

# Assignment 1:

**Let's assume you are the system administrator of department of Computer Science and Engineering at IIT Guwahati. A system administrator had to do lots of things to handle the system properly and efficient way. This assignment has several parts that gradually take you to the several steps of works for a system administrator.**

## Part 1:

Every year lots of students are admitted in the department, in B Tech, M Tech and PhD. We have a central server where we need to make user accounts for these users. For the batch of 20XX, there are groups such as btechXX, mtechXX and phdXX. For example M Tech 2010 batch students will be under the group mtech10, and they can only view the files which have access permission for mtech10 groups. The user-name of a user is derived from his name, and the initial password is a random string. If Vishal Jha is a student of B. tech 2010 batch, his user name can be like j.vishal, v.jha, vishal or vishal10 ( as like your normal IITG user accounts, but you can use innovative ideas to derive user names), and in this case his home directory will be like /home/btech/btech10/j.vishal if the user-name is j.vishal. You will be provided a text file that contains the complete name of all the newly admitted students. As a system administrator, you need to automate the process using a shell script. So, your script should do the followings;

- i) Create a group for each newly admitted batch, as described above.
- ii) Create suitable user name and initial password for all the students of that batch, create their home directory too.
- iii) Their home directory should have a maximum quota limit of 10mb.

- iv) To make the system secure, a user can have read and write access to only the files under his home directory. He should not be able to read directory of files of other users.

So you have to write the script “createusers.sh”, so when you execute the script;

```
createuser.sh btech10
```

it will read the file btech10 that contains all the name of the students of batch B Tech 2010, and create the user accounts as described above. The script will also produce a csv (comma separated value) file in <name, user ID, password> format.

## **Part 2:**

A student should be able to log in the system using SSH. Configure your system such that the SSH server runs always. Now configure the system such that if a student logs in at the system, but remains idle for 5 minutes, (s)he will be automatically logged out.

## **Part 3:**

Now the CS 240 course instructor wants to make the attendance procedure automated. He decided that a student should be marked as present, if (s)he satisfies following rules:

- i) The lab timing is Friday, 2:00 p.m. to 5:00 p.m. A student should log in at the terminal before 2:15 p.m. and (s)he cannot leave the terminal before 4:45 p.m.
- ii) If a student is logged in at the terminal, but is idle for more than 5 minutes (is not doing anything in the terminal), (s)he will be marked as absent.

Design a script that will generate a student attendance report after each lab session. There will also be a main report that will

contain the total attendance details of all the students. Remember, you should not start the script every Friday, rather it should be executed automatically at every Friday to log the attendance details.

## **Part 4:**

Sometimes you need to take a backup of all users' files and then shut down the system for maintenance. Before shutting down, it is necessary to inform all the users about these details, and you should take a backup of all files of the users. So design a script as follows. When you execute it, it should send a message to all the users as follows:

**The Machine will be shut down today at 14:30 hrs. The backup will be at 13:30 hrs.**

Now it should schedule a backup procedure at the specified time, and then shut down the system at that time. The backup procedure should create a hard link of all users' directory and files inside /tmp/ directory.

## **Part 5:**

Let CS201 course instructor asked you to design an automated system for the assignment checking. All the assignments will have a common input-output format as follows;

The input of the assignments will be stored in a file named input.data. Let there are 20 test cases, each test cases will be written in a line, with all the inputs as comma separated value format. Let the assignment be a program to implement bubble sort. Then, a simple test case will have the following format;

```
20,10,11,10,12,19
21,11,11,10,18,19,21
4,9,8,7,10
1,1,1,1,1,1
```

The output file, named output.data will have a similar format; with each line corresponds to the output of the given test case. A sample output.data file will have following format:

```
10,10,11,12,19,20
10,11,11,18,19,21,21
4,7,8,9,10
1,1,1,1,1,1
```

Your task is to design a system written in shell script, that will take the input.data file as input, generate the output, and finally check whether the program is correct or not, by comparing the output with the output.data file. ( REMEMBER, the student will not do their program using file handling, and so their program can take a single input, and generate the corresponding output.)

To check this program, a sample C program named sort.c is attached with this assignment. Let the compiled executable file for sort.c is sort, so you need to run the file as follows;

```
./sort 20 10 11 10 12 19
```

The output will be;

```
10 10 11 12 19 20
```

## **Part 6:**

After solving all the 5 parts, as described above, your task is to design a menu driven list, that will execute the appropriate shell script to perform specific task.

If your script is admin.sh; then a sample run will be as follows:

```
$bash admin.sh
```

Welcome to CSE,IITG Server Admin !!!

- 1) create new user account
- 2) execute ssh server
- 3) start attendance for cs 240
- 4) stop attendance for cs 240
- 5) Display attendance report for cs 240
- 6) system maintenance
- 7) cs 201 assignment checking

Enter Your Choice ::

Now if the user enters 3, then the script for part 3 will be executed. Again, if user enters 4, then that script will be stopped.

# Assignment 2:

“Make” utility is a very useful utility to compile a complete package, written using some programming languages, and generate the final executable file. You can have a good tutorial for “make” utility in this website :

<http://www.eng.hawaii.edu/Tutor/Make/>

In this assignment, you have to deal with “make” utility to compile and build a project. To do this assignment, you need to do following things first:

1. Check whether lex is installed in your system. To check this, use the command “lex”. If not installed, install it (or install flex).
2. Check whether yacc is installed in your system. To check this, use the command “yacc”. If not installed, install it (or install bison).
3. Download the file nothing.tar.gz from attached with this assignment :
4. Extract the file, you will have a folder named “nothing”

Now the folder nothing has two files, prog.l and prog.y . The compiler sequence for these files are as follows;

I) compile prog.l with lex;  
lex prog.l

II) compile prog.y with yacc  
yacc -dv prog.y

III) Above two step will generate two C files lex.yy.c and y.tab.c, one C header file y.tab.h , and one system file y.output. Now your task is to compile lex.yy.c and y.tab.c using gcc, and

generate the output file.

```
gcc *.c -o nothing
```

Now, write a “makefile” that can utilize make tool to build the project and generate the executable file “nothing”. Also modify your make file such that following functionalities are included;

- I) make clean ; will clean all the .c,.h and .output files as well as the executable files.
- II) make install; will install the executable file “nothing” to /usr/bin directory, and add a soft-link to /usr/sbin.
- III) If the build, install and clean succeeds, then it should show the success message.