**LINUX EXERCISE (LAB 09)**

**Backup operations for Linux systems**

Exercise 1. Tar command

1. What are the advantages of tar command when comparing with cp command to backup files and directories?
2. Log in as root and change directory to /var/log
3. How much disk space is being used by this directory? Record this number (Hint: use du)
4. Create the archive in root's home directory

*tar cvf /root/logfiles.tar*

1. Change directory back to root's home and view the archive you just made.
2. Note the names of the files in the archive. Are they absolute or relative?
3. How large is this tarball? How does it compare to the disk usage recorded above?
4. Extract from this archive the messages file using the following command

*tar xvf logfiles.tar messages*

1. Why doesn't this work?
2. Try again with: *tar xvf logfiles.tar ./messages*
3. How does the messages file you just extracted compare with the messages file in the /var/log directory? Are they the same? Look at dates, permissions, ownership as well as contents
4. Create a second tarball by archiving the /var/log directory as an absolute name *tar cvf logfile2.tar /var/log*
5. List the files in this archive and notice how the names are specified. Are they absolute or relative?
6. See if you can extract the messages file from this archive
7. Where will the file be extracted?

Exercise 2. CPIO

1. What is the difference between cpio and tar command?
2. Log on as root, or su to the super-user account
3. If we were performing the seventh day of a differential backup, then we would want to archive all the files on the root file system that have been modified within the last 7 days. To see which files these are, run the following find command

*find / -mount -mtime -7 | more*

1. What does the -mount option do?
2. Since cpio reads its input of filenames from stdin and writes the archive to stdout, we can use a pipeline to archive these files to file with the path /tmp/level-7.bak
3. Verify the backup using cpio (with the options “-vitB”)
4. Where would these files be restored if we were to extract this archive?
5. Change to the / directory, and extract this archive
6. Notice whether the archive file remains even after you extract its contents

Exercise 3. DUMP

1. Add a new SCSI hard drive to your Linux system (VM or real system). Make it 5GB in size and do not allocate all disk space now. This will be our virtual backup device
2. Start up your Linux system and login as root
3. Create a single primary partition on the new drive, /dev/sdb
4. Change directory to /
5. Run the dump command to save the first partition of your current working hard-drive (/dev/sda1) to the backup drive (Note: /dev/sda1 should be your root partition. You need to check it)
6. When the dump is complete, (about 5 minutes), try to restore first 100 files from the backup file to a directory (Ex: /root/lab09) using **restore** command
7. Change directory to /etc and do a long listing of grub.conf. Note what kind of file this is. Will a restore bring back a symbolic link, or the file it is pointing to?
8. Move the grub.conf file in /etc to the /tmp directory
9. Use the interactive mode of the restore program to extract the file you just moved

* Hint: use the man page or Lesson Powerpoints to see how this is done.
* When restore asks you which volume do you want restored, select 1
* If asked to set owner/mode, select "n"

Note: To actually restore the files from the backup, you must be in the top directory of the file system, in this case /

1. When you are done with the restore, how does the file you restored compare to the file you moved to the /tmp directory? How about the inodes?