

Lab 1: DrHJ Setup, Async-Finish Parallel Programming

Instructor: Vivek Sarkar

1 Course Resources

Web Page : <https://wiki.rice.edu/confluence/display/PARPROG/COMP322>

Staff Email : comp322-staff@mailman.rice.edu

Clear Login : ssh <your-netid>@ssh.clear.rice.edu and then login with your password

All setup and execution for today's lab can be done on your laptop or on a lab computer. We recommend doing lab exercises in pairs, but you also have the option to do it individually on your laptop. Priority for the lab machines will be given to pairs rather than individual users.

To check if your laptop can run the course infrastructure, check that you have version 1.6.x of JDK installed. (You can type "java -version" on the command line to check the version.)

NOTE #1: It's possible that your turn-in userid is not properly set up as yet. If you discover that's the case, you can ignore the remainder of Section 2 and move to Section 3.

NOTE #2: There are known issues with running DrHJ on some Windows machines. While DrHJ works properly on some Windows computers, the following errors may be encountered on others:

- The HJ compiler is not available as an option when selecting compilers.
- The HJ compiler is unable to locate standard classes such as java.util.*.

If your Windows machine exhibits any of the above problems, then we suggest that you run DrHj on a CLEAR machine, and Xming on your Windows machine to work with the DrHJ display. Instructions on installing Xming can be found at <https://docs.rice.edu/confluence/display/ITTUT/SSH+with+X11+forwarding+on+Windows>.

2 Instructions on homework submission

You will have to use 'turnin' to submit all homeworks, including written assignments. 'turnin' is a generic script that uses Rice's CLEAR subversion system to submit your homeworks. Here are the instructions to submit each homework. As mentioned earlier, our grading scripts require that each homework be placed in a folder named *hw_n*, where *n* is the homework number. So, your solution for the first homework should be placed in a folder named "hw_1" (not HW_1, Hw_1, hw1, HW1, Hw1, etc).

You should follow the steps below to submit all your homeworks using turnin on CLEAR. Note that you can do your homework on any machine. The instructions below include steps to copy them to CLEAR and to submit them.

1. Go to the folder (in your machine) that contains all that you need to submit for your homework.
2. Zip the files you want to submit. Here 'n' refers to the homework number.

```
zip -r hw_n.zip *
```

OR

```
zip -r hw_n.zip <files/folders> // If you want to select some files/folders only
```

3. Use sftp to copy the zip file to CLEAR.

```
sftp <your-netid>@ssh.clear.rice.edu
```

```
<your-password>
```

You should have the sftp prompt '*sftp>*' now.

```
cd comp322
```

Create a new folder with the name 'hw_n' for this homework number *n*

```
cd hw_n
```

```
put hw_n.zip
```

This should show that the zip file has been copied into the folder hw_n

4. Login to CLEAR

```
ssh <your-netid>@ssh.clear.rice.edu
```

```
<your-password>
```

5. Go the folder hw_n

```
cd hw_n
```

6. Unzip the file

```
unzip hw_n.zip
```

7. Delete the zip file

```
rm hw_n.zip
```

8. Go one level up, to the "comp322" folder

```
cd ..
```

9. Turnin the folder hw_n

```
turnin comp322-S12:hw_n
```

This should show all your files being added to the subversion.

10. Your submission is complete. If you have problems, just email your submission zip file to comp322-staff@mailman.rice.edu.

To find out more about the turnin command type the following while logged in to CLEAR: *turnin -help*

3 DrHJ Setup

3.1 Download and Install

DrHJ is a pedagogic IDE for HJ that will be used at the start of COMP 322. You can use DrHJ to edit, compile and run HJ programs on whichever machine DrHJ is launched on. In later labs, you will be exposed to command-line interfaces to compile and run HJ programs on different parallel machines.

- Download the jar file for DrHJ from
<http://www.cs.rice.edu/~vsarkar/downloads/drjava-r5450-hj-1.3.a.jar>
- A link to the above jar file can be obtained by following these links from the course web page: "HJ Info" → "HJ Download and Setup", and then searching for "Download the jar file corresponding to DrJava-HJ"
- Below, we will refer to the jar file as 'drjava-hj-version.jar'.

3.2 Testing

Here are the instructions to compile and run HJ programs using the DrHJ IDE.

- Download the HelloWorld.hj program from the Code Examples link for Lab 1 (not Lecture 1) in the course web page
- Open the DrHJ IDE
java -Xms2000m -Xmx2000m -jar drjava-hj-version.jar
(The “2000m” amounts in the -Xms and -Xmx options specifies the Java heap size as 2GB. You can omit these options if the default memory settings work fine for you.)
- Now you should have the DrHJ IDE running.
- Open an HJ program.
Click on the open button in the top panel
Navigate to the folder containing HelloWorld.hj
Select HelloWorld.hj and click open
- Compile the HJ program
Click on the Compile button in the top panel
- The 'Compiler Output' tab in the bottom panel should show 'Compilation Completed'.
- Go to the 'Console' tab in the bottom panel and check if there were any errors during compilation. All errors will be shown in RED.
- Go to the 'Interactions' tab in the bottom panel. Run the program by typing the following.
run HelloWorld

4 ReciprocalArraySum Program

We will now work with the ReciprocalArraySum program introduced in Monday's lecture.

- Download the ReciprocalArraySum.hj program from the Code Examples link for Lab 1 (not Lecture 1) in the course web page
- Compile the program by clicking on the Compile button.
- Run the program by typing the following in the Interactions page:
run ReciprocalArraySum
- Experiment with smaller array sizes (specified as an integer N):
run ReciprocalArraySum N
- What speedups do you see for different values of N? (Watch out for OutOfMemoryError's when experimenting with values of N > 50,000,000.)

5 PrimeSieve Program

We will now work with the PrimeSieve program introduced in Wednesday's lecture.

- Download the PrimeSieve.hj program from the Code Examples link for Lab 1 (not Lecture 1) in the course web page

- Compile the program by clicking on the Compile button.
- Run the program by typing the following in the Interactions page:
run PrimeSieve
- Experiment with smaller array sizes (specified as an integer N):
run PrimeSieve N
- This is a sequential program with timing routines inserted for the sieve computation. Can you transform this into a parallel program that runs faster than the sequential program? (For today's lab, assume that your computer only has two processors.)