Homework #3

Due: May 11, 2026 **Points:** 100

This homework is designed to teach you about Linux file permissions. You are to write a program called "access" that will say what rights a user or group has over a file or directory.

Command

The command for your program is the following:

`access[ -g ] name file1 ...`

In this command, *name* is either a name or a non-negative integer. If the `-g` flag is given, *name* is interpreted as a group name or GID, and if that flag is not given, it is interpreted as a user name or UID. It is an error to give a non-existent user or group name, but it is *not* an error to give a UID or GID without an associated name.

Example Output

In these examples, "filex" is a file and "dirx" is a directory. Assume "filex" has owner bishop, group owner src, and protection mode 0654, and the directory "dirx" has the same owner and group and the protection mode 0751. Then the command

```
access bishop filex
```

prints

```
The user bishop (UID 917) can read and write the file filex
```

The command

```
access sshd filex
```

prints

```
The user sshd (UID 106) can read the file filex
```

and the command

```
access 40 filex
```

prints

```
The user with UID 40 can read the file filex
```

but the command

```
access -g 40 filex
```

prints

```
Members of the group src (GID 40) can read and execute the file filex
```

The difference is that, without the `-g` option, the 40 is interpreted as a UID, and there is no associated name. But with that option, the 40 is interpreted as a GID, and that has the associated name "users".

If any of the named files are directories, you are to print "list the contents of", "modify", or "search" rather than "read", "write", and "execute", respectively. So, the command

```
access bishop dirx
```

prints

```
The user bishop (UID 917) can list the contents of, modify, and search the directory dirx
```

(all on the same line), the command

```
access -g src dirx
```

prints

```
Members of the group src (GID 40) can list the contents of and search the directory dirx
```

and the command

```
access -g ntp dirx
```

prints

```
Members of the group ntp (GID 140) can search the directory dirx
```

If the named user has UID 0, all access is permitted regardless of the permission settings. This user is usually named *root*, but any user with a UID of 0 has such access.

Finally, sometimes the user or group will have no access to the named file or directory:

```
access bishop /root
```

prints

```
The user bishop (UID 917) can do nothing with the directory /root
```

on the standard output *stdout* (or file descriptor 1), and ***not*** the standard error *stderr* (or file descriptor 2).

Error Messages

If the file (or directory) does not exist, print the file (or directory) name followed by the message from *perror*(3) on the standard error *stderr* (or file descriptor 2):

```
access bishop filexy
```

prints

```
filexy:  No such file or directory
```

If the user does not exist, print the user name followed by a colon and a space, then "no such user" on the standard error *stderr* (or file descriptor 2):

```
access bishopric filex
```

prints

```
bishopric:  no such user
```

Similarly, if the group does not exist, print the group name followed by a colon and a space, then "no such group" on the standard error *stderr* (or file descriptor 2):

```
access -g unknown filex
```

prints

```
unknown:  no such group
```

Note that if the user or group is given as an integer, your program should proceed as in the previous section and not print an error message, even if there is no name associated with that UID or GID.

Checking Your Output

A reference version of this program is available in the directory ~`bishop/ecs153` on the CSIF; it is the executable file "access". Your output, and especially your error output, is to match that of this program.

Submitting Your Program

Please turn in the source code, the *Makefile* (you must include one) and any related information (such as manual pages and README files). Put these in a directory called *lastname-firstname/hw3* (where "lastname" and "firstname" are your last and first names, respectively), create a *tar*(1) file called *hw3.tar*, and submit that on Canvas.

We will grade your program on the CSIF, so be sure it runs there. To grade it, we will unpack the tar file, run *make*, and then run some tests. Your *Makefile* must create an executable called "access" as that is the command we will use to run the tests.

If you use an interpreted language such as python, be sure you make the script executable in such a way that it can be executed by invoking it as "access".