

Pre-processor macros to aid in debugging

Brian MacKie-Mason

Oct. 17, 2020

One of the hardest parts of troubleshooting a scientific code that crashes is determining where the crash occurred. Most low-level programming languages do not provide enough help to get started, thus further delaying a developer's efforts in tracking down an error. This note briefly describes a *new to me* methodology for clearly informing a developer-user the location of a crash. This technique only works where there is a conditional inside the code that would cause the code to crash, so it is a somewhat limited debugging technique. However, it has still been very useful to me lately so I thought I would share it.

Take, for example the following code snippet.

```
if (condition 1) then
    stop
end if
```

The code will perform a hard stop with no direct information about where the error occurred except whatever was printed to the screen prior to the error, but that print statement could have come from any part of the code. The `__FILE__` and `__LINE__` pre-processor macros can be used in a case like this very effectively. Consider the following, very similar, code snippet.

```
if (condition 1) then
    write(*,*) 'Exiting from ' \__FILE__\__ ' at line number ' \__LINE__\__
    stop
end if
```

Adding this one simple line will at least tell the user-developer what condition was not met and give a clue as to where to start the debugging process. I hope that little tip helps.