Lab 00 - Introduction to UNIX-Like Operating Systems

Objectives

1. Learn about UNIX-like operating systems.
2. Learn how to run the terminal program.
3. Learn a few useful commands.
4. Learn how to write a basic bash script.
5. Install Ubuntu on your computer at home.

Pre-reading

The following material contains commands and/or other information that could be essential to completing this and future labs. Students are strongly encouraged to take a look at the material below *before* coming to the lab, since knowing this material before you start will likely make the task of completing the lab much faster and easier.

Web Links

Ubuntu: http://www.ubuntu.com
VirtualBox: http://www.virtualbox.org

UNIX-Like Operating Systems

The UNIX operating system was originally developed in the 1970s by Ken Thompson and Dennis Ritchie at AT&T Bell Laboratories. Since then, many operating systems have been developed that are based on the original UNIX design. These are generally referred to as **UNIX-like** operating systems.

Linux-based operating systems are a particular family of UNIX-like operating systems that are based on the Linux kernel. The Linux kernel was originally developed by Linus Torvalds in the 1990s. Contrary to what many people believe, Linux is *not* an operating system! Linux is a *kernel*, and operating systems that use this kernel are called *Linux-based*.

The operating system we'll be using in this course is **Ubuntu**. Ubuntu is an open source Linux-based operating system. It has been installed on all the computers in the lab, but we strongly recommend you install it on your own computer as well (preferably using virtual machine software, such as VirtualBox).

The Terminal Emulator

Most modern UNIX-like operating systems have a Graphical User Interface similar to Windows. In this course, however, we're more interested in using a command line interface instead of a GUI. Luckily, most of these operating systems come with an emulator program that emulates a traditional UNIX terminal (i.e. command line) interface. Search for a program called **Terminal** to get started.

The terminal program only emulates the traditional terminal hardware. The actual command prompt you see *within* the terminal window is generated by your default shell program. A **shell** program is a command interpreter. That is, it accepts your commands and executes them. This program is run when you first start the terminal. The default shell in Ubuntu is usually bash. There are several other popular shells as well, including sh, csh, and korn, each offering a different syntax and range of functionality. In this course, we'll assume you're using the bash shell.

A Short List of Common Commands

The most important command on any UNIX-like operating system is the **`man`** command. Man is short for manual. When you type it in, followed by a command that you don't understand, it will bring up the manual page for that command. For example

```
$ man ls
```

will bring up the manual page for the **`ls`** command. When you are viewing a man page, you may press the up/down arrow keys to scroll, and the q key to quit.

The following is a short list of some commonly used commands:

date	view or change the time/date
cal	view a calendar
which	locate the actual file that is executed by a command
cd	change directories
ls	list the files in a directory
cat	print the contents of a file to the screen
echo	print some text to the screen
pwd	prints the current directory

Bash Shell Scripting

In this course, you'll primarily be writing programs in C. However, writing a bash script is almost as powerful as programming in C, and often far easier. There are some major differences however. Shell scripts are not compiled; they are interpreted by the shell program when executed (like running a batch (.bat) file in Windows).

The following is a simple bash script:

```
#!/bin/bash
#example script
#Author: your lab TA
echo “hello world!”
exit 0
```

In this script, obviously the lines starting with # are comments. The exception to this is the shebang statement, the line starting with #*!*. This statement specifies what program should be used to execute this script file (in this case, it's bash).

Each non-comment line is interpreted by the shell program and executed in sequence. You have seen the echo command before, but not exit. Exit ends the script and its argument is the return status of the script. By convention, zero is success and non-zero is failure.

When you want to run a shell script file, use the bash command. For example,

```
$ bash myscript.sh
```

Installing Ubuntu at Home

We strongly recommended you install the Ubuntu operating system on your home computer. But, rather than erasing or partitioning your hard drive, we suggest a much easier and risk-free alternative: **VirtualBox**.

VirtualBox is a (free) program you can use for running any operating system of your choice within a virtual machine. The "guest" operating system (in this case, Ubuntu) runs within a window on your "host" operating system.

The links to both the Ubuntu and VirtualBox websites are listed at the start of this lab.

There are many instructions online (with pictures) for installing Ubuntu with VirtualBox. You can use those, or you can follow the instructions below.

1. Download and install VirtualBox.
2. Run VirtualBox and click on the "New" button at the top of the VirtualBox Manager window. A wizard will pop up to guide you through setting up a new virtual machine (VM). Supply the following information:
   - VM Name and OS Type:
     - Name: Ubuntu.
     - Operating System: Linux.
     - Version: Ubuntu.
   - Memory:
     - RAM Memory Size: 1024 MB (or more).
   - Virtual Hard Disk:
     - Select "create new hard disk" and click next. A wizard will pop up. Supply the following information:
       - File Type: VDI.
       - Storage details: dynamically allocated.
       - Size: 15 GB.
3. Select your new virtual machine from the left side of the manager window and press start. The first start wizard should pop up. Insert the Ubuntu install disk and select it as your boot media (a burnable image is available on the Ubuntu website). The Ubuntu installation should begin.
4. Follow the instructions presented to you during the Ubuntu installation. The installation process is straightforward.

Procedure

Install Ubuntu on a VirtualBox virtual machine on your computer at home.

Review Questions

1. **(10 marks)**
   Enter the command
   ```
   $ cat /proc/version
   ```
   into a terminal session. What version of the Linux kernel is your system using?

For the next two questions, give *only* the commands you would use to achieve the goal.

2. **(10 marks)**
   Determine where the `mkdir` command is located.

3. **(10 marks)**
   List all of the files in your current directory alphabetically.

4. **(10 marks)**
   Where is your home directory (that is, the default directory when the terminal program starts up)?

5. **(10 marks)**
   What is the most important command? Why?

Deliverables

1. **(50 marks)**
   Answers to the above five review questions in a file called **answers.txt**

2. **(50 marks)**
   Proof that you've successfully installed Ubuntu at home. There are two ways to prove this:
   - Bring the computer in and show the TA.
   - Take a picture of you next to the computer screen running Ubuntu in VirtualBox. The picture should clearly show you next to the computer screen, with Ubuntu running in a VirtualBox window in the background, and the *About VirtualBox* dialog box (*Help Menu → About VirtualBox*) in the foreground.