

find.zip

Relazione Sistemi Operativi Laboratorio 22/02/2020

Alex Citeroni - Davide Perozzi

❖ Struttura e architettura

➤ Directory

▪ Source

- main.c
 - ◆ Il programma inizia da qui, infatti il main prende quello che viene scritto a riga di comando dall'utente e verifica se sono presenti eventuali errori, in caso di assenza, se possibile, eseguirà le operazioni richieste
- kmp.c
 - ◆ Implementa l'algoritmo di Knuth-Morris-Pratt, restituendo tutte le posizioni della parola cercata all'interno del file
- find.c
 - ◆ Esegue la ricerca delle parole all'interno dei file, generando un report con tutte le informazioni cercate
- posix.c
 - ◆ Definisce la struttura dati, i parametri e le operazioni possibili
- lists.c
 - ◆ Definisce la struttura della lista dei file e delle parole e le operazioni possibili
- Include
 - ◆ kmp.h
 - ◆ find.h
 - ◆ posix.h
 - ◆ lists.h

- Output
 - File di output generati a Runtime
- Makefile

➤ **Librerie**

- Standard degli header file del C
 - <ctype.h>
 - ◆ Contiene funzioni utilizzate per la classificazione dei caratteri
 - <errno.h>
 - ◆ Contiene definizioni di macro per la gestione delle situazioni di errore
 - <limits.h>
 - ◆ Contiene delle costanti definite che indicano le proprietà specifiche dell'implementazione dei tipi interi, come l'intervallo dei numeri rappresentabili (_MIN, _MAX)
 - <stdio.h> (standard input-output header)
 - ◆ Contiene definizioni di macro, costanti e dichiarazioni di funzioni e tipi usati per le varie operazioni di input/output
 - <stdlib.h> (STanDard LIBrary, libreria standard)
 - ◆ Contiene funzioni e costanti di unità generale come conversioni, generazione di numeri pseudo-casuali, allocazione di memoria, controllo del processo, variabili d'ambiente, segnali, ricerca ed ordinamento
 - <string.h>
 - ◆ Contiene definizioni di macro, costanti e dichiarazioni di funzioni e tipi usati nella manipolazione delle stringhe e della memoria
 - <time.h>
 - ◆ Fornisce un accesso standardizzato alle funzioni di acquisizione e manipolazione del tempo
- <dirent.h>
 - Contiene funzioni per manipolare le directory del file system

- `<unistd.h>`
 - Definita da POSIX, fornisce l'accesso all'API del sistema operativo POSIX
- `<getopt.h>`
 - Utilizzata per analizzare le opzioni della riga di comando (Unix/POSIX)
- `<sys/stat.h>`
 - Acquisisce informazioni su un file

❖ Implementazione delle funzionalità

- **Main:** esegue tutti i comandi possibili specificati nell'help, utilizzando gli altri file e i metodi contenuti al loro interno
- **Posix:** singola occorrenza di una parola
 - Formata da:
 - Riga
 - Carattere
 - Puntatore all'elemento successivo
 - Le possibili operazioni sulla singola occorrenza sono:
 - Concatenazione
 - Eliminazione
 - Creazione
 - Stampa
- **Lists**
 - Operazioni Possibili:
 - Stampa
 - Creazione
 - Eliminazione
 - Aggiunta (in ordine alfabetico)
 - FileList: posix all'interno di uno specifico file
 - Memorizza:

- ◆ Percorso assoluto
- ◆ Numero di occorrenze
- Formata da puntatori a:
 - ◆ Posix
 - ◆ Elemento successivo (un altro file)
- WordList: insieme di parole all'interno dei file
 - Memorizza:
 - ◆ Stringa della parola
 - ◆ Numero di occorrenze
 - ◆ File dov'è memorizzata la parola

➤ **Generazione e analisi del report**

- Contiene i file analizzati ed il loro percorso, le parole, il numero totale delle occorrenze delle parole in ogni file e la loro posizione nel file (tramite le righe e la posizione del carattere nella riga)
- **Algoritmo di KMP (Knuth-Morris-Pratt)**
 - `searchWord(filename, word)`: restituisce tramite posix una lista ordinata di occorrenze della parola all'interno del file
 - ◆ Viene utilizzato il metodo `KMPSearch()` per ogni linea presente nel file
 - `KMPSearch(word, text, rows, headPosition)`: cerca all'interno di una riga le occorrenze di una parola, per scoprire la sua posizione all'interno della riga
- **Find**
 - `findWord(filePath, fileWord, excludeExtension, output, isVerbose)`: dopo essersi accertato che i percorsi siano validi, questo metodo ritorna la lista delle parole in ordine alfabetico, stampando a video o in un file specificato che verrà generato se non esiste o sovrascritto se esiste già (tramite `-o` o `--output <file_path>`) il report

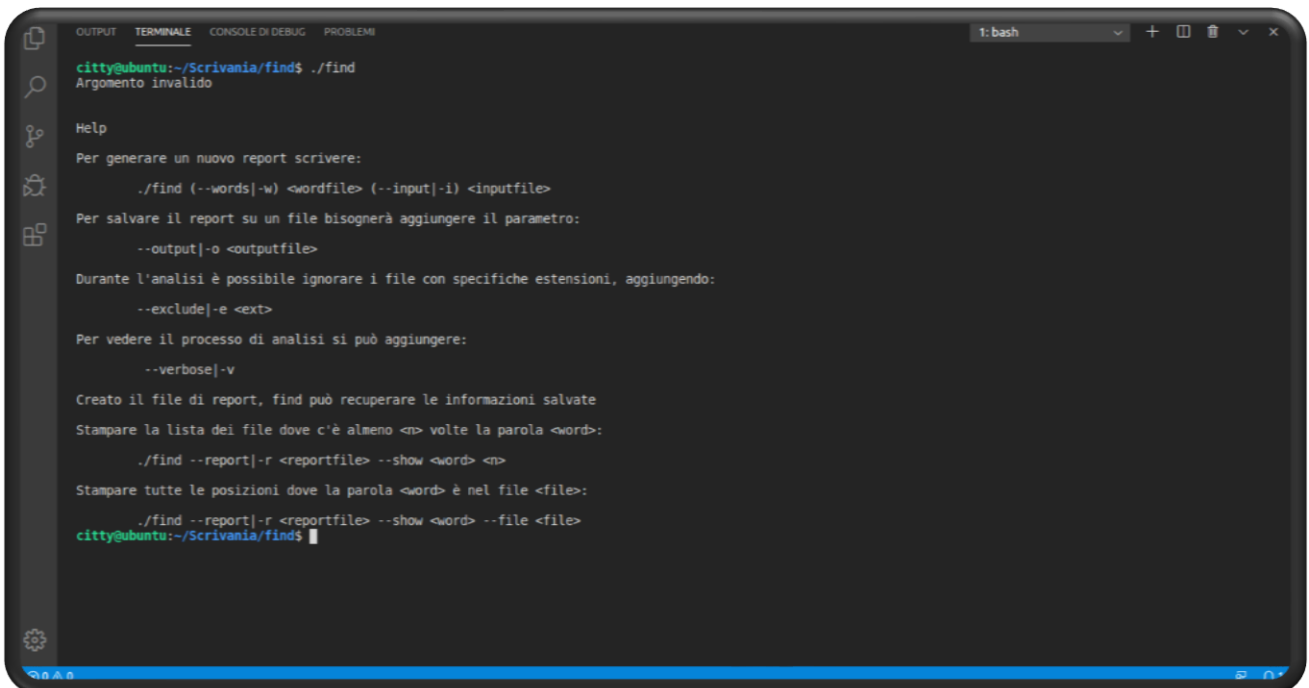
- ◆ Richiama il metodo `elaborateWord(file_path)` per ogni parola nella lista
- `elaborateWord(word, filePath, exclude, isVerbose)`: scorre ogni riga del file e restituisce una lista formata dalla parola, dalla listax di file dove si è cercato, le occorrenze (specificando la riga e la posizione nella riga) e il totale delle occorrenze in tutti i file visti
 - ◆ Richiama il metodo `searchWordInDirectory(file_path)` se la riga è uguale ad una directory
 - ◆ Richiama il metodo `searchWordInFile(file)` se la riga rappresenta un file valido, se invalido da errore
- `searchWordInDirectory(basePath, recursively, headPathList, word, exclude, isVerbose)`: restituisce una lista ordinata di file, contenente il numero di occorrenze e le posizioni della parola
 - ◆ Viene utilizzato il metodo `searchWordInFile()` per tutti i file presenti nella cartella
 - ◆ Se l'utente vuole accedere a sottocartelle presenti nella cartella, è possibile specificarlo e accedervi grazie a questo metodo che viene richiamato ricorsivamente
 - ◆ Se l'utente vuole escludere dalla ricerca certi tipi di file, può farlo grazie al comando `-e` o `--exclude`, scrivendo le estensioni da escludere, ad esempio `"-e txt"` (per escludere una singola estensione) o `"--exclude txt_pdf"` (per escludere più estensioni)
 - ◆ Se l'utente vuole visualizzare maggiori informazioni durante la ricerca, può utilizzare il comando `-v` o `--verbose`
 - Inizio dell'elaborazione della parola, del file e della cartella

- Durata dell'elaborazione della parola, del file e della cartella
- Fine dell'elaborazione della parola, del file e della cartella
- `findOccurrencesFromFile(input, report, file)`: analizza ogni riga del report ricercando una parola dentro il file e ritorna il numero di occorrenze della parola nel file
- `findPathFromReport(input, report, occurrences)`: scrivendo `--show <word> <n>` viene stampato un elenco di tutte le occorrenze e file che contengono un numero di occorrenze maggiore o uguale a quello specificato in `<n>`, questo avviene analizzando ogni riga contenuta nel report e cercando la parola inserita

❖ Test

- I test sono stati effettuati su macchina virtuale con sistema operativo Linux, utilizzando l'ultima versione di Ubuntu disponibile (Ubuntu 19.10)

Help

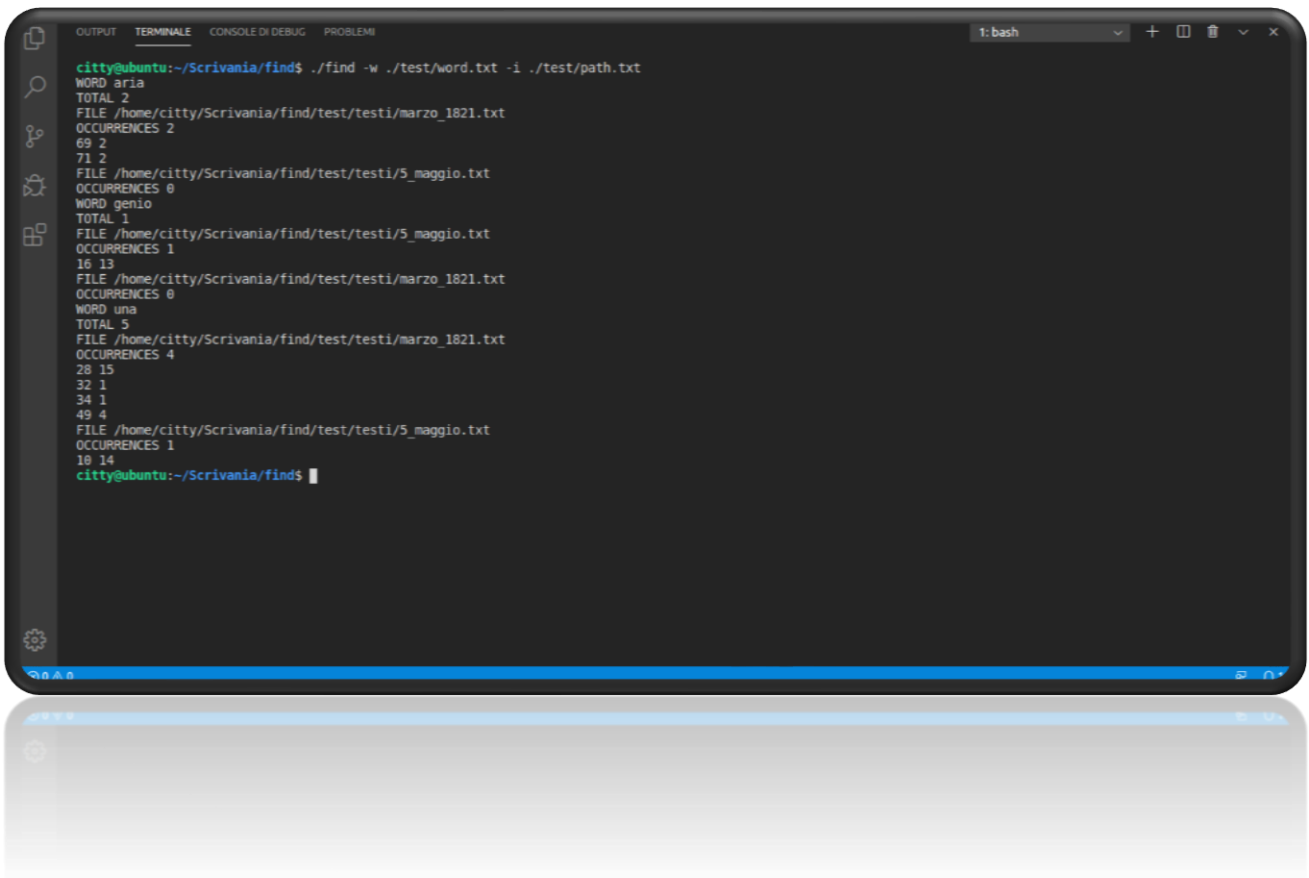


A terminal window titled "1: bash" showing the help text for a program named "find". The user has entered the command `city@ubuntu:~/Scrivania/find$./find`, which resulted in the error "Argomento invalido". The help text is displayed in Italian and includes the following information:

```
Help
Per generare un nuovo report scrivere:
    ./find (--words|-w) <wordfile> (--input|-i) <inputfile>
Per salvare il report su un file bisognerà aggiungere il parametro:
    --output|-o <outputfile>
Durante l'analisi è possibile ignorare i file con specifiche estensioni, aggiungendo:
    --exclude|-e <ext>
Per vedere il processo di analisi si può aggiungere:
    --verbose|-v
Creato il file di report, find può recuperare le informazioni salvate
Stampare la lista dei file dove c'è almeno <n> volte la parola <word>:
    ./find --report|-r <reportfile> --show <word> <n>
Stampare tutte le posizioni dove la parola <word> è nel file <file>:
    ./find --report|-r <reportfile> --show <word> --file <file>
city@ubuntu:~/Scrivania/find$
```

Comandi principali del programma stampati a video

Find



```
cityty@ubuntu:~/Scrivania/find$ ./find -w ./test/word.txt -i ./test/path.txt
WORD aria
TOTAL 2
FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
OCCURRENCES 2
69 2
71 2
FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
OCCURRENCES 0
WORD genio
TOTAL 1
FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
OCCURRENCES 1
16 13
FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
OCCURRENCES 0
WORD una
TOTAL 5
FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
OCCURRENCES 4
28 15
32 1
34 1
49 4
FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
OCCURRENCES 1
19 14
cityty@ubuntu:~/Scrivania/find$
```

Principale comando di questo programma che cerca e stampa, tramite una lista di parole e un altro file contenente il percorso, le occorrenze delle parole presenti nei diversi file

Verbose

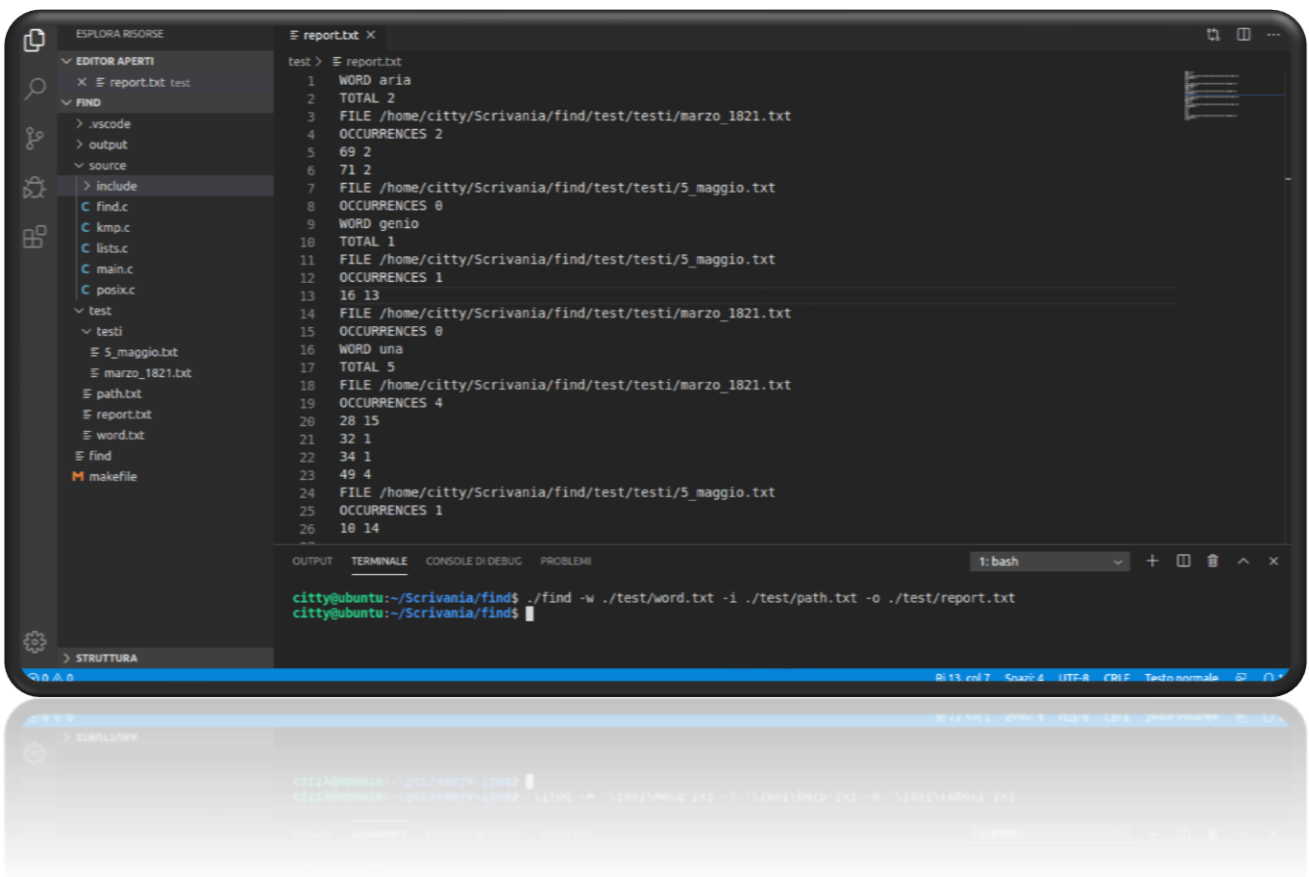
```

citty@buntu:~/Scrivania/find$ ./find -w ./test/word.txt -i ./test/path.txt -v
Inizio elaborazione parola: genio
Inizio elaborazione directory: ./test/testi
Inizio elaborazione file: ./test/testi/marzo 1821.txt
Fine elaborazione file: ./test/testi/marzo 1821.txt (0.000176 sec)
Inizio elaborazione file: ./test/testi/5 maggio.txt
Fine elaborazione file: ./test/testi/5 maggio.txt (0.000057 sec)
Fine elaborazione directory: ./test/testi (0.000466 sec)
Fine elaborazione parola: genio
Inizio elaborazione parola: una
Inizio elaborazione directory: ./test/testi
Inizio elaborazione file: ./test/testi/marzo 1821.txt
Fine elaborazione file: ./test/testi/marzo 1821.txt (0.000157 sec)
Inizio elaborazione file: ./test/testi/5 maggio.txt
Fine elaborazione file: ./test/testi/5 maggio.txt (0.000054 sec)
Fine elaborazione directory: ./test/testi (0.000364 sec)
Fine elaborazione parola: una
Inizio elaborazione parola: aria
Inizio elaborazione directory: ./test/testi
Inizio elaborazione file: ./test/testi/marzo 1821.txt
Fine elaborazione file: ./test/testi/marzo 1821.txt (0.000157 sec)
Inizio elaborazione file: ./test/testi/5 maggio.txt
Fine elaborazione file: ./test/testi/5 maggio.txt (0.000053 sec)
Fine elaborazione directory: ./test/testi (0.000325 sec)
Fine elaborazione parola: aria
WORD aria
TOTAL 2
FILE /home/citty/Scrivania/find/test/testi/marzo 1821.txt
OCCURRENCES 2
69 2
71 2
FILE /home/citty/Scrivania/find/test/testi/5 maggio.txt
OCCURRENCES 0
WORD genio
TOTAL 1
FILE /home/citty/Scrivania/find/test/testi/5 maggio.txt
OCCURRENCES 1
16 13
FILE /home/citty/Scrivania/find/test/testi/marzo 1821.txt
OCCURRENCES 0

```

Vengono visualizzati alcuni passaggi effettuati dal programma

Output



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure with folders like 'include', 'test', and 'testi'. The main editor area displays the contents of 'report.txt'. The file contains the output of a 'find' command, listing files and their occurrence counts. At the bottom, the terminal shows the command being executed: `./find -w ./test/word.txt -i ./test/path.txt -o ./test/report.txt`.

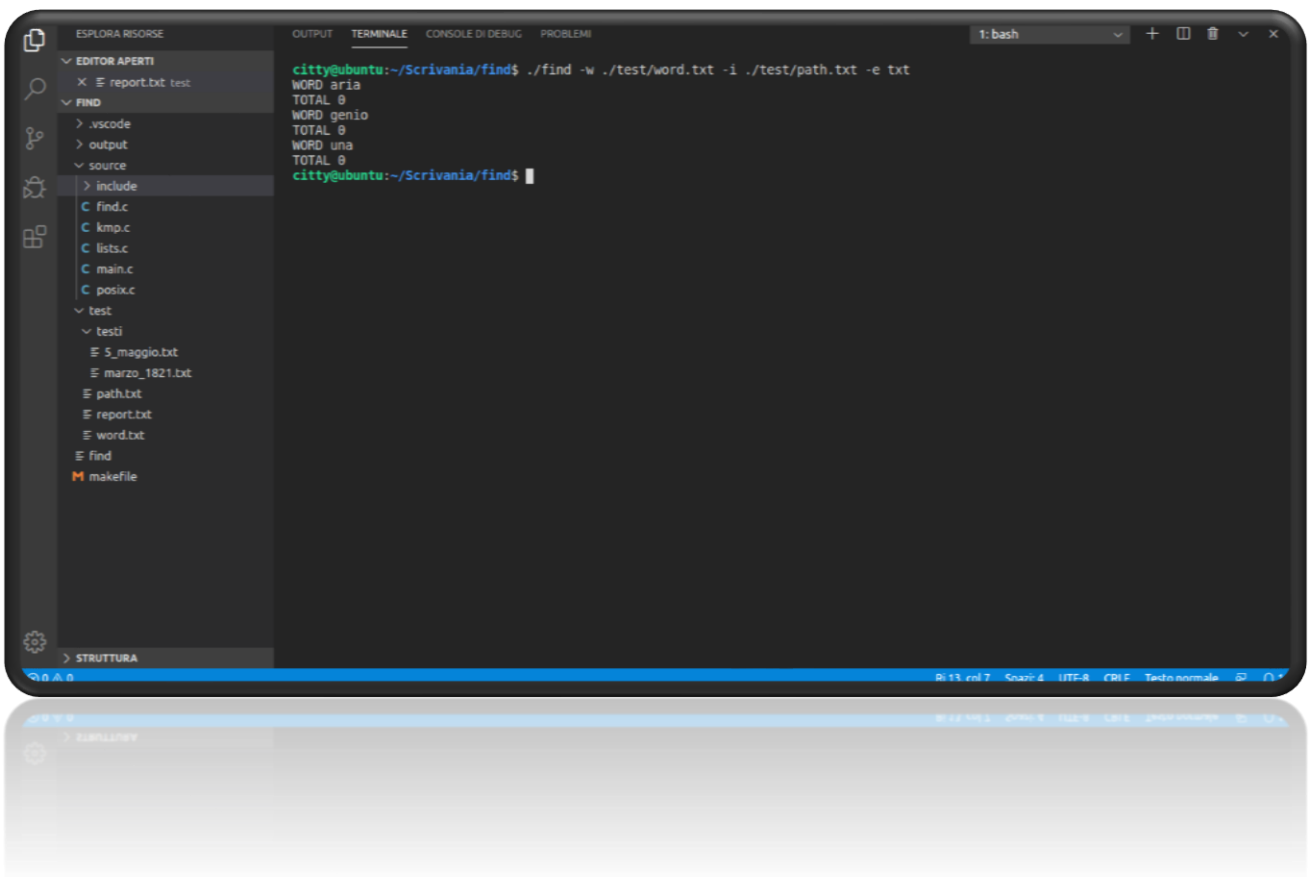
```
test > report.txt
1 WORD aria
2 TOTAL 2
3 FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
4 OCCURRENCES 2
5 69 2
6 71 2
7 FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
8 OCCURRENCES 0
9 WORD genio
10 TOTAL 1
11 FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
12 OCCURRENCES 1
13 16 13
14 FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
15 OCCURRENCES 0
16 WORD una
17 TOTAL 5
18 FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
19 OCCURRENCES 4
20 28 15
21 32 1
22 34 1
23 49 4
24 FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
25 OCCURRENCES 1
26 10 14
```

```
citty@ubuntu:~/Scrivania/find$ ./find -w ./test/word.txt -i ./test/path.txt -o ./test/report.txt
citty@ubuntu:~/Scrivania/find$
```

Viene creato (se non è già presente) o sovrascritto (se esiste già) un file di report contenente quello che viene stampato a video quando viene eseguito il comando:

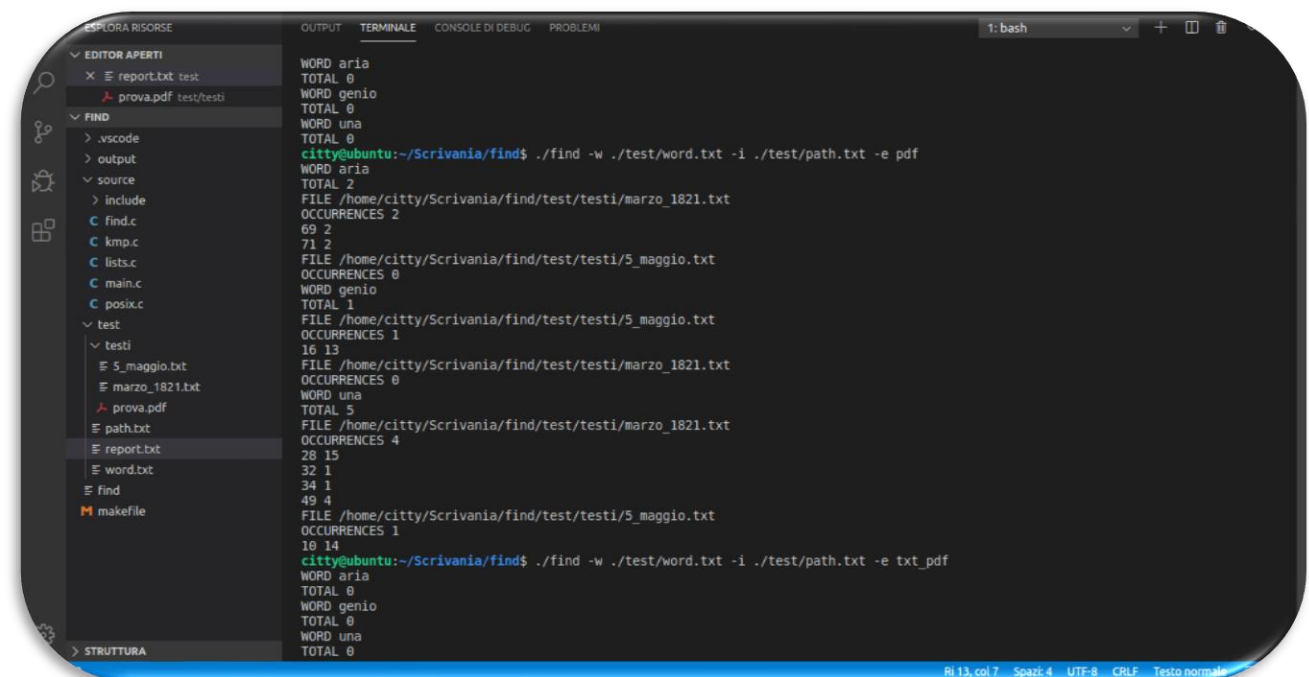
“./find -w ./test/word.txt -i ./test/path.txt”

Exclude con un solo tipo di estensione

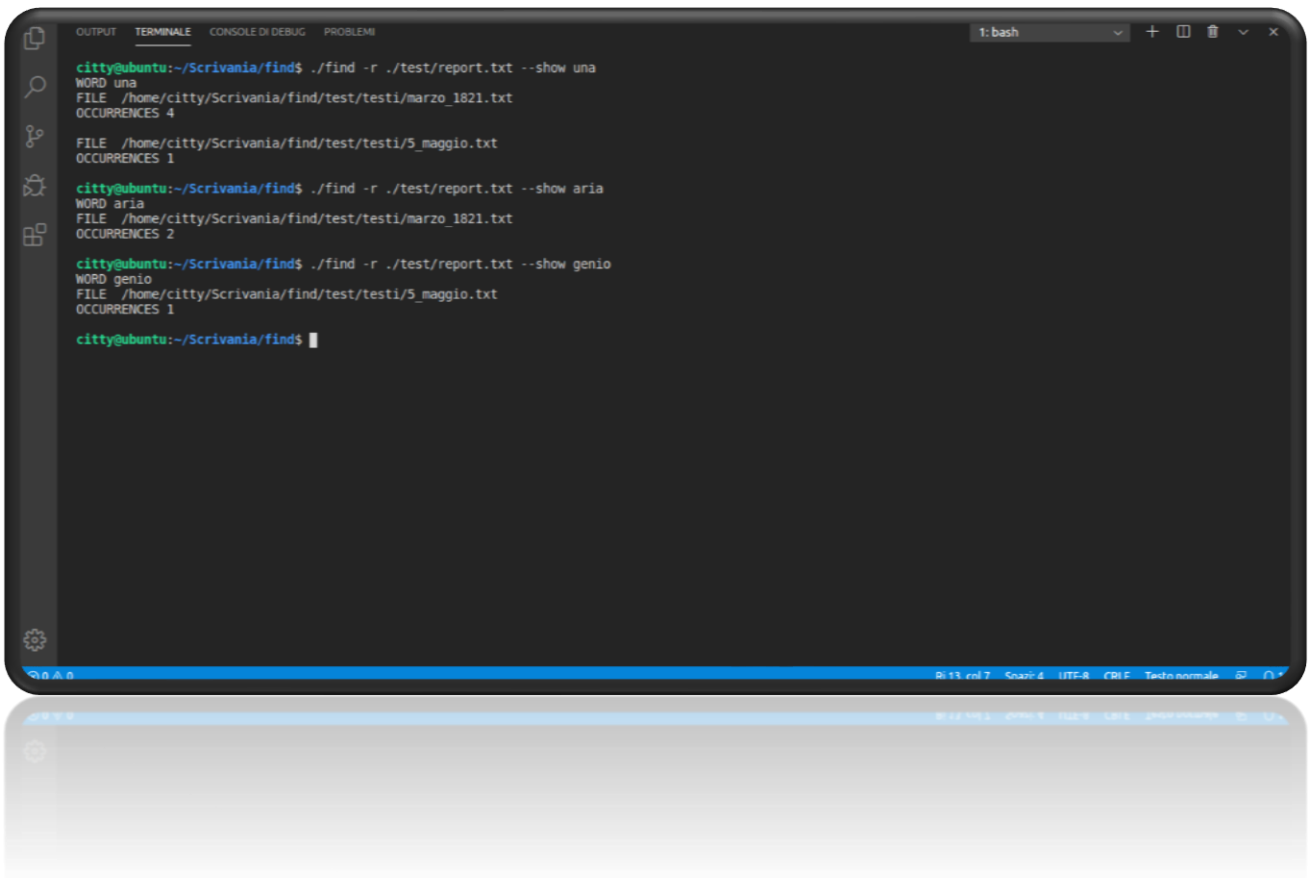


Vengono escluse dalla ricerca uno o più tipi di estensione del file (ad esempio il txt o pdf)

Exclude con più file con estensioni diverse



Show



```
citty@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show una
WORD una
FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
OCCURRENCES 4

FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
OCCURRENCES 1

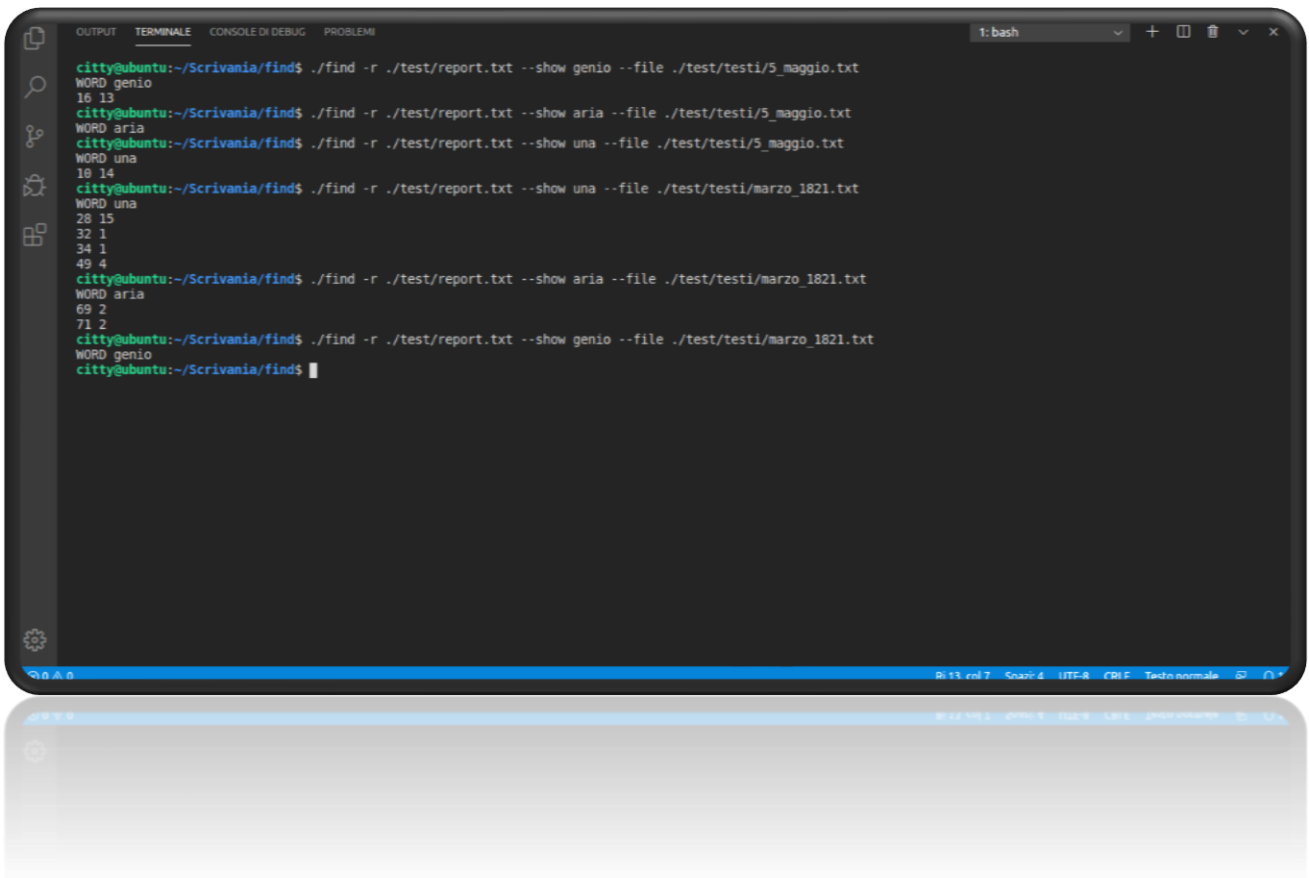
citty@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show aria
WORD aria
FILE /home/citty/Scrivania/find/test/testi/marzo_1821.txt
OCCURRENCES 2

citty@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show genio
WORD genio
FILE /home/citty/Scrivania/find/test/testi/5_maggio.txt
OCCURRENCES 1

citty@ubuntu:~/Scrivania/find$
```

Vengono stampate a video le occorrenze di una parola presenti nei file, se non viene scritto nessun numero dopo la parola vengono stampate quelle maggiori uguali a 1, altrimenti può essere specificato un numero minimo

File



```
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show genio --file ./test/testi/5_maggio.txt
WORD genio
16 13
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show aria --file ./test/testi/5_maggio.txt
WORD aria
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show una --file ./test/testi/5_maggio.txt
WORD una
10 14
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show una --file ./test/testi/marzo_1821.txt
WORD una
28 15
32 1
34 1
49 4
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show aria --file ./test/testi/marzo_1821.txt
WORD aria
69 2
71 2
city@ubuntu:~/Scrivania/find$ ./find -r ./test/report.txt --show genio --file ./test/testi/marzo_1821.txt
WORD genio
city@ubuntu:~/Scrivania/find$
```

**Vengono stampate le occorrenze di una parola un uno
specifico file**