

## Homework #5 (20 points)

Due, Friday, March 31

Your program takes a single argument *P*.

First, it prints a message stating whether or not *P* is executable by its owner. (Just assume *P* is a file readable by you. You don't need to check that.) If is not executable, then stop. Otherwise, continue.

Second, it prints *P*'s magic number and a message stating if the magic number is good or bad. If it is bad, then stop. Otherwise, continue.

Third, it prints out all the names in *P*'s symbol table. (All executable object files (see figure 7.20, page 219) have a section containing a symbol table.)

In order to make life bearable, the file `/unc/brock/home5/read_sym.o` contains a compiled C subroutine `read_symbol` that when called with an integer `f` and a pointer `sym_name` to a character string buffer will:

- (1) read the symbol in the executable object file opened at file descriptor `f` beginning at the present offset,
- (2) place the ASCII name of that symbol in the buffer `sym_name`,
- (3) move the offset of file `f` to the beginning of the next symbol, and
- (4) return 0, if successful, and -1, otherwise.

So, all you've got to do is figure out how to seek `f` to the beginning of the first symbol and how many symbols to read. Incidentally, `/unc/brock/home5/read_sym.c` contains the source for `read_symbol`.

The program `/unc/brock/home5/home5` is my *compiled* solution. You can run it to get some idea of what your program's output should be. By the way, the C program that produced `home5` was 43 lines long.

### Warning

Not much code needs to be written but you've got a major task. You must understand the format of executable object files on `napoleon`. Most of the information you need will come from the manual page for `a.out` and Figure 7.20 (p. 219) of your textbook. Look at both of these carefully before you start.

You are also going to have to become a C guru. You'll need to know how to use structures of included files and how to combine compiled modules with your own code. Hint – I used the following:

```
cc -g -o home5 home5.c read_sym.o |& more
```

to compile my program.

### *Rules of engagement*

Turn in a printout of your program.

You may work in groups of two *except* both members of the team cannot work for the Department of Computer Science *and* team members must split the work, both intellectual and grunt, equally.