Running SML/NJ

whm, 8/21/06

Running sml on lectura

The first step to is to login to lectura via a secure shell (ssh) connection.

On departmental Windows machines, you can use Secure Shell Client on the Start menu (or desktop) to establish an ssh connection to lectura. After starting the application, press <ENTER> to display the “Connect to Remote Host” dialog. Specify lectura as the Host Name and your CS Account login id as the User Name. Click Connect and enter your password in the resulting dialog.

You'll then see a window with contents something like this:

```
SSH Secure Shell 3.2.9 (Build 283)
[...more...]
Last login:  Mon Aug 21 20:44:53 2006 from ...
[whm@lectura ~]$
```

The last line is the shell prompt. The prompt above ends with a dollar sign but yours might end with a percent, greater than sign, or something else.

To start SML/NJ, just type sml:

```
[whm@lectura ~]$ sml
Standard ML of New Jersey v110.57...
- 3 + 4;
val it = 7 : int
```

To exit sml, just type ^D (control+D). If sml becomes non-responsive—perhaps there’s infinite recursion in progress—type ^C to interrupt it.

To terminate the shell and close the ssh connection, type exit. Or, just close the ssh client.

If you’d like to try a different ssh client, PuTTY is my favorite. You can find it at
www.chiark.greenend.org.uk/~sgtatham/putty

More information on ssh can be found here:
www.cs.arizona.edu/computing/access/ssh.html

If you haven’t used a UNIX machine before you might find my CSc 352 UNIX slides to be useful. They’re on the Resources page on the class website.

Editing sml source files for use on lectura

One option for editing sml source files is to use an editor on lectura. Some simple editors available there are pico, nano, and joe. Many like the power of vim but others find its bi-modal operation maddening. My preferred editor is Emacs. It’s somewhat complicated but it is widely used among professional developers for editing outside of IDEs. A set of slides on Emacs that I wrote for CSc 352 can be found on the Resources page on the class website.

If you don’t want to edit on `lectura`, you can certainly use a Windows-based editor, TextPad perhaps. Things are arranged so that on the departmental Windows machines, `H:\` corresponds to your home directory on `lectura`. If you create the file `H:\x.sml`, you’ll see that file on `lectura` and will be able to load into `sml` with `use “x.sml”`.

If you create a file like `H:\372\x.sml`, use `cd 372` on `lectura` to make that your working directory. Then run `sml`. (`sml`’s `use` function looks in the current directory if no path is specified.)

Watch out for problems with file name extensions when using a Windows editor. For example, TextPad will sometimes add an extension to a name, and produce a name like `x.sml.txt`. To avoid that, select a file type of “All files” when you first save the file.

Installing and running SML/NJ on your own Windows machine

Go to `smlnj.org/dist/working/110.57/index.html` and under Microsoft Windows in the Downloads section, which is the first thing on the page, save `smlnj.zip` on your machine.

NOTE: If you browse around the `smlnj.org` site you’ll see a “A self-installing .EXE for Windows”. DO NOT GET IT. It is for a much older version of the system. It’s important that you use 110.57 because it’s the version on `lectura` and it’s the version that we’ll use for grading assignments.

Open `smlnj.zip` from your browser or perhaps with Explorer. Copy the `bin` and `lib` folders into a new directory that will serve as the root of the SML/NJ tree. With XP you should see "Extract all files" under Folder Tasks, on the left. *For discussion, we'll assume you're using `c:\smlnj` as that directory.*

Two adjustments are needed to your environment variables, which can be accessed with Start>Control Panel>Performance and Maintenance>System>Advanced (tab)>Environment Variables. (That's with Control Panel's "Category View".) Using the environment variables dialog, do two things:

(1) Under User Variables, create a new variable named `SMLNJ_HOME` with the value `c:\smlnj` (or wherever you unloaded the zip).

(2) If you already have a user variable called `PATH`, add `;c:\smlnj\bin` to the end of it. (The semicolon separates entries.) If you don't have a `PATH` variable, add it, with the value `c:\smlnj\bin`.

(3) Exit the maintenance dialogs.

You should now be able to open a command prompt window (All Programs>Accessories), type `sml` at the prompt and see it start up:

```
c:\> sml
Standard ML of New Jersey v110.57 ...
-
```

The up-arrow key can be used to recall previously typed expressions. To exit `sml`, type `^Z` (control+Z) <ENTER>

Source files for use with `sml` are plain ASCII text files—no formatting codes, markup, etc. To create an SML source file, say `x.sml`, use Notepad or any of the many third-party text editors available for

Windows. If you use Notepad, on the "Save As..." dialog, pick “All Files” as the file type. (If you don't you'll end up with a name like `x.sml.txt`.)

You should be able to load the file into `sml` with `use "x.sml"` but BE SURE that when you issue the `sml` command, your current directory is the directory where `x.sml` resides.