

CS1110 Introduction to Systematic Programming

Second Practical Class in Week 5

Printing Files on a Line-Printer from UNIX machines

The standard UNIX print commands `lp` and `lpr` have been disabled on most UNIX machines. Too many students were attempting to print binary files and wasting reams of paper and often getting the line-printer into peculiar states.

Instead the UNIX shell script `print` must be used. To print a text-file (such as an Ada source file or a results listing) on a line-printer (as required for the ISP coursework) use a command of the form:

```
print printer-name file-name
```

There are two CS line-printers both in MB265 called `mb265dr` and `mb265dr2`.

For example the command

```
print mb265dr cwkl.adb
```

would print the file `cwkl.adb` on the line-printer `mb265dr` whereas the command

```
print mb265dr2 results.txt
```

would print the file `results.txt` on the line-printer `mb265dr2`. Note that there is no charge for line-printing.

Please collect your output promptly from the printer room to prevent others stealing your program listing and copying it. For the same reason retain unwanted print-outs until well after the submission deadline.

On the day of an ISP coursework submission line-printer queues can be very long; allow plenty of time (up to 2 hours or so) for your output to appear or better still print your submission at least one day before the hand-in deadline.

Compiler Listings

If your program fails to compile correctly, then hand-in a full compiler listing of your program instead of a results listing. Assuming your Ada source file is called `cwkl.adb`, you can produce a compiler listing with the Unix command

```
gnatmake -gnat1 cwkl.adb >! cwkl.lst
```

and then print the file `cwkl.lst` as described above. Note the last character in `-gnat1` is an ell. A listing of the compiler error messages alone is not sufficient.

Practical Skills Check-list

You should spend the rest of your time in this lab catching up on previous lab hand-outs if you are behind in your work. After that you may start work on the first coursework assignment.

In the practical classes during the induction week and the first four weeks of the ISP you should have become familiar with the the PC-UNIX systems, Emacs and the Gnat Ada compiler. You will need to have mastered a number of practical skills in order to complete the first coursework assignment.

If you are unsure about any of the topics below you should revise the material indicated. All these hand-outs may be found on-line on the CS Intranet (only accessible from within Aston) follow links from the CS Home Page <http://www.cs.aston.ac.uk/>

Intranet → Programme Information → CS1110 → Practical Material

Intranet → Programme Information → CS1110 → Lecture material

Note if you are using Netscape on a UNIX machine then you should view the PostScript (`.ps`) version of the hand-outs. On Windows or Macintosh you may view either the MS-

Word version (.doc) or the PostScript version. However the MS-Word version is preferred on these systems. Note a quick way to move from the practicals directory to lecturenotes is to select **Parent Directory** link and then click on the **lecturenotes** link.

All the Ada features needed for the coursework assignment are covered in Units 1-8 but Unit 9 may also be helpful. These Units are available as paper and PostScript copies only. Spare paper copies of these and other hand-outs are available from outside MB212 (while stocks last).

Check-list

Logging on/off. Initial system configuration and use of Email. Password synchronisation (see `InductionLab.doc/InductionLab.ps`)

Basic Use of CDE (Window system) and the Emacs editor.
(see `Lab1.doc/Lab1.ps`)

Compiling/running an Ada program from within Emacs. Compiling/running an Ada program from a terminal window. Managing multiple buffers in Emacs, scrolling X-term windows.
(see `Lab2.1.doc/Lab2.1.ps`)

See also `Compiler.doc/Compiler.ps` in lecturenotes directory

Converting an algorithm to a full Ada program
(see `Lab2.2.doc/Lab2.2.ps`) See Unit 3 of the main ISP notes for full details

Correcting compilation errors and obtaining compiler error listings. Filename completion
(see `Lab3.1.doc/Lab3.1.ps`)

Managing UNIX Files and Directories
(see `Lab3.2.doc/Lab3.2.ps`)

See also `BasicUnix.doc/BasicUnix.ps` in lecturenotes directory

Emacs Ada Mode (using special features of Emacs for formatting Ada programs)

File based I/O in Ada

(see `Lab4.1.doc/Lab4.1.ps`)

See also `EmacsAdaMode.doc/EmacsAdaMode.ps` in lecturenotes directory

The Run-time Debugger GDBTK

Tracing program execution: (see `Lab4.2.doc/Lab4.2.ps`)

Finding logical errors in running programs: (see `Lab5.1.doc/Lab5.1.ps`)

Common Ada Run-time Exceptions

See `AdaExceptions.doc/AdaExceptions.ps` in lecturenotes directory

Printing Files on a Line-Printer

This hand-out (see `Lab5.2.doc/Lab5.2.ps`)