CS4760 **Operating Systems**
Assignment # 1 Due Date: February 10, 2020

Important: Please do all assignments on `hoare`

Linux System Calls and Library Functions

The goal of this homework is to become familiar with the environment in hoare while practising system calls. I'll like to see the use of `perror` and `getopt` in this submission.

Do Exercise 5.8:**Traversing Directories** (p. 179) in your text by Robbins/Robbins. You only need to do the Example 5.38, or breadth-first traversal.

The programming task requires you to create a utility to traverse a specified directory in breadth-first order. Breadth-first search explores all the nodes at a given level before descending lower in the tree. Breadth-first search is implemented with a queue. As the program encounters each directory node at a particular level, it enqueues the pathname for later examination. You can use the following pseudocode which makes use of queue ADT. You will have to provide the code for queue ADT operations yourself.

```
breadthfirst ( root )
{
    enqueue ( root );
    while ( ! empty ( queue ) )
    {
        next = dequeue ( queue );
        for each node directly below next
        {
            visit ( node );
            if ( isa_directory ( node ) )
                enqueue ( node );
        }
    }
}
```

Use the output format specified in Example 5.38. The executable should be called `bt`. The program will be invoked by:

```
bt [-h] [-L -d -g -i -p -s -t -u | -l] [dirname]
```

The options are to be interpreted as follows:

h Print a help message and exit.

L Follow symbolic links, if any. Default will be to not follow symbolic links.

t Print information on file type.

p Print permission bits as `rwxrwxrwx`.

i Print the number of links to file in inode table.

u Print the UID associated with the file.

g Print the GID associated with the file.

s Print the size of file in bytes. If the size is larger than 1K, indicate the size in KB with a suffix K; if the size is larger than 1M, indicate the size in MB with a suffix M; if the size is larger than 1G, indicate the size in GB with a suffix G.

d Show the time of last modification.

l This option will be used to print information on the file as if the options `tpiugs` are all specified.

If the user does not specify `dirname`, run the command using current directory and print the tree accordingly. The output will appear as follows:

```
$ bt proj
proj
proj/bi_scan
proj/include
proj/bi_scan/CVS
proj/bi_scan/Makefile
proj/bi_scan/Makefile.Linux
proj/include/CVS
proj/include/cluster.h
proj/include/config.h
proj/bi_scan/CVS/Entries
proj/bi_scan/CVS/Repository
proj/bi_scan/CVS/Root
proj/include/CVS/Entries
proj/include/CVS/Repository
proj/include/CVS/Root
```

```
$ bt -l proj
drwx------  10 sanjiv   faculty    4K  Nov 25, 2019 proj
drwx------   3 sanjiv   faculty    4K  Jan 06, 2020 proj/bi_scan
drwx------   3 sanjiv   faculty    4K  Nov 25, 2019 proj/include
drwx------   2 sanjiv   faculty    4K  Nov 25, 2019 proj/bi_scan/CVS
-rw-------   1 sanjiv   faculty   712  Nov 25, 2019 proj/bi_scan/Makefile
-rw-------   1 sanjiv   faculty    1K  Nov 25, 2019 proj/bi_scan/Makefile.Linux
-rw-------   1 sanjiv   faculty    5K  Nov 25, 2019 proj/include/cluster.h
-rw-r--r--   1 sanjiv   faculty    5K  Jan 22, 2020 proj/include/config.h
drwx------   2 sanjiv   faculty    4K  Nov 25, 2019 proj/include/CVS
-rw-------   1 sanjiv   faculty   336  Nov 25, 2019 proj/bi_scan/CVS/Entries
-rw-------   1 sanjiv   faculty    24  Nov 25, 2019 proj/bi_scan/CVS/Repository
-rw-------   1 sanjiv   faculty    15  Nov 25, 2019 proj/bi_scan/CVS/Root
-rw-------   1 sanjiv   faculty   650  Nov 25, 2019 proj/include/CVS/Entries
-rw-------   1 sanjiv   faculty    24  Nov 25, 2019 proj/include/CVS/Repository
-rw-------   1 sanjiv   faculty    15  Nov 25, 2019 proj/include/CVS/Root
```

With the use of `perror`, I'll like some meaningful error messages. The format for error messages should be:

```
bt: Error: Detailed error message
```

where `bt` is actually the name of the executable (`argv[0]`) and should be appropriately modified if the name of executable is changed without recompilation. These error messages should be sent to `stderr` using `perror`.

It is required for this project that you use version control, a `Makefile`, and a `README`. Your `README` file should consist at a minimum of a description of how I should compile and run your project, any outstanding problems that it still has, and any problems you encountered. Your `Makefile` should use suffix-rules or pattern-rules and have an option to clean up object files.

What to handin

Create your programs in a directory called *username*.1 where *username* is your user name on hoare. Once you are done with developing and debugging, *remove the executables and object files*, and issue the following commands:

```
% cd
% chmod 755 ~
% ~sanjiv/bin/handin cs4760 1
% chmod 700 ~
```

Do not copy and paste those commands from the PDF *of the assignment. Type in the commands.*

Do not forget `Makefile` (with suffix or pattern rules), your versioning files, and `README` for the assignment. If you do not use version control, you will lose 10 points. I want to see the log of how the program files are modified. Therefore, you should use some logging mechanism and let me know about it in your `README`. You must check in the files at least once a day while you are working on them. Omission of a `Makefile` (with suffix rules) will result in a loss of another 10 points, while `README` will cost you 5 points. I do not like to see any extensions on `Makefile` and `README` files.

Before the final submission, perform a `make clean` and keep the latest source checked out in your directory.

You do not have to hand in a hard copy of the project. Assignment is due by 11:59pm on the due date.