

## Sistemi Operativi – a.a. 2024/2025

prova di laboratorio

– 12 settembre 2025 – laureandi –

Creare un programma **find.c** in linguaggio C che accetti invocazioni sulla riga di comando del tipo:

```
find <word> <dir-1> <dir-2> ... <dir-n>
```

Il programma dovrà ricercare all'interno delle directory indicate i file regolari e conteggiare al loro interno le occorrenze della parola indicata per poi riportarne il totale.

Al suo avvio il programma creerà  $n+1$  thread:

- $n$  thread DIR-i che si occuperanno di scansionare, senza ricorsione, la cartella assegnata alla ricerca di file regolari direttamente contenuti in essa;
- un thread SEARCH che si occuperà di ricercare all'interno di un file il numero di occorrenze della parola (case-insensitive).

Gli  $n$  thread DIR-i agiranno in parallelo e inseriranno, per ogni file regolare incontrato, il pathname dello stesso all'interno di un buffer condiviso di capienza prefissata (10 pathname). Il thread SEARCH estrarrà, uno alla volta, i pathname da tale buffer ed effettuerà la ricerca all'interno del suo contenuto andando a conteggiare il numero di occorrenze della parola in modalità case-insensitive (assumiamo siano file di testo). Se trovata almeno una occorrenza, la coppia di informazioni (pathname, occorrenze) sarà passata, attraverso un'altra struttura dati (un buffer con 10 elementi), al thread principale MAIN che si occuperà di mantenere un totale globale di occorrenze.

I thread si dovranno coordinare opportunamente tramite mutex e semafori numerici POSIX: il numero (minimo) e la modalità di impiego sono da determinare da parte dello studente.

Tutti i thread dovranno terminare spontaneamente alla fine dei lavori e non si dovranno usare strutture dati con visibilità globale. E' necessario rispettare fedelmente la struttura dell'output riportato nell'esempio a seguire. Codici sorgente con errori che bloccano la compilazione e pregiudicano la generazione del codice binario con compilatori standard (gcc o clang) non saranno valutati.

**Tempo:** 2 ore e 30 minuti

La struttura dell'output atteso è la seguente:

```
$ ./find Hello /usr/include/linux /usr/share/dict

[DIR-1] scansione della cartella '/usr/include/linux'...
[DIR-2] scansione della cartella '/usr/share/dict'...
[DIR-2] trovato il file 'american-english' in '/usr/share/dict'
[DIR-2] trovato il file 'cracklib-small' in '/usr/share/dict'
[SEARCH] parola da cercare: 'Hello'
[SEARCH] il file '/usr/share/dict/american-english' contiene 5 occorrenze
[DIR-1] trovato il file 'acct.h' in '/usr/include/linux'
[SEARCH] il file '/usr/share/dict/cracklib-small' contiene 3 occorrenze
[MAIN] con il file '/usr/share/dict/american-english' il totale parziale è di 5
occorrenze
[MAIN] con il file '/usr/share/dict/cracklib-small' il totale parziale è di 8
occorrenze byte
[DIR-1] trovato il file 'acrn.h' in '/usr/include/linux'
[SEARCH] il file '/usr/include/linux/acct.h' non contiene occorrenze

...
[MAIN] il totale finale è di 18 occorrenze
```