
Subject: Using wait in IDL from C

Posted by [Karl Young](#) on Fri, 11 Oct 1996 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

I'm using the following code to call IDL from C and it works fine except that I can't seem to make IDL wait for keyboard input (e.g. to terminate). For example if I stick a getchar in, it gets executed before all the idl commands, regardless of where I put it in the program and so is useless. It seems the compiler is separating all the code I send through the pipe and I apparently need to include some kind of synchronization.

More specifically what I want to do is call IDL and put up a plot, then wait for some keyboard input to terminate, so you can view the plot as long as you want. My current workaround is really ugly; run the process in the background, send IDL a really long wait after putting up the plot and then closing the plot window when done viewing (what a trainwreck if a lot of people tried to use this at once !)

Here's a simple example:

```
-----  
  
#include <stdio.h>  
FILE *idl;  
  
start_idl()  
{  
    idl = popen("idl >/dev/null 2>&1", "w");  
  
    if (idl == NULL) {  
        printf ("error invoking idl\n");  
        exit(1);  
    }  
}  
  
stop_idl()  
{  
    pclose(idl);  
}  
  
main()  
{  
    char *cmd;
```

```

start_idl();
/*
 * send commands to idl...
 */
cmd = "num = FindGen(40)*10\n";
fputs(cmd, idl);
cmd = "line = Sin(num * !DtoR)\n";
fputs(cmd, idl);
cmd = "Plot, num , line\n";
fputs(cmd, idl);
/*
 * here's where I'd like to pause but currently
 * just send a long wait, e.g. wait, 180. If I
 * put something like:
 *      getint = getchar();
 * it gets executed before the above idl calls,
 * so the program initially just waits and then
 * zooms through all the idl calls after any
 * keyboard input.
 */
cmd = "exit\n";
fputs(cmd, idl);

stop_idl();
}

```

Any suggestions greatly appreciated !

-- Karl Young