Assignment # 1 Linux System Calls and Library Functions

Details

The goal of this homework is to become familiar with the environment in HOARE while practicing system calls. I’ll try to show the use of perror and getopt in this submission.

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 level, it enqueues the pathname for later examination.

We can use the following pseudocode which makes use of queue ADT.

breadthfirst ( root )

{

enqueue ( root );

while ( ! empty ( queue ) )

{

next = dequeue ( queue );

for each node directly below next

{ visit ( node );

if ( isa\_directory ( node ) )

enqueue ( node );

}

}

}

The executable will be called bt. It will be invoked by:

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

example

./bt -h

|  |  |
| --- | --- |
| **Option** | **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.

With the use of **perror** with 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.