Homework #6

Due:Friday, December 5, 1997, at 11:59PM
Points: 100

The C Programming Language

Submitting: Put your program into files named as indicated and submit them as described in the **All About Homework** handout. Remember, we'll compile and run them on the DECs, so be sure they work there! (**Note**: if you use other file names, we will deduct 5 points.)

Write a program that determines whether a user can access a file to perform the specified operation. The operation(s) are indicated by command-line options, and are:

`-r` read

`-w` write

`-x` execute

For example, if your executable is named *access*, then

```
access -r bishop /etc/motd
```

should print

```
bishop can read /etc/motd
```

Options may be combined; for example:

```
access -rwx bishop /etc/motd
```

should print

```
bishop can read /etc/motd
bishop can write /etc/motd
bishop cannot execute /etc/motd
```

Only one user and one file may be named per command. If the user is given as an integer, assume that is a UID. For example:

```
access -rw 917 /etc/passwd
```

should print

```
User 917 can read /etc/passwd
User 917 cannot write /etc/passwd
```

If the user's UID is 0, print the message

```
That user is the superuser!
```

and exit. If the object named as a file is a symbolic link, give the results that correspond to the target of the symbolic link (in other words, follow them).

Hints: Use *stat*(2). The program *user.c* discussed in class (see the lecture notes for Lecture #9) gives an example of translating from names to UIDs and how to handle groups. The program *finfo.c* in *~cs40/samples* shows you how to access the directory modes. Also, use *getopt*(3) for handling the options; it makes that part quite simple. The program *prs.c* (see the notes for lecture #8) gives an example of that also.