

SYSTEM PROGRAMMING

Project Report

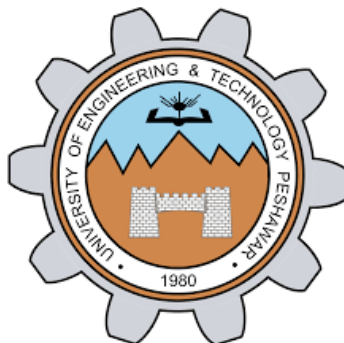
Searching a Keyword in a File System

by

Name: SYED MUHAMMAD USMAN

Roll Number: 16PWCSE1444

Submitted to: Dr. TARIQ KAMAL



Department Of Computers Systems Engineering
University of Engineering and Technology, PESHAWAR

Searching a Keyword in a File System

INTRODUCTION

We have all been through a time when we write some code or some text. But we forget where we placed it in our PC. We also come across the moment when we save a file or directory with a certain name but don't exactly remember where we stored. This is where the project of 'Searching a Keyword in a File System' comes handy.

The Project is implemented in C language and uses some system functions to achieve the task of searching the keyword.

The project does two things:

- 1) Finds the file/Sub-directory name that matches the keyword and returns it's path.
- 2) It opens the all the readable files in the directory provided and returns the path if the keyword is present within particular file(s).

In a bit technicals terms,

- 1) We provide **Directory** and **path** to the program as **input** and in return we get the the file **paths that matches** the keyword as **output**.
- 2) We also provide the program the following arguments,
 - I) **'-a'** ---- Which means we search keyword in file/Dir names only.
 - II) **'-b'** ---- Which means we search the keyword also within the files in the Directory.

Program Synopsis:

\$/Project [arg1] [arg2] [arg3]

where,

arg[1] : Option '-a' or '-b' , as discussed on the previous page.

arg[2] : is the directory name where we want to search. (We can provide root or home directory path here).

arg[3] : is the Keyword that we want to search, it must be more than 3 characters.

Output:

-list of matched filepath(s) and the count of it.

A Possible Application

- suppose you wrote a code in any language let's say in python. The code you wrote was a month ago and you placed it somewhere in the file system that you don't remember. But you do remember the name of a function the you used in your code. With this program you simply provide the **Directory** you want to search let's say we provide the root directory in this case and the function as the **keyword**. We use option **'-b'** since we want to search within the files.

-The Program would return the **path of program** that contains the particular function.

Header files used

```
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
#include <string.h>
```

Functions

1) int **keyword_file_lookup**(char *path1, char *keyword1, int size_path, int size_keyword);
-Function looks up if keyword is present within file of given file path or not.

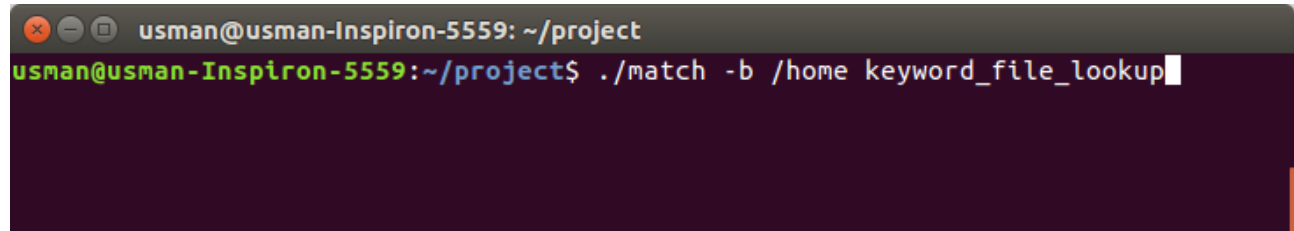
2) int **match**(char path1[],char keyword1[],int size_path,int size_keyword);
-Function that finds if keyword is present in the path or not.

3) void **find_path**(char *name,char *keyword);
-Function that passes list of all files in dir (including those in SubDirs) to match function.
-Uses recursion to open Sub-Dirs.

4) void **find_inside_file**(char *name,char *keyword);
-function that passes list of all files in dir (including those in SubDirs) to the keyword_file_lookup.
-Uses recursion to open Sub-Dirs.

Working Screen Shots

Using Option -b



```
usman@usman-Inspiron-5559: ~/project
usman@usman-Inspiron-5559:~/project$ ./match -b /home keyword_file_lookup
```

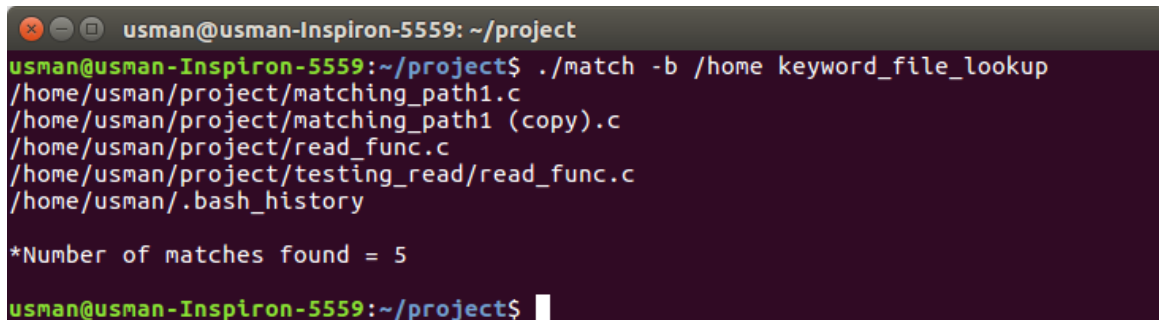
./match : program executable

-b : Option2, search keyword within the files present in Dir /home.

/home : The Dir where we are searching.

keyword1_file_lookup : It is the keyword that we are searching, in this case it is the function we used in our program. It returns the paths of c files having this function.

OUTPUT:



```
usman@usman-Inspiron-5559: ~/project
usman@usman-Inspiron-5559:~/project$ ./match -b /home keyword_file_lookup
/home/usman/project/matching_path1.c
/home/usman/project/matching_path1 (copy).c
/home/usman/project/read_func.c
/home/usman/project/testing_read/read_func.c
/home/usman/.bash_history

*Number of matches found = 5
usman@usman-Inspiron-5559:~/project$
```

Using Option -a

```
usman@usman-Inspiron-5559: ~/project
usman@usman-Inspiron-5559:~/project$ ./match -a /home browsefile
```

./match : program executable

-a : Option1, search keyword in the file paths of all files in given Dir /home.

/home : The Dir where we are searching.

browsefile : It is the keyword that we are searching

OUTPUT:

```
usman@usman-Inspiron-5559: ~/project
usman@usman-Inspiron-5559:~/project$ ./match -a /home browsefile
/home/usman/project/test/browsefile2.py
/home/usman/project/test/browsefile1.c
/home/usman/project/test/browsefile.txt

*Number of matches found = 3

usman@usman-Inspiron-5559:~/project$
```

As output we get list of file paths and the count of file paths matched.