Search and Find secret word from “Oracle” using threads

Savvas Rostantis

Table of Contents

DESCRIPTION OF FILES 2

DESCRIPTION OF PROGRAM IMPLEMENTATION 3

DESCRIPTION OF STRUCTURES AND THREADS 4

LIST 4

BLOOM FILTER 4

THREAD SERVER 4

THREAD WORKER 4

THREAD FIND WORD 4

EXAMPLES OF IMPLEMENTATION 5

DESCRIPTION OF FILES

Το project αποτελείται από τα εξής αρχεία:

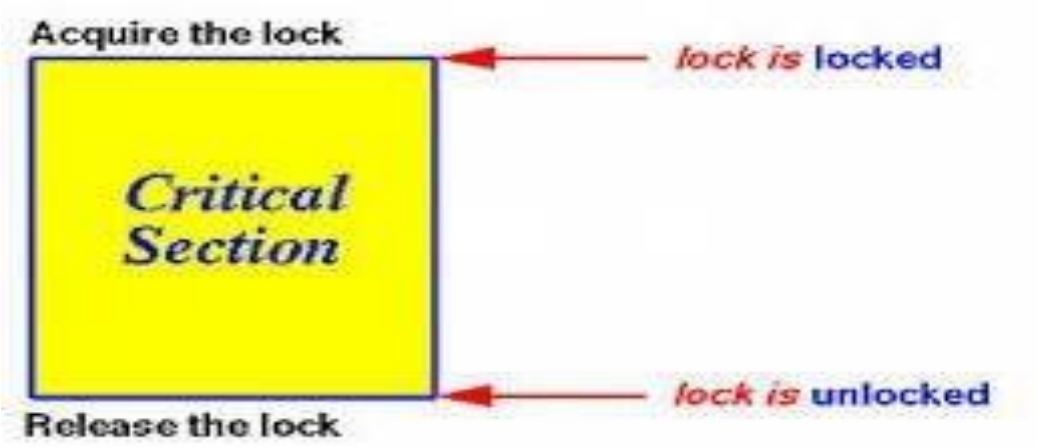
- Code:
  - Main.cpp
  - Main_functions.cpp
  - List.cpp
  - Bloom_Filter.cpp

- Headers:
  - Main_functions.h
  - Oracle.h
  - List.h
  - Bloom_Filter.h
  - Hash.h
  - Libhash.a
  - Liboracle.a
  - Operations.h

Main.cpp is the start of the program, where we create the threads and locks, checking for the correct arguments, connecting to another machine, as well as appropriate terminations. Also, there we have the code for the three types of program threads. In Main_functions. cpp (Main_functions.h) have print jobs for our program. In List.cpp (List.h) we have the list structure and its functions and finally in Bloom_Filter.cpp (Bloom_Filter.h) we have the filter structure and its functions. The Oracle.h Hash.h Libhash.a files Liboracle.a is provided as part of the exercise and it deals with the implementation of the oracle and the hash functions and the corresponding libraries. Finally, Operations.h contains data and libraries for the program.

DESCRIPTION OF PROGRAM IMPLEMENTATION

Initially, a check is made for the correct input of user arguments and the program is set if the user wants to connect to another machine. If the user wants to connect to another machine they are checked if their arguments are the same as the other machine and the other's filter is loaded otherwise the program terminates. Then it becomes appropriate initializing variables and creating threads and locks. The server thread that waits for new connections is created, blocking program flow when they are made. Each thread searches for the hidden word, words that are checked are filtered in the same filter. When a thread finds the hidden word it updates all the other threads that terminate, before terminating each thread writes some statistics into a common folder. The program waits for all threads to terminate as well as being responsible. to terminate the thread server. The whole program is based on the principle of thread synchronization using locks where they secure variables from simultaneous use of more than one thread. Finally, the program ensures that it is appropriate freeing up reserved space and proper destruction throughout it. It is also consistent in properly printing errors.



DESCRIPTION OF STRUCTURES AND THREADS

LIST

The list has the classic one-word information list structure. Consists of word input functions, a printer, deleting the list, returning the next word to main, and returning total statistics nodes. The utility is a list of pointers to the tree nodes for printing and deleting it.

BLOOM FILTER

The filter consists of a table, a number of hash functions, and an indication of whether it has been filled. It includes initializing, that is, the table in bytes, checking whether it has been filled, deleting it, returning the rate it has filled for statistics, and checking the import word. Bitwise operators and a mask are used for this control. First we run hash functions to find the byte and the bit, that is, the position, which is in the table. Make a byte mask in the appropriate position (eg byte 2- > bit 4 -> mask = [to array [2] <-00001000). We check if the byte is changed, if so then we entered the table, if not no login and returns accordingly to the tree.

THREAD SERVER

Server thread is about listening to external connections from other machines and creating worker threads to process those connections It communicates with customers in the following stages: First it listens to the connection, then sends its arguments to the client and waits for a response, if the arguments match it sends the filter to pieces or terminates the connection.

THREAD WORKER

The worker thread is about carrying out the procedure described above.

THREAD FIND WORD

The search word thread refers to the search for the hidden word. Initially, it inserts a random word in each iteration, that is. Upon finding the word it informs the other yarns that it has found it. Thread words are filtered by the same filter.

EXAMPLES OF IMPLEMENTATION

```
linux07:~/Desktop/SavvasRostantis_1115201000149-project1> ./invoke-oracle-multithreaded 120000 10 20 1223 file1
>>Insert seed for oracle function : 1
-1417412570 1875 0
>>Server Thread :[3078474560] listening for connections to port: 1223
>>Thread :[3053296448] could not found secret word ,exiting...
>>Thread :[3013593920] could not found secret word ,exiting...
>>Thread :[3070081856] could not found secret word ,exiting...
>>Thread :[3038772032] could not found secret word ,exiting...
>>Thread :[2996808512] could not found secret word ,exiting...
>>Thread :[3005201216] could not found secret word ,exiting...
>>Thread :[2988415808] could not found secret word ,exiting...
>>Thread :[3030379328] could not found secret word ,exiting...
>>Thread :[3061689152] could not found secret word ,exiting...
>>Thread :[3021986624] could not found secret word ,exiting...
>>Server Thread :[3078474560]  exiting...
>>Bloom filter deleted...
>>////////////////////////////
>>//--->Word not founded<----//
>>////////////////////////////
>>----------[Statistics]----------
>>--->You have inserted<----------
>>-->Size of bloom filter        : 120000 bytes
>>-->Number of threads           : 10 threads
>>-->Times of thread restarts    : 20 restarts
>>-->Port identification         : 1223
>>-->Logfile name                : file1
>>-->Seed entered from user      : 1
>>-->Number of hash functions    : 3 hash functions
>>------------------------------
>>--->Connection established     : NO
>>--->Execution time             : 104.48 sec
>>--->Total words                : 1045183 words
>>--->Total words used           : 486835 words
>>--->Total words rejected       : 558348 words
>>--->Percentage words used      : 46%
>>--->Percentage bloom filter full: 89.0336%
>>----------[Statistics]----------
>>Program terminated successfully..!
linux07:~/Desktop/SavvasRostantis_1115201000149-project1>
```