Language Processors Lab Week 1

These sheets introduce you to the Linux/Unix Java programming environment to be used on the module by getting you to compile and run a program. It is also a gentle introduction to Language Processors since you will compile and execute your first interpreter: The Reverse Polish Notation (RPN) Calculator.

You will be expected to be able to copy and manipulate program files and directories or web locations named in lab sheets, modify and compile program files, and execute the compiled programs. If you are not familiar with the Linux labs you should also get a copy of any *Computing Services* documentation available.

The Linux Environment

Working with the KDE windowing system under Linux

Each computer is attached by a network to several file systems. Under Linux, the fact that there are multiple file systems is something you don't normally have to think about. All the available systems just appear as different directories (folders) in a large'tree' of directories. The tree is upside-down with its root at the top. The Root Directory is called "`/`". We will use the words 'directory' and 'folder' to mean the same thing.

The Linux labs use Suse Linux and the KDE windowing system as its main user interface. You will notice that KDE has a Windows-like **taskbar** (the control panel) at the bottom of the screen with a button with the Suse logo (K Menu) on the bottom-left. As on Windows, clicking on this reveals a menu that launches applications and utilities.

Your CSD home directory and the KDE File Manager

When logged in as a Linux user you have a *home directory*. This is a place in the file system which belongs to you: the place where you can create, modify and delete files and folders. When you start the file manager (the button labelled with a small house icon on the control panel is the easiest way to start it), your home directory is the one it will display. You can also get the file manager to take you to your home directory at any time simply by clicking on the home icon on the task bar at the top. Use the file manager to go to your home directory now. Does it contain any files? Does it contain any hidden files?

Unix sessions, Terminals, and your directories

A Unix session is made up of a series of commands typed and executed by the user from the command line at a *Unix terminal* or *Unix shell* window. A Unix shell window can be fired up from the K menu or from one of the icons on the taskbar: the one with the terminal screen on it, marked with **Konsole**.

Fire up a shell window. The first question you may want to ask is 'Where am I?'. You can answer this question by typing:

```
pwd
```

The `pwd` command displays the name of the *current directory*, that is, the name of the directory where you are currently located. By default you will always start a Unix session from your home directory. Therefore, the first `pwd` command should display a directory name starting with '`/`' (the root directory) and ending with your user name. In my case, I got the following:

```
/u4.bristol/s00/sbbc287
```

Now type:

```
ls
```

The `ls` command displays the contents of the current directory - the previos command shoul've listed the contents of your CSD home directory. You can also display the contents of any directory by passing a directory name as argument to the `ls` command. For example, in order to see the contents of the `KDesktop` directory simply type:

```
ls KDesktop
```

Besides arguments, Unix commands can also accept special options listed after a the '-' (minus) symbol. For example, these are valid `ls` options:

`ls -l`	Lists permissions, ownership and last update info
`ls -a`	Lists hidden files as well
`ls -al KDesktop`	Combined options

You can see a brief explanation of almost all Unix command by using the `man` and `info` commands/ For example, type `man ls` and `info ls`.

Creating a directory

In this section you will learn how to create new directories and move through the directory tree. The command `cd dirname` changes the current directory to the directory name specified as argument i.e. `dirname`. First, make sure the current directory is your home directory. You can always move to your home directory by typing the following:

```
cd
```

The `cd` command with no arguments will always get you to your home directory. Now, execute the following series of commands:

`ls`	Check `KDesktop` is one of the names listed
`cd KDesktop`	The current directory should be now be `KDesktop`
`pwd`	Check your current directory
`ls`	Inspect the contents of `KDesktop`
`cd ..`	Move to the parent directory (one directory up)
`pwd`	Check you are back to your home directory

New directories can be created with the `mkdir` command. We would like to create the `LanguageProcessors` directory - you can pick any name you want - in order to store all our lab work. Do the following:

`mkdir LanguageProcessors`	Creates new directory `LanguageProcessors`
`cd LanguageProcessors`	You should be in `LanguageProcessors now`

Java environment and programming

The Java environment we will use is Sun's JDK (Java Development Kit, now also known as the SDK), which can be considered to be the reference Java environment, and provides (amongst other things) a compiler, interpreter and debugger. These are not integrated into a closed environment, instead you need to use the Unix file system as above, a Unix editor to be able to create and modify programs (see below), and learn how to compile and execute Java programs by invoking the correct Java SDK commands from the command line. In particular, we will use the `javac` and `java` commands to compile and execute Java programs.

Before we continue, make sure your current directory is `LanguageProcessors`[1].

Compiling Java programs on Unix

Next, we need to get the source Java files that make up the RPN Calculator. The files have been packaged into an archive file named `rpn.tar` by the Unix utility program `tar`. Do the following:

[1] Or the directory you have chosen for your Language Processors work

- Download the `rpn.tar` file from the module's Moodle directory `lab1` or from the following link `http://www.soi.city.ac.uk/~sbbc287/rpn.tar`.
- Make sure `rpn.tar` has been copied to your `LanguageProcessors` directory. It may be that the file system unpacks the tar file automatically. If that's the case, copy the `rpn` directory to the `Language Processors` directory.
- If the file system has not unpacked the `rpn.tar` file automatically, unarchive the files with the following command:

  ```
  tar -xvf rpn.tar
  ```

  This creates the `rpn` directory and copies all related `.java` files into it.
- Change to the `rpn` directory (`cd rpn`) and list its contents (`ls`). You should get the following:

  ```
  Lexer.java              StackMachineException.java
  LexerException.java     Token.java
  RPN.java                TokenKind.java
  StackMachine.java
  ```

  The RPN Calculator (lexical analyzer and stack machine) is made up of 7 `.java` files.

The Java compilation command is `javac`, which will compile single or multiple `.java` files and generate `.class` (Java bytecode) files. In order to compile the RPN Calculator execute the following command:

```
javac RPN.java
```

which will compile the `RPN.java` source file, and all Java source files needed by `RPN.java`. We are compiling `RPN.java` because we know that it contains the `main` method (see next section). *Your first compilation will take a long time; subsequent ones take much less time.*

The `javac` command generates a new `class` file per compiled class. Succesful execution of the command above should generate 10 new `.class` files. Verify this by executing the `ls` command.

Running Java programs on Unix

Java programs are executed with the `java command`:

```
java  MainClass
```

where `MainClass` is any class which contains the `public static void main(String[] args*)` method. In our example, the `main` method is part of the `RPN` class. You can quicly check this by invoking the `cat RPN.java` or `more RPN.java` commands - with the latter, press the spacebar in order to see the rest of the file. Thus, the command

```
java RPN
```

will execute the RPN Calculator

The Reverse Polish Notation (RPN) Calculator

Typical calculators use Infix notation, that is, they work with arithmetical expressions where operators are placed between its operands. For example, the arithmetical expression `3 + 5` has operator `+` between operands 3 and 5. This notation is easier to read but has a few shortcomings:

- Complex expressions require the use of parenthesis e.g. if we want to multiply the result of `3 + 5` to the result of `11 - 4` we need to write `(3 + 5) * (11 -4)`.

- The calculator needs to read the whole expression before it starts its computation. It may also need to store temporary results, as with the previous example.

These shortcomings are addressed by Reverse Polish Notation (RPN). RPN is a mathematical notation where operators are placed after their operands. It is also called Postfix notation. For example, the infix expression `3 + 5` is written `3 5 +`, and `(3 + 5) * (11 - 4)` can be expresed by RPN expression `3 5 + 11 4 - *`.

When executed, the RPN Calculator displays a couple of greeting lines, followed by the interpreter's prompt `Enter Expression >` as shown below:

```
Reverse Polish Notation Calculator
Enter RPN expression after prompt and hit Enter
Or just hit Enter to Exit
```

Try a few examples:

```
Enter expression > 6

Enter Expression > 3 5 +

Enter Expression > 3 5 + 11 4 - *
```

Are the results what you expected? The calculator also reports on errors due to wrongs input. For example `4 +` should report the wrong number of arguments.

Modifying the Calculator

The calculator has a small error: it performs division in the wrong order. For example, `8 2 /` gives us 0 insted of 4.

We need to modify the StackMachine.java file. Follow this steps:

- Fire up one of the text editors provided by the system (check K Menu/Utilities/Editors) and load the StackMachine.java file.
- Look for the place in the program where division is performed `op1 / op2` and change the order of the operators.
- Save the file.
- Re-compile the program with `javac RPN.java`, and run it again (`java RPN`).

Doe sit work?

Logging Out

Always remember to log out when you have finished working at a machine. To logout from a Linux machine, press the button (K button?) at the left of the control panel at the bottom of the screen, and select `Logout` from the menu.

If you leave a machine without logging out you are putting your own work at risk.