

# CS1110 Introduction to Systematic Programming

## First Practical Class for Week 8

### Problems with transferring Text-Files between UNIX and other systems

Sometimes there can be problems when text-files are transferred between UNIX and a PC, Mac or other non-UNIX system. Either the file will appear without line-breaks (and so appear to be 'all on one line') or for transfers in the opposite direction control characters (usually ^M) will appear at the end of each line. This problem occurs because different systems use different end-of-line markers (some use LF (linefeed), others use CR (carriage return) and others use both!). Other problems can occur if the document contains characters not in the basic ASCII character set (i.e. character codes  $\geq 128$ ) as these may be 'mangled' since there is little standardisation of these extended character sets.

Note that Emacs (unlike much other software) is sufficiently sophisticated to be able to handle all the common text-file formats correctly. If on UNIX you are viewing a text-file created on a Windows system the Emacs status line will show (DOS) just before the file name. Similarly if in Windows you are viewing a text-file created on a UNIX system the Emacs status line will show (UNIX) just before the file name. Emacs also recognises the Mac text-file format.

UNIX provides commands for converting text-files to/from DOS format. To convert a text-file `cwk3.adb` created on Windows (or DOS) to a file `cwk3_unix.adb` (say) so that it is suitable for use on UNIX, do

```
dos2unix cwk3.adb cwk3_unix.adb
```

To convert a text-file `cwk3.adb` created on UNIX to a file `cwk3_dos.adb` (say) so that it is suitable for use on Windows, do

```
unix2dos cwk3.adb cwk3_dos.adb
```

### Sparc UNIX machines

So far in this course it has been assumed that the user has been using a workstation directly by sitting down at one of dual-boot machines (Intel Solaris/Windows) in MB357, MB268 or in MB202 and logging in. There are also some Sparc Solaris machines in MB264/6 -- these are not dual boot machines and they are UNIX only machines and cannot run Windows. **Important: do not switch off the power to these machines or attempt to reboot them in any other way.** They are intended to be left switched on permanently.

To use these machines simply log on to UNIX in the normal way -- if the monitor screen is blank, press the Shift key to dismiss the screen-saver and get the login screen to appear. These machines run CDE and provide a working environment almost identical to that on Intel Solaris machines. Student users home directories are mounted on both Intel and Sparc machine and so files created on Intel are available on Sparcs and vice-versa. **However there are some differences of which one should be aware:**

the underlying machine architecture is different so the object files (`.o`) and executable files are incompatible with those for Intel machines. Thus Ada programs compiled on Intel machines will need to be completely re-compiled before running on Sparc machines (and of course vice-versa).

The directory `/usr/local/` on Sparcs is distinct on that on Intel machines (as it contains executable system software). The sub-directory `/usr/local/staffstore/` is also different and in particular `/usr/local/staffstore/cs1110` is not available on Sparc machines.

Staff home directories are not mounted on Intel Solaris machines (for security reasons) but they are mounted on Sparc Solaris machines (as these are more secure).

A later version of the Gnat Ada compiler (3.15p) is installed whereas only version 3.12p is installed on Intel Solaris machines. In particular the debuggers (`gdbtk` or `gvd`) are much more robust in Gnat 3.15p

## Command-Line Logins

Sometimes it is not possible to login directly to CDE even if a console is available. The most usual reasons for this are that

- you have exceeded your disk-quota (currently 30MBytes)
- or you have an error in your `.login` or `.tcshrc` files.

Either of these situations will usually result in the window-based login procedure being aborted, however you need some way of logging in so that you can correct the problems. In this situation find a vacant machine. Select "Command-Line Login" from the "Options" menu and then login in the normal way. This provides you with a simple command-line interface to UNIX. Issue the command

```
tcsh
```

to start your normal shell. Then at the normal TC-shell prompt type

```
quota -v
```

This gives you information about your disk-usage (figures are in KBytes) If you are at or near your limit then delete (or compress) some files. In particular delete executable Ada programs, any core dump files (called `core`) and `.o` and `.ali` files plus any editor back-up and autosave files (`somefile.adb~` and `#somefile.adb#` etc.). A simple way to do this is to issue the command `clean` in each of your home directory and in any sub-directories. Look for any particular large files by issuing the command

```
ls -l
```

and delete any unwanted large files (graphics files etc. that you may have downloaded, or results files produced by runaway Ada programs containing infinite loops).

It might also be necessary to delete your Netscape cache.

```
rm -r .netscape/cache
```

and if you use the File Manager, empty the Trash:

```
rm -r .dt/Trash
```

Check again with the `quota` command and make sure you have at least 1 MByte (1000KBytes) or so free.

If your quota looks OK, it may be that you have corrupted your UNIX initialisation files, reinstall suitable `.login` or `.tcshrc` by issuing the set-up command as in your first lab

```
/usr/local/bin/envsetup
```

Now logout by typing

```
logout
```

then log back in again using the window-login screen after selecting CDE from the "Options" menu.

To avoid such problems in future before they stop you logging in, periodically delete unwanted mail messages **and empty the mail Trash Folder**. Set the Netscape cache size to zero (or 1 MByte) using the Netscape Preferences menu. If you use file manager, remember to empty the Trash folder regularly. Periodically delete unwanted compilation products and core dumps using `clean` and delete any unwanted Ada executables etc<sup>1</sup>.

## Using UNIX Workstations Remotely

If one is sitting at a machine and using it directly, we say that the user is logged in on the **console**. Using UNIX from a console is not the only method of using the UNIX machines.

## Remote Logins

There are several other ways of logging into UNIX systems:

- 1) when already logged in on one UNIX computer one can log in to another by using the commands:

```
rlogin host
```

---

<sup>1</sup> and if you the the File manager use the command **Shred** more frequently.

or

```
telnet host
```

where *host* is the name of the other computer that you wish to use.

- 2) by using a terminal emulator such as NCSA Telnet on a Macintosh or a telnet client such as `Putty` on an IBM PC **provided that machine is connected to the network.**

### The `rlogin` command

Some software is only available on certain workstations on the network and not on others; some commercial software is only licensed for use on a specified machine or group of machines. Furthermore certain public domain software is only available on the Sparc machines and not on Intel machines (and vice-versa). Also there are several general UNIX labs around the University but these generally do not have Gnat Ada installed.

Occasionally UNIX users may wish to use a piece of software which is not directly available on the computers that they are currently using. Rather than walk to another computer (perhaps in another room across the Campus) they can login to another UNIX machine via the network. Remote login is also useful if the particular lab that you want to use is in use for classes but machines in another lab are available.

To do a remote login, login to a UNIX machine in the normal way and start a fresh terminal window and then issue the command at the usual UNIX prompt

```
localhost-name ~> rlogin cs-hydra
```

This calls up the specified Aston UNIX computer (in this case `cs-hydra`), over the network and logs in automatically. Depending on the remote computer selected, users may or may not be required to type their UNIX passwords to gain access to the remote machine.

The normal message of the day will be displayed and then the C-shell prompt for `cs-hydra` will appear. Then start a TC-shell:

```
cs-hydra% tcsh
```

and the normal tcsh prompt will appear

```
cs-hydra ~>
```

You can then run the desired piece of software on the remote machine and have the results displayed on the workstation at which you are sitting.

Another use of `rlogin` is to copy files from directories which are not mounted on the local host, but which are mounted on the remote machine `cs-hydra` (say). Then you could work with the file on the local machine as it has been copied to the Ada sub-directory of your home directory (which is mounted both on the local and remote hosts).

When work on the remote machine is complete, logout by typing

```
cs-hydra ~> exit
```

and then complete the logout by doing

```
cs-hydra% logout
```

which returns you to the local machine and you can then continue work on that machine.

Of course if you initiate an `rlogin` command from a `xterm` window while running CDE you can (say) edit a file with `emacs` on the local machine in one window whilst working with another package on the remote machine in the `xterm` window.

### Using X windows software remotely.

So far we have only described how to use basic UNIX software on a remote machine. To use X--Windows based applications software on a remote machine you must 'tell' the remote application to display its window(s) on the screen of the local machine (rather than on the screen of the remote machine). You do this by setting the environment variable `DISPLAY` (in the login shell on the remote machine).

For example to run the X application only available on Sparc machines whilst logged onto to host in say MB357:

```
localhost-name ~> rlogin cs-hydra
```

Now start tcsh:

```
cs-hydra% tcsh
```

Now tell the X-system to use console screen of the local host

```
cs-hydra ~> setenv DISPLAY localhost-name:0.0
```

where, of course, `localhost-name` is the actual name of the computer at which you are sitting. Now run a window-based application (`matlab`, say) in the background

```
cs-hydra ~> matlab &
```

### Notes

1. If the environment variable `DISPLAY` is not set, `matlab` will attempt to use the default display for `cs-hydra`: namely a console screen somewhere in 264/6. However you will not 'own' the display and the system will respond with the error message

```
Can't open display.
```

Imagine the chaos if you were allowed to write willy-nilly to a display that was being used by another innocent and unsuspecting user! The X-window system prevents this, but it **does allow** you to use the console screen on the local host because you 'own' it as you are logged on via the local host console provided that you have set the environment variable `DISPLAY`.

2 Note that only software started from the command-line on the remote machine `cs-hydra` will run on that machine. Software started from the CDE Front Panel, from the CDE Root Menu or by typing commands in other terminal windows will run on the local host.

3 Some software (e.g. `emacs`) checks to see if the environment variable `DISPLAY` is set as it starts up. If so it runs as a X-window-based application. If not a non-WIMP interface is provided; when using Emacs in this mode menus and the mouse are not available, Emacs quick-keys must be used exclusively.

4 As most Intel Solaris machines are dual-boot machines it is only possible to `rlogin` into them (if at all) when they are booted up in Solaris.

### The telnet command

The `telnet` command also allows you to log in remotely to either UNIX<sup>2</sup> or non-UNIX computers provided of course you have an account on the remote machine. For example you can access the some remote computer from a workstation by the command

```
cs-hydra ~> telnet <some-computer>
```

If you 'telnet' to a machine requiring a userID and password you will be prompted to type them in the normal way and then will be logged in to the remote machine. Note window-based applications cannot easily be run over Telnet connections.

Using Telnet you can login into Aston UNIX machines from off-campus. If you have a PC with telnet software installed and if you have an Internet connection you can login to

```
external.aston.ac.uk
```

and this enables you to access your home UNIX directory. From this machine you can then log in to a CS Intel machine `cs-sandbox` in order to compile an Ada program. Access to other CS Intel machines and to CS Sparc machines is not currently possible from outside Aston. Note that Netscape and Internet Explorer each have a telnet client built in; to access `external.aston.ac.uk` from off-campus you would use the URL:

```
telnet://external.aston.ac.uk/
```

---

<sup>2</sup> For UNIX machines `rlogin` is more convenient.