computer, regardless of whether they have a digital signature.

Warn to display a warning message whenever an installation program attempts to install a device driver without a digital signature. This is the default behavior for Windows.

Block to prevent an installation program from installing device drivers without a digital signature.

If you are a logged on as an administrator or as a member of the Administrators group, under Administrator option, click Apply setting as system default to apply the selected setting as the default for all users who log on to this computer.

Question 39: SCSI

What is a SCSI?

A: SCSI stands for Small Computer System Interface. It's a standard for connecting peripherals to your computer via a standard hardware interface which uses standard SCSI commands. The SCSI standard can be divided into SCSI (SCSI1) and SCSI2 (SCSI wide and SCSI wide and fast) and now SCSI-3 which is made up of at least 14 separate standards documents.

SCSI2 is the most popular version of the SCSI command specification and allows for scanners, hard disk drives, CD-ROM players, tapes and many other devices. SCSI-3 resolves many "gray areas" and adds much new functionality and performance improvements. It also adds new types of SCSI busses like fiber channel which uses a 4 pin copper connection or a pair of glass fiber optic cables instead of the familiar ribbon cable connection.

Question 40: Choosing SCSI Devices

I want to use a SCSI interface but how do I choose what devices should I utilize?

A: I suggest for you to have devices which use LVD, automatic termination and SCAM (SCSI Configured Automatically). Most new devices fit this pattern. Also make sure that you consider where these devices are going to go, since you need to consider the total length of the cable. And check to see if you will need an adapter for an external device (50 to 25 pin or 68 to 50 pin).

Buy quality SCSI cards and PCI if possible. Make sure it will support the bus speed you have in mind (SCSI-3 U2W for example). I recommend Adaptec cards as the best on the market but I have also heard good things about the Diamond cards. Also make sure you buy high quality cables. Granite Digital makes great cables and other accessories and is well known for their quality.

A SCSI hard drive is set-up just like any other drive. Depending on your BIOS settings, you can set the system to boot from the SCSI drive or from an IDE drive. There is no problem in using a mixed drive setup. Seagate, IBM and Quantum are all well known for their SCSI drives.

A great number of manufacturers offer SCSI burners (CD-R or CD-RW) but just plain CD players are harder to come by.

A good high-end scanner will be a SCSI scanner. Check around to see what kind of results people are getting with the model you have in mind.

Zip Drives, LS120s, Jaz Drives and many other devices of this type are available in the SCSI interface. You can add any SCSI device into your chain as long as you have not exceeded the number of devices for your card.

Question 41: Setting up a SCSI chain

How do I set up a SCSI chain?

A: Install the SCSI card. Most cards today are plug_and_play and Windows will find and configure them quickly and easily. The next step is to cable the devices together and set the I.D.s and termination (if necessary). Each device on a SCSI chain requires its own I.D.

As an example (example is an 8 device chain):

ID#0 - hard drive
ID#1 - CD-ROM
ID#2 - CD-R
ID#3 - Internal Jaz
ID#4 - Scanner
ID#5 -

Hard drive
ID#6 - Zip
ID#7 - SCSI Card

If two devices are set to have the same I.D., the whole chain may not work or possibly just the conflicting devices. Always make sure each device has a different I.D. Most devices have jumpers or a small dial for setting the I.D. number. The hard drive at ID#0 should be your boot drive (if you are booting SCSI), and the CD-ROM is set at the lowest available I.D.# in order to have it assigned a drive letter before the CD-R.

Termination is a separate issue, but it also affects your SCSI bus a great deal. Each device at the far ends of a SCSI chain must be terminated. This refers to their physical location on the chain and NOT their ID number. In the above example, the scanner is probably at one end of the chain and possibly a hard drive is at the other with the card in the middle, like so:

Scanner

ZIP

Card

CD-ROM

CD-R

Jaz
Hard
drive
Hard
drive

So, unless the devices are auto-terminating, that hard drive and the scanner must be set to terminate usually by jumpers or a switch on the device. In older devices, this might actually require placing a terminator on the end of the chain. Generally, that is no longer necessary. All items between the ends must have their termination disabled or the chain will not operate properly. If you have all internal devices or all external devices, then the card itself will be an

end device and must be terminated. If you have a couple of auto termination devices, and a few which are not, you can try to set the auto termination devices at the ends of the chain to make setup easier.

Despite the reputation of being difficult to configure, I have found that SCSI is extremely easy to get up and running. Pay attention to the issues above and you should be fine. Just in case, here are a few troubleshooting tips:

Check Termination First.

Then check it again. Check

ID numbers

Use quality cables. If you have a problem especially with a fast hard drive, try changing the cable.

Use the shortest connections possible. Pay attention to the maximum total length.

Make sure you have the newest ASPI drivers (Adaptec site if you have an Adaptec card).

Make sure the cables are plugged in

correctly and firmly: pin 1 to pin 1. If you

are not sure your devices are SCAM

compliant, turn it off.

Check for IRQ conflicts.

Question 42: Changing screen resolution

I installed mdv 2006 and it works fine at 1024X768. I wanted to raise the resolution but when I do, the window is larger than my screen. I have to drag the mouse pointer to see the top or bottom of the screen or left and right.

What is the best way to change the screen resolution?

A: There are a couple of things you can do:

1. Install all the updates available. There are several updates to the graphical system called "X" or "xorg". Mdv2006 shipped with a developmental version of xorg that caused problems with some nvidia based cards. Later updates fixed that.

2. The configuration file for your graphical system is *etc*X11/xorg.conf. This is a fairly lengthy file divided into sections.

3. After you installing all the updates, rerun the graphical configuration utility. You can do so by opening a console and run:

$ su
<enter root
password> #
XFdrake

A window will pop up with XFdrake, the mandriva graphics setup utility. Check that you have the right card selected and double check your monitor setting. There is a button you can press to test the configuration changes you make before committing them. Keep trying until you get something that tests out properly. Actually, you might want to try running XFdrake before doing anything else. By the way, XFdrake automatically edits xorg.conf for you thus simplifying the graphics setup process.

4. Mandriva ships with the open source driver for your nvidia based graphics card called "nv".

That's probably what you will see in your xorg.conf "Device"

section for the driver. Nvidia makes an excellent closed source Linux driver but the installation is a little daunting for newbie's. You need the nvidia driver for 3d acceleration necessary to run many games. After you get things running properly with the open source driver, you may want to consider installing the nvidia closed source driver.

Question 43: Modem

What is a Modem?

A: A modem is a device that lets one send digital signals over an ordinary telephone line not designed for digital signals. If all telephone lines are digital, then you wouldn't need a modem. But sometimes, a substitute for an analog modem, connected to a digital phone line is inaccurately called a "digital modem". A modem permits your computer to connect to and communicate with the rest of the world. When you use a modem, you normally use a communication program or web browser to utilize the modem and dial-out on a telephone line. Advanced modem users can set things up so that others may phone in to them and use the computer remotely. This is called "dial-in".

Oversimplified, there are four basic types of analog modems for a PC: external serial (RS-

232), USB (= external USB), internal, and built-in. The external serial and USB set on your desk outside the PC while the other two types are not visible since they're inside the PC. The external serial modem plugs into a connector on the back of the PC known as a "serial port". The USB modem plugs into a USB cable. The internal modem is a card that is inserted inside the computer. The built-in modem is a chip on the motherboard used primarily in laptops. Internal modems will generally apply also to built-in modems. Internal modems are further subdivided into PCI, ISA, and AMR depending on whether they are designed for the PCI or ISA bus or for an AMR slot.

When you get an internal or built-in modem, you also get a dedicated serial port (which can only be used with the modem and not with anything else such as an external modem or console terminal). In Linux, the common serial ports are named ttyS0, ttyS1, *etc.* (or tts/0, tts/1 for the device file system (devfs).

Modems usually include the ability to send Faxes (Fax Modems) while "Voice" modems can work like an automatic answering machine and handle voicemail.

Question 44: Identify software modem

How do you determine if an internal modem is a software modem?

A: First see if the name, description or even the name of the MS Windows driver for it indicates it's a software modem: HSP (Host Signal Processor) , HCF (Host Controlled Family), HSF (Host Signal Family), controller less, host-controlled, host-based, and soft modem. If it's one of these modems, it will only work for the cases where a Linux driver is available. Since software modems cost less, a low price is a clue that it's a software modem.

If you don't know the model of the modem and you also have Windows on your Linux PC, click on the "Modem" icon in the "Control Panel". Then see the modem list. If the above doesn't work, you can look at the package the modem came in. Read the section on the package that says something like "Minimum System Requirements" or just "System Requirements".

A hardware modem will work fine on old CPUs (such as the 386 or better). So if it requires a modern CPU (such as a Pentium or other "high speed" CPU of say over 150 MHz) this is a clue that it's an all-software modem. If it only requires a 486 CPU (or better) then it's likely a host-controlled software modem.

You may have a hardware modem if it fails to state explicitly that you must have Windows. By saying its "designed for Windows" it may only mean that it fully supports Microsoft's plug-and-play which is okay since Linux uses the same plug-and-play specs. Being "designed for Windows" thus gives no clue as to whether or not it will work under Linux. You might check the Website of the manufacturer or inquire via email. Some manufacturers are specifically stating that certain models work under Linux. Sometimes they are linmodems that require you to obtain and install a certain linmodem driver.

Question 45: sound application error

The sound applications do not work properly

when I run Linux then boot DOS. Why does this

happen?

A. That happens after a soft reboot to DOS. Sometimes the error message misleadingly refers to a bad CONFIG.SYS file. Most of the current sound cards have software programmable IRQ and DMA settings. If you use different settings between Linux and MS-DOS/Windows, this may cause problems. Some sound cards don't accept new parameters without a complete reset.

A fast solution to this problem is to perform a full reboot using the reset button or power cycle rather than a soft reboot (e.g. Ctrl-Alt-Del).

The correct solution is to ensure that you use the same IRQ and DMA settings with MS-DOS

and Linux (or not to use DOS).

Question 46: Configure Linux to support sound

What are the correct steps to configure Linux to support sound?

A: Follow these steps:

1. Installing the sound card: Follow the manufacturer's instructions for installing the hardware or have your dealer perform the installation. Older sound cards usually have switch or jumper settings for IRQ, DMA channel, etc; note down the values used. Use the factory defaults if you are not sure. Try to avoid conflicts with other devices (e.g. Ethernet cards, SCSI host adaptors, serial and parallel ports) if possible. Usually you should use the same I/O port, IRQ, and DMA settings that work under DOS. In some cases, you may need to use different settings to get things to work under Linux. A trial and error may be needed.

2. Configuring Plug and Play if applicable: Some

sound cards use the ISA Plug and Play protocol to configure settings for I/O addresses, interrupts, and DMA channels. If you have a newer PCI-bus type of sound card or one of the very old ISA sound cards that uses fixed settings or jumpers, then you can skip this procedure. The preferred way to configure Plug and Play cards is to use the isapnp tools which ship with most Linux distributions or download them from Red Hat's web site http://www.redhat.com/.

First check the documentation for your Linux distribution. It may already have Plug and Play support set up for you or it may work slightly differently than described here. If you need to configure it yourself, the details can be found in the man pages for the isapnp tools. Briefly the process you would normally follow is:

Use pnpdump to capture the possible settings for all your Plug and Play devices, saving the result to the file *etc*isapnp.conf.
Choose settings for the sound card that do not conflict with any other devices in your system

and uncomment the appropriate lines in *etc*isapnp.conf. Don't forget to uncomment the (ACT Y) command near the end.

Make sure that isapnp is run when your system boots up, normally done by one of the startup scripts. Reboot your system or run isapnp manually.

If for some reason you cannot or do not wish to use the isapnp tools, there are a couple of other options. If you use the card under Microsoft Windows 95 or 98, you can use the device manager to set up the card, then soft boot into Linux using the LOADLIN program. Make sure Windows and Linux use the same card setup parameters. If you use the card under DOS, you can use the icu utility that comes with SoundBlaster16 PnP cards to configure it under DOS, then soft boot into Linux using the LOADLIN program. Again, make sure DOS and Linux use the same card setup parameters. True ISA PnP support is implemented in the 2.4 and later kernels. Some of the sound card drivers now support automatically detecting and configuring the cards without the isapnp tools. Check the documentation for the card's driver for details.

3. Configuring and building the kernel for sound support: You need the appropriate device drivers for your sound card to be present in the kernel. The kernel running on your system may already include the drivers for your sound card. In most cases the drivers would have been built as kernel loadable modules. You can check which drivers are available as modules by looking in the *lib* modules directories. For the 2.4.4 kernel, the sound drivers would normally appear in *lib* modules/2.4.4/kernel/drivers/sound/. If you see the driver(s) for your sound card, you can try using the module directory and skip recompiling the kernel.

If the sound drivers are not already built, you will need to configure and build a new kernel. You can either build the sound drivers into the kernel or build them as kernel loadable modules. In most cases building as modules is preferred, as it allows you to easily experiment with loading different drivers if unsure which one to use and the drivers can be unloaded when not needed, freeing up memory. Building the drivers into the

kernel itself may be desirable if you are unfamiliar with kernel modules and want a simpler solution.

If you have never configured the kernel for sound support before it is a good idea to read the relevant documentation included with the kernel sound drivers, particularly information specific to your card type. The files can be found in the kernel documentation directory and usually installed in

*usr*src/linux/Documentation/sound. If this directory is missing, you likely either have a very old kernel version or you have not installed the kernel source code. Follow the usual procedure for building the kernel. There are currently three interfaces to the configuration process. A graphical user interface that runs under X11 can be invoked using make xconfig. A menu-based system that only requires text displays is available as make menu config. The original method, using make config, offers a simple text-based interface.

When configuring the kernel there will be many choices for selecting the type of sound card you have and the driver options to use. The on-line help within the configuration tool should provide an explanation of what each option is for. Select the appropriate options to the best of your knowledge. After configuring the options you should compile and install the new kernel as per the Kernel HOWTO.

4. Creating the device files: For proper operation,

device file entries must be created for the sound devices. These are normally created for you during installation of your Linux system. A quick check can be made using the command listed below. If the output is as shown (the date stamp will vary) then the device files are almost certainly okay.

% ls -l *dev*dsp
crw-rw-rw-1 root root14,3 Jan 25 1997 *dev*dsp

Note that having the right device files there doesn't guarantee anything on its own. The kernel driver must also be loaded or compiled in before the devices will work.

In rare cases, if you believe the device files are wrong, you can recreate them. Most Linux distributions have a *dev* MAKEDEV script which can be used for this purpose.

Take note that if you are using the devfs filesystem support in the 2.4 kernels, the sound device files are found in *dev* sound, but there

will be symbolic links to the older devices, such as *dev*dsp.

5. Booting the Linux kernel and testing the installation: You should now be ready to boot the new kernel and test the sound drivers. Follow your usual procedure for installing and rebooting the new kernel.

If you are using loadable kernel modules for sound, you will need to load them using the modprobe command for the appropriate drivers, *e.g.* run the command modprobe sb for a SoundBlaster card.

After booting or loading the kernel modules, check for a message such as the following using the dmesg command:

```
Sound blaster audio driver Copyright (C) by
Hannu Savolainen 1993-1996 sb: Creative
SB AWE64 PnP detected
sb: ISAPnP reports 'Creative SB AWE64
PnP' at i/o 0x220, irq 5, dma 1, 5 SB 4.16
detected OK (220)
sb: 1 Sound blaster PnP card(s) found.
```

Crystal 4280/46xx + AC97 Audio, version 1.22.32, 10:28:40 Apr 28 2001 cs46xx: Card found at 0xf4100000 and 0xf4000000, IRQ 11
cs46xx: Thinkpad 600X/A20/T20 (1014:0153) at 0xf4100000/0xf4000000, IRQ 11
ac97_codec: AC97 Audio codec, id: 0x4352:0x5914 (Cirrus Logic CS4297A rev B)

The message should indicate that a sound card was found and match your sound card type and jumper settings (if any). The driver may also display some error messages and warnings if you have incorrectly configured the driver or chosen the wrong one.

Previous versions of this HOWTO suggested checking the output of *dev*sndstat. This is no longer supported in the 2.4 and later kernels.

Now you should be ready to play a simple sound file. Get hold of a sound sample file, and send it to the sound device as a basic check of sound output, *e.g.*

% cat end of the world > *dev*dsp

% cat crash.au > *dev*audio

(Make sure you don't omit the ">" in the commands above).

Note that using cat is not the proper way to play audio files; it's just a quick check. You'll want to get a proper sound player program that will do a better job.

If the above commands return "I/O error", you should look at the end of the kernel messages listed using the "dmesg" command. It's likely that an error message is printed there. Very often the message is "Sound: DMA (output) timed out - IRQ/DRQ config error?" The message means that the driver didn't get the expected interrupt from the sound card. In most cases it means that the IRQ or the DMA channel configured to the driver doesn't work. The best way to get it working is to try with all possible DMAs and IRQs supported by the device.

Another possible reason is that the device is not

compatible with the device the driver is configured for. This is almost certainly the case when a supposedly "SoundBlaster (Pro/16) compatible" sound card doesn't work with the SoundBlaster driver. In this case you should try to find out the device your sound card is compatible with.

If you have sound input capability, you can do a quick test of this using command such as the following:

```
# record 4 seconds of audio from microphone
% dd bs=8k count=4 <dev
audio >sample.au 4+0
records in
4+0 records out
# play back sound
% cat sample.au >dev audio
```

Obviously for this to work you need a microphone connected to the sound card and you should speak into it. You may also need to obtain a mixer program to set the microphone as the

input device and adjust the recording gain level.

If these tests pass, you can be reasonably confident that the sound D/A or A/D hardware and software are working.

Some Linux distributions provide a sound driver configuration utility that will detect your sound card and set up the entire necessary configuration files to load the appropriate sound drivers for your card. Red Hat Linux, for example, provides the config utility. If your distribution provides such a tool I suggest you try using it.

Source: From Internet

Follow Me For next series of Questions coming

https://www.linkedin.com/in/mukeshkumarrao/

Visit at: www.ineuron.ai
For Live Training courses and Job Assurance guidance from our mentors

********** THANK YOU **********