

Find – Relazione Progetto

Librerie utilizzate:

- **<stdio.h>**: è l'header file della libreria standard del C che contiene definizioni di macro, costanti e dichiarazioni di funzioni e tipi usati per le varie operazioni di input/output. Il loro funzionamento è garantito su ogni piattaforma per la quale esiste un compilatore C.
- **<getopt.h>**: è una libreria C utilizzata per definire, validare e analizzare facilmente le opzioni da riga di comando dal resto degli argomenti.
- **<stdlib.h>**: (**ST**an**D**ard **LI**Brary) è l'header file che all'interno della libreria generale del C (glibc) dichiara funzioni e costanti di utilità generale: allocazione della memoria, controllo dei processi, e altre funzioni generali.
- **<string.h>**: è una libreria standard C che contiene dichiarazioni di funzioni e tipi usati non solo nella manipolazione delle stringhe ma anche nella manipolazione della memoria.
- **<unistd.h>**: accesso alle API dei sistemi operativi POSIX.



- **<dirent.h>**: è una libreria che contiene funzioni che facilitano l'attraversamento e la gestione delle directory.
- **<errno.h>**: contiene definizioni di macro per la gestione delle situazioni di errore. Restituisce un valore significativo per segnalare eventuali errori.
- **<stdarg.h>**: consente alle funzioni di accettare un numero indefinito di parametri.



Descrizione della struttura

Per avere un codice più leggibile, organizzato e modulare ho diviso il progetto in più file:

- Main.c;
- Find.c, Find.h;
- Logs.c, Logs.h;
- KMP.c, KMP.h;
- MakeFile.

Main.c

Nel Main mi occupo di leggere ed organizzare i parametri di chiamata del programma. Questo mi è possibile grazie alla libreria **<getopt.h>**.

Find.c

Una volta acquisiti i parametri il Main chiama il Find.

Questo è il gestore di tutto il progetto, tra le varie funzionalità le più importanti sono:

- **CheckParam():** verifica che i parametri passati dall'utente siano validi ed indirizza il programma verso il funzionamento che l'utente ha richiesto;
- **takeWords ():** si occupa di leggere e memorizzare le 'words' da ricercare;

- **ScanPaths():** gestisce la ricerca delle 'words' nei file o nelle directory richiamando il **KMP**, sarà possibile fare ricerche anche su directory (parametro *r* *nel pathfile*), in questo ultimo caso il programma eseguirà una scansione ricorsiva della directory indicata;
- **Gestisce le esclusioni:** l'utente avrà la possibilità di indicare una determinata estensione file da escludere durante la ricerca;
- richiamare **printResult()** (vedi Logs.c);
- **FromReport():** ricerca una 'word' in un file di report;
- **FromFile():** "estende" FromReport() con la possibilità di specificare un path assoluto di un file dove andare a ricercare la 'word'.

Logs.c

Questo file si occupa principalmente di tre funzioni:

- Se richiesto dall'utente, stampa a video lo stato dell'elaborazione del programma;
- Stampa a video eventuali codici di errore nelle operazioni di I/O dei file;
- **printResult():** questa funzione, oltre a stampare a video il risultato finale del Find, si occupa (se richiesta dall'utente) della generazione del file di output.



KMP.c

Il KMP (Knuth Morris e Pratt) è un algoritmo di confronto tra stringhe.

Il KMP.c è formato da due funzioni principali:

- **get_matches()**: richiamato dal Find.c, si occupa di ricercare le occorrenze delle 'words' nel path passato dal programma chiamante tramite la funzione KMPcheck;
- **KMPcheck()**: fulcro vero e proprio della ricerca dell'algoritmo.

Utilizzo del programma

Scansione delle 'words'

Comando:

```
find -w <word.txt> --i <pathfile.txt>
```

Descrizione:

Il programma andrà a scansionare il file <word.txt> per memorizzare il suo contenuto; in seguito andrà a ispezionare i path indicati nel file <pathfile.txt> per cercare



corrispondenza con i dati memorizzati dal file <word.txt>.

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find --w word.txt --i path.txt
WORD aria
TOTAL 2
FILE /home/arlind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2
69 1
71 1
FILE /home/arlind/Scrivania/find/trova/5maggio.txt
OCCURRENCES 0
WORD destre
TOTAL 2
FILE /home/arlind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2
14 8
14 32
FILE /home/arlind/Scrivania/find/trova/5maggio.txt
```

Inoltre l'utente ha la possibilità di seguire il processo aggiungendo il parametro `-v`.

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find --w word.txt --i path.txt --v
Inizio elaborazione file: /home/arlind/Scrivania/find/trova/marzo1821.txt
Fine elaborazione file: /home/arlind/Scrivania/find/trova/marzo1821.txt
Inizio elaborazione file: /home/arlind/Scrivania/find/trova/5maggio.txt
Fine elaborazione file: /home/arlind/Scrivania/find/trova/5maggio.txt
WORD aria
TOTAL 2
FILE /home/arlind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2
69 1
71 1
```



Scansione Ricorsiva

Se a fianco di un path contenuto dentro il file `<path.txt>` aggiungiamo *r* la scansione di quel path sarà ricorsiva.

```
path.txt
1 /home/arlind/Scrivania/find r
```

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find -w word.txt -i path.txt
WORD aria
TOTAL 10
FILE /home/arlind/Scrivania/find/new2.txt
OCCURRENCES 1
1 5
FILE /home/arlind/Scrivania/find/.vscode/launch.json
OCCURRENCES 1
13 54
FILE /home/arlind/Scrivania/find/verbose.c
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/find.o
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/knuth-morris-e-pratt.h
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/.vscode/settings.json
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/.vscode/tasks.json
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/knuth-morris-e-pratt.c
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/find
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/don2.txt
OCCURRENCES 1
1 5
FILE /home/arlind/Scrivania/find/word.txt
OCCURRENCES 1
1 0
FILE /home/arlind/Scrivania/find/path.txt
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/knuth-morris-e-pratt.o
OCCURRENCES 0
FILE /home/arlind/Scrivania/find/main.o
OCCURRENCES 0
```



File di Output

L'utente ha la possibilità di memorizzare il risultato finale in un file di **output** aggiungendo il comando `-o <outputfile.txt>`.

```
Arilind Canga - arlind.txt
~/Scrivania/Progetto Arlind Canga/find

WORD aria
TOTAL 2
FILE /home/arilind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2
69 1
71 1
FILE /home/arilind/Scrivania/find/trova/5maggio.txt
OCCURRENCES 0
WORD destre
TOTAL 2
FILE /home/arilind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2
14 8
14 32
FILE /home/arilind/Scrivania/find/trova/5maggio.txt
OCCURRENCES 0
.....
```

Escludere estensione specifica

`find -w <word.txt> --i <pathfile.txt> --e <txt>`

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find --w word.txt --i path.txt --e txt
WORD aria
TOTAL 0
WORD destre
TOTAL 0
WORD immobile
TOTAL 0
WORD morte
TOTAL 0
WORD sospiro
TOTAL 0
WORD spade
TOTAL 0
WORD sponde
TOTAL 0
```




Generazione di Report

Una volta generato il file di output, il programma permette di analizzarlo indicando informazioni sulla 'word' specificata, si potrà scegliere di visualizzare dove la 'word' è presente n volte.

Comando:

```
find -r <reportfile.txt> --show <word> n
```

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find --r arlind.txt --s aria 2
WORD aria
FILE /home/arlind/Scrivania/find/trova/marzo1821.txt
OCCURRENCES 2

arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$
```

Si ha la possibilità di aggiungere `-f <absolutePath>` per visualizzare le occorrenze di una data parola in un determinato path.

```
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$ ./find --r arlind.txt --s aria --f /home/arlind/Scrivania/find/trova/marzo1821.txt
WORD aria
Linea:69 Posizione:1
Linea:71 Posizione:1
arlind@ubuntu:~/Scrivania/Progetto Arlind Canga/find$
```