# Lab 4

You'll need to work with `examples4.tar.bz2` for this lab, which is on the website.

1. You'll be needing `http://www.drchip.org/astronaut/drchipdbg/` and "get the source", compile it to object form (`gcc -c dbg.c`), and link it in when needed. For those using Ubuntu, you may have to remove the prototype for `getline()` in `xtdio.h`.
   To use this library, you'll need to tell the compiler two things:
   - where the library is: cc ... *path*/xtdio.a
   - where the header file is: cc -I*path* ...
   - so your compiling command will resemble:
     cc -I*path* ... *path*/xtdio.a

2. Use `openseek`; this creates a file `openseek.dat`. Referring to `openseek.c`; look at the file `openseek.dat` and explain what you see. Then explain the output to the terminal from `openseek`.

3. Use `cheesey`; this creates a file `cheesey.hole`. Referring to `cheesey.c`; use vim to look at the file `cheesey.hole` and explain what you see.

4. This step exercises the `open` program: (which you may have to run using `./open`). This program uses the `int open(const char *pathname,int flags,mode_t mode)` function.
   (a) Use `man open` to see the man pages for the `open` function
   (b) Type `open` to get a listing of the commands that `open` supports
   (c) Try to write to a file: `./open w lab4` . Explain the message.
   (d) Try to create a file: `./open wc lab4` . Then try `cat lab4`. Explain the message. Type `ls -lsa lab4`; note the quantity of bytes in the file `lab4`. Remove the `lab4` file.
   (e) Try again to create a file, this time: `./open wctRuWu lab4` . What is this command doing? Type `ls -lsa lab4`. Type `cat lab4` and see the result. Explain the permissions.
   (f) Try `./open r lab4` and see the result.