

Sistemi Operativi II Modulo

Progetto (*Homework*)

Canale A – L (A.A. 2022-23)

Scadenza: **23:59, 09 giugno 2023**

Prof. Paolo Zuliani (zuliani@di.uniroma1.it)

L'obiettivo è implementare un programma C che trasformi un testo in italiano da una colonna a più colonne su più pagine (come ad es. per una pagina di quotidiano). Sono richieste **due versioni** del programma: una a singolo processo e una multi-processo con almeno tre processi concorrenti.

Dati in ingresso e requisiti:

- un file di testo in codifica Unicode (UTF-8) contenente un testo in italiano strutturato in paragrafi separati da una o più linee vuote (stringa contenente solo il carattere '\n'). Ogni paragrafo è costituito da una o più stringhe terminate da '\n' (il testo in ingresso è tutto su una sola colonna);
- il numero di colonne per pagina su cui distribuire il testo;
- il numero di linee per ogni colonna;
- larghezza di ogni colonna (numero di caratteri per colonna);
- la distanza in caratteri tra una colonna di testo e l'altra.

Dati in uscita e requisiti:

- un file di testo in codifica Unicode (UTF-8) contenente il testo di input incolonnato e impaginato secondo i parametri espressi dall'utente. Le pagine sono separate dalla stringa "\n %%% \n";
- ogni paragrafo è separato da una linea vuota;
- la prima colonna di testo è a sinistra;
- tutte le righe, tranne l'ultima, di ogni paragrafo dovranno essere allineate ad entrambi i margini della colonna;
- l'ultima riga di ogni paragrafo è solo allineata a sinistra;
- le parole in una linea dovranno essere separate da almeno un carattere di spazio;
- la sillabazione di eventuali parole troppo lunghe non è necessaria.

Requisiti generali del progetto:

- ogni file .c/.h dovrà essere **ben commentato**: per ogni funzione commentare brevemente i parametri di ingresso/uscita e il suo funzionamento generale; nel corpo di ogni funzione commentare le linee di codice più importanti;
- fornire un **makefile** per compilare il progetto con un semplice `make`;
- fornire un file di testo **README** con una breve spiegazione dei file inclusi e un breve manuale utente;

- la versione multi-processo del programma dovrà essere costituita da almeno **tre processi concorrenti** ed intercomunicanti: uno per leggere il file di ingresso, uno per creare le colonne, ed infine uno per la scrittura del file di output. (Ulteriori suddivisioni del carico di lavoro in più di tre processi sono ammesse.)
- **NON è ammesso** l'uso di librerie esterne con funzioni per la manipolazione di stringhe o testo che non siano quelle standard del C. **Eccezione:** è ammesso l'uso di librerie esterne per l'analisi delle opzioni della linea di comando per il vostro programma (ad es. <https://gflags.github.io/gflags/> o <https://www.argtable.org/>).

Suggerimenti:

- usare nomi di variabili e funzioni corrispondenti al loro significato/utilizzo;
- usare stdin e stdout per rispettivamente il testo in ingresso e in uscita;
- sviluppare prima la versione mono-processo del programma, poi quella multi-processo;
- per l'analisi della linea di comando si consiglia getopt, di uso molto semplice e che fa parte della libreria GNU standard del C;
- concentratevi prima sulla *correttezza* dei programmi, cioè assicuratevi con più testi di input che i programmi producano un output incolonnato secondo le richieste. Se avete tempo alla fine, ottimizzate il codice per uso di CPU e/o RAM, mantenendone la correttezza;
- potete usare wc per verificare che il numero delle parole in ingresso e in uscita dei vostri programmi sia lo stesso (usando stdin e stdout nei vostri programmi, questo semplice test si riduce ad una sola linea di comando con l'operatore pipe | della shell).

Il punteggio massimo è 6/30, articolato come segue:

- correttezza del programma (3/30)
- architettura del programma e commenti (2/30)
- usabilità e istruzioni utente (1/30)

Il progetto dovrà essere inviato in un solo file .zip tramite la pagina Google Classroom del corso <https://classroom.google.com/c/NTk3NDYwOTg2Njc4>.