

CSE 398/498
Exercise 1:
Lab Dates:

Robocup
Programming Environment Orientation & the Tekkotsu Tutorial
20-23 Jan 06

A. Objectives:

1. To become familiar with the Linux programming environment and build procedures for developing code on the Sony Aibos.
2. To progress through the *Tekkotsu* tutorial - the CMU developed coding environment for developing robocup applications for the Sony Aibo.

B. Tasks

1. Build Procedure: The Aibo's operate by running an executable file installed on the Sony memory stick. These build procedures outline the necessary steps for transferring code developed on your desktop to the Sony Aibo.

WARNING - the build procedures do NOT unmount the disk. You will be reminded of this each time with a warning/error reading that the stick failed to unmount. You must unmount it yourself by right clicking the icon and selecting unmount. The light on the stick reader should come on briefly, after the light goes out, it is safe to remove the stick.

- a. You are set up with a Linux account on the second computer from the door (FangornForest). Your user name is `robocup` with password `tekkotsu`.
- b. In your home directory, there is a directory called `project`, which is the main Tekkotsu project.
- c. Adding files to this project is as simple as creating a `.cc` file in that directory. The `makefile` automatically searches the project directory for all files and compiles them in.
- d. If you should happen to destroy your project directory, you can recover the valid copy by

```
cp -r /etc/skel/project
```

which will copy the project directory into the current directory.
- e. Building the project is as simple as typing `make`. This will generate all of the binary files for your project.
- f. Installing these files onto you memory stick requires generating a new stick, installing those files and then syncing the project with the stick.
 - i. To generate a new stick, destroying the files on your old stick, you would simply type

```
make newstick
```
 - ii. After generating a new stick, you should install the binaries with

```
make install
```

- iii. Then, the project and the stick must be synced with a `make update`
 - g. Once you have created a valid stick, you can update the software by repeating the `make update` command which will build all of your changed files and update the stick with the differences.
- 2. Environment Configuration: There are 2 files which determine the build environment (there are many but only two which concern us).
 - a. The first is the `Environment.conf` file in your home directory. This has a number of options which determine the build options for the robot's program. The two options that maybe of importance are:
 - i. `MEMSTICK_ROOT` - this determines the mount point for the memory stick, which in our case should be `/media/usbdisk`
 - ii. `TEKKOTSU_ALWAYS_BUILD` - this tells the makefile to try and build the framework itself, which is protected, so it will most likely fail. This should be set to the empty string
 - b. The second is the `wlandflt.txt`, in the `ms/open-r/system/conf/` directory. It determines the IP address information of the AIBO. The IP addresses for the dogs are `192.168.0.XXX`, where the last three digits should be affixed to the individual dog.
- 3. Running Programs: Once you've implemented a new behavior and installed in on the memory stick, you need to instruct the dog to execute it. This will be accomplished via the `ControllerGUI` interface described below.

WARNING - You may execute a behavior that causes the Aibo to injure itself. You can suspend any behavior by emergency stop. This can be accomplished by hitting the STOP sign on the ControllerGUI or double-tapping the center back button of the dog.

- a. After building a valid stick, you can place it in the dog and power it up using the button on the dog's back by the base of its neck. It should generate a reassuring growl, which means it is booted up and working correctly.
- b. There are visual interfaces to the dog from the desktop. These are through telnet and a GUI window.
 - i. To connect, first open a new terminal window and type `telnet 192.168.0.XXX 59000`
This will open up a terminal connection to the dog showing all of the standard out information.
 - ii. In another terminal window, the GUI can be attached by typing `/usr/local/Tekkotsu/tools/mon/ControllerGUI 192.168.0.XXX`
 - iii. The telnet window provides information useful for debugging programs. The `ControllerGUI` application provides an

interface for executing behaviors. We will use these extensively throughout the course.

- iv. There are numerous sample behaviors in the `ControllerGUI` worth exploring. The sample code for these is also available. These will be useful to review for generating your own behaviors in the future.

4. The Tekkotsu Tutorial:

- a. The best place to start learning about programming in Tekkotsu is via the Tekkotsu tutorial. The main Tekkotsu page is at <http://www.cs.cmu.edu/~tekkotsu/>
- b. You can find the Beginner's Tutorial at <http://www.cs.cmu.edu/~dst/Tekkotsu/Tutorial/>
- c. You should go through AND IMPLEMENT all of the sample code exercises on the dog through Section 12. Those without extensive programming experience, this may seem intimidating. However there is extensive documentation (step 2 of the tutorial), and you should bookmark the online reference. You should be referring to it quite often.
- d. Work as a team, and TAKE YOUR TIME GOING THROUGH THE TUTORIAL. It is not a race to see how quickly you can get through it. You are only competing against yourself, and in the end you will understand the programming model better if you take the time to understand the code fragments presented.