C Notebook

David Spencer takes over the chair for this month s regular round up of C hints and tips.

ALLOCATING STORAGE

Perhaps one of the most common mistakes made by programmers new to C is to declare a pointer to an object when in fact they needed to declare the object itself. This is particularly common when passing pointers to functions. For example:

```
int *data;

void process_number(int *number) {
     ... body of function }

*data = 100;
process_number(data);
printf("Result is %d/n", *data);
```

This all compiles correctly, but will most probably crash on execution because data is never actually set to point to a place in which an int can be stored. Bugs like this can be very hard to track down.

The correct method would have been to declare data as an int, and then to replace *data (twice) by just data , and data by &data . This will keep all the pointer logic correct, but the declaration of data will now actually create a hole into which an int can be stored.

EXECUTION PROFILING

One seldom used feature of Desktop C is the ability to produce a profile of how many times each line of code in a program is executed. This can be useful in locating areas that could profitably be optimised. To use profiling, the Profile option in the compiler s menu must be turned on. Your program should then be modified to include a call to either _mapstore() or _fmapstore(filename) before completion. This will either display, or save to the named file, the execution profile, which is in the form of a line number followed by the number of times the line was executed.

One point to note is that to use profiling, your program must be linked with ANSILib from Clib.o as well as with the Stubs file. Also, profiling slows execution down so don t leave it on in final versions!

USING wimpt_Checkmode

The wimpt module of RISC OS_Lib provides some useful functions that can be used within redraw code to read details about the screen mode. However, for any of these to work, it is vital that the redraw code calls the function:

```
BOOL wimpt_checkmode(void)
```

each time it is invoked. This forces wimpt to re-cache its stored information.

wimpt_checkmode() returns True if the mode has changed since the last call, and False otherwise.

A NEW C?

Acorn are working, in conjunction with ARM Ltd, on both a new version of the C compiler and also a new Desktop Development Environment, including a much improved version of DDT and a new template editor which makes the creation of 3D dialogue boxes much easier. The new C, release 5, offers improvements both in compilation time, and also speed and size of the compiled code.

A beta-test form of the product has already been released to registered developers who use C, but there is as yet no word as to when the final product will be released to the public.