CST-321 Activity 1 Guide

Contents

Tools Installation

Overview

In this activity, students will install the Linux Virtual Machine and validate the C programming tools with a simple "Hello World" webpage.

Execution

Execute this assignment according to the following guidelines:

1. Install the Linux Ubuntu Desktop Virtual Machine:
    a. Download and install VirtualBox from here for your operating system. The instructions for this activity are written for VirtualBox. You also have the option to install VMWare from your Microsoft CSET on the Hub account. You can refer to Appendix B if you cannot get VirtualBox to work or choose to install VMware rather than VirtualBox. One other option to try if you cannot get VirtualBox to work is to try an older version of VirtualBox, which can be downloaded from the "VirtualBox older builds" section of the VirtualBox downloads page.
    b. Download Ubuntu Desktop ISO (or download the 64-bit Ubuntu Desktop iso from here). If you are on Windows, you will need to make sure you have virtualization enabled in your BIOS in order for a 64-bit Guest OS to be installed and also make sure to enable Hyper-V in Windows 10 settings if you are on Windows 10 Pro edition and use this link if needed.
    c. Install the Linux Ubuntu desktop operating system:
        - Start VirtualBox.

- Click the *New* icon, give your VM a name (like Ubuntu Desktop), select *Linux* type, select *Ubuntu 64-bit*, setup your VM configuration as noted below, and click *Continue or Next*. **Note**: if VirtualBox contains an option to Skip Unattended Installation mode, this mode should be selected so you have complete control over the default User creation. Your VM configuration should be set up as follows:
    i. Select the Ubuntu Desktop ISO file that was downloaded from the step 1b.
    ii. Increase your Memory size to at least 4GB (or half of your host operating system).
    iii. Make sure you are running at least 2 virtual CPUs in the VM.
    iv. Select *Create a virtual hard disk now* option, increase your disk size to 12 GB, select *VDI hard disk* type, and then select *Dynamic*.
- Click the *Start* icon.
    i. **Note**: if your guest screen size display ends up being very small, shut your VM down and adjust the Scale Factor to 200% by going to the Settings and selecting the Display icon.
- Complete the Ubuntu Desktop installation (select the 'Install Ubuntu' from the first install screen and don't install updates from the second screen).
    i. **Important**: for the 'Who are you' screen, make sure you remember your username and password and do not log in automatically.
- The install will take about 3–5 minutes.
- Restart Ubuntu (leave the ISO image mounted if prompted) and when prompted, log in.
- Once Ubuntu Desktop starts up, log in, and then wait for the Software Updater to finish updating the operating system. This will take 3–5 minutes to complete.
- When prompted, click the *Restart Now* button and then log back into the Ubuntu Desktop.
- Adjust the display resolution and location of the Dock as desired by selecting the *Settings* menu option from the System menu located in the top right corner of the screen.

2. Install the VirtualBox Guest Additions (this is mandatory for Folder Sharing to work):
    a. Install Guest Additions:
        - Start the Ubuntu virtual machine and log in.
        - Select the *Devices > Insert Guest Additions CD image...* menu from the VirtualBox VM application. Click the *Run* button and enter your password when prompted.
        - The installation will take about 1–2 minutes.
        - Shutdown Ubuntu by clicking the *Gear* (on the top right) and selecting the *Shut Down* option.
    b. Set Up Shared Folders:

- Start the Ubuntu virtual machine but do not log in.
- Select the *Settings* > *Shared Folders* menu option.
- Click the Add icon:
    i. Set Folder Path from Windows/Mac for your folder to share.
    ii. Set Folder Name to *MySharedFolder* for your folder in Ubuntu.
    iii. Select *Auto-mount* and *Make Permanent* options.
    iv. Click the *OK* button to dismiss all prompts.
- Start the Ubuntu virtual machine and log into Ubuntu.
- Access your Share Folder by opening the *Files* application. Your Shared Folder will be listed under *Computer* and in the *media* folder.

c. Share Files:
- Start the Ubuntu virtual machine and log in.
- Run the *Files* application.
- You will see a *sf_[Your Shared Folder Name]* mounted under *Computer* in the *media* folder.
- First, you need to set permissions on this folder so you can use it.
    i. Right-click on your Shared Folder and select Properties; notice who owns and has access to this folder.
    ii. Click the *Show Application* icon from the Launcher/Favorites bar. Search for terminal. Select the *Terminal* application. To pin the *Terminal* application to the favorites in your Launcher. Right-click on the *Terminal* application icon and select the *Add to Favorites* menu option.
    iii. Open terminal window and run the following:

        sudo usermod -aG vboxsf <yourusername>

    iv. This will add you to the vboxsf Group, which has access to create and delete files from your Shared Folder.
    v. Log off and log back into Ubuntu.
- Test Your Shared Folder:
    i. Click the *Show Application* icon from the Launcher/Favorites bar and search for Text. There is a simple text editor called *Text Editor*. Select it.
    ii. Add some text to the blank document.
    iii. Create folder on your desktop (right-click on Desktop and select *New Folder*) called *Assignments*.
    iv. Save your text file to the *Assignments* folder as a file named *Test.txt*.
    v. Open the *Assignments* folder and drag/drop the *Test.txt* file to your Shared Folder.
    vi. Go to your Host OS (Windows/Mac) and validate you can view and edit/update this file to and from your Guest OS (Ubuntu).

d. Optional Applications (use the Ubuntu Software application to install):
- Install Atom, which is a programmer's editor.

- Install GitKraken, which is a GIT Client.
- Pin these as favorites to your Dock as desired.

e. Take a screenshot of the Ubuntu Desktop running in your virtual machine.

C Compiler Validation

Overview

In this activity, you will program your first "Hello World" C program.

Execution

Execute this assignment according to the following guidelines:

1. Validate the C compiler is installed:
    a. Start the Ubuntu virtual machine.
    b. Open up the *Terminal* application.
    c. Type *gcc* in the terminal prompt. You might see an error message indicating that 'gcc' is currently not installed.
    d. If 'gcc' is not installed, run the following commands from the terminal prompt:
        sudo apt-get update
        sudo apt-get install gcc
    e. Type *gcc* again in the terminal prompt. You should see that 'gcc' ran but is looking for input files to compile. This is OK.
2. Create your first C program:
    a. Open up the Text Editor application.
    b. Type in the following C program:

```
/* Hello World program */
#include<stdio.h>
main()
{
    printf("Hello World\n");
}
```

    c. Save the file as *HelloWorld.c* into your Applications folder created on your desktop.
    d. Compile your program:
        i. Open the *Terminal* application.
        ii. Enter the following command:
            gcc HelloWorld.c -oHelloWorld.out
        iii. Run a list command (enter *ls*) from the terminal prompt. You should now see the compiled binary output called *HelloWorld.out.*
        iv. Enter the following in the terminal prompt:
            ./HelloWorld.out
        v. This should print the "Hello World" message to the terminal/console window. Take a screenshot of the Terminal and Console window output.
        vi. Edit the *HelloWorld.c* file by changing the printf string to "*Hello World.\n I now know how to compile and run a C program!*" then recompile the program

and run the program. What difference do you see?

More C Programming Practice

Overview

In this activity, you will program your second C program.

Execution

Execute this assignment according to the following guidelines:

1. Read the following videos and tutorials found in Class Resources:
    a. View the video "Getting Started Programing C."
    b. View the video playlist "Learning C."
    c. Download the Intro to C Programming training deck and study Modules 1 and 2.
2. Create a new C program that supports the following capabilities:
    a. Variable declarations (you need to research variables, declarations, and data types).
    b. Getting input from a user (you need to research the scanf() function).
    c. Operators (you need to research the operators available in C).
    d. Printing formatted output (you need to research the scanf() function).
    e. Run your new C program.
    f. Take a screenshot of the Terminal and Console window output.
3. Some additional programming tools you may want to explore include a real programming editor and a GIT Client. Both can be installed in Ubuntu Desktop using the command line interface and the Terminal application. You are also free to explore and leverage other programming editors. Other popular programming editors include, but are not limited to, VI, Sublime, and Visual Studio Code for Linux (see this link for more information, https://code.visualstudio.com/docs/setup/linux).
    a. GitKraken GIT Client:
        wget https://release.gitkraken.com/linux/gitkraken-amd64.deb
        sudo dpkg -i gitkraken-amd64.deb
    b. Atom Code Editor:
        sudo snap install atom --classic

Getting to Know the Terminal Windows

Overview

In this activity, you will learn some common Linux terminal commands.

Execution

Execute this assignment according to the following guidelines:

1. Read the following tutorials and video found in Tutorials Point and Class Resources:
    a. Go to https://www.tutorialspoint.com/unix/.
    b. Complete the following tutorials:

        i. Home, Getting started, File Management, Directories, Permission, and Useful Commands.
    c. Reference the video playlist "Linux Command Line Playlist" in Class Resources as needed.
2. Practice the following Linux commands (these will also work on your Mac, but you should practice these on your Ubuntu Desktop, so you get familiar with that desktop environment). You can get help on any of these commands by accessing the built-in manual for any command by entering: man [cmd]. Take a screenshot of the Terminal and Console output demonstrating the use of each of the following commands:
    a. List Command: ls, ls -a, ls -all, ls -l
    b. Change Directory: cd, cd .., cd ~, pwd
    c. Make or Remove (Empty) Directory: mkdir, rmdir
    d. Copy or Move or Remove Files: cp, mv, rm, rm –r
    e. Find Files: find –iname [filename to find]
    f. Find Text in Files: grep "string to search for" [file to search in]
    g. Process: ps, top
3. Practice the following Windows commands so you can compare how the same Windows commands work in the Linux environment. In Windows, you will need to run the commands by opening a DOS/Command window. You can get help on any of these commands by accessing the built-in manual for any command by entering: help [cmd]. This step can be ignored if you have a Mac. Take a screenshot of the Command Window output demonstrating the use of each of the following commands:
    a. Directory Command: dir, dir /AD, dir /O:GSD
    b. Change Directory: cd (displays current directory), cd .., cd ~, pwd
    c. Make or Remove (Empty) Directory: mkdir, rmdir
    d. Copy or Move or Delete Files: copy, xcopy, move, del
    e. Find Files: dir "file to find, use * for wildcards" /s
    f. Find Text in Files: find "string to search for" [file to search in]
    g. Process: tasklist

Bash Shell Scripting – The Basics and Variables

Overview

In this activity, you will learn basics in bash shell scripting by creating a useful C programming compiler script and working with environment variables.

Execution

Execute this assignment according to the following guidelines:

1. Read the following tutorials found in Tutorials Point:
    a. Go to https://www.tutorialspoint.com/unix/.
    b. Complete the following tutorials:
        i. Shell Scripting, What is Shell, Using Variables, and Special Variables.

2. Create a bash script that does the following:
    a. Create a text file 'compile.sh' in the text editor (or Atom or vi editors)
    b. Indicate the Shell/Script type in line 1: #!/bin/sh
    c. You can print text to console by using: echo "My Text" command
    d. Enter the required commands to compile and run your Hello World C program
    e. Exit your Text Editor (make sure to save your work)
    f. Run your script 'compile.sh' from a Terminal session by entering: ./compile.sh
        i. What happened?
    g. Give the script permission using: chmod 777 test.sh command and run the script.
    h. Take a screenshot of the Terminal and Console window output.
3. Remove the hard code file names and use a bash script argument:
    a. Open the 'compile.sh' script in the text editor (or Atom or vi editors)
    b. Save the program to compile script that is passed as a script argument in a variable ($1 will be your first bash argument).
    c. Substitute the variable where you currently have the 'HelloWorld' program name text
    d. Test your script.
    e. Take a screenshot of the Terminal and Console window output.
4. Add error checking:
    a. Open the 'compile.sh' script in the text editor (or Atom or vi editors).
    b. Check that the script arguments are not empty.
    c. Check the return value from 'gcc.'
    d. Test your script.
    e. Take a screenshot of the Terminal and Console window output.
5. Working with environment variables:
    a. Create a 'var1.sh' script.
    b. Create a variable in var1.sh script.
    c. Call the printenv command to print the environment variables out.
    d. Call var2.sh from this script.
    e. Create a 'var2.sh' script.
        i. Call the printenv command to print the environment variables out.
    f. Run 'var1.sh' script.
    g. Take a screenshot of the Terminal and Console window output.
    h. Modify 'var1.sh' script and prefix the variable with the word export.
    i. Run 'var1.sh' script again.
    j. What differences do you see in the environment for each script?
    k. Take a screenshot of the Terminal and Console window output.

Research Questions

Research Questions: For traditional ground students, answer the following questions in a Microsoft Word document:

a. What is the difference between kernel and user mode? How does the dual mode approach (kernel and user) impact the design of an operating system?
b. What does the following Linux shell pipeline do?

    grep nd xyz | wc –l

    Write an alternative command (or commands) that can accomplish the same thing and explain which alternative is better and why. Provide screenshots showing both commands executed.

Submission

1. In a Microsoft Word document, complete the following for the activity report:
    a. Cover sheet with your name, the name of this assignment, and the date.
    b. Section with a title that contains all theory of operation write-ups, answers to questions asked in the activity, and any screenshots taken during the activity.
    c. Section with a title that contains the answers to the research questions (traditional ground students only).
2. Submit the activity report to the digital classroom.

Appendix A – Sample Programs

The following can be used as guidance to program the C programs and shell scripts in the activity.

C Compiler Validation Example

```
/* Hello World program */
#include<stdio.h>
main()
{
    printf("Hello World\n");
}
```

C Programming Example

```
#include <stdio.h>

/*
 * Add 2 numbers and print result to the screen
 */
int main()
{
    // Declare 2 variables to keep our number input.
    int number1, number2;

    // Prompt the user for input values
    printf( "Enter a your first number to add: ");
    scanf("%i", &number1);
    printf( "Enter a your second number to add: ");
    scanf("%i", &number2);

    // Add 2 numbers and save the result in a variable calle result
    int result = number1 + number2;

    // Print the result to the screen
    printf("The result of adding my 2 numbers is: %d", result);

    return 1;
}
```

Bash Script Examples

```
#!/bin/sh

# Save script arguments and validate we have inputs (or could use $# -eq 0)
myprog=$1
if [ -z $myprog ]; then
    echo "Input program name to compile is required for this script."
    exit 1
fi

# Compile the program
echo "Compiling my program....." $myprog
gcc $myprog.c -o$myprog.out
if [ $? -ne 0 ]; then
    echo "Compile failed"
    exit 2
else
    echo "Done"
fi

# Run the program
echo "Running my program....." $myprog
./$myprog.out
echo "Done"
```

```
#!/bin/sh

# Create a variable and export it for this shell as well as all other processes
MYVAR="This is a test"
export MYVAR;
echo "############################## "
echo "######## VAR1.SH ############# "
echo "############################## "
printenv

# Call another script
exec ./var2.sh  # Run in same process and wait for it to complete
#exec ./var2.sh  &  # Run in the background (and not block)
#exec ./var2.sh > test.txt
```

```
#!/bin/sh
echo "################################## "
echo "######### VAR2.SH ############### "
echo "################################## "
printenv

exit 0
```

Appendix B – VMware Install Instructions

The following can be used as guidance to install VMware, which is a commercial virtualization application with only a 12-month license, in the event that you cannot get VirtualBox to work.

1. You can download and run VMware for 12-months for free with your *CSET On The Hub* account, which can be accessed using the following link.
    a. https://e5.onthehub.com/WebStore/ProductSearchOfferingList.aspx?srch=Vmware&ws=9c07565e-a326-e711-9427-b8ca3a5db7a1&vsro=8
2. Click the Sign In button and sign in using your **GCU credentials**.
3. You will then get an email for you to finish the registration process.
4. Once you have registered and logged into *CSET On The Hub* do a search for VMware.
    a. You will need the latest version of VMware Workstation Pro if you are on Windows.
    b. You will need the latest version of VMware Fushion if you are on the Mac.
    c. Download the appropriate VMware for your operating system.
5. Remember to keep your license key once you have "bought" the product and downloaded it.
6. Once you have VMware installed, you can just create a new VM, use the ISO that you downloaded for Ubuntu Desktop, and proceed with the new VM Wizard. It is really straightforward, with very few questions you need to answer during the VM setup.