CREATING PROGRAMMES IN AN MS-DOS WINDOW

PHYSICS 359E

INTRODUCTION

Some computer programmes, for example compilers (which translate the computer programme that you wrote in FORTRAN or C into instructions that your computer can understand), do not have nice graphical interfaces. That is, they cannot be made to run by pointing and clicking as most Windows programmes do. Instead, they must be run from the simpler operating system that underlies older versions of the Windows operating system, which is known as MS-DOS (Microsoft Disk Operating System). To run such compilers, you need to understand a little about MS-DOS.

You can usually access MS-DOS in two different forms. If you click on Start -> Programs on your main Windows screen, you will probably see two entries with MS-DOS on them. One is usually called MS-DOS Prompt while the other will be MS-DOS Real Mode. The MS-DOS Prompt command will simply open a small window on your screen through which you can issue MS-DOS commands to the operating system. The MS-DOS Real Mode will actually restart your computer, allowing MS-DOS to access drivers (e.g. for your mouse) that it cannot access in the MS-DOS Prompt mode. You probably want to use the MS-DOS Prompt mode, since it is much easier to move back and forth between MS-DOS and your more familiar Windows environment. (This way you can keep open the Notepad editor to modify a programme you are working on, and easily go into MS-DOS whenever you want to try compiling or running it.) So select Start -> Programs -> MS-DOS Prompt.

A small window will open on your screen. At the top there will probably be some information about Microsoft’s copyright, and then a new line saying something like C:\WINDOWS>

This innocent looking line is your tool for talking to MS-DOS. It is what is called a “command line”, since you type commands to the operating system on it, and the C:\WINDOWS> text is what is called a prompt – it is a request from the operating system for your next order. In the past, the command line was a very common way of interacting with an operating system (and it still is in UNIX systems), but you may have had little experience with this if you have a modern operating system on your computer.

This prompt may have some useful information in it. In this case it tells you that at the moment you are working on disk C, in the folder (directory) with the name Windows. You can easily find this directory in the file system tree of your computer using the Windows Explorer. If you move to another folder, the prompt will normally continue to inform you of where in the directory tree you are.
SOME USEFUL COMMANDS

Now, you actually *do* things in MS-DOS by typing commands after the prompt. You need to know what the available commands are; they are not presented in a menu anywhere. (MS-DOS used to have a very useful command *help* that listed all the other available commands, but some years ago Microsoft removed this command from MS-DOS, perhaps to discourage people from using MS-DOS.) In this write-up we will introduce some of the most useful commands, enough to allow you to get to compiling your computer programmes.

As a first step, you probably want to go to the folder where your computer programme is being written. It might well be somewhere in the folder *My Documents* or a sub-folder of that folder such as *My Computer Programs*. You will do this by using the “change directory” or *cd* command. To get to *My Documents*, you need first to go up the directory tree to the level above the *Windows* folder, which you do by writing the command after the prompt

C:\WINDOWS> cd ..

This command takes you up one level, to C:\, the top of the directory tree on disk C.

You can go directly to the top level of the directories on disk C by typing

C:\WINDOWS> cd \)

To go to another disk than C, for example the floppy disk A, you would type simply

C:\> A:

then you would be at the top of the directory system on your floppy disk.

Now to get from C\ to C:\My Documents, you will need to understand that MS-DOS is not able to read folder names with several words, or that are longer than eight letters. As a result, MS-DOS thinks that the folder that you want to go to is called *mydocu~1*, and this is the name you must use in discussing the matter with MS-DOS. So you type the command

C:\> cd mydocu~1

and you should be rewarded with a new prompt

C:\My Documents>

indicating that MS-DOS has moved to the new folder. (If you have an message about an error, look carefully at what you have typed. MS-DOS is *very* picky.)

To see what you have in the My Documents folder, you should try the command

C:\My Documents> dir

which asks MS-DOS to list the files and folders (labelled with <DIR>) in the current folder. The list will also tell you the normal Windows name of the folders and files, so that you can identify the MS-DOS name for each of your files.

If you look at the list of files on the screen, you may see that it is only the end of the list, and that some have vanished off the top of the screen. You can see the full list by setting a flag on the *dir* command, thus:

C:\My Documents> dir /p

which will cause MS-DOS to list one screenful of files, and then wait for you to hit any key to see the next screenful. When you reach the bottom of the list, you will get back the normal command prompt again.

If you didn’t know that *dir* could take the flag /p, how would you find out? By using the *help* facility. If you want to know about the flavours of a command such as *dir*, type

C:\My Documents> dir /?
which prints a short description of `dir` and its flags on the screen. However, again you probably observe that you are only getting the end of a file. You need to force MS-DOS to print one screenful at a time. You do this by feeding the output of the `dir /?` command into the paging command, which has the name `more`, by means of a "pipe" symbol, |, thus:

```
C:\My Documents> dir /? | more
```

This command tells MS-DOS to print out the description file but to send it not to the screen, but to `more`, which in turn sends it to the screen, one page at a time. Type any key, and you will see the next screen, until you have finished the text.

With the help text you can see that, for example, you would type

```
C:\My Documents> dir /od
```
to list the files in the directory in date order from oldest to youngest, rather than alphabetically.

Several other commands are very useful.

```
C:\My Documents> del useless.doc
```

allows you to delete the Word file `useless.doc`.

```
C:\My Documents> rename fred.doc agnes.doc
```

renames the Word file `fred.doc` as `agnes.doc`.

```
C:\My Documents> copy curvel.for curvel.bak
```

copies the current version of the file `curvel.for` into another (backup) file which you can refer to if you mess up `curvel.for` too badly.

```
C:\My Documents> find banana *.txt
```

This command tells MS-DOS to search for all occurrences of the word "banana" in any file (in the current directory) with the extension `.txt`. Notice the very useful symbol "*", called a "wild card": it allows you to specify all files with the `.txt` extension rather than just one of them. Many commands can understand this symbol.

## Compiling a FORTRAN programme

Often you will open MS-DOS to work on a FORTRAN programme. It is not necessary to edit the file you are working on outside of MS-DOS. You can start the MS-DOS native editor with the command

```
C:\My Documents> edit curvel.for
```

which will open a simple full-screen editor in your MS-DOS window and import the file `curvel.for` if it exists in the directory where you are working, or start a new file with this name if none already exists. The functioning of this text editor is very similar to that of Notepad, with familiar pull-down menus at the top of the window. When you are finished editing your file and want to try to compile and run it, go to **File -> Save** (if the file already has a name) and then **File -> Exit**, and you will be back in the regular MS-DOS window.

To compile a FORTRAN programme with the Salford compiler available on the computers in the Phys359 lab (which you may copy onto your own machine at home from the disk available for loan from the lab), you will usually need to do two steps. First you must run the compiler itself on your programme (which needs to have the extension `.for`) by executing the command

```
C:\My Documents> ftn77 curvel.for
```

(replace `curvel.for` with the name of your programme). You will need to do this step for each `.for` programme file you need (the FORTRAN programme `nlfit.for` requires a second
programme called \texttt{nlfsub.for}, so this is an example of a single programme that resides in two files. This step of compiling with \texttt{ftn77} \emph{must be repeated each time you modify a programme}, but only on the files that you have modified. The result will be that you will find in your directory a file named \texttt{curve1.obj}, and corresponding \texttt{.obj} files for other FORTRAN files you compile. These are intermediate files; only the compiler can use them.

The second step is to link these files together (even if there is only one) with the command

C:\My Documents> \texttt{slink curve1.obj}

or if you have two files \texttt{curve1.for} and \texttt{curvesub.for}, use

C:\My Documents> \texttt{slink curve1.obj curvesub.obj}

This step will produce a single file \texttt{curve1.exe} (named for the first file you list in your \texttt{slink} command) which is now \emph{executable}. It is run by simply typing its name:

C:\My Documents> \texttt{curve1}

It may seem that nothing happens (depending on the length of the programme) but if everything has gone well, you will see screen output, or find new output files that the programme has created in your directory.

If your programme does not seem to compile correctly (you find error messages and no new \texttt{.obj} or \texttt{.exe} file), read the error messages carefully and use them to figure out what is wrong with your programme. Your lab instructors can help you to interpret these messages, and there is a manual for the Salford compiler that contains much useful information, including some explanation of error messages, in a green binder in the lab.

Sometimes you will want to keep the screen output of a programme in a file for later reference (for example, to check the correctness of the screen output of your programme). This can be easily done by “re-directing” the output from the screen into a file with the > command. To try this out, type the command

C:\My Documents> \texttt{dir > listing.txt}

Nothing will appear on the screen but a new prompt. However, if you examine the contents of the newly created file \texttt{listing.txt} with more,

C:\My Documents> \texttt{more listing.txt}

directory listing will be taken from the file and listed on the screen. This same trick can be used to capture the screen output from your computer programme \texttt{curve1.for}, which after compilation has produced the executable programme \texttt{curve1.exe}. To run the programme and capture the screen output, simply type

C:\My Documents> \texttt{curve1 > output1.txt}

and the output will be placed into the file \texttt{output1.txt} where you can examine it at your leisure.

Finally,

C:\My Documents> \texttt{exit}

exits MS-DOS and closes the little window.