

An Introductory Guide to Using
Linux

Redeemer University College

Department of Computer Science

Version 2.03

September 2007

Derek Schuurman

Contents

1	Introduction	3
2	Getting Started	3
2.1	The CS Server and the Science Computing Lab	3
2.2	Opening a Terminal Window	3
2.3	User ID and Password	4
2.4	Logging On	4
2.4.1	Remotely logging in to the CS Server	4
2.4.2	Logging on directly to a Linux Workstation	5
2.5	Logging Off	5
2.6	Disk Quotas	5
3	Basic Console Commands	5
3.1	Managing Files	6
3.2	Managing Processes	6
3.3	Miscellaneous User Commands	7
4	Printing	7
4.1	Submitting a Print Job	7
4.2	Managing Print Jobs	8
5	E-Mail	8
5.1	Checking Your E-Mail	8
5.1.1	Text-Based E-mail Access	9
5.2	Forwarding Your E-mail	9
6	Setting up Web Space	9
7	Using a Text Editor	9
7.1	Using L ^A T _E X	11
7.2	Formatting Source Files	11
8	Using USB Flash Drives and CD-ROMs	12
9	X-Windows Applications	12
10	Scientific and Mathematical Applications	13
10.1	Maple	13
11	Compiling and Running a C/C++ Program	13
11.1	Compiling	14
11.2	Running a Program	14
11.3	Logging the Program Output	14
11.4	Using a Debugger	14
11.5	Library Documentation	15

11.6 Using Makefiles	16
11.7 Other Tools	16
11.8 C++ Integrated Development Environments	16
12 Creating and Running JAVA Programs under Linux	16
12.1 Compiling a JAVA Program	16
12.2 Running a JAVA Program	16
12.3 JAVA Integrated Development Environments	16
12.4 Other Tools	17
13 Utilities	17
14 CS Account and Lab Policies	18
15 Other Sources of Information	18

1 Introduction

An operating system is software to control the hardware of a computer. Many different Operating Systems have been developed over the years. *Linux* is free Unix-type operating system originally written by a graduate student named Linus Torvalds at the University of Helsinki in Finland. Linux grew over time with support from developers around the world and continues to grow increasingly popular. It is distributed under the GNU general public license (GPL) and is freely available. This document is by no means complete and is provided as-is and is strictly intended as an introduction for students enrolled in Computer Science at Redeemer University College.

2 Getting Started

Before you can use the CS computing facilities at Redeemer University College, you must have an account set up by the Computer Science Department. Students enrolled in a computer science course will all be given an account on the CS server. In addition, there is a dedicated computer Lab in room 223 which provides students with access to computer workstations. This lab serves not only as a teaching resource but as a place for computer science students to congregate after hours. To ensure each computer science student has ample access to computing resources, the use of the lab is restricted to CS, math and physics majors (and non-majors enrolled in science courses). Access to the lab is restricted using a combination lock. The combination will be provided by your professor.

2.1 The CS Server and the Science Computing Lab

An account on the CS server enables students to log in remotely from the dorms or over the Internet or you may log on to one of the workstations in the Science Computing lab. Your password and “home” directory files will follow you where ever you log in. The computers in the Science Computer Lab have been named after various people linked (some more closely than others) with the reformation, and particularly the “reformed” wing thereof, and are networked to the CS Server as illustrated in Figure 2.1. A laser printer is available in the lab for student use, but you will need to provide your own paper. The laser printer is suitably named *laser* and is available as the default printer.

2.2 Opening a Terminal Window

the Linux desktop environment provides many menu-driven tools for clicking options and launching programs. However, most things can be also be accomplished by typing commands directly within a *Terminal Window*. Many of the commands described in this manual are intended to be entered within a Terminal Window. To open a terminal window, simply right click on your desktop and select “Open Terminal”. The terminal window launches a *shell* that provides a prompt from which commands can be typed or programs launched. You may wish to setup a shortcut in your task bar to open a terminal window by clicking on an icon.

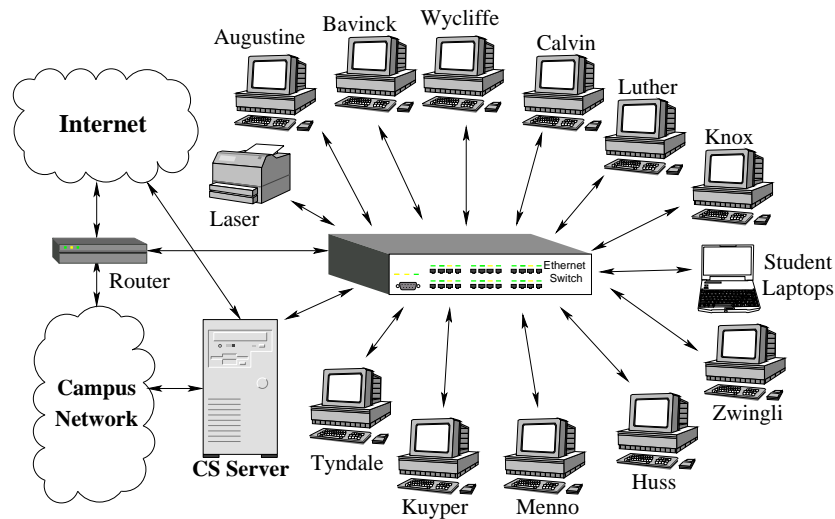


Figure 1: Diagram of the Science Computing Lab.

2.3 User ID and Password

Typically, your User ID is comprised of your first initial followed by your surname. For example, *Martin Luther* would have the User ID *mluther*. Your password will be assigned with your User ID. You should change your password as soon as you log into your new account.

If you are using one of the workstations in the CS Lab, you may change your password in a terminal window by *logging on to the CS server* and then typing:

```
passwd
```

Note that it may take a while before the password change propagates throughout all the workstations in the CS and robotics lab. You must use a secure password, which excludes most common dictionary words and names (so you won't be able to use the name of your pet hamster). Passwords protect your computer account from unauthorized access. Passwords are comprised of letters, numbers, and punctuation and should be at least 6 characters in length.

2.4 Logging On

Users may login to a Linux system directly at a Linux workstation or they may log-in remotely. Users may log in remotely using secure shell (*ssh*).

2.4.1 Remotely logging in to the CS Server

Students should be aware that using programs such as *telnet* over the Internet is insecure since when you log-on your *userid* and *password* are sent in the clear. This means that

it is possible for someone to intercept your password and use it to gain access to the server. Students who wish to log-in from home over the Internet must use a secure-shell (SSH) client. Windows SSH clients, such as *Putty* are also available allowing you to remotely log in from home. SCP (secure copy) programs are also available for transferring files between home and the CS server. *WinSCP* is an example of a free SCP client that is available for Windows. Ask your instructor if you require more information on secure shell clients.

2.4.2 Logging on directly to a Linux Workstation

To use a Linux workstation you must also have a valid userid and password. Your userid and password will be the same for any machine in the Linux Lab as it is on the CS server. The files in your *home directory* and your e-mail will also be available from any computer in the Linux Lab. Using a Linux Workstation allows the user to work in X-Windows which provides a friendly graphical user interface. You may also use secure shell to connect to other Linux machines from any Linux workstation.

2.5 Logging Off

Once you are finished using Linux you must properly log-off the computer. To log-off simply issue the command `logout` or `exit` or click the appropriate icon. If the workstation locks-up or malfunctions simply move to another terminal and send an e-mail describing the problem to the CS sys admin person. Note that improperly shutting down a workstation may cause damage to the computer's file system.

2.6 Disk Quotas

Due to limited disk space, the CS server uses disk quotas to prevent the home partition from filling up and to ensure that the space does not exceed the disk capacity.

When disk quota is exceeded you will typically be unable to log into Linux via the GUI interface and you will not be able to create or save new files. To check your quota, log into the CS Server and type `quota` in a shell window. If your disk usage exceeds your allocated quota, you must delete some files. As always, please keep an eye on you disk space and remove any unnecessary files. If you need space, try removing core files and cleaning your web browser cache.

If you temporarily need *a lot* of space (on the order of several gigabytes) then use the `/tmp` directory on the CS workstations. Note that this directory can be read by all users and is cleaned regularly. You may also use USB flash drives and burn CD/DVDs if necessary.

3 Basic Console Commands

When you open a terminal windows or log in remotely, you will be running in a Linux *shell*. In simple terms, a shell is a command interpreter which provides a rich set of

commands and can also be used to launch a wide variety of programs. The default shell in the lab is the *bash* shell, but other shells are also available.

Your default shell can be changed to a different shell other than *bash*. For example, to change your default shell to the Z-shell, type:

```
sh -s /bin/zsh
```

Various other shells that are available include the C-shell and the TC-shell.

An on-line manual referred to as the *man* (manual) pages provides help on the many commands and programs that can be used. The syntax for invoking the *man* utility is as follows:

```
man command-name
```

and information regarding the specified command will be displayed on the screen. To find out more about the *man* utility simply type

```
man man
```

from the command prompt. There are several basic system commands which are commonly used. Many of the commands listed below have additional options which may be invoked. Consult the *man* pages for more information.

3.1 Managing Files

Command	Description
<code>cd <directory></code>	Changes the current working directory to <directory>
<code>pwd</code>	Displays the name of the current working directory
<code>mkdir <directory></code>	Create a new directory called <directory>
<code>rmdir <directory></code>	Removes the directory called <directory>
<code>ls</code>	Display a list of files find Search for a file
<code>cp <f1> <f2></code>	Copy a file from the source<f1> to the destination <f2>
<code>rm <filename></code>	Remove a file <filename>
<code>mv <f1> <f2></code>	Move a file from <f1> to <f2>
<code>ftp <host></code>	Transfer files to and from <host>
<code>diff <f1> <f2></code>	Display the difference between file <f1> and file <f2>
<code>cat <filename></code>	Displays the contents of <filename>
<code>more <filename></code>	Displays contents of <filename> pausing when screen is filled
<code>head <filename></code>	Displays contents at the beginning of <filename>
<code>tail <filename></code>	Displays contents at the end of <filename>
<code>wc <filename></code>	Displays the number of newlines, words, and bytes in <filename>

3.2 Managing Processes

Command	Description
<code>ps</code>	Display a list of processes
<code>kill <pid></code>	Terminate the process with process ID <pid>

3.3 Miscellaneous User Commands

In addition to the commands described above there are numerous other commands some of which are summarized in the following table:

Command	Description
<code>clear</code>	Clears the terminal display.
<code>date</code>	Display the current date and time.
<code>df</code>	Report the amount of free disk space available
<code>ncdu</code>	Show a summary of disk utilization
<code>du</code>	Report the amount of disk usage
<code>find <path> <conditions></code>	Utility for finding files
<code>locate <search string></code>	Utility for locating files
<code>finger <user></code>	Display data about the specified user
<code>free</code>	Report information about memory usage
<code>hostname</code>	Display the name of the current host system
<code>talk <user></code>	Talk to user on your machine or on another host
<code>uptime</code>	Display the current time stats
<code>who</code>	Display the users currently logged-on the system
<code>rwho</code>	Display the users currently logged-on various computers
<code>clamscan</code>	scan for viruses
<code>whoami</code>	Display your User ID

These commands only represent a portion of the user commands available under Linux.

4 Printing

The CS server also acts as the the print server for the CS lab and all print jobs are spooled to that computer before being printed.

Please remember that printers are to be shared by all students in the lab. All students must provide their own paper for printing.

4.1 Submitting a Print Job

To print to the printer in the Linux Lab, select the “print” option from the menu. To print a text file from the command line, simply type:

```
lpr -P printername filename
```

which will send the file specified by `filename` to the printer specified by `printername` (currently, *laser* is the only usable option).

Alternatively, users may wish to use “enscript” which provides nicer output by converting text files to Postscript. Enscript provides many different options for headers, footers, and special formatting to customize printouts. To use enscript type:

```
enscript filename
```


which will send the specified file to the default laser printer. Enscript provides a number of filters that can be used to “pretty print” your code. For example, to print a Java program type:

```
enscript -G -Ejava filename
```

To save trees, you can print two pages at a time in landscape mode using the following command:

```
enscript -r -2 -G -Ejava filename
```

4.2 Managing Print Jobs

To check the status of the print queue type:

```
lpq
```

Printer jobs may also be managed and monitored using a web browser. To monitor the CS Lab printer, open the URL:

```
http://cs.redeemer.ca:631/jobs
```

To remove a job from the print queue type:

```
lprm job-number
```

where the print job number is as shown in the queue.

To remove all print jobs from the printer type:

```
lprm -
```

5 E-Mail

With your account on the CS server you will also receive an e-mail account. Your e-mail address will be:

```
userid@cs.redeemer.ca
```

where *userid* is your userid you use to log-in on the CS server. Please check your CS e-mail regularly since any computer science class announcements will be sent your computer science e-mail account.

5.1 Checking Your E-Mail

There are two ways you can check your e-mail. You may use a text-based e-mail client or you may use a web-based e-mail client.

5.1.1 Text-Based E-mail Access

You can send and receive e-mail using the *Alpine* mail client which can be invoked by simply typing `alpine` at the command line on the CS server. Pine provides a simple interface that runs directly in your shell on the CS server. Once in Pine you can check your e-mail inbox by typing "L" or compose a new message by typing "C".

5.2 Forwarding Your E-mail

If you wish to forward your e-mail to another account or to your main campus e-mail account simply create a text file in your home directory called `.forward` containing the e-mail address to which you wish to forward your e-mail as follows:

```
user@host.domain
```

where `user@host.domain` is the e-mail address to which you want your e-mail forwarded.

6 Setting up Web Space

The CS Server is running the open-source Apache web server to publish web pages. Each account on the CS server includes some personal web space. This will allow you to create web pages which may be required in some of your courses.

To begin, you will need to create a subdirectory called `public_html` in your home directory. Transfer your HTML and possibly any Java Applet CLASS files into this directory. Ensure that the web files and the `public_html` directory are world readable by using the following Linux command:

```
chmod 755 public_html/
```

You will need to ensure that each new file placed in this folder will also have the necessary permissions.

You are able to view your files by pointing your browser to the URL:

```
http://cs.redeemer.ca/~userid/
```

where `userid` is your user-id on the CS server.

7 Using a Text Editor

Text editors are used to edit files such as program source files. There are several editors available which you may use. Some common editors include X-Emacs and `vi`. The editor you use is your choice but it is recommended that you use X-Emacs for editing program files. To help get started, a very brief summary of the basic commands for several different editors are summarized below. These editors possess more powerful and advanced features which the student may wish to learn once he/she has mastered the basics. For more information consult the man pages or the web.

If you are using a Linux Workstation, it is recommended that you use `gedit` for editing program source files. `Gedit` is a general-purpose editor that provides some useful features for editing programs including syntax highlighting.

Another editor you may wish to consider using is Emacs, although Emacs is much more than just an editor. When running under X Windows, X-Emacs provides a toolbar menu of the various commands. However, when running in a text terminal the user must use various control keys to issue the editor commands.

The list of basic commands is summarized in the table below:

Command	Description
Ctrl-x Ctrl-s	Save
Ctrl-x Ctrl-c	Exit
Ctrl-h	Help
Ctrl-h t	Emacs tutorial
Del	Delete preceding character
Ctrl-g	Cancel Command
Ctrl-x	Search (in the forward direction)
Ctrl-x	Move down one page
Ctrl-x u	Undo last change
Esc <	Move to the beginning of the file
Esc >	Move to the end of the file

Use the help command (`ctrl-h`) to list additional commands or for more information.

Another popular editor is the *vi* editor. To edit a file in *vi* type:

```
vi filename
```

vi has two basic modes of operation: command mode and input mode. Text may only be entered while in input mode. The user must hit the escape key to exit input mode and enter command mode before any editing commands are entered. The list of basic commands is summarized below.

Command	Description
$\uparrow\downarrow\rightarrow\leftarrow$	Move up, down, right, left (if terminal emulation is configured right)
0	Move to the start of the current line
\$	Move to the end of the current line
escape key	Ends input mode and returns to command mode.
a	Appends text after current position and enters input mode
A	Appends text at the end of the current line and enter input mode
i	Inserts text at current position and enters input mode
o	Inserts a line below the current line and enters input mode
rc	Replaces the current character with c
R	Replaces the current text with the text you type after R
x	Deletes current character
dw	Deletes a word
dd	Deletes current line of text
:q	Quits and exits from vi (only if no changes were made)
:q!	Quit and exits without saving
:w	Saves the file you are editing
:wq	Saves the file you are editing and quits
/pattern	Searches the file for text matching pattern

7.1 Using L^AT_EX

LaTeX is a document preparation system for high-quality typesetting and for creating “beautiful documents”. It often used for making mathematical, technical or scientific documents, but it can be used for almost any type of document. There are numerous resource available on the web describing how to edit a LaTeX document file.

It is recommended to give your L^AT_EX file a `.tex` file extension. Once you have made your L^AT_EX file, you can create a DVI (Device Independent file format) file. To do this, run the following from the command line:

```
latex file.tex
```

If all goes well, this should generate DVI file (along with some other files). The DVI file can the be converted to a postscript file as follows:

```
dvips -o file.ps file.dvi
```

7.2 Formatting Source Files

Some editors (such as X-Emacs) will automatically indent and format your program source files but many editors do not. To ensure that the format of your program source files comply with the *CS department coding standards*, you may use the `indent` program to “beautify” your source code. This program will automatically adjust the spacing and indentation of your source file as required. To use this program type:

```
indent -bli0 -i3 <source-file>
```

where `<source-file>` is the name of your source file.

8 Using USB Flash Drives and CD-ROMs

You may choose to transfer files to and from the CS Lab using a USB Flash Drive or CD-ROM (floppy disks are fragile and are discouraged). Simply insert the CD-ROM or USB Flash Drive and, after a few moments, the device should be automatically *mounted* and a link to the device should appear on your desktop. Once a drive is mounted, one can access files in the directory where the drive has been mounted. The device will logically appear in the `/media` directory on the computer.

USB Drives are a convenient way of transferring your work between home and school. The workstations in the lab are equipped with USB ports on the front panel where USB drives may be inserted. After plugging in a USB device, you may confirm that the OS has recognized the new device by typing:

```
/sbin/lssusb
```

to list all the USB devices that are currently detected.

To view a list of currently mounted filesystems, simply type:

```
mount
```

Once you have finished using a drive it must be unmounted. Media may be unmounted by right-clicking the icon on the desktop and selecting the “Unmount Volume” option. Alternatively, devices may be unmounted using the `umount` command.

Make sure you have unmounted any drives you have used before logging out of a workstation.

9 X-Windows Applications

If you are using Linux in the computer science lab you will be running in the X-Windows environment. X-windows provides a friendly graphical interface and many useful applications. X-Windows can be used with several different window managers such as KDE or Gnome. The window manager you wish to use is a matter of personal preference. You may also customize your X-window desktop with your own icons and backgrounds.

Some of the many X-Window applications that are installed include the following:

Program	Description
xemacs	emacs text editor with a graphical user interface
gedit	A text editor
eclipse	An integrated development environment
evince	A PDF document viewer
firefox	A world wide web browser
xcalc	A calculator
gimp	A powerful image processing program

In the CS Lab, it is recommended that you set the browser cache directory to a local temporary directory rather than using your home directory. Your home directory is a network drive, so it is better to cache temporary files locally on your workstation. To do this in firefox, type:

```
about:config
```

In the address bar, right-click in the parameter list and add a new string parameter called *browser.cache.parent_directory* and set the value of this parameter to */tmp*.

There are several office programs available on the Linux workstations including LibreOffice. LibreOffice is a freely available office productivity suite available from:

```
http://www.libreoffice.org
```

It includes key desktop applications such as a word processor, spreadsheet, presentation manager, and drawing program. It has numerous features and uses a friendly user interface. LibreOffice includes the following components:

Writer A word processor

Calc A spread sheet

Impress A presentation program

Draw A drawing program

Math A formula editor

To start LibreOffice you may select the office programs from within the main menu. LibreOffice is also capable of reading many popular office document file formats. Consult the on-line help for more information.

10 Scientific and Mathematical Applications

There are several scientific applications available on the Linux workstations.

10.1 Maple

Maple is an advanced, mathematical program which is capable of powerful symbolic computation. To start Maple simply type:

```
maple
```

To start Maple in graphical mode simply type:

```
xmaple
```

Maple is also available from the mouse-driven drop-down menu (under the "Education" category).

11 Compiling and Running a C/C++ Program

Before a program can be compiled, a source file must be created using one of the text editors described in section 7. Once a program source file is complete, it must be compiled before it can be run. The compiler which we will use under Linux is the GNU C Compiler (`gcc`). There is also a C++ compiler available (`g++`).

11.1 Compiling

To compile your source code type the following at the prompt in a terminal window:

```

  compiler          Sets output filename
  {gcc}  { -Wall }  { -o myprogram } { myprogram.c }
  warnings on                                     Source filename
```

The gcc compiler has numerous other command line options which you may use. For more information consult the man pages. Note that the C++ compiler can be invoked by using g++ in place of gcc.

11.2 Running a Program

To run a program, simply type the output filename. If no output filename was provided to the compiler the default filename will be a.out. To run a file in your current working directory do not forget to specify a ./ in front of the program name to specify the path as the current directory.

11.3 Logging the Program Output

Most computer science assignments will require that you submit a log of the actual input and output of your completed program. This can be accomplished by using the script command. When you are ready to log your program simply type:

```
script logfile
```

where logfile is the name of the logfile. Start logging before running your program and all your subsequent output text will be logged to the log file you have specified. To append to an existing log file use:

```
script -a logfile
```

To end your logging session simply type exit. If you are running script from within an X-Window console you should be aware that any colours that appear may cause strange control characters to appear in the log file. This can be avoided by using a simple console window such as a plain xterm window.

11.4 Using a Debugger

A debugger allows you to see what is happening “inside” a program while it is executing. The GNU debugger can be used to debug your C and C++ programs.

To use the debugger, compile using the -g option as follows:

```
gcc -Wall -g myprogram.c -o myprogram
```

where myprogram.c is the C source file. The -g option tells the gcc compiler to include information in the output file for use by the debugger. To start the GNU debugger, type:

```
gdb myprogram
```

where `myprogram` is the name of the executable file you wish to debug. At the prompt in the debugging window you can now enter various commands. For example, to set a breakpoint for the start of the `main()` function by type:

```
break main
```

A breakpoint stops program execution allowing you trace your program step-by-step. To begin running your program type:

```
run
```

You may begin single-stepping through each line of your code by typing “step” (or just “s”) in the debugging window. When your program ends, you may quit the debugger by typing:

```
quit
```

The debugger includes several additional commands that are useful when debugging code. Some of the `gdb` commands which can be entered after the (`gdb`) prompt are listed in the table below:

gdb Command	Description
<code>run</code>	Start running a program
<code>break ;function;</code>	Set a breakpoint at the start of a function
<code>break ;line number;</code>	Set a breakpoint at a linenumber
<code>info break</code>	list all breakpoints
<code>disable ;breakpoint;</code>	disables a breakpoint
<code>enable ;breakpoint;</code>	enables a breakpoint
<code>continue</code>	resume running the program
<code>list</code>	list the next source lines
<code>next</code>	execute the next statement
<code>step</code>	same as next, but step into a function
<code>print ;variable;</code>	print the value of a variable
<code>display ;variable;</code>	display the value of a variable after each step
<code>set ;variable;=;value;</code>	assign a new value to a variable
<code>help</code>	display a list of <code>gdb</code> commands
<code>help command</code>	display help on a specific <code>gdb</code> command
<code>quit</code>	quit <code>gdb</code>

The debugger can also be used to examine core dumps when segmentation faults occur. For more information on using the debugger, type `help` at the debug prompt or type `man gdb` for more information.

11.5 Library Documentation

The `man` pages include documentation on most of the functions found in the various C Libraries. To display information on a C function type the following:

```
man -S3 function_name
```

where `function_name` is the name of the function you want documentation for. Documentation also includes the names of the `#include` header files that must be included for a given function.

11.6 Using Makefiles

The `make` utility can be used to automatically determine which pieces of a large program need to be recompiled. Type `man make` for more information.

11.7 Other Tools

There are many other tools that have been developed to assist C programmers. One such tool is `splint` which is a tool for statically checking C programs for security vulnerabilities and coding mistakes. To use `splint` type:

```
splint <source-file>
```

where `<source-file>` is the name of your source file.

11.8 C++ Integrated Development Environments

Most of your coding will not be done from the command line, but will be done with the help of an IDE (Integrated Development Environment). In the CS Lab, the *Eclipse* IDE has been installed for C/C++ coding and development.

12 Creating and Running JAVA Programs under Linux

It is also possible to work with JAVA programs with Linux.

12.1 Compiling a JAVA Program

To compile a JAVA program type the following at the prompt:

```
javac myprogram.java
```

12.2 Running a JAVA Program

To run a JAVA program type the following at the prompt:

```
java myprogram
```

12.3 JAVA Integrated Development Environments

BlueJ is a IDE that is well-suited for students learning Java for the first time. To start the BlueJ IDE simply type:

```
bluej
```

Eclipse provides a full-featured development environment for Java. To start the Eclipse IDE simply type:

```
eclipse
```

Many of your courses will require you to submit a Java JAR file. To create a JAR file in Eclipse, do the following:

- Select File/Export in Eclipse and select JAR file
- Select the project you want to export and choose the options to include class files as well the Java source files
- Enter a file name for the output JAR file
- Select the option to compress the contents of the JAR file
- Specify the class of the application entry point where the "public static void main" method resides
- Click "Finish"

You can test your JAR files by trying to run them by typing:

```
java -jar myprogram.jar
```

To learn more about eclipse, see: <http://www.eclipse.org/>.

12.4 Other Tools

There are also other tools to support object-orientated program design. In particular there are several tools available for drawing UML diagrams. Two programs available in the CS Lab that provide UML drawing capabilities are *umbrello* and *violet*.

Javadoc is a tool for automatically generating documentation from Java source code. The Javadoc tool is also available in the CS Lab.

13 Utilities

As you become more familiar with Linux you may wish to explore some of the more advanced utilities. Some of these are listed below:

Command	Description
grep exp file	Searches for expression exp in file
make	Keeps a set of programs current
aspell	Spell checker
gv	Postscript viewer
gzip	compress and uncompress files
cal	display a calendar
tar	Creates/extracts an archive of files

14 CS Account and Lab Policies

Students who receive computer science accounts are expected to use their computer accounts in accordance with the policies established by college. Students with accounts for the computer lab may use the room at any time that the building is open. In addition there are several common sense rules that must be observed:

1. People who need to work on programming assignments have priority for using the computers
2. Do not give the door combination code to anyone
3. Computer workstations are to be kept free of excess clutter
4. No food is allowed in the lab and only drinks in sealed containers are allowed in the lab
5. *Workstation consoles should not be locked for more than 10 minutes*
6. Use headphones if you wish to listen to music

Students who do not respect the rules or abuse the equipment will have their access privileges revoked. Anyone who observes inappropriate use of the computing facilities can report incidents to:

`abuse@cs.redeemer.ca`

15 Other Sources of Information

This document is meant to be a brief introduction to Linux and is by no means complete. For more information the student is encouraged to seek other resources. Many reference books have been recently published on Linux. In addition, Linux includes a set on online man pages which contain details on most commands. Finally, there are a variety of Internet resources some of which are listed on the course homepages.