

RELAZIONE HOMEWORK 1

Antonio Pietro Romito mat.193500

Sistemi Operativi IIa.a.2022

COMPILAZIONE ED ESECUZIONE DEL PROGRAMMA

Per compilare il programma digitare il comando: `>make`

O in alternativa: `>gcc input.c diff.c hw.c -o diff`

Per eseguire il programma invece digitare il comando:

`>./diff [-g] [-s] [-d] [-u] [-v] [-o NOME_FILE_OUTPUT] F1 F2`

STRUTTURA DEL CODICE

Il codice è diviso in 3 file principali:

- **hw.c** che contiene la funzione principale.
- **input.c** che contiene le funzioni utilizzate per analizzare il comando dato in input da terminale, individuando e validando gli argomenti e le varie opzioni. In caso di input scorretti le funzioni fanno terminare il programma stampando sullo standard error la spiegazione dell'errore.
- **diff.c** ossia la parte centrale del programma che contiene le funzioni utili a calcolare il diff tra i due file forniti in input in base alle opzioni.

Sia per input.c sia per diff.c sono inoltre presenti due file header contenenti le intestazioni delle funzioni con una breve spiegazione a fronte.

IMPLEMENTAZIONE

La prima cosa che il programma effettua è l'**elaborazione dell'input** attraverso la funzione `input`: essa scorre le stringhe passate come argomento in `argv` (partendo dalla 1 dato che la 0 è comando `./diff`) e in base ad esse inizializza i seguenti array passati come parametro:

- **options:** array booleano che rappresenta le opzioni (tranne -o) dove ogni posizione rappresenta un'opzione. Se `options[i] = true` significa che quell'opzione è stata passata come argomento al comando.
- **path1 e path2:** stringhe che rappresentano i percorsi ai file di cui calcolare il diff.
- **path_out:** stringa che rappresenta il percorso al file di output, se fornito con l'opzione -o.

Successivamente **viene validato l'input** controllando che i percorsi forniti corrispondano a file esistenti e la correttezza delle opzioni in base a quanto specificato nel punto 3 della consegna dell'homework.

Se viene fornito un **file di output** con l'opzione -o viene aperto associandolo allo standard output (che invece viene chiuso). Il file di output invece viene chiuso appena prima della terminazione del programma.

Infine viene **calcolato il diff** tra i due file in base alle opzioni attraverso il seguente algoritmo:

- Vengono **aperti i due file** da analizzare, se l'apertura genera errore viene stampata la descrizione dell'errore sullo standard error e interrotto il calcolo del diff.
- Vengono **lette al più 50 righe** da entrambi i file con la funzione *leggi_n_righe (1)* e a seconda delle opzioni fornite vengono elaborate diversamente: se i file fino a quel punto sono uguali ed è presente l'opzione -g e/o -s allora **confronta i file** con la funzione *f_uguali (2)*; se è presente l'opzione -d e/o la -u allora **stampa le righe** con la funzione *enum_righe (3)*. Questo passaggio viene ripetuto finché vengono lette esattamente 50 righe da entrambi i file, dato che con molta probabilità ce ne saranno delle successive. Le righe lette vengono poi deallocate dalla memoria.
- Come ultima cosa vengono **chiusi i file e stampati i risultati** delle opzioni -g e/o -s se presenti.

FUNZIONI PER IL CALCOLO DEL DIFF

1. **leggi_n_righe:** riceve in **input** lo stream dal quale leggere le righe, un array di puntatori a stringhe (che punteranno alle righe lette) e il numero di linee da leggere n. Finché non viene letto l'end of file oppure se vengono lette n righe, **alloca la memoria** con *malloc(3)* per una riga (lunghezza stimata di 80 caratteri), assegna il puntatore alla prossima posizione libera dell'array, assegna alla memoria allocata la linea letta con *getline(3)* (se i byte allocati non bastano *getline*

utilizza *realloc(3)*), incrementa il numero di righe lette. **Restituisce** il numero di righe lette.

2. **f_uguali**: riceve in **input** gli array con i puntatori alle righe e il numero di righe lette di entrambi i file. Se il numero di righe lette dei file è diverso o se c'è almeno una riga diversa ritorna false, altrimenti ritorna true.
3. **enum_righe**: riceve in **input** gli array con i puntatori alle righe e il numero di righe lette di entrambi i file, un flag che indica se è presente l'opzione -v e uno che se true indica che è presente l'opzione -u, se false -d. **Confronta** ad una ad una le righe corrispondenti di entrambi i file (se il numero di righe lette è diverso effettua il confronto per il numero di righe più piccolo) a seconda dei flag stampa i numeri di righe uguali o di quelle diverse. Se l'opzione -v è presente stampa anche il contenuto delle righe.

ASSUNZIONI

- Ogni file di testo dato in input al programma finisce con un accapo (\n).
- I file con lunghezze diverse, dove tutte le n righe del file più breve f1 sono uguali alle prime n del file più lungo f2, sono considerati diversi (dato che f2 a più righe che non vengono analizzate nel confronto).
- Per lo stesso motivo i file con lo stesso contenuto di cui uno con una o più righe vuote successive sono considerati diversi.

RISULTATI DI ALCUNI TEST

I risultati dei seguenti comandi si trovano nei file riportati ma possono essere eseguiti nuovamente. I file utilizzati come input invece si trovano tutti nella directory test.

#TEST	COMANDO	FILE OUTPUT
1	./diff -o output/1.txt test/2.txt test/4.txt -dv	output/1.txt
2	./diff -o output/2.txt test/2.txt test/4.txt -uv	output/2.txt
3	./diff -o output/3.txt test/2.txt test/4.txt -d	output/3.txt
4	./diff -o output/4.txt test/2.txt test/4.txt -u	output/4.txt
5	./diff -o output/5.txt test/empty.txt test/empty.txt -gs	output/5.txt