I. Lab Objectives

This Lab work will introduce students to the fundamentals and primary components of modern operating systems and corresponding design trade-offs. The emphasize will be on the organization of the Linux-based operating systems and their responsibilities using examples and lab experiments.

II. Lab Outcomes

At the end of this lab work the students will have sufficient knowledge to describe the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems. Discuss the operating system concepts with its functionalities along with the various file and disk management strategies. Apply various CPU scheduling algorithms for problems.

III. Text Book:

- Silberschatz, Abraham. Operating System Concepts, 9th Edition. Austin, John Wiley and Sons, 2013. http://www.cs.nthu.edu.tw/~ychung/slides/CSC3150/Abraham-Silberschatz-OperatingSystem-Concepts---9th2012.12.pdf

IV. Reference Books:

- Stallings, William. Operating Systems: Internals and Design Principles, 7th Edition. US, Prentice Hall, 2009. https://repository.dinus.ac.id/docs/ajar/Operating_System.pdf

- Tanenbaum, Andrew. Modern Operating Systems, 3rd Edition. UK, Prentice Hall, 2008. http://stst.elia.pub.ro/news/SO/Modern%20Operating%20System%20-%20Tanenbaum.pdf

Docker Deep Dive V4, Nigel Poulton, 2017
https://drive.google.com/file/d/142rH04LjnqR2ffFEzbRSJicMpvpZ_29V/view

Lab # 1

Introduction to Linux operation system

I. THEORY

This is an introduction to the major features of Linux that helps to get acquainted with them. It does not go into detail or cover any advanced topics, as is done in later lab exercises. Instead it is intended to give a head start in understanding what Linux is, what Linux offers, and what is needed to run it. Linux is the clone of UNIIX operating system that runs on machines with an Intel 80386 processor or better, as well as Intel-compatible CPUs. such as AMD and Cyrix.

Linux is not UNIX, as UNIX is a copyrighted piece of software that demands license fees when any part of its source code is used. Linux was written from scratch to avoid license fees entirely though the operation of the Linux operating system is based entirely on UNIX. It shares UNIX's command set and look-and-feel, so if you know either UNIX or Linux, you know the other too.

Linux supports a wide range of software, from TeX(a text formatting language) to (a graphical user interface) to the GNU C/C++ compilers to TCP/IP networking.

II. Major Features of Linux

- Multiuser: Multiple users can run Linux side by side.
- Multitasking: Run spreadsheet, Email, browse internet on the same time.
- Wide variety of compatible software.
- Ability to run on older hardware save upgrade cost.
- Easy to learn, similar features to Windows.

III. Major Distributions

- Top Ten Distributions are:

1. Ubuntu
2. Fedora
3. OpenSUSE
4. Debian
5. Mandriva
6. Slackware
7. Getoo
8. CentOS

- In the Enterprise

1. Red Hat Enterprise Linux
2. Novell SUSE Linux Enterprise

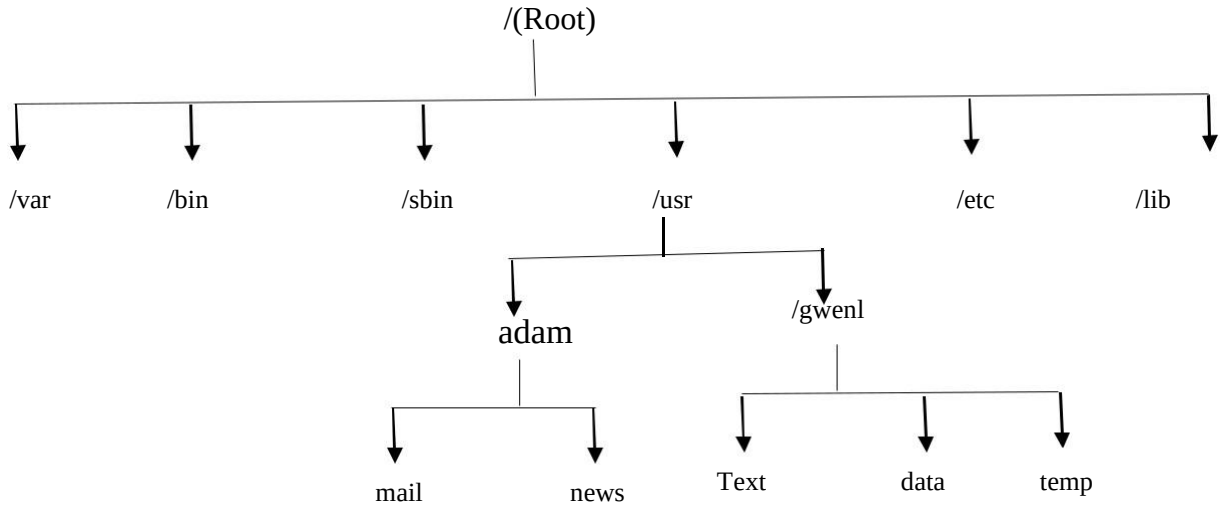
IV. What comes with Linux OS

- Every Linux distributions comes with basic required softwares like Photo Viewer, Video player, Document Editor, Text Editor, Web Browser, Games etc.
- Many Linux distributions come with programming languages installed like PHP, Java, C, C++ etc.

V. Linux File System

Under Linux, the file space that is visible to users is based on an inverted tree structure, with the root at the top. The various directories and files in this space branch downwards from the root. The top directory '/' is known as the Root directory.

The Linux operating system is made up of several directories and many different files. Depending on how you selected your installation, these directories have different file systems. Typically most of the operating systems resides on two file systems: the root file system known as '/' and a file system mounted under/user.



1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.

2. /bin – User Binaries

- Contains binary executables.
- Common linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here. For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

Just like /bin, /sbin also contains binary executables.
But, the linux commands located under this directory are used typically by system administrator, for system maintenance purpose. For example: iptables, reboot, fdisk, ifconfig, swapon

4. . /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. . /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. . /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process.
- For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. . /var – Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.
- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. . /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs.
- /usr/sbin contains binary files for system administrators
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

- Home directories for all users to store their personal files.
- For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

Contains boot loader related files.

Some of the frequently used Linux commands are : su , pwd , cd , ls , more , less , find , grep and man.

VI. Install Linux on a virtual machine

Required Resources

- Computer with a minimum of 2 GB of RAM and 10 GB of free disk space
- High-speed Internet access to download Oracle VirtualBox and Linux OS image, such as Ubuntu Desktop

Instructions

Part 1: Prepare a Computer for Virtualization

In Part 1, you will download and install desktop virtualization software and a Linux OS image. Your instructor may provide you with a Linux OS image.

Step 1: Download and install VirtualBox.

VMware Player and Oracle VirtualBox are two virtualization programs that you can download and install to support the OS image file. In this lab, you will use the VirtualBox application.

a. Navigate to https://www.virtualbox.org/. Click the download link on this page.

b. Choose and download the appropriate installation file based on your operating system.

c. After the VirtualBox installation file is downloaded, run the installer and accept the default installation settings.

Step 2: Download a Linux Image.

d. Navigate to the Ubuntu website at http://www.ubuntu.com. Click the Download link on this page to download and save an Ubuntu Desktop image.

Step 3: Create a New Virtual Machine.

a. Click **Start** and search for **Virtualbox**. Click **Oracle VM VirtualBox** to open the manager. When the manager opens, click **New** to start the Ubuntu installation.

b. In the **Name and operating system** screen, type **Ubuntu** in the **Name** field. For the **Type** field, select **Linux**. In the **Version** field, select the corresponding downloaded version. Click **Next** to continue.

c. In the **Memory size** screen, increase the amount of RAM as long as the amount of RAM for the virtual machine is in the green area. Going beyond the green area would adversely affect the performance of the host. Click **Next** to continue.

d. In the **Hard disk** screen, click **Create** to create a virtual hard disk now.

e. In the **Hard disk file type** screen, use the default file type settings of **VDI (VirtualBox Disk Image)**. Click **Next** to continue.

f. In the **Storage on physical hard disk** screen, use the default storage settings of **dynamically allocated**. Click **Next** to continue.

g. In the **File location and size** screen, you can adjust the hard drive and change the name and location of the virtual hard drive. Click **Create** to use the default settings.

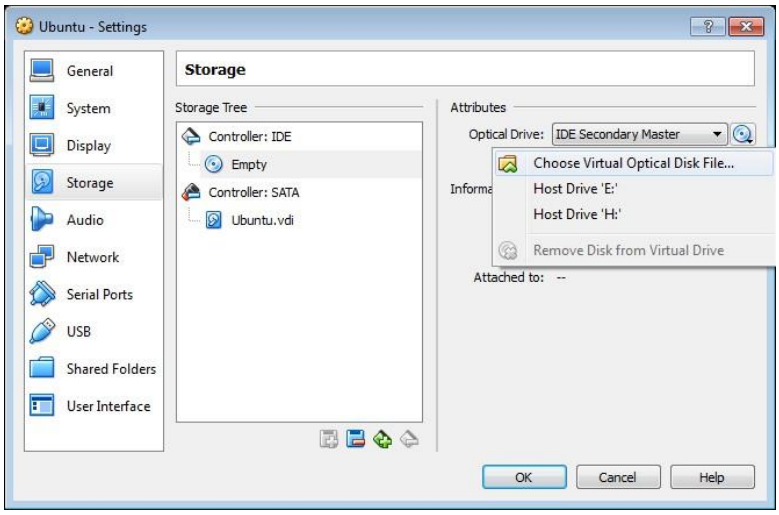
h. When the hard drive creation is done, the new virtual machine is listed in the **Oracle VM VirtualBox Manager** window. Select **Ubuntu** and click **Start** in the top menu.

Part 2: Install Ubuntu on the virtual machine

Step 1: Mount the Image.

a. In the **Oracle VM Virtualbox Manager** window. Right-click **Ubuntu** and select **Settings**. In the **Ubuntu – Settings** window, click **Storage** in the left pane. Click **Empty** in the middle pane. In the right pane, click the CD symbol and select the file location of the Ubuntu image. Click **OK** to continue.

b. In the **Oracle VM VirtualBox Manager** window, click **Start** in the top menu.



Step 2: Install the OS.

a. In the **Welcome** screen, you are prompted to try or install Ubuntu. The try option does not install the OS, it runs the OS straight from the image. In this lab, you will install the Ubuntu OS in this virtual machine. Click **Install Ubuntu**.

Follow the on-screen instructions and provide the necessary information when prompted.

Note: If you are not connected to the Internet, you can continue to install and enable the network later.

b. Because this Ubuntu installation is in a virtual machine, it is safe to erase the disk and install Ubuntu without affecting the host computer. Select **Erase disk and install Ubuntu**. Otherwise installing Ubuntu on a physical computer would erase all data on the disk and replace the existing operating system with Ubuntu. Click **Install Now** to start the installation.

c. Click **Continue** to erase the disk and install Ubuntu.

d. In the **Who are you?** screen, provide your name and choose a password. Use **iteuser** for **Your Name** and **ITEpass!** for the password. You can use the username generated or enter a different username. If desired, you can change the other settings. Click **Continue**.

e. The Ubuntu OS is now installing in the virtual machine. This will take several minutes. When the **Installation is complete** message displays, return to the **Oracle VM Virtualbox Manager** window. Right- click **Ubuntu** and select **Settings**. In the **Ubuntu – Settings** window, click **Storage** in the left pane. Click the mounted Ubuntu image in the middle pane. In the right pane, click the CD symbol and click **Remove Disk from Virtual Drive**. Click **OK** to continue.

f. In the Ubuntu VM, click **Restart Now**.

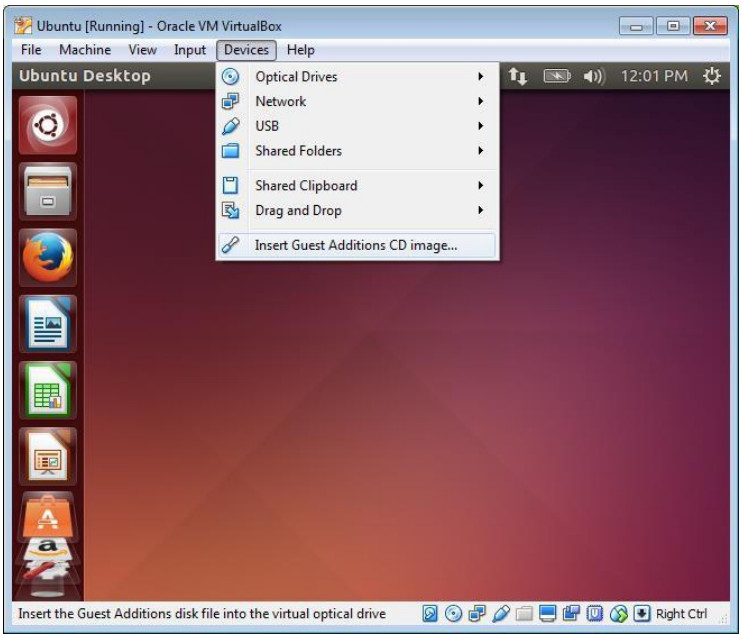
Part 3: Explore the GUI

In this part, you will install the VirtualBox guest additions and explore the Ubuntu GUI.

Step 1: Install Guest Additions.

a. Log on to your Ubuntu virtual machine using the user credentials created in the previous part.

b. Your Ubuntu Desktop window may be smaller than expected. This is especially true on high-resolution displays. Click **Device > Insert Guest Additions CD image…** to install the Guest Additions. This allows more functions, such as changing the screen resolution in the virtual machine.



c. Click **Run** to install the additions. When prompted for a password, use the same password that you used to log on. Click **Authenticate** to continue.

d. If the computer was not connected to the Internet during the installation, click **Devices > Network Settings** in the Oracle VirtualBox menu. Enable network

adapters and configure the proper setting for network connections as necessary. Click **OK.**

e. When the installation of the additions is done, restart the virtual machine again. Click the menu in the upper-right corner and click **Shut down**. Click **Restart** to restart Ubuntu.

Step 2: Open a web browser.

a. Log on to Ubuntu again. After you are logged on again, you can resize the virtual machine window.

b. Open a web browser. Depending on the Linux distribution, you may need to search for a web browser or there is a link to a web browser already on the Desktop.

c. Locate a terminal emulator to access the command line interface. You will be using a terminal emulator in later labs.

d. Explore the installed Linux distribution and locate a few applications that you may use.

VII. Executing a Linux commands

1. pwd

Use the pwd command to print the working directory (the current directory you are in).

Example

$ pwd
/home/john

2. cd

You can change your current directory with the cd command (Change Directory).

Example

$ cd /etc
$ pwd /etc

3. cd ..

To go to the parent directory (the one just above your current directory in the directory tree), type cd .. .

Example

cd ..
$ pwd /usr/share/games
$ cd ..
$ pwd /usr/share

4. cd –

Another useful shortcut with cd is to just type cd - to go to the previous directory.

Example

```
$ pwd
/home/bilal
$ cd /etc
$ pwd
/etc
$ cd –
/home/bilal
$ cd –
/etc
```

5. ls

You can list the contents of a directory with ls.

Example

```
$ ls
allfiles.txt dmesg.txt  httpd.conf stuff summer.txt
```

6. ls –a

A frequently used option with ls is -a to show all files. Showing all files means including the hidden files.

Example

- $ ls
  allfiles.txt dmesg.txt httpd.conf stuff summer.txt
- $ ls -a
  allfiles.txt .bash_profile dmesg.txt .lesshst stuff .bash_history .bashrc httpd.conf .ssh summer.txt

7. ls –l

Many times you will be using options with ls to display the contents of the directory in different formats or to display different parts of the directory. Typing just ls gives you a list of files in the directory. Typing ls -l (that is a letter L, not the number 1) gives you a long listing.

Example

```
$ ls -l total
23992
-rw-r--r-- 1 bilal bilal 24506857 2006-03-30 22:53 allfiles.txt
-rw-r--r-- 1 bilal bilal 14744 2006-09-27 11:45 dmesg.txt -rw-r--r-
- 1 bilal bilal  8189 2006-03-31 14:01 httpd.conf drwxr-xr-x 2
bilal bilal 4096 2007-01-08 12:22 stuff -rw-r--r-- 1 bilal bilal
        0 2006-03-30 22:45 summer.txt
```

8. ls –lh

Another frequently used ls option is -h. It shows the numbers (file sizes) in a more human readable format.

Eample

$ ls -lh total
24M
-rw-r--r-- 1 bilal bilal 24M 2006-03-30 22:53 allfiles.txt
-rw-r--r-- 1 bilal bilal 15K 2006-09-27 11:45 dmesg.txt
-rw-r--r-- 1 bilal bilal 8.0K 2006-03-31 14:01 httpd.conf drwxr-xr-x 2 bilal bilal 4.0K 2007-01-08 12:22 stuff -rw-r--r-- 1 bilal bilal 0 2006-03-30 22:45 summer.txt

VII. Creating a Directory

1. mkdir p

To create a new directory, use the mkdir command. The syntax is mkdir <name>, where <name> is replaced by whatever you want the directory to be called. This creates a subdirectory with the specified name in your current directory:

Example

$ mkdir MyDir
$ cd MyDir
$ls

2. mkdir -p

When given the option -p, then mkdir will create parent directories as needed.

Example

$ mkdir -p MyDir2/MySubdir2/ThreeDeep
ls MyDir2
MySubdir2
$ ls MySubdir2
ThreeDeep

VIII. Removing Directories

1. rmdir

When a directory is empty, you can use rmdir to remove the directory. Before you can remove a directory, it must be empty (the directory can't hold any files or subdirectories).

Otherwise, you see rmdir: <directory>:
Directory not empty

2. rmdir –p

And similar to the mkdir -p option, you can also use rmdir to recursively remove directories.

Eample

$ mkdir -p dir/subdir/subdir2
$ rmdir -p dir/subdir/subdir2

Tasks

1. List all files in the current directory and their sizes: use multiple columns and mark special files.

2. List the contents of directories /bin and /etc.

3. Count all files in the current directory.

4. Display your current directory.

5. Change to the /etc directory.

6. Go to the parent directory of the current directory. Go to the root directory.

7. List the contents of the root directory.

8. Stay where you are, and list the contents of ~

9. List all the files (including hidden files) in your home directory.

10. List the files in /boot in a human readable format.

11. Create a directory with your name in your home directory.

12. Create in one command the directories ~/dir1/dir2/dir3 (dir3 is a subdirectory from dir2, and dir2 is a subdirectory from dir1).

13. Remove the directory created with your name.