1. Semester and Year : 2012, FALL
2. Course Number : CS-227
3. Course Title : Linux & System Software
4. Work Number : LA-04
5. Work Name : Lab 4
6. Work Version : Version1
7. Long Date : Sunday, 23, September, 2012
8. Author(s) Name(s) : Jake Waffle

Task 2.5: Configure Disk Devices

Identifying Your Hardware
Linux acts as if all disk devices belong to either the ATA or SCSI category. And ATA disks have device filenames similar to /dev/sda1, while SCSI device filenames are similar to /dev/hda1 ("a" and "1" in each filename are objective.)

After using the df command to check all of the disks on a school computer. It resulted with one disk that falls under the ATA category, a udev filesystem (pseudo-filesystem) and a couple of entries with a filesystem label of "None."

Then I used the command "fdisk -l /dev/sda" to look up sda's partition table. And the resulting table says that there is just a root partition and a swap (which is on an extended partition, which makes it a logical partition I think.)

I also looked up a complete list of SCSI devices by printing /proc/scsi/scsi to the screen (cat /proc/scsi/scsi.) And the resulting text tells me that there are two SCSI devices. One of them is the virtual box hard disk and the other is the cd-rom.

Testing Disk Performance
I tested the raw disk input/output speed and cached disk input/output speed using command "hdparm -tT /dev/sda". The resulting speeds were 1715.88mb/sec for the cache disk speed and 74.66mb/sec for the raw disk speed.

Tweaking Disk Performance
The hdparm command also has an option (-X) for changing a disks Direct Memory Access(DMA) transfer mode. And you are able to use sdmax simple DMA level x, mdmax multiwird DMA level x and udmax ultra DMA level x. I did not change any disk's DMA transfer mode to refrain from making unreasonable changes.

Task 3.4: Perform System Backups

Selecting Backup Hardware

The book says that almost anything that stores data can be used as a backup medium (page 130.) But random access devices (CD-ROMS) are faster than sequenctial access devices (tapes.)

Planning a Backup Schedule

There are different things one needs to keep in mind while planning a backup schedule. The first consideration I'd say is the size of the system that needs backed up. If the system is large, then an incremental backup technique may need to be used instead of a full backup (a full backup may be unnecissary when it needs updated frequently.) Or if the system is small, a full backup should suffice. What needs to be backed up is another – maybe only a single directory.

Choosing Backup Software

There are complex programs that handle backup scheduling tools and simple commands that can be assessed when choosing backup software.

Command cp is capable of copying a system onto disks, but the command needs to be run by the user each time in order to update the backup (this could simplify each backup update, but requires the backup updates to be applied manually.)

AMANDA is a network-based backup tool that may simplify the backup procedure on a network, but it requires a lot of effort to configure AMANDA on both the backup server and its clients. Backing up a network is probably a lot more complex than just an individuals computer.

The tar command is also used for formatting backed up files to reduce its size (it's comparible to Window's .zip files.)

Preparing Backup Commands

The tar command is used as an example in the book (page 134) to create a regular backup and an incremental backup of a list of directories and save it on a specified device

(regular command: "tar cvlf /dev/st0 / /home /usr"
and incremental command: "tar cvplf /dev/st0 –listed-incremental /var/log/incr.dat / /home /usr")

There really isn't much to preparing your backup tar command though. The options are well optional and after looking through them with "tar –help" it's easy to see what is needed or just helpful. Like –create for example, it is needed in order to create a new archive. And –file is needed to select the archive device or file that the new archive will be stored in. --verbose is a common option present in many terminal programs. It is just a generally helpful option to include in a command, because it prints out what the program actually does! And finally –listed-incremental is needed if an incremental backup is what the user is going for. Other than that, the directories to be backed up need to be specified.

Planning a Restore Strategy

There seems to be two general strategies for system restore. One is to restore the entire system back to a previous state (when it was backed up) and the other is to just fix a part of the system that has been corrupted.

A bootable linux system (like partedMagic) can be used to reimage a system with a backed up image. And it also has tools that can be used to fix just part of the system. These systems are pretty versatile from what I've seen.

A similar method of doing this could be to just reinstall a minimum system and recover the system with a backed up image.

My Backup Plan

My plan for backing up my system is simple. I will only backup the /home directory (where I store my important files: school work, projects, collected information) and I will use an external harddrive as the storage medium. I will use the tar command ("tar cvplf /externalHarddrive –listed-incremental /var/log/increment.dat /home") to compress the /home directory and save the file onto the external harddrive each week (to make sure the files are up-to-date.) Then when an important file gets corrupted, restoring the file is as easy as getting it off of the external harddrive. And when the whole system goes down, a fresh install with the extraction of the important files is enough (programs can always be downloaded again.)

Task 3.8: Schedule Jobs to Run in the Future

Creating User Cron Jobs

When trying to check my crontab file with the "crontab -l" command. I found out that I don't have crontab file on my computer. So I created a simple crontab file with "vi my-crontab" and saved it as my user's crontab file with "crontab my-crontab." Now I just need to think of a command that I may need run every so often (like a backup command) for this user and add it to the crontab.

Creating System cron Jobs

The system cron job scripts are stored within /etc/crontab.interval to be ran in correspondence with the interval provided (scripts in /etc/crontab.daily will be ran everyday.) To create a new system cron job, just save the according script in the corresponding /etc/crontab.interval directory.

Running a Single Future Job

When a command or script will only need to be ran once, the at command is useful. It allows for a command or script to be ran at a time specified in the future. Note that the time can be specified differently -- it can be specified as now + a duration, MM/DD/YY, or the time of day. If I was wanting to run script.txt on my birthday at teatime, I could use "at -f script.txt teatime 02/13/93".

Task 4.2: Monitor System Performance

Testing Your CPU

The linux kernal provides a means to test the CPU speed and is called BogoMIPS. The BogoMIPS can be found by searching for Bogo in the dmesg output (dmesg | grep Bogo.) The BogoMIPS CPU speed is dependent on the internal kernel timing loops, so it isn't really decipherable at first glance. My computer's BogoMIPS value is 4788.66 (found with "dmesg | grep Bogo".)

Testing Your Memory

Free is a command provided by linux that allows a user to see the amount of total memory and memory in use for the RAM, buffers/cache and swap space at a single point in time (top outputs average memory use.)

My results from the free command:

	total	used	free
RAM	3.8gb	3.5gb	0.3gb
Buffers/Cache		884mb	3gb
Swap	262140b	20b	262120b

Testing Your Disks

The hdparm utility allows testing for disk input speed. The book says that the Timing cached output ". . . is more of a measure of RAM performance than disk performance" (page 167), while the Timing buffered disk outputs a reasonable disk performance.

My computer after inputting "hdparm -tT /dev/sda" outputted 3297.63mb/s for the Timing cached line and 103.23mb/s for the Timing buffered disk line. So according to what the book said about the performance measure of each line, the RAM performance on my computer is much faster than the disk performance.

Df is command that displays information on the amount of available space on each of the computer's mounted partitions. And this is relevant to testing your disk, because issues can arise when one runs out of space on a disk. So with the df command, one can check to see if either of the mounted partitions is almost full.

All of my partitions are well under 80% used, except the /host which is at 89%. (I have a dual-boot(Windows/Ubuntu,) with my ubuntu install on its own partition (along with some other files on the Windows-side,) which I think is the /host.) So I should either make this partition bigger or delete some of the files I've put alongside the Ubuntu install.

Performing Ongoing Tests

Through use of cron and a script, one can perform a test for system performance with commands like the ones gone through in this chapter (i.e. df, free, hdparm.) But the example in the book looks rather lame and requires a human's analysis to find any problems. I bet it would be simple enough to apply some of the logic in this chapter to make a script that would warn the user if something looks off (as defined by the script.)

Task 4.4: Use System Documentation

Start with man

Tha man-pages allow one to look up information on utilities, configuration files and other things. And it is really simple to use, one only needs to type the utility/other thing following "man ".

Additional man-Related Tools

When it is unclear what is needed to be looked up in the man-pages, there are a couple of ways to go about this. One is to search through the man-pages with the whatis command (we've encountered this one before) and output a summary of each similar command. And the other is to search through the man-pages with the apropos command and output a little more detail of each similar command than the previous method.

Using the CUPS HTML Documentation

In most man-pages there is a section entitled "See Also" and under it is either related commands or a URL pointing to documentation on your computer. The URL can be entered into a Web browser and automatically be handled by the Common Unix Printing System (CUPS.) The CUPS server will ultimately show several links to documentation on your computer.

Looking for Additional Documentation

It is also possible to use the rpm package manager to output a list of the files that are contained within a specific package (the -qfl options passed to rpm will produce the described effect.)

Using GUI Programs' Documentation

GUI programs exist with their own documentation separate from that of the man-pages. And this is pretty much the same as how Windows treats its documentation for GUI programs (besides with readme files.)