

## Unix Shell Environment and Basic Commands

This lab introduces the UNIX shell, its basic commands to browse through the file system, running programs and some common utilities and their usage manuals.

1. Open up a shell prompt and type in the following commands in order and analyze the output of each
  - a) `echo $SHELL`
  - b) `echo $PATH`
  - c) `uname`
  - d) `uname -r`                      # also try `uname -m`, `uname -s` etc
  - e) `/bin/bash`
  - f) `var=137.207.148.123`    # no spaces around the '=' sign
  - g) `echo $var`
  - h) `exit`
  - i) `echo $var`
2. Now create a directory hierarchy on the file system to organize all the work pertaining to this course. Use the following commands in order. Note that the # symbol and the text following it are comments and not a part of the command.
  - a) `pwd`                              #shows current directory. It should be your home directory
  - b) `mkdir 60-256; cd 60-256`        # creates a dir 60-256 and changes to it
  - c) `pwd`
  - d) `mkdir lab1; mkdir lab2; mkdir lab3` # creates directory named lab1, lab2 etc. There will be 9 or 10 labs. You can also create separate directories for the assignments.
  - e) `ls`
  - f) `ls -l`
  - g) `cd lab1`
  - h) `touch lab1`
  - i) `ls -l`
  - j) `cd lab1 &`                      # you can also try `gedit lab1 &`

3. Once nedit opens up the lab1 for editing enter the following text into it and save it. Analyze each line of code. It explains some basic programming constructs of bash. You can also download it using [this link](#) from the lab site

```
#!/bin/bash
echo "Good Morning: This is my first shell script"
today=`date`
echo "Today is $today"
dateFields=(`date`)
echo "Total field count is ${#dateFields[@]}"
for field in ${dateFields[@]}
do
    echo $field
done
echo "Month = ${dateFields[1]}"
echo "Time = ${dateFields[3]}"
echo "=====
echo "Lets try some arithmetic"
echo -n "Type a number:"
read a
echo -n "Type another number:"
read b
sum=$((a+b))
echo "a+b=$sum"
if ((a==b)); then
    echo "a is equal to b"
elif ((a>b)); then
    echo "a is greater than b"
else
    echo "a is less than b"
fi
```

4. Now lets try to run the program. Issue the following command
- a) lab1 # it shouldn't run. Because of the permission issues
  - b) ls -l # check the permissions' bits
  - c) chmod 744 lab1
  - d) stat -c%a lab1
  - e) lab1
5. Try to explore the man command.
- a) man date
  - b) man mailx > mailxManual.txt
  - c) nedit mailxManual.txt &
6. Now close everything and launch mozilla. Google for Unix stat command