Breeha Qasim 08283

Lab 01: Introduction to Bash Fall 2024

1. Introduction

This lab is designed to introduce you to Linux, C language, and some simple examples on how the operating system works as a 'resource manager.'

This lab derives its contents from the following two books:

a. B. Wajid, H. Iqbal and M. Jamil. "Linux programming for the faint of heart," Sabz Qalam, 2020 (ISBN #: 978-969-7941-00-1).
b. R. H. Arpaci–Dusseau and A. C. Arpaci–Dusseau. "Operating Systems: Three Easy Pieces," Arpaci–Dusseau Books, LLC, 2019.

Here book (a) is written by the faculty and two of his undergraduate students, specifically for an audience like you, i.e., undergraduate students. The book is available on HLMS.

HW 1 – Lab 1: You are expected to read "Part I – Getting Started," (Chapters 1 and 2) at home [1]. Don't worry these are just 18 pages written using a large font size, on a standard small book size.

2. Shell Scripting (ETC: 1.5 hours)

Go through Chap. 3 of the book titled "Introduction to Shell Scripting" [1]. Everyone must run the codes written on the terminal themselves.

3. C Coding (ETC: 0.5 hours)

Chap. 2 of the book [2] introduces a simple C code, reproduced herein below, that continues to print a string passed by the user, until the program is forcefully terminated (by pressing CTRL+C).

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <assert.h>

int main (int argc, char *argv[])
{
        if (argc != 2) {

                fprintf( stderr, "Usage: cpu <string> \n" );
        fprintf( stderr, "Example: cpu \"University\" \n" );

                exit(1);
        }

        char *str = argv[1];
        int sleep_seconds = 2;

        while (1) {
                sleep( sleep_seconds );
                printf("%s\n", str);
        }

        return 0;
}
```

Write the above code using a text editor. Save the code as “cpu.c”. Once saved, open the terminal, and navigate yourself to the folder where the file is present. Once there, type the following:

```
> gcc -o cpu cpu.c -Wall
```

This will generate an executable file called “cpu.” Now run the file by typing the following on the terminal:

```
> ./cpu
```

This will give an error. Now, run the code again, this time giving a string as an input. This string could be anything you want to repeat. For example:

```
> ./cpu “Habib University”
```

The code will continue printing the string every 2 seconds. To “kill” the program, press CTRL+C. Now run multiple instances of the same code by typing the following on the terminal. Note an instance like “<Your name>” means that you are expected to type your name here.

```
> ./cpu “Habib University” & ./cpu “<Your name>” & ./cpu “<Your thoughts>”
```

Note, even though we have only one processor, all three instances are running concurrently.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <assert.h>
int main (int argc, char *argv[])
{
   if (argc != 2) {
      fprintf( stderr, "Usage: cpu <string>
\n" );
      fprintf( stderr, "Example: cpu
\"University\" \n" );
      exit(1);
   }
   char *str = argv[1];

   while (1) {
      for (int i=0;i<=1000;i++){

         printf("%s\n", str);
      }
   }
   return 0;
}
```

Q1: The main function here, had two arguments (`int argc, char *argv[]`), explain their use?
Answer: The first parameter tells us about the number of command line arguments passed and first argument is always the program name while the second parameter is an array of strings pointers which will hold actual arguments passed content

Q2: How do we increase the delay in printing strings?
Answer: To increase the delay we will need to change the value of variable sleep seconds, it is initialized with 2 value we can increase it to 6 or so to increase delay in printing values. It will then wait for that many seconds to print next.

4. Linux Files and Filesystem (ETC: 1.5 hours)

Go through Chap. 4 of the book titled “Linux Files and Filesystem” [1]. Everyone must run the codes written on the terminal, and navigate through the file systems themselves.

Q3: What is a shell, and which shell are you in?
A shell is a command interpreter that receives command, that translates them into binary instructions and then sends them to OS’s kernel. The shell we are using is Bash shell.

Q4: What is a “Home Directory,” and what is your Home Directory?
A home directory is personal directory assigned to user on UNIX and it lets user to store files, scripts etc. By entering Shift and ~ we will get name of our directory on terminal, my directory is /Users/bq08283:

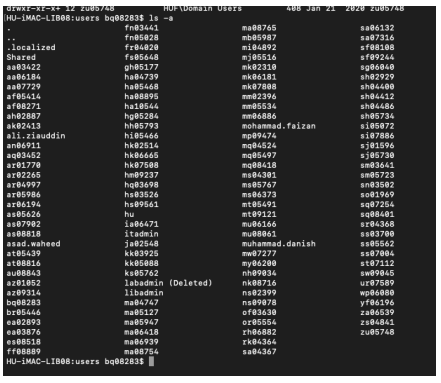
Q5: What’s a Working Directory and which directory are you in?
Working Directory is directory we are currently inside and working in. Current working directory can be checked by using pwd (print working directory) command. My directory is /Users/bq08283

Q6: Differentiate between an ‘Absolute Path’ and a ‘Relative Path’?
An absolute path specifies location to file or directory in relation to root directory and it can be easily identified by command ‘/’ while Relative Path specifies location to a file or directory, in relation to directory in which user is currently working on.

Q7: What’s the largest file inside the directory “/usr/bin”?
By using command “ls –l” we get list of files in long format then to check the largest file I did it manually and what I get is
0 drwxr-xr-x+ 32 itadmin staff 1088 May 31 2023 itadmin

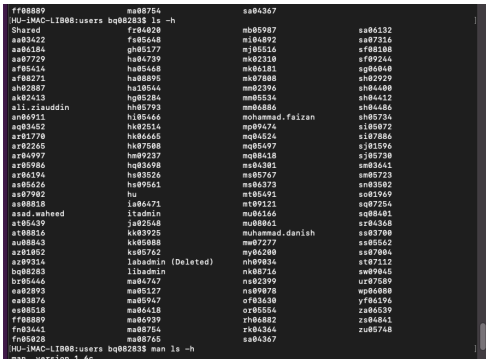
Q8: What’s the most recently created file inside the directory /usr/bin?
The most recent file created is
drwxr-xr-x+ 11 hm09237 HUF\Domain Users 374 Jan 25 2024 hm09237

Q9: List all the hidden files and directories in your home directory.
Using “ls –a” we can get all hidden directories and files.

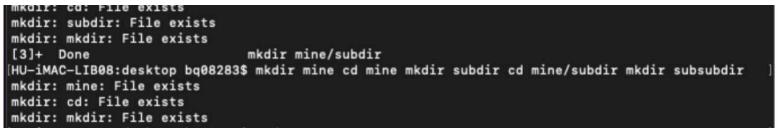


Q10: What does the command ‘file’ do?
The command file is to determine file type. It examines the content of the file and then provide information about file’s format and type.

Q11: Search for the "-h" option of "ls." What do they do? Use them.
The –h option displays file in human readable form.



Q12: Make the directory "mine/subdir/subsubdir" using one command only.



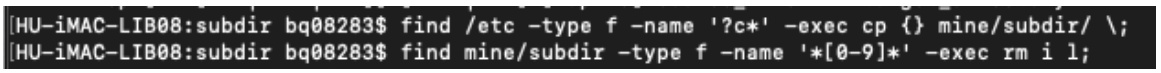
Q13: While staying in your home directory, create an empty file dummy.txt in mine/subdir/subsubdir.
We can use touch command specifying the path where we want to create file then specify file type to .txt
"touch mine/subdir/subsubdir/dummy.txt"

Q14: While staying in your home directory, copy the files zip, zipgrep, zipinfo from /usr/bin to mine/subdir/subsubdir
To copy items, a cp command is used. It will copy the files zip, zipgrep, zipinfo from /usr/bin to mine/subdir/subsubdir
"cp /usr/bin/zip /usr/bin/zipgrep /usr/bin/zipinfo mine/subdir/subsubdir/"

Q15: Move all files from mine/subdir/subsubdir to mine/subdir/
To move all files from mine/subdir/subsubdir to mine/subdir/ we will use mv command.
"mv mine/subdir/subsubdir/* mine/subdir/"

Q16: List all the files in /etc whose second letter is c.
To list all the files /etc whose second letter is c we will use find command.
"find /etc-type f-name '?c*' "

Q17: Copy all of them to mine/subdir. Then delete all files that contain a digit.



Q18: Delete the mine/subdir/ directory

References

[1] B. Wajid, H. Iqbal and M. Jamil. "Linux programming for the faint of heart," Sabz Qalam, 2020 (ISBN #: 978-9697941-00-1).

[2] R. H. Arpaci–Dusseau and A. C. Arpaci–Dusseau. "Operating Systems: Three Easy Pieces," Arpaci–Dusseau Books, LLC, 2019.