Introduction to Debugging

Introduction to Debugging

If your program exits abnormally, such as with a "bus error,", then there is almost certainly a logical error (a bug) in your program. 99% of programming is finding and removing these bugs. Here are some tips to help you get started. Remember, though, that debugging is an art, and takes practice to do well.

Before going on, it is necessary to reiterate the standard OLC policy on program debugging: Do **NOT** ask OLC for help debugging a program. This stock answer is intended to give you some tips on how to get started in this area; however, in general, program debugging requires more time and effort than consultants are usually able to provide.

The first step is to find the exact line where the program exits. One way of doing this is with print statements scattered through your code. For example, you might do something like this in your source code:

```
myplot(int x, int y)
 {
  printf("Entering  myplot()\n"); fflush(stdout);
  ---- lots of code here ------
  printf("Exiting myplot()\n");  fflush(stdout);
  return;
 }
```

The fflush() command in C ensures that the print statement is sent to your screen immediately, and you should use it if you're using printf()'s for debugging purposes.

Once you have narrowed down the line where your bug occurs, the next step is to find out the value of your variables at that time. You will probably find that one of your variables contains very strange values. This is the time to check that you have not done the following things:

- assigned an integer value to a pointer variable; or
- written to a subscript that is beyond the end of an array (remember that in C array subscripts go from 0 to N-1, not from 1 to N.)

Other mistakes also cause bugs. Make sure that your loops test correctly for their end conditions, for example.

Other kinds of bugs (programs not exiting, incorrect output) are debugged using similar methods. Again, find the line where the first error occurs, and then check the values of your variables. Once you fix a bug, recompile your program, run it again, and then debug it again as necessary.

Using printf()s is a primitive method of debugging, but sometimes it's the only one that will work. Usually, though, it is quicker and easier to use a debugger. Athena has several sophisticated debugging tools available. Saber is the tool of choice for C programmers. Gdb and dbx may also come in handy, and both of these work with Fortran as well as with C. There are stock answers that introduce Saber and dbx, and Saber even comes with a tutorial.