

CS1110 Introduction to Systematic Programming

2nd Practical Class Week 3

By now you should be familiar with

the basics of the CDE graphical user interface.
how to edit, compile, link and run simple Ada programs.

You should know how to convert an algorithm into a complete Ada program (as in the worksheet for Lab2.2) and how to read error messages and obtain complete compiler listings (as in the hand-out for Lab3.1). It is important that you master these skills before you start the first coursework assignment which will be distributed during week4.

If you are not familiar with the above topics, work through the previous lab hand-outs in the labs **and in your own time** to catch up. **If you are having difficulties, ask the demonstrator in your lab (or one of your fellow students) for help.**

Basic UNIX Commands

During this class it may help to refer to the lecture notes on Basic UNIX Commands. This is available on-line on the CS Web site if you do not have a hard-copy with you.

Listing the Contents of a Directory

In a terminal window, use the `ls` command to get a listing of all the files in your home directory:

```
ls
```

Depending on how many of the lab exercises you have completed in the past two weeks or so, you should see a considerable number of files.

Also try listing invisible 'dot' files¹ with the command

```
ls -a
```

and try getting more information on the files in your directory (size, date of last modification, owner etc.) with the command

```
ls -l
```

Viewing the Contents of a File

Choose a **text** file and experiment with displaying its contents in a terminal window using the commands `cat` and `less`. For example

```
cat prog1.adb  
less .tcshrc
```

Note when using the `less` command you can page through a file by pressing the Spacebar and you quit from `less` by pressing the Q key. For more information on commands available within `less` press the H key.

Deleting Files

Files with the extensions `.o` and `.ali` are object files and Ada Library Information files produced by the Ada compiler when building executables. These take up valuable disk space and can be deleted when the compilation process is complete. Try deleting a few of the object files using the UNIX `rm` command, for example:

```
rm prog1.o
```

Always pause for a moment to check you have typed the command correctly before using the `rm` command; once you have pressed the <Return> key the files will be erased permanently. Take care not to delete source files such as `prog1.adb` by mistake. Check that the file has been deleted by using the `ls` command

¹ **Important** do not delete any 'dot' files or you may corrupt your UNIX set-up and some software will cease to work correctly.

Using Wildcards

A quicker way of deleting a group of files is to use wildcard characters; for example to delete all Ada Library Information files in the current directory use the command

```
rm *.ali
```

Take extreme care that there is no space between the * and the dot!²

Delete any bind files (those whose names begin with b~) in the current directory with the command

```
rm b~*.*
```

If you have any Emacs back-up files (those whose names end with ~) or Emacs autosave files (those whose names begin and end with #) or any executable files (those with no extension), try deleting them with suitable `rm` commands. Note that as long as you keep the Ada source files (.adb files) you can always recreate the deleted files by recompiling.

Check that the files have been deleted by using the `ls` command.

Changing Directories

You can navigate around the UNIX file system using the `cd` (change directory) command. Experiment with this, for example make the home directory of user `csweb5` by using the command

```
cd ~csweb5
```

Note how the UNIX prompt changes to display the current working directory. Now use the `ls` command to view the contents of this directory. Change directory to the sub-directory `CSLib` of this directory

```
cd CSLib
```

and use `ls` command again³. View the contents of a few of these files using the `less` command.

Change to the directory `/usr/local/staffstore/cs1110` using the `cd` command and inspect the contents of this directory and any sub-directories using the `ls` and `less` commands. This directory contains a number of files related to the ISP course and more files will be placed there as the module progresses. In particular the sub-directory `unit-programs` contains copies of the programs that appear in the Ada units.

Move to the parent directory `/usr/local/staffstore` by using the command

```
cd ..
```

and inspect the contents of this directory using the `ls` command. You will see a number of directories related Ada and to other modules

Return to your own home directory using the command

```
cd
```

Creating a Directory (or Folder)

After a few weeks most users will have created dozen of files. In order to keep track of these it is best to create a number of sub-directories within your home directory and to keep related files together in separate folders. For example you might create a directory called `ada` (or perhaps `ISP`) and keep all your work for this module in that directory. Note that you already have a folder called `nsmail`⁴ in which Netscape stores your email messages.

²(The command `rm * .ali` would attempt to delete ALL visible files in the current working directory plus a hidden file called `.ali`.

³This directory contains the source files for a number of Ada libraries including `CS_Int_IO`, `CS_File_IO` and `CS_Flt_IO`.

⁴ This folder is created automatically by Netscape Messenger.

After a while you may need to create sub-directories of your Ada folder as the number of files you create increases. For example you might choose to create a sub-directory `cwk1` (say) of the Ada folder to store all your files related to the first coursework assignment.

For the moment we will simply create a folder called `Ada` and move all the Ada files created so far into this folder. Proceed as follows:

create a folder called `Ada` in your home directory:

```
cd    # make your home directory into the current working directory
mkdir Ada
```

now move all your `Ada` files into this sub-directory

```
mv *.adb Ada
```

Now check that the files have been moved by using suitable `ls` commands to list the contents of your home directory and of the sub-directory `Ada`.

When an application is started from the a terminal window, the current working directory of the application is the directory that was current in the shell from which the application was launched.

Thus if you change directory to the folder `Ada` in a terminal window and then launch `emacs` from that terminal window, then Emacs' current working directory will be the `Ada` folder not your home directory. Emacs will, by default, then save files in the `Ada` folder not in your home directory.

For applications launched from the CDE Root Menu or from the CDE Front Panel, the current working directory is always the user's home directory.

Thus if you start Emacs in this way and you want to save an edit in a file `prog1.adb` (say) in your `Ada` sub-directory, you must specify the file name as

```
Ada/prog1.adb
```

If you just give the name as `prog1.adb`, Emacs will save it in your home directory.

Exercise

Bearing in mind the above points, try editing, saving, recompiling and running one of the `Ada` programs that you have just moved to your `Ada` sub-directory.

Filename completion with Tab

Note to save typing you can use **Tab completion** feature of the shell: type

```
cp /usr/local/sta
```

then press the `<Tab>` key and magically the directory (or folder) name will be completed to `/usr/local/staffstore/` now type the `uni` and press the `<Tab>` key again. This time the directory name will be completed to `/usr/local/staffstore/cs1110/unit-programs/`

Filename completion with `<Tab>` works as soon as you have typed enough characters to uniquely specify a file or directory. If there are several possible completions, the shell will display a list of matching files type a few more characters of the required name to make the completion unique and press `<Tab>` again. For example type

```
cp /usr/local/staffstore/cs1110/unit-programs/fact
```

and press `<Tab>` and the shell will display `factors04.adb` etc. as possible completions and complete the command as far as

```
cp /usr/local/staffstore/cs1110/unit-programs/factors0
```

Type an `'4'` and press `<Tab>` again and the file name will be completed to `factors04.adb`.

Complete the rest of the command by typing a space and a full stop and press return.

Note if no file names match what you have typed when you press <Tab>, the system will beep to indicate an error.

Copying Files

Copy one or more of the example programs that appear in the Ada lecture notes from the directory `/usr/local/staffstore/cs1110/unit-programs` into your Ada sub-directory. For example use a command of the form:

```
cp /usr/local/staffstore/cs1110/unit-programs/factors04.adb ~/Ada
```

Hint: use the <Tab> completion feature of the shell (as described above) to cut down on the typing of the long pathname OR `cd` change directory using the command

```
cd /usr/local/staffstore/cs1110/unit-programs
```

and then use the command

```
cp factors04.adb ~/Ada
```

Try editing, saving, compiling and running this program -- remember to `cd` to `~/Ada` before launching `emacs` or `gnatmake`.

Afterwards remove the compilation products from the current working directory by using the command (actually a UNIX alias)

```
clean
```

which automatically deletes most of the files produced when you compile an Ada program. Periodically throughout the term you should clean up your directories by removing unwanted files with `clean` or `rm`.

Using the File Manager

CDE provides an alternative means of managing your files: namely the **File Manager**. This provides a graphical user interface to many of the commands discussed above. Although slower and less flexible than using UNIX commands directly, novice users may find it easier to use the File Manager GUI.

Start the File Manager by selecting it from the CS Root Menu or by clicking the filing cabinet icon on the CDE Front Panel. You can change directory by clicking on the required folder icon, you can select files and rename, copy or delete them using commands from the **Selected** menu. You can change directory and create new files or folders using commands from the **File** menu. For more details on the File Manager click on **Help** in the File Manager window and select **Overview** or **Table of Contents**.

Important Note

Moving a file to Trash does not actually delete it from disk. It merely moves the file to a hidden UNIX folder called `.dt/Trash`. **If you use the File manager you should periodically empty the Trash folder otherwise you will quickly run out of disk space.**

Do this by clicking on the **Trash** icon on the CDE Front Panel and then selecting **Shred** from the **File** menu in the Trash window. This physically deletes the files in the Trash folder from disk (by doing a `rm *` command).

Optional Exercises

Familiarise yourself with the File Manager by working through the Help screens and performing some of the file operations discussed earlier in this hand-out using the File Manager.