Linux Assignment (Day 1 and Day 2)

Day 2 Assignment

Shell Scripting to perform basic arithmetic operations

```
#!/bin/bash
#====================================================================
# SCRIPT NAME: operationscript.sh

# PURPOSE: Demonstrate simple Bash programming concepts using arithmetic operations

# REVISION HISTORY:

# AUTHOR                  DATE            DETAILS
#------------------------- --------------- --------------------------
# Anita Baral             2021-07-30      Initial version

# LICENSE: leapfrog-technology v2.0
#====================================================================
read -p "Enter two numbers:  " x y
read -p "Enter the operator (+, -, *, /, %):  " z
#echo $x $y $z
if [ "$z" = "+" ] ; then
echo "Sum: " $x + $y = $((x + y))
elif [ "$z" = "-" ] ; then
echo "Subtraction: " $x - $y = $((x - y))
elif [ "$z" = "*" ] ; then
echo "Multiplication: " $x "*" $y = $((x * y))
elif [ "$z" = "/" ] ; then
echo "División: " $x / $y = $((x / y))
elif [ "$z" = "%" ] ; then
echo "Modulo: " $x % $y = $(( x % y))
else
echo "Unable to perform the specified operation"
fi
```

Explore curl command

curl is a command line tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP or FILE). *curl* is powered by Libcurl. This tool is preferred for automation, since it is designed to work without user interaction. curl can transfer multiple files at once.

curl features: Multi-function tool, support multiple network protocols, security, support gzip compression technology.

Basic Syntax:

```
curl [options] [URL...]
```

Basic example:

```
curl https://www.google.com
```

How to make get and post request with curl

Post

The general form of the curl command for making a POST request is as follows:

```
$ curl -X POST [options] [URL]
```

The following example shows how to make a POST request to a form that has “name” and “email” fields:

```
$ curl -X POST -F 'name=linuxize' -F 'email=linuxize@example.com' https://example.com/contact.php
```

When the -F option is used, curl sends the data using the multipart/form-data Content-Type.

Another way to make a POST request is to use the -d option. This causes curl to send the data using the application/x-www-form-urlencoded Content-Type.

```
$ curl -X POST -d 'name=linuxize' -d 'email=linuxize@example.com' https://example.com/contact.php
```

Specifying the Content-Type

To set a specific header or Content-Type use the -H option. The following command sets the POST request type to application/json and sends a JSON object:

$ curl -X POST -H "Content-Type: application/json" \   -d '{"name": "linuxize", "email": "linuxize@example.com"}' \   https://example/contact

Uploading Files

To POST a file with curl, simply add the @ symbol before the file location. The file can be an archive, image, document, etc.

$ curl -X POST -F 'image=@/home/user/Pictures/wallpaper.jpg' http://example.com/upload

GET

The simplest and most common request/operation made using HTTP is to GET a URL. The URL could itself refer to a web page, an image or a file. The client issues a GET request to the server and receives the document it asked for. If you issue the command line
$ curl https://curl.se
you get a web page returned in your terminal window. The entire HTML document that that URL holds.
All HTTP replies contain a set of response headers that are normally hidden, use curl's --include (-i) option to display them as well as the rest of the document.

Options:
-o: saves the downloaded file on the local machine with the name provided in the parameters.
Syntax:
$ curl -o [file_name] [URL...]
Example:
$ curl -o hello.zip ftp://speedtest.tele2.net/1MB.zip

-O: This option downloads the file and saves it with the same name as in the URL.
Syntax:
$ curl -O [URL...]

Example:

$ curl -O ftp://speedtest.tele2.net/1MB.zip

-C : This option resumes downloading which has been stopped due to some reason. This is useful when downloading large files and was interrupted.

Syntax:

$ *curl -C - [URL...]*

Example:

$ *curl -C - -O ftp://speedtest.tele2.net/1MB.zip*

--limit-rate : This option limits the upper bound of the rate of data transfer and keeps it around the given value in bytes.

Syntax:

$ curl --limit-rate [value] [URL]

Example:

$ curl --limit-rate 1000K -O ftp://speedtest.tele2.net/1MB.zip

-u : curl also provides options to download files from user authenticated FTP servers.

Syntax:

$ curl -u {username}:{password} [FTP_URL]

Example:

$ curl -u demo:password -O ftp://test.rebex.net/readme.txt

To download via a proxy server

$ curl -x proxy.example.com:3128 http://www.tutorialspoint.com/unix/

Day 1 Assignment
Different desktop environments:

1. KDE

KDE is one of the most popular desktop environments out there. You may also refer to it as the “**Plasma**” desktop. Even though it’s not my primary choice, it is highly customizable and extremely lightweight.
KDE also makes it easier to connect your phone with your Linux system using KDE Connect. You will also find Plasma’s browser integration that connects your phone directly to your browser for establishing quick communication.

Overall, it looks like KDE is an incredibly lightweight desktop environment while being one of the most flexible as well.

Some Linux distros using KDE as the default are openSUSE, Kubuntu and KDE Neon.

- Modern and polished user interface
- Highly customizable and flexible experience
- Several useful tools built-in
- Extremely lightweight
- The customization options and tools might be a little too overwhelming for newbies, leading to potential confusion.

2. MATE

MATE Desktop Environment is based on GNOME 2. MATE was initially developed for the users who were disappointed with the latest iteration of GNOME shell — GNOME 3.

Even though it's based on the good-old GNOME 2, the MATE team has improved the desktop environment on a lot of grounds. To get some idea, you might want to check out what Ubuntu MATE 20.04 has to offer.

If we take the example of Ubuntu MATE 20.04, MATE desktop is suitable for almost everyone. And, of course, especially for the ones who loved GNOME 2 but hate the new GNOME. In addition to the user experience, it is also worth noting that it is a lightweight desktop environment as well.

MATE comes with a collection of basic applications and includes a number of built-in useful tools.

Ubuntu MATE is one of the official flavors of Ubuntu that utilizes the MATE desktop. Some other popular Linux distributions like Linux Mint, Manjaro, etc, also offer MATE editions of their distributions.

- Easy to use and robust experience
- Lightweight
- Simple yet Customizable
- May not offer the most intuitive user experience

3. GNOME

GNOME is arguably the most popular desktop environment out there. Many of the popular Linux distros use GNOME as their default desktop environment and it has some popular forks, such as Cinnamon.

GNOME is designed to be easy to use and customizable. The user interface aims to provide a unique experience (kind of tailored for both mobile and desktops).
Unfortunately, GNOME isn't a lightweight desktop environment. So, it's not a great choice to go with if you are looking to install a Linux distribution on older computers or systems with less than 4 Gigs of RAM.

It's good to see that GNOME is also focusing on the performance side of things with their recent GNOME 3.36 release.

So, if you want a good user experience with something that looks different from the likes of a traditional Windows layout, GNOME should be the perfect pick.
Some major distros using GNOME are Debian, Fedora, openSUSE and Ubuntu. Not to forget Pop OS 20.04 also features many good things along with the GNOME desktop environment.

- Modern and touch-friendly UI
- Can extend functionalities through GNOME Shell Extensions
- Customizable

- Modern and touch-friendly UI
- Can extend functionalities through GNOME Shell Extensions
- Customizable

4. Cinnamon

   Cinnamon, a fork of GNOME 3, was initially developed to be and is the default desktop environment for Linux Mint. It is known for its similarities with the Windows user interface which helps new Linux users get comfortable using it easily.

   Cinnamon tries to present itself as a modern desktop environment while offering a traditional user interface. And, being somewhat light on resources makes Cinnamon a balanced choice for many Linux users.

   - Sleek and polished look
   - Familiar interface
   - Pretty customizable
   - May not offer the most intuitive user experience

5. Budgie
6. LXQt
7. Xfce
8. Deepin

Different Linux Distros

RPM-based

Red Hat Linux and SUSE Linux were the original major distributions that used the .rpm file format, which is today used in several package management systems. Both of these were later divided into commercial and community-supported distributions. Red Hat Linux was divided into a community-supported but Red Hat-sponsored distribution named Fedora, and a commercially supported distribution called Red Hat Enterprise Linux, whereas SUSE was divided into openSUSE and SUSE Linux Enterprise

DEB-based

Debian Linux is a distribution that emphasizes free software. It supports many hardware platforms. Debian and distributions based on it use the .deb package format and the dpkg package manager and its frontends (such as apt or synaptic).

- **Ubuntu-based**

Ubuntu is a distribution based on Debian, designed to have regular releases, a consistent user experience and commercial support on both desktops and servers.[

Pacman-based

Pacman is a package manager that is capable of resolving dependencies and automatically downloading and installing all necessary packages. It is primarily developed and used by Arch Linux and its derivatives.

- **Arch Linux**

The linux booting process

1. BIOS

BIOS stands for Basic Input/Output System. In simple terms, the BIOS loads and executes the Master Boot Record (MBR) boot loader.
When you first turn on your computer, the BIOS first performs some integrity checks of the HDD or SSD.
Then, the BIOS searches for, loads, and executes the boot loader program, which can be found in the Master Boot Record (MBR). The MBR is sometimes on a USB stick or CD-ROM such as with a live installation of Linux.
Once the boot loader program is detected, it's then loaded into memory and the BIOS gives control of the system to it.

2. MBR

MBR stands for Master Boot Record, and is responsible for loading and executing the GRUB boot loader.

The MBR is located in the 1st sector of the bootable disk, which is typically /dev/hda, or /dev/sda, depending on your hardware. The MBR also contains information about GRUB, or LILO in very old systems.

3. GRUB

Sometimes called GNU GRUB, which is short for GNU GRand Unified Bootloader, is the typical boot loader for most modern Linux systems.

The GRUB splash screen is often the first thing you see when you boot your computer. It has a simple menu where you can select some options. If you have multiple kernel images installed, you can use your keyboard to select the one you want your system to boot with. By default, the latest kernel image is selected.

The splash screen will wait a few seconds for you to select and option. If you don't, it will load the default kernel image.

In many systems you can find the GRUB configuration file at /boot/grub/grub.conf or /etc/grub.conf.

4. Kernel

The kernel is often referred to as the core of any operating system, Linux included. It has complete control over everything in your system.

In this stage of the boot process, the kernel that was selected by GRUB first mounts the root file system that's specified in the grub.conf file. Then it executes the /sbin/init program, which is always the first program to be executed. You can confirm this with its process id (PID), which should always be 1.

The kernel then establishes a temporary root file system using Initial RAM Disk (initrd) until the real file system is mounted.

5. Init

At this point, your system executes runlevel programs. At one point it would look for an init file, usually found at /etc/inittab to decide the Linux run level.

6. Runlevel programs

Depending on which Linux distribution you have installed, you may be able to see different services getting started.

User and Group Management

There can be multiple users on a Linux system. If you want to list all users in Ubuntu, the information is 3present in the **/etc/passwd** file. This file stores the list of users on the system along with important information regarding these users. **/etc/passwd** file is used at the time of login.

We can obtain the list of users by displaying the content of **/etc/passwd** file.

To display content of **/etc/passwd** file simply use the cat command.

```
$ cat /etc/passwd
```

Creating a new user

Apart from knowing how to list all users in Ubuntu, it is important also to know how to add a new user. To create a new user in Linux, use the following command:

```
$ useradd username
```

Setting password for the new user

```
$ passwd username
```

Deleting the user

```
$ userdel username
```

Get information about a particular user

You can use grep command to get information about a particular user from the **/etc/passwd** file. This is useful if there are a lot of users in the **/etc/passwd** file. This can help in checking for the existence of a particular user.

```
$ grep username /etc/passwd
```

There are two types of groups in Linux, they are

1. Primary group
2. Secondary or Supplementary group

Primary group

To add a user to a Primary group, use the following user mod command as root,

Syntax

```
usermod -g [groupname] [username]
```

Secondary Group

A user can be added to a secondary group using the following command.

Syntax

```
usermod -G [groupname] [username]
```

Deleting a group

To delete a group, use the following User management command

Syntax

```
groupdel [groupname]
```

Cut command

The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by **byte position, character and field**. Basically the cut command slices a line and extracts the text. It is necessary to specify options with commands otherwise it gives errors. If more than one file name is provided then data from each file is **not preceded** by its file name.

Syntax:

cut OPTION... [FILE]...

Cut Text Based on Byte Positions

The b option provided by the cut utility allows us to cut sections of a text-based on their byte positions. We need to use the cut command with the flag -b followed by the byte numbers for this purpose.

```
$ echo "cutting text from input" | cut -b 1
```

The above command echoes the string *"cutting text from input"* to the standard output and pipes it as an input to the cut command. The cut command will cut just the first byte(*c*) from this string as only 1 was provided with the **-b** flag.

Cut Text Based on Characters

The cut command in Unix allows users to cut a section of text based on characters. When handling large file processing tasks, you'll need to do this quite often.

```
$ echo "cutting text from input" | cut -c 1
```

The above command cuts the first character from the standard input and displays it in the terminal. In this case, it is "*c*". Change your string to something different to understand this clearly.

Cut Text From Columns using Fields and Delimiter

The cut command allows users to cut sections of a text very easily. For this, we need to use both the d and the f flag of cut. The d flag stands for delimiters and f for fields. Delimiters are special characters that separate sections of a text from others. Common examples include '-', ':', and " " (space). The reference file we're using has ':' as the separator.

Cut the First Section of Input Stream

```
$ echo "Let's cut this input stream section by section" | cut -d ' ' -f1
```

The above cut command will cut the first section of text("*Let's*" in this case) from the input stream. Note that the value to the delimiter flag **-d** is a single space.