University of Pennsylvania

Electrical & Systems Engineering Undergraduate Laboratories

ESE 112: Introduction to Electrical & Systems Engineering

Lab 3: Introduction to Linux, Java, and the EduBot Environment

Sansern Somboonsong
Edited by Diana Palsetia
2008-2009

Objective

Provide students with a basic understanding and usage of the Linux operating system including navigating the directory structure, using remote shell connection, writing, compiling and running java programs using non-IDE environment, and using EduBot simulation environment.

Background

Computers
Computers have become an essential part of our everyday life in the past few decades. Computers are complex combination of electronic devices which consists mainly of a computational, memory, communication, and human interface units. The intangible aspect of computers is the software that enables the users to (knowingly or not) utilize the computer hardware. The complexity of how each device in the computer functions need not be passed on to the everyday user and is usually hidden away in a piece of software called the operating system (OS). The operating system handles all of our interaction with the computer hardware, schedule multiple tasks, and allocate resources between processes. There are several operating systems available today, but the commonly used are Windows XP, Windows Vista, Mac OS X, UNIX, and Linux.

Linux Operating System
In 1969, a group of engineers at Bell Labs developed one of the first widely-used operating systems called UNIX. As the UNIX system evolved, it spawned off many other operating systems and in 1991, Linux was born. The main reasons for its popularity are the open source license, multiuser capability, reliability, and high levels of configurability. For these reasons, Linux is preferred by many engineers, computer scientists, and those in the research field. Natively, Linux is a command line based operating system, however, graphical (Windows like) environment has been developed and numerous programs in Linux now run graphically. The graphical environment in Linux now appeals to new users thus increasing its user base. The command line

interface is still useful and is heavily used by experienced users who can type in text commands a lot faster than moving the mouse across the screen and clicking buttons.
Due to its configurability, the EduBot is running a stripped down Gentoo Linux distribution. The EduBot is in essence a computer with legs. A keyboard and monitor can be connected and it would function just like any Linux computer with limited but sufficient resources.

Java
Java is a high-level, object oriented programming language developed by Sun Microsystems. The only requirement to run Java is a Java Virtual Machine (JVM) which can work on most operating systems, thus making Java platform independent. The content of JAVA is covered in ESE112 lecture and will not be reiterated in the lab.

Edubot Simulation Environment
A simulation is a computer generated model of some real-life system. Systems parameters can be changed without the need to use a real EduBot. Before attempting to physically implement certain behaviors on robots, engineers often use a simulation in order to verify that the robot will behave properly and potentially avoid any damages resulting from adverse responses. Once the simulation runs properly, the experiment is then performed using the real robot. The EduBot simulation environment is used to serve these purposes. However, one should always keep in mind, that the simulation is not a perfect replication of the physical world. Running the same piece of code in the simulation and the real robot may result in different behaviors.

Prelab

1. What are some command line text editors in Linux? For any two text editor search for keyboard short cut to saving a file.
2. What is a Linux "distribution" and give 4 examples?
3. What are relative paths and absolute path?
4. What is open-source software?
5. Look up the command that lists the content of a directory in Linux. What options do you have to specify to get a listing of all the files including the hidden files?

Material

Computer running Linux operating system
Internet connection
Wireless USB adaptor
EduBot Robot

Lab Instructions

Part I - Using Linux Commands

1. Log on to the Linux machine with username **`ese112`**
    - The password will be given in class
2. Open a shell terminal
    - Locate and click on the icon that looks like monitor screen. 
    - You should see a screen with ese112@MOR204-## (where ## is the computer number)

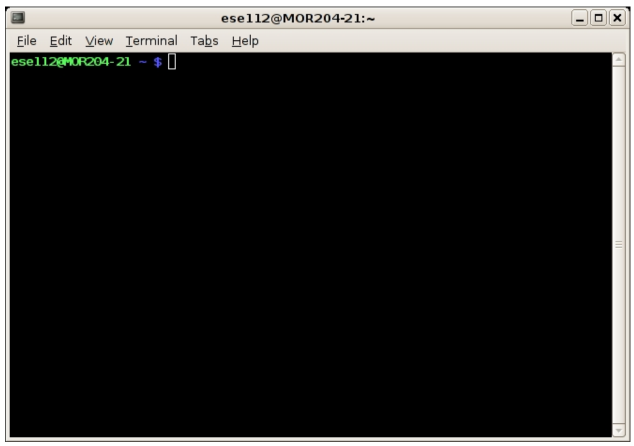

Figure 1 - Terminal

Note: To look up a command in Linux, use the manual or man pages using the command `man` followed by the command you want to look up. For example `ls` is a command to list the content of a directory. To get the manual for this command type the following command (without the #) and hit enter.

```
# man ls
```

3. Use the **`pwd`** command to output the absolute path of your current directory
4. At the prompt invoke command "emacs" that will open a text editor called emacs
    - You want to note down some handy keyboard shortcuts shown at the first screen.
5. Write a file named **`test.txt`** In that file, write your group members' names.
6. Save the file and exit emacs
    - Where did it get saved ?

7. Create a directory named **TEST**
8. Copy the file you just saved into the directory and delete the original copy of the file.
9. Change directory into **TEST** where the file is now located and rename the file to **`eseXXXX.txt`** (where xxxx is the initials of the names of the people in your group)
10. List the contents of the directory
11. Go back to the original directory
12. Copy the file to your eniac account using **`scp`**
    - the address for the eniac machine is **`eniac.seas.upenn.edu`**
13. Remotely connect (using **`ssh`**) to eniac.
14. Search for eseX.txt to verify that it copied
15. Delete the file and exit.

Part II - Running the EduBot simulation.

1. Open a terminal window and Change directories into "~/EduBot/Software/RobotCode/Demo/ESE112"
   ```
   # cd ~/EduBot/Software/RobotCode/Demo/ESE112
   ```
2. Run the supervisor and the window in figure 2 should pop up and and EduBot should appear.
   ```
   # ./DemoSup
   ```

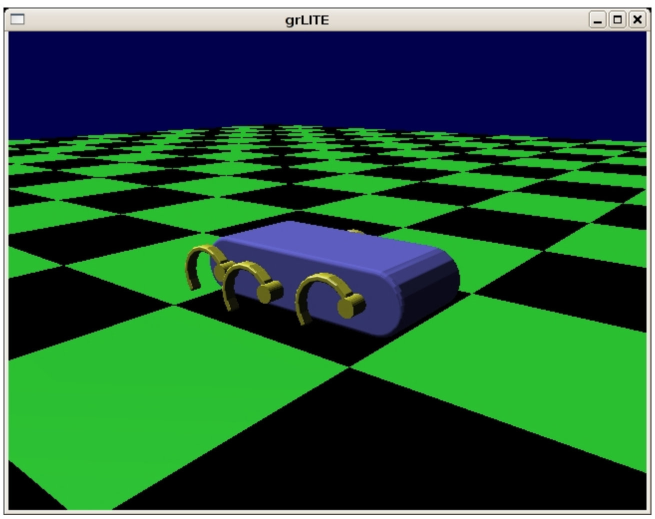

Figure 2 – Simulation Window

3. In a **<u>different</u>** terminal and change directories to "~/EduBot/Software/Operator"
   ```
   # cd ~/EduBot/Software/Operator
   ```

4. Run the command

```
# ./main_gui 3000 localhost
```

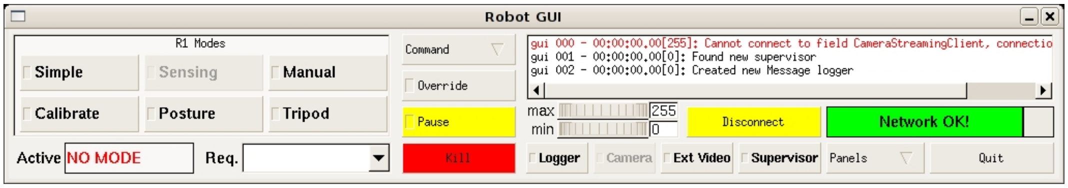

Figure 3 – Robot GUI

5. Walk the robot with the GUI
   a. The simulation environment should now open.
   b. Click the **Calibrate** button on the EduBot GUI window and select **Ground** and **Start** to calibrate the robot. Wait a few seconds until the robot is calibrated. You should see a message on the terminal.

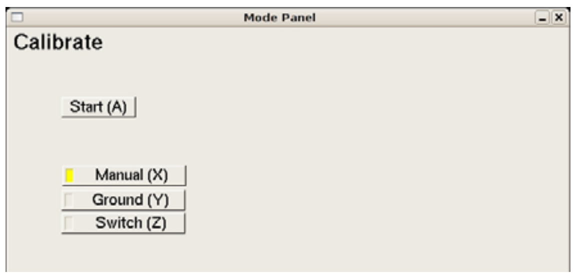

Figure 4 – Calibrate Mode Panel

   c. Click **Posture** mode and click **Stand** to make the robot stand up.

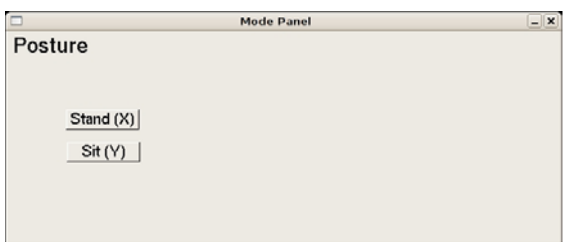

Figure 5 – Posture Mode Panel

   d. Go into **Manual** mode. In the Manual Mode Panel, under PD Gains, set kd = 0.10 and click on the **Apply Profile** button. Increase the slider speed and click on the up arrow. The robot should start walking in the simulation. Click the button in between the up arrow and down arrow with the black square (not the red stop button) to stop the robot.

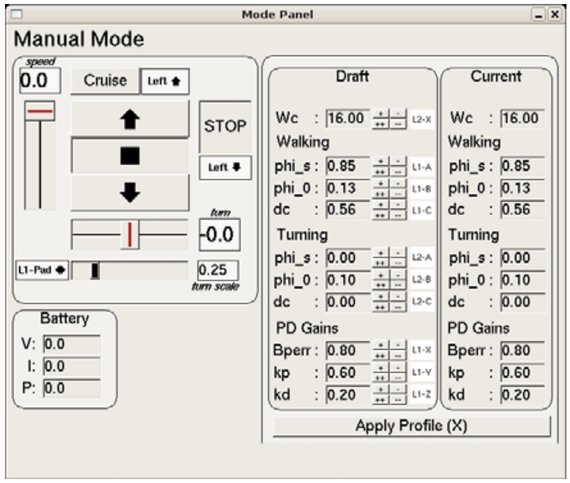

Figure 6 – Manual Mode Panel

Part III - Running the EduBot

1. Insert a wireless USB adaptor into the computer and then run the following script in a terminal.

   ```
   # setnet-rhex
   ```

2. In a terminal, ping the router to make sure that the network is up on your computer

   ```
   # ping 192.168.1.1
   ```

3. In a terminal, ping the robot to make sure that the network is up on the robot

   ```
   # ping <robot_ip_address>
   ```

   Note that the robot's IP address is on the label on each robot.

4. If the network connection is present, login into the robot using secure shell. The password will be provided by the instructor.

   ```
   # ssh root@<robot_ip_address>
   ```

   Now you are remotely connected to the robot just like how you were remotely connected to the eniac server.

5. While being remotely logged into the robot, use the same terminal to change the directory and start the supervisor on the robot. Note: the robot should be placed on flat ground 3 feet from any obstacles.

   ```
   # cd /home/edubot/EduBot/Software/RobotCode/Demo/ESE112
   # ./DemoSup
   ```

   `Note: If you cannot navigate to Demo/ESE112 directory then in current directory type ./g`

6. In a **different** terminal, change directory and run the main GUI

```
# cd ~/EduBot/Software/Operator
#./main_gui 3000 <robot_ip_address>
```

7. Once the GUI starts, repeat step 5 of Part II to walk the robot around. If any unexpected behavior occurs, immediately pick up the robot from the ground without obstructing the motion of the legs.

Part IV – Writing Java Programs in non-IDE Environment

Engineers and scientists write programs to exploit the power of modern computers and solve real world problems. These problems are based on mathematical models of some physical occurrence. You will write a Java application (so you will need the main method) to approximate the derivative of the exponential function (e^x) at a given point (x =*a*) using the Taylor Series given below.

$$e^x \cong \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots \text{ for all x}$$

The program should be procedural based (i.e. make all methods static) and uses the concepts taught in programming lecture. The parameters *a* and the accuracy measure should be variables easily changed before compilation. The program should output the approximated value and the number of steps it took to arrive at the answer. Use emacs or other non-graphical test editor (instead of IDE like Dr. Java) and work in the terminal environment. Use "javac" command to compile your code and "java" command to run your program (HINT: use the man page). You will be asked demonstrate the compiling and running the program in a non-IDE environment.

Questions

1. What are two advantages of graphical and inline text editing?
2. What are the discernible differences between the simulation and the actual running of the robot?
3. What were the calculated values of e^1 and e^4?
4. How did you approach the Taylor series problem? Explain how you used functions to do the highly repetitive work.
5. How accurate was your program and to what power were you able to calculate? Which data type did you choose to implement and why?
6. Include your program as an appendix. Ensure that your program has adequate commenting and style usage.
   Also submit your program to Blackboard through the digital dropbox under the appropriate lab section (101/102). Be sure to comment your code liberally so that the reader can easily understand the purpose of your code. Put your .java file in the folder named ese112_Linux_XX_XX_XX (where XX is the first and last initial of a group member). Then zip the folder to finally send the file **ese112_Linux_XX_XX_XX.zip**