Installing the StreamIt Compiler

January 29, 2007

This is for the January 29, 2007 release of the StreamIt compiler and runtime system, version 2.1.1.
`$Id: install.tex.in,v 1.19 2006/09/08 19:58:15 thies Exp $`

1 Compilation

Unpack the StreamIt source tar file somewhere; this can be in your home directory, or in a shared directory. Run `./configure` from the top-level directory, which will look for several programs needed to use the compiler and create files needed by the build process. If it complains that certain programs are missing, you should install them before continuing; a list of programs required is included in the `README` file.

1.1 Source Release

For the source release, you need to add several directories to your Java `CLASSPATH`; these include the jar file for the ANTLR LL(k) parser generator, a tree for various third-party packages used by the compiler, and the StreamIt compiler and runtime proper. You also need to add the top-level StreamIt directory to your `PATH`. If you use bash, add to your .bashrc file:

```
### Location of unpacked StreamIt tree
STREAMIT_HOME=$HOME/streamit-src-2.1.1

### Location of ANTLR jar file (test: 'java antlr.tool')
ANTLRJAR=/usr/share/java/antlr.jar

### Update CLASSPATH
CLASSPATH=.:${CLASSPATH}
CLASSPATH=${CLASSPATH}:${ANTLRJAR}
CLASSPATH=${CLASSPATH}:${STREAMIT_HOME}/src
CLASSPATH=${CLASSPATH}:${STREAMIT_HOME}/3rdparty
CLASSPATH=${CLASSPATH}:${STREAMIT_HOME}/3rdparty/JFlex/jflex.jar

### Update the shell path
```

```
PATH=${PATH}:${STREAMIT_HOME}

export STREAMIT_HOME CLASSPATH PATH
```

or if you use tcsh, add to your .cshrc file

```
### Location of unpacked StreamIt tree
setenv STREAMIT_HOME  ${HOME}/streamit-src-2.1.1

### Location of ANTLR jar file (test: 'java antlr.tool')
setenv ANTLRJAR       /usr/share/java/antlr.jar

### Update CLASSPATH
setenv CLASSPATH     .:${CLASSPATH}
setenv CLASSPATH      ${CLASSPATH}:${ANTLRJAR}
setenv CLASSPATH      ${CLASSPATH}:${STREAMIT_HOME}/src
setenv CLASSPATH      ${CLASSPATH}:${STREAMIT_HOME}/3rdparty
setenv CLASSPATH      ${CLASSPATH}:${STREAMIT_HOME}/3rdparty/JFlex/jflex.jar

### Update the shell path
setenv PATH           ${PATH}:${STREAMIT_HOME}
```

Alternatively, the release comes with provided dotfiles that set up all of the possible environment variables. Note that the settings in these files *must* be customized according to your local infrastructure. In this case, you can "source" `$STREAMIT_HOME/include/dot-bashrc` or `${STREAMIT_HOME}/include/dot-cshrc`, as appropriate, after editing the files. If you want to override the settings the provided scripts set (for example, the Raw starsearch directory), set the relevant variables after reading the dot-bashrc or dot-cshrc file.

If you are trying to install both StreamIt and the Raw tools, set up Raw first; the setup process requires that the `TOPDIR` environment variable be unset. After running `make setup` in the starsearch directory, edit the dot-bashrc and/or dot-cshrc files to have the correct setting for `TOPDIR`.

You need to reload your shell's dotfiles; an easy way to do this is to log out and log in again. You can compile the source tree: run `./configure`, then `make`, from the top-level directory of the StreamIt release. After this completes, you should be able to run the compiler.

1.2 Binary release

Add the StreamIt jar file to your Java `CLASSPATH`, and the top-level StreamIt directory to your `PATH`. If you use bash, add to your .bashrc file

```
### Location of StreamIt
STREAMIT_HOME=$HOME/streamit-2.1.1

### Update CLASSPATH
```

```
CLASSPATH=.:${CLASSPATH}:${STREAMIT_HOME}/streamit.jar

### Update the shell path
PATH=${PATH}:${STREAMIT_HOME}

export STREAMIT_HOME CLASSPATH
```

or if you use csh, add to your .cshrc file

```
### Location of StreamIt
setenv STREAMIT_HOME $HOME/streamit-2.1.1

### Update CLASSPATH
setenv CLASSPATH     .:${CLASSPATH}:${STREAMIT_HOME}/streamit.jar

### Update the shell path
setenv PATH          ${PATH}:${STREAMIT_HOME}
```

Compilation for the various back ends requires extra setup. If you are compiling for Raw, you need to set up the Raw tools separately. The uniprocessor back end requires a runtime library, which can be compiled by running `./configure` and then `make` from the top-level directory of the StreamIt release.

After you complete these steps, you should be able to run the compiler.

2 Testing the Compiler

A simple test of the compiler is the "hello world" program, which prints a series of increasing integers. To run this test, type the following:

```
cd /tmp
cp $STREAMIT_HOME/apps/examples/misc/hello/HelloWorld.str .
strc HelloWorld.str
./a.out -i 10
```

This should print the numbers 0 through 9, in order.

You can also test the Java library back end using `strc`:

```
strc --library HelloWorld.str -i 10
```

This will convert the StreamIt program to Java, compile it against the Java library, and run it for ten steady-state iterations.