CS1110 Introduction to Systematic Programming

Week 2 First Practical Class

If you did not attend the practical last week or if you did not complete the worksheet, work through the hand-out for that practical BEFORE working on this sheet.

Open the file prog1.adb that you created in the first practical class. Remember you can do this in several ways:

1. Type in an xterm window
```
        emacs prog1.adb &
```
2. Select Emacs from the **Editors** submenu of the CS Root menu and then open the file by selecting **Open File** from the Emacs **Files** menu.
3. Start Emacs plus filename prompt from the **Editors** submenu of the CS Root menu and then type the name of the file in the dialogue box that appears.

Compiling an Ada Program

The computer cannot execute an Ada program such as prog1.adb directly; first it needs to be converted into machine code that the computer can execute by a process known as compiling, binding and linking.

1) To compile the Ada program prog1.adb (say) from within Emacs, from the Emacs **Ada** menu select **Build**[1]. This invokes the GNAT Ada compiler, binder and linker to convert your Ada program, to binary machine code suitable for execution by the computer. The Emacs window will split into two panes and the lower pane will show the results of the compilation process.

If the program compiles successfully, this pane will show the progress of the compilation process which will end with a message of the form[2]

```
        Compilation finished at Wed Oct  9 12:47:03
```

If the compilation is successful an executable file called prog1 is created. You can run this by selecting **Run** from the Emacs **Ada** menu. Type in two numbers and the program will add them up and display the result. Select **Run** again if you want to run the program again with different input.

Note the Ada menu belongs to the pane containing the Ada program and is not accessible when the compilation pane has the focus. If the Ada menu is not visible, click in the upper pane of the Emacs window so that the Ada program pane has the focus.

If the compilation fails, the compiler outputs one or more error messages in the compilation pane followed by a message of the form

```
Compilation exited abnormally with code 4 at Wed Oct  9 14:46:58
```

This means that the program contains one or more Ada errors (or bugs) and that it needs **debugging**. In the Emacs window carefully correct any typos and other errors; you will need to look at the hand-out for practical 1 for the correct version of prog1.adb.

[1] Note that if **Compile** is selected from the **Ada** menu, the program is compiled only -- the binder and linker are not invoked. Subsequently the program can be bound and linked by selecting **Build**.

[2] The compiler will issue a warning message about the file and the unit names not matching -- do not worry about this.

Use the compiler error messages to help locate and correct the errors. Error messages take the form:

```
prog1.adb:2:01: misplaced "with"
prog1.adb:8:10: "Frst" is undefined
```

indicating that there is an error in line 2 column 1 of the file `prog1.adb` and that it has something to do with the `WITH` command and another error near column 10 of line 8 and that the variable `Frst` has not been defined. (In fact the programmer mistakenly typed `Frst` instead of `First`).

When you have corrected all the errors, save the modified file[3] (choose **Save Buffer** from Emacs' **Files** menu) and try compiling it again as above.
If the program now compiles without error, run it as described above. If it still contains errors, repeat the debugging process.

When the program has been run succesfully, exit Emacs

2) We will now compile the program `prog2.adb` (which you should have typed in the last lab class). First we must open the file in Emacs. Do this as described at the start of this lab sheet[4].

When the buffer containing the file `prog2.adb` isvisible, compile the program by select **Build** as in (1) above. Then run the program, debugging if necessary.

Managing Frames and Windows in Emacs

When we compile a program from within Emacs the Emacs window splits into two panes. When a window is split keyboard input is directed to the pane which is active. Similarly the menu commands available depend on which pane is active and the menu commands normally only affect the active pane. We can make a pane active simply by clicking with the mouse inside the pane.

At any stage you can split a window into two panes by selecting **Split Window** from the Emacs **Files** menu. Initially the two panes will contain the same text, but later two different files can be displayed in the two panes by opening a new file in one of the panes by selecting **Open File** from the **Files** menu.

If at any stage you can remove the split from a window by selecting **One Window** from the Emacs **Files** menu. The active pane then occupies the whole window. The contents of the other pane are no longer visible, but they are not lost. They can be made visible by using the **Buffers** menu to select the buffer required.

Rather than splitting a window, we can open a new window frame in Emacs by selecting **Make New Frame** from the Emacs **Files** menu. An existing buffer can be displayed in

[3] Note that the changes you make to file in an Emacs buffer are not saved to disk until you save the file. Since the Ada compiler works with the file on disk not the temporary version in the Emacs buffer and so you must save your changes before compiling.

[4] **Note** We **do** need to start a new Emacs process, due to a feature of Emacs Ada-mode. If we were to use the Emacs process which was used to compile `prog1.adb` in (1) above, then the **Build** command would recompile `prog1.adb` even when invoked in the buffer of `prog2.adb`. So remember to start a new Emacs process whenever you start working on a different Ada program. Note having two (or more) Emacs processes running simultaneously is wasteful of machine resources and slows the system down, so always kill an old Emacs process before starting another one. Moreover if you attempt to edit the same file in two different Emacs processes, chaos and confusion can quickly ensue.

the new frame by using the **Buffers** menu or a new file opened in the new frame by selecting **Open File** from the **Files** menu in the normal way.

A frame can be deleted by selecting **Delete Frame** from the Emacs **Files** menu.

Exercise

Experiment with splitting windows and opening new frame in Emacs using the above commands.

Compiling and Running a Program from a Terminal Window

Rather than invoking the Ada compiler from within Emacs, we can compile an Ada program by invoking the compiler by entering commands in an `xterm` window

To compile, bind and link a program `someprog.adb` we use the command

```
gnatmake someprog.adb
```

then to run the program we use the command

```
someprog
```

Experiment with running prog1 and prog2 from an xterm window. Copy the Ada program `factors04.adb` (which appears in Unit 4 of the ISP notes) from the `unit-programs` folder of the **Lecture Material** link on the CS1110 site on the CS Intranet

```
unit-programs
```

into your own directory as follows:

> enter the CS Intranet as described in the Induction Week Lab.
> Follow links to the **CS1110** module page.
> Then follow the **Lecture Material** link,
> then click on the **unit-programs** folder,
> click on **`factors04.adb`**
> and then choose **Save as..** from the Netscape **File** menu and save the file as `factors04.adb` in your home directory.

Now compile, bind and link `factors04.adb` -- this time using the `gnatmake` command in an `xterm` window:

```
gnatmake factors04.adb
```

and then run the program entering the command (in an `xterm` window)

```
factors04
```

Scrolling Text in Windows[5]

Sometimes an Emacs window is not big enough to display all the information in a file and some information is hidden. Similarly in an `xterm` window output may fill the window and information disappears off the top of the window. In these cases the scroll-bar at the side of the window becomes highlighted indicating that the window may be scrolled so this information becomes visible. The length of the scroll-bar indicates the whole of the document; the top of the scroll-bar representing the start of the document. The greyed-out portion of the scroll-bar (sometimes called the '**thumb**') represents the portion of the document that is currently visible.

[5] The summary of scrolling in this section refers to basic X-windows scroll-bars such as those used in `xterm` and `emacs`. Scrollbars in certain other applications (for example various SunTools such as File Manager, Netscape and Xemacs) behave differently but perhaps more intuitively.

To scroll to a different section of the document we need to move the scroll-thumb - imagine that the document is held fixed and that the scroll-thumb is a window which moves to reveal a different portion of the document. Scrolling may be done in a number of ways:

Scroll down one 'page' Move the mouse to near the bottom of the scroll-bar and left-click.

Scroll up one 'page' Move the mouse to near the bottom of the scroll-bar and right-click

Scroll down/up a fraction of a 'page'

Move the mouse the required fraction of the way down the scroll-bar and then left-click (scroll down) or right click (scroll up). For example to scroll half a page, click half-way down the scroll-bar.

The following additional operations are available on three-button mice on Sparc Solaris machines:

Continuous Scrolling Move the mouse to the thumb and middle-drag the thumb until the required text is visible in the window in the file.

Scroll to the start of the file Middle-click at the top of the scroll bar

Scroll to the end of the file Middle-click at the bottom of the scroll bar

Scroll to a specified part of the file Middle-click at the corresponding point in the scroll-bar.

For example to scroll to about halfway through the file, middle-click halfway down the scroll bar. To move to a third of the way through the document, middle-click a third of the way down the scroll-bar and so on.

Note with a two-button mouse you middle-click by pressing both buttons simultaneously -- this requires a little practice to master.

Exercises

Open the file `factors04.adb` in Emacs and practice scrolling the window up and down.