OPS245 Lab 3

Contents

LAB PREPARATION

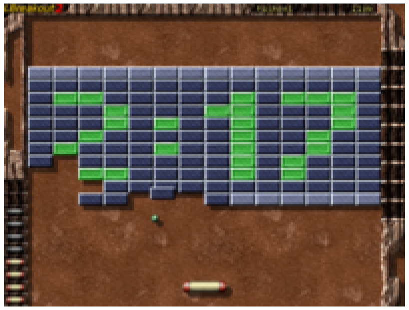
Purpose / Objectives of Lab 3

In this lab, you are going to identify potential problems such as running out of space to run a GUI, or forgetting your root password. You will also learn to conserve hard disk space by learning how to compress and decompress files that are stored on your computer server. In addition, you will learn alternative methods of how to install applications (i.e. programs).

Main Objectives

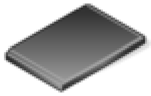
- Learn to boot your CentOS VM into **Single User mode** for rescue purposes
- Create and extract **archive** files (**tar** and **tar.gz**)
- Install applications using various methods:
  - **Download**, **decompress** and archive file from the Internet, then **compile source code** in order to install an application.
  - Use **yum** utility to install software from **local source programs** (on DVD drive)
  - Use **yum** utility to install software from **repositories** (online storage areas)
  - Learn how to **add repositories** (online software storage areas) to install less common applications.
- Demonstrate the use of Bash Shell scripts to automate routine tasks (generate a customized software report)

Learning how to conserve disk space is essential for the efficient operation of a Linux computer system. You will learn how to manipulate the size of Linux file-system sizes (via **LVM**) in Lab 5.

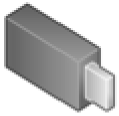


You will download and compile the game called **lbreakout2** which is only available as compressed source code.

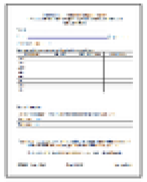
Minimum Required Materials



Solid State Drive



USB key
(for backups)



Lab3 Log Book

Linux Command Reference

Archiving / Compiling

tar (http://man7.org/linux/man-pages/man1/tar.1.html)
gzip, gunzip (http://linuxcommand.org/lc3_man_pages/gzip1.html)
make (http://man7.org/linux/man-pages/man1/make.1.html)

Software Manangement
rpm (https://linux.d

Miscellaneous

ls (http://man7.org/linux/man-pages/man1/ls.1.html)
more (http://man7.org/linux/man-pages/man1/more.1.html)
wget (http://man7.org/linux/man-pages/man1/wget.1.html)
chmod (http://man7.org/linux/man-pages/man1/chmod.

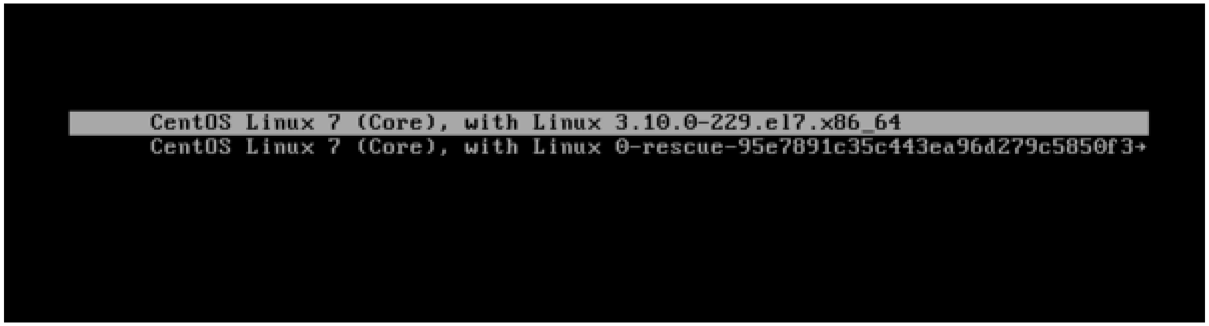
Matrix On-line Tutorials:

- Shell Scripting Basics - Part 4 (The sed Utility): **/home/ops235/scripting-4**

ie.net/man/8/rpm) yum (http://man7.org/linux/man-pages/man8/yum.8.html)	1.html) vi (http://ss64.com/vi.html)

INVESTIGATION 1: TROUBLESHOOTING BOOTING & FORGOTTEN ROOT PASSWORD

Part 1: Common Booting Issues



```
CentOS Linux 7 (Core), with Linux 3.10.0-229.el7.x86_64
CentOS Linux 7 (Core), with Linux 0-rescue-95e7891c35c443ea96d279c5850f3→
```

Press **e** at Grub Boot Menu to edit

There are a few "classic problems" that students can encounter with their virtual machines and their host machine after performing lab2. Although all OPS245 students may not encounter these problems, it is good to be aware of what a potential problem may look like, and know how to solve that problem.

A few common problems are:

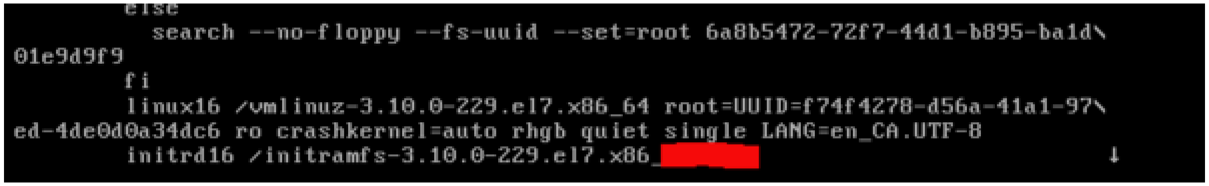
- I Can't boot into Graphical Mode on my **c7host** machine
- I Forgot My **Regular User** Password
- I Forgot My **root** Password
- I Can't Start My Virtual Machine

Troubleshooting consists of 3 basic elements:

- **Asking Questions** (eg. what was done recently prior to problem?)
- **Eliminating** what the problem **IS NOT**
- Having **Tenacity** and **patience** (ability to focus on finding the cause of the problem)

Part 2: Booting into Single-User Mode

If you cannot log graphically into your machine (first two common problems), there is a trick to boot your system into **single-user** mode. This mode does not provide networking, graphics, or logging in as other regular users, but can connect to a text-based interface as the **root** user. This method will only work if a GRUB (**Gr**and **U**nified **B**oot-loader) password has not been set, or that you haven't forgotten the GRUB password.



```
          else
            search --no-floppy --fs-uuid --set=root 6a8b5472-72f7-44d1-b895-ba1d\
01e9d9f9
          fi
          linux16 /vmlinuz-3.10.0-229.el7.x86_64 root=UUID=f74f4278-d56a-41a1-97\
ed-4de0d0a34dc6 ro crashkernel=auto rhgb quiet single LANG=en_CA.UTF-8
          initrd16 /initramfs-3.10.0-229.el7.x86_                                ↓
```

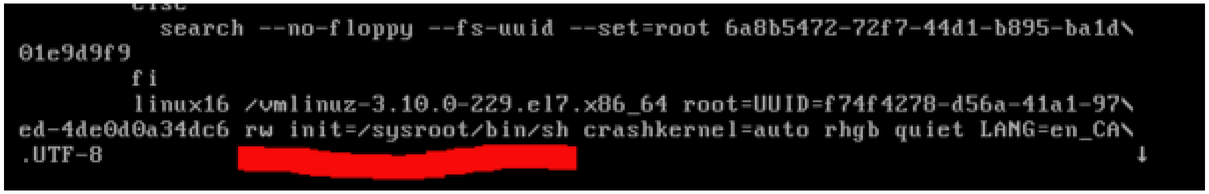
Add the boot option **single** and then press **ctrl-x** to boot into single user mode

Perform the following steps:

1. Launch the VMware application.
2. Select Open Virtual Machine, and select the VM called c7host on your Solid State Drive.
3. Launch your **c7host** machine, and login as as a regular user.
4. Boot-up your **centos1** VM. **When the Grub Boot menu appears**, press the letter `e` (for "edit").
5. Using your arrow keys, scroll to next screen to **linux**, or **linux16**, or **linux-efi** command and type the word `single` as an argument after **quiet** (see diagram for reference) and then press `ctrl-x` to boot.
6. The system should boot into text-based mode. Enter your **root** password.
7. One thing to look at is partition space usage. Issue the command: `df -h`
8. If you notice 0% free space in the / partition, then there is a problem. This most likely is caused by not following steps to create a compressed copy of the VM image. If that is the case, locate the large image backup and remove it. Do NOT remove the image in */var/lib/libvirt/images* directory!
9. You can use the **passwd** command to reset your **regular user** password (eg. `passwd regularuserid`). You can press **ctrl-c** if you wish to abort (i.e. not change password).
10. To restart in graphical mode, simply enter the command `reboot`.

Part 3: Resetting Forgotten Root Password

Unfortunately, booting into Single-User Mode does not work if you forgot your **root** password, since the procedure in the previous part requires that you enter your root password. In order to reset your root password, you need to perform a different procedure (shown below).



```
          else
            search --no-floppy --fs-uuid --set=root 6a8b5472-72f7-44d1-b895-ba1d\
01e9d9f9
          fi
          linux16 /vmlinuz-3.10.0-229.el7.x86_64 root=UUID=f74f4278-d56a-41a1-97\
ed-4de0d0a34dc6 rw init=/sysroot/bin/sh crashkernel=auto rhgb quiet LANG=en_CA\
.UTF-8                                                                          ↓
```

Add **rw init=/sysroot/bin/sh** as shown and then press **ctrl-x** to boot into single user no root password mode

Perform the following steps:

1. The procedure to reset root's password is different than shown above.
2. Make certain that your centos1 VM that you used in Part 2 has been shutdown.
3. Start your centos1 VM and press `e` at the Grub boot menu.
4. Using your arrow keys, move to **linux** or **linux-efi** command and replace the argument **ro** with the argument `rw init=/sysroot/bin/sh` (see diagram for reference) and then press `ctrl-x` to boot.
5. The system should boot into text-based mode without prompting for root's password.
6. Issue the command: `chroot /sysroot`
7. Issue the command: `passwd root` in order to change your root password (press **ctrl-c** if you wish to abort - i.e. not change password).
   NOTE: if you are using SELinux in **enforcing** or **permissive** mode, you will also need to issue the command: **touch /.autorelabel**
   This may take some time depending on the amount of files you have on the file system (usually about 2 minutes).
8. To restart in graphical mode, simply enter the command `exit`, then `reboot`.

What To Do If Reboot Doesn't Work
In this mode, the reboot command may not work. If it doesn't, go to the top of the centos1 VM window, select Send Key -> Ctrl+Alt+Delete.

Catastrophic Boot Problems

Not being able to start your **c7host** due to **Kernel Panic** or some sort of **catastrophic disk failure** is not as easy to fix. You might be able to boot from your **Centos LIVE DVD, open a terminal and mount the partition** via command line and look for possible problems (setup files like **/etc/fstab**). **Lab5** will discuss more about mounting and the /etc/fstab file. The "worst-case scenario" would be to purchase a new hard disk, **perform lab1 completely, perform lab2 to install and set-up virtualization software**, then **restore your VM image and xml file backups** (eg. decompressing images, issuing virsh define commands for .xml files). That is why consistently **performing backups of ALL of your VMS at the end of each lab is absolutely essential! You have been warned!**

Answer INVESTIGATION 1 observations / questions in your lab log book.

INVESTIGATION 2: ARCHIVING & RESTORING FILES

Part 1: Creating a File Archive

One method to help prevent running out of hard disk space is to compress or **archive** files so they take up less space on your computer system. Archiving files is also a popular method to **bundle files together** into one smaller file for fast transfers to other computer servers.

Perform the following steps:

1. Boot up your **centos3** VM.

2. Change your working directory to **/usr/share/doc/sudo***
3. Use the tar (tape archiver) command to create an archive file named "/tmp/archive1.tar" by issuing the following command:
   ```
   tar cvf /tmp/archive1.tar .
   ```

> **Warning!**
> Don't miss the `.` at the end of the `tar` command (where indicated). It specifies what should go into the archive: the contents of the current directory.

4. What do the options **c**, **v**, and **f** mean?
5. Record the archive file size.
6. Compress the file using **gzip** by issuing the command: `gzip /tmp/archive1.tar`
7. Record the archive file size after compression.
8. Make sure you're still in **/usr/share/doc/sudo*** and then create a compressed archive by issuing the following command:
   ```
   tar cvzf /tmp/archive2.tar.gz .
   ```
9. What does the **z** option do?
10. Compare the sizes of **/tmp/archive1.tar.gz** and **/tmp/archive2.tar.gz**. Why are they so close in size?

Part 2: Restoring Files From an Archive

Perform the following steps:

1. Remain in your **centos3** VM.
2. Create the directory **/tmp/extract1**
3. Change to the **/tmp/extract1** directory.
4. Move the file **archive1.tar.gz** to your current directory.
5. Unzip the first archive you created by issuing the command: `gunzip archive1.tar.gz`
6. Extract the files from the first archive by issuing the command: `tar xvf archive1.tar`
7. Are all the files there?
8. Compare **/tmp/extract1/README** and **/usr/share/doc/sudo*/README**. Are they exactly the same? Why?
9. Create the directory **/tmp/extract2**
10. Move the file **archive2.tar.gz** to the **/tmp/extract2** directory.
11. Extract the files from the second archive by issuing the following command: `tar xvzf /tmp/extract2/archive2.tar.gz`
12. Note that this time a separate `gunzip` command was not needed. Why?
13. Repeat the previous command, leaving out the option **z**. Does it work? Why?
14. Compare the **README** file in this directory with the original file. Are they exactly the same?

Answer Investigation 2 observations / questions in your lab log book.

INVESTIGATION 3: MANAGING SOFTWARE PACKAGES

Part 1: Managing Software and Repositories with Yum

We will learn how to install software packages with the **yum** utility. This command is useful for installing software since it automatically resolves software dependencies prior to installing the software. Upon your Centos OS install, a link to a software repository is automatically made to allow for easy software updates.

Internet Connection
In order for the **yum install** command to work you require a connection to the Internet.

Unfortunately, there is no guarantee that a particular software package is contained on the default repository. In order to overcome this problem, you can add other repositories that may contain the application that you wish to install.

Perform the following steps:

1. Make certain that you are in your **c7host** machine.
2. Issue the command: `sudo yum install elinks`
3. Now issue the command: `yum info elinks`
4. How can you tell if the elinks package has been installed?
5. To remove the elinks package issue the command: `sudo yum remove elinks`
6. Verify that the elinks package has been removed. Also verify that the application called: **xchat** is not installed.
7. Use the wget command to download the xchat rpm binary from the following URL:

   `http://mirror.centos.org/centos/7/os/x86_64/Packages/xchat-2.8.8-25.el7.x86_64.rpm`
   What do you think is the purpose of the numbers also contained in the filename?

8. Issue the command: `sudo yum localinstall xchat-2.8.8-25.el7.x86_64.rpm`
9. Verify that the **xchat** command has been installed.
10. Run the xchat application.

11. We will now look at how we can add different repositories to our **c7host** machine.
12. Issue the following command: `yum repolist`
13. Take a few moments to review the output. Do you see which repositories are used by the **yum** command? Write down the repositories in your lab logbook.
14. View the following link to see a general listing of repositories:
    [Available Repositories for Centos (http://wiki.centos.org/AdditionalResources/Repositories)].
15. To add the epel repository, issue the command:
    `sudo yum install epel-release`
16. To verify that you have added the repository, you can issue the command: `yum repolist`
17. Unused and unneeded software can present a security risk and ties up disk space needlessly. Find **at least 4 other packages** to remove on your c7host machine(for

IRCs: A Great Tool for Linux Administrators
Although Search Engines (like Google) are a system administrator's good friend, Internet Relay Chats (IRCs) are also a great tool for system administrators to help obtain information. Many website offer information on how to connect to IRCs (both nodes (eg. FreeNode) and irc channel (eg. #linux). You can now use your newly-created application called **xchat** to do this.

<u>**A few tips to consider with IRCs**</u>:

example: **sound & video**, **games**, etc) that you're not using on your system. Be careful to ensure that removing those packages does not also remove other software that you do need.

- Do your Homework (read docs first!)
- Ask specific questions
- Not all chats are friendly
- Be patient when asking questions (use courtesy)
- The advice is free (you get what you pay for!)

Part 2: Compiling Source Code from a Downloaded Archived File

Sometimes, some software may not be available on repositories to install with the **yum** command. They may be available from web-pages to download, and compile. Now that you know how to create and decompress "zipped tarball archives", we will demonstrate how to install applications from websites containing these types of archives. Although this method is not as "user-friendly" as using the yum command, this method is useful if the application is NOT contained in regular software repositories...

In order to build software from source code, you must have the appropriate software development tools (such as make and gcc) and libraries (such as GTK) installed. The required tools will vary depending on the computer languages used in the software being built.

Perform the following steps:

1. Remain in your **c7host** VM, but make certain that you are logged in as a regular user (i.e. NOT root!).
2. Issue the command: `which lbreakout2`
   Is there a program called **lbreakout2** on your system?
3. Perform an Internet search for the pattern: `lbreakout2 tar.gz`
4. Go to an appropriate webpage and download a "zipped tarball" (compressed source code) for the **lbreakout2** game (filename should be something like: **lbreakout2-2.6.5.tar.gz**)
5. Change to the directory where you downloaded that file (most likely *~/Downloads*).
6. Use the **tar** command to decompress the "zipped tarball" called something like (not may be exactly): **lbreakout2-2.6.5.tar.gz**.
7. Change to the directory that contains that *lbreakout2* source code.

8. Issue the following command:
   ```
   ./configure && make
   ```
   Most likely, you will get an **ERROR!** What do you think the error is telling you?
9. Issue the following command to install the files for application development by issuing the command:
   ```
   sudo yum groupinstall "Development Tools"
   ```
10. Issue the command in step 7. You should get another ERROR! What dependency is missing?
11. Issue the following command to install the library files for SQL by issuing the command:
    ```
    sudo yum install SDL-devel libpng-devel zlib-devel
    ```

Installing Development Libraries
In the future, remember the above procedure whenever installing software from source. Sometimes, you need to install additional tools or libraries in order to compile a particular software package

Using && and || Conditional Statements
A very useful trick when running several commands in sequence is to use conditional statments. The **&&** conditional will run the subsequent command only if the previous command returns a true (0) exit status, for example:
./configure && make. The **||** conditional will run the subsequent command only if the previous command returns a false (non-zero) exit status,

12. Reissue the `./configure && make` command. Were you successful?
13. Run the command `lbreakout2`. Were you successful?
14. You need to run a command to make this program available regardless of your current directory (as opposed to running command in the directory that contains the program). So, issue the command: `sudo make install`
15. Run the command: `lbreakout2`. Did it work? Issue the command: `which lbreakout2`. What do you think that the **make install** command did?

Part 3: Using sed to Filter Output from Commands

We will continue with using commands to create a Software Information Report that manipulates output generated by the **rpm** command. The **sed** and **awk** commands are very useful tools in shell scripting to manipulate text. In this lab, we will be using *sed* to allow the user to select certain portions from the rpm command (options -qi).

Perform the following steps:

1. Use your **c7host** VM for this section.
2. Open a Bash terminal and run the command:
   ```
   # Use sed with extended regular expressions to only print those matching report elements
   rpm -qi xchat
   ```
   Notice that using the **-i** option with **-q** causes rpm to generate more output than we have seen from it so far.
3. Pick any five packages and run the `rpm -qi` command on all five at once. This should produce dozens of lines of output. Consider the output it would generate if we used -a instead of picking just a few packages.
4. Run `rpm -qia` now.
5. That is way too much output for us to reasonably search through, but we have commands that can filter this down to a much more readable amount.
6. Try the command:
   ```
   rpm -qia | sed -nre '/^Name[[:space:]]+:/ p' -e '/^Install Date[[:space:]]*:/ p'
   ```
   Instead of printing every line, this will only display the name and install date of each package.
7. Try modifying that command to display only the package Name and License.

Answer the Investigation 3 observations / questions in your lab log book.

for example:
./configure || echo "Houston, we have a problem". These conditional can be combined, for example:
./configure && make || echo "Houston, we have a problem"

Bash Shell Tips:

Using sed to Manipulate Text

- **The Linux command** sed **stands for** Streaming Editor **which is an effective way to manipulate a text file, output sent from a command, or from within a "here document". This command can manipulate matching text on a variety of criteria (such as** line number(s), regular expression match**, etc). Commands can then be used for manipulation such as** omitting, printing, substituting, adding, **and** inserting **text.**

- **The sed option** -n **suppresses display of text so the print** (p) **command can be used; otherwise, the text will be displayed (with edits via the sed command instructions).**

- **Results of text manipulation with sed can be stored in a variable using command substitution, or redirected to a file.** NEVER redirect the stdout from a sed command to the same input file (or the input file will be destroyed)!

INVESTIGATION 4: USING LOGICAL STRUCTURES IN PYTHON

Now that we have added some powerful logical structures to our python scripts, we can make our scripts much more adaptable. Our python scripts can now respond to different situations by executing different code, or repeating the same code several times (e.g. keep asking the user for input until they give us something useful). In this investigation you will write a python script that will prompt a user for a directory to archive, ask them if they want the archive compressed, and (only if they do want compression) what type of compression to apply.

- **Examples**

  sed 's/|/ /g' **<<+**
  I|like|weekends!
  +

  sed 's/$/\n/g' **<<+**
  This text
  should be
  double-spaced!
  +

1. In your bin directory, create the file tarchiver.py, and populate with our standard beginning:
   ```
   #!/usr/bin/env python3
   # tarchiver.py
   # Purpose: Creates a tar archive of a directory
   #
   # USAGE: ./tarchiver.py
   #
   # Author: *** INSERT YOUR NAME ***
   # Date: *** CURRENT DATE ***
   import os
   ```
2. Next, add prompts using the **input()** function to ask the user for:
   - The directory they wish to archive.
   - The name they want to give the archive.

   Remember to store their responses in variables.
3. Add a line that uses the values provided by the user to execute a tar command and create and archive of the directory they requested.
4. Test your script to make sure it works.
5. Add a third prompt (immediately after the other two prompts and before the tar command), asking the user if they want the archive to be compressed. You'll need an if statement to run your tar command with gzip compression if they answered yes.
6. Test your script to make sure it works.
7. Add a fourth prompt asking the user what type of compression they want (present them with gzip, bzip2, and xzip as options). Note that this prompt should only be shown if the user opted for compression (if they don't want compression, there is no point asking what type they don't want).
8. Expand the if statement around your actual tar command to use whichever type of compression the user asked for.
9. Test your script to make sure it works.
10. Test your script again, but feed it nonsense data (e.g. answer 'very much so' instead of 'y' or 'n' when prompted about compression).

11. Add loops around your prompts for whether or not the user wants compression, and the compression type to make the script repeat each prompt until the user gives a response your script can actually use. Don't worry about doing anything with the archive name or path.
12. Now test your script again, with good data and with nonsense.
13. When you are confident your script works, you are ready to submit the lab.

LAB 3 SIGN-OFF (SHOW INSTRUCTOR)

Follow the submission instructions for lab 3 on Blackboard.

> **Time for a new backup!**
> If you have successfully completed this lab, make a new backup of your virtual machines as well as your host machine.

Perform the Following Steps:

1. Make certain that your **c7host** and **centos3** VMs are running.
2. Switch to your **c7host** machine, open a shell terminal as a regular user, and change to the directory: **~/bin**
3. Download the Lab 3 check script: `wget https://raw.githubusercontent.com/OPS245/labs/main/lab3-check.bash`
4. Give the **lab3-check.bash** file execute permissions (for the file owner).
5. Run the shell script with `sudo`, and if any warnings appear, fix the issues noted by the script and rerun the check until you receive the "Congratulations!" message.
6. Arrange proof of the following on the screen:

   ✓ **centos3**:

   - Archived files **created**
   - Archive files **restored**

   ✓ **c7host**:

   - **One repository added** for yum
   - Run the **lab3-check.bash** script (must have all `OK` messages)

   ✓ **Lab 3** logbook pages filled out.
7. Upload a screen of the proof from the previous step, along with the file generated by **lab3-check.bash**, your log book, and your **tarchiver.py** script to blackboard.

Practice For Quizzes, Tests, Midterm & Final Exam

1. What is the purpose of booting into single-user mode?
2. List the steps in order to boot into single-user mode.
3. List the steps to reset a forgotten root password.
4. What is the difference between a .tgz file and a .tar.gz file? What do these stand for?
5. What is the purpose of a repository?
6. What is source code?
7. How do you build software from source code?
8. What does yum do that compiled source code does not?
9. Other than running an installed package, what commands can you issue to verify that the package was installed?
10. List the steps to remove a package via yum command.
11. List the steps to install a package using the yum command.
12. List the steps to add a repository to be used with yum.
13. What is the command to list all repositories associated with the yum command?
14. What is the command to view all repositories that have been added to-date?
15. How do you use the **sed** command to modify text?
16. What is the **Here-Document**, and how can it be used in issuing command and shell scripting?

Retrieved from "https://wiki.cdot.senecacollege.ca/w/index.php?title=OPS245_Lab_3&oldid=158471"