Sistemi Operativi II Modulo
Progetto

Canale A – L (a.a. 2023-24)

Scadenza: **23:59, 31 maggio 2024**

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

L'obiettivo è implementare un programma ANSI C che:

Compito 1:

1. dato un testo in Italiano, produca una tabella contenente per ogni parola del testo le parole immediatamente successive e la loro frequenza di occorrenza nel testo;

Compito 2:

2. generi un testo in maniera casuale usando la frequenza delle parole calcolata nel punto 1.

Sono richieste **due versioni** del programma: una a singolo processo e una multi-processo con almeno tre processi concorrenti. Entrambe le versioni dovranno dare all'utente la possibilità di eseguire indipendentemente i compiti 1 e 2.

Dati in ingresso e requisiti generali:

- *Compito 1*: un file di testo in codifica Unicode (UTF-8) contenente un testo in Italiano strutturato in frasi terminate dai caratteri ., ?, o ! (altri caratteri di punteggiatura possono essere presenti);
- *Compito 2*:
  1) un file di testo in formato CSV (*comma-separated values*: ogni riga del file rappresenta una riga di una tabella, in ogni riga gli elementi sono separati da una virgola). Il file contiene una tabella in cui ogni riga riporta una parola e le parole immediatamente successive con le loro frequenze;
  2) il numero di parole da generare;
  3) (opzionale) una parola precedente da cui iniziare la generazione.

Dati in uscita e requisiti generali:

- *Compito 1*: la tabella della frequenza delle parole, come file di testo in formato CSV (vedere il punto a. di cui sopra);
- *compito 2*: un file di testo in codifica Unicode (UTF-8) contenente le parole generate casualmente.

Requisiti specifici:

- le punteggiature ., ?, ! devono essere trattate come parole separate; gli apostrofi fanno parte della parola; la rimanente punteggiatura può essere scartata. Vedere il file allegato per alcuni esempi di testo di input e la corrispondente tabella;
- maiuscole/minuscole non hanno effetto: ad es., 'oggi'= 'Oggi' = 'OGGI' = etc.
- la prima parola del testo si intende preceduta dal carattere punto . ;
- nel generare il testo casuale si può specificare una prima parola, altrimenti il programma seleziona una punteggiatura a caso tra ., ?, ! (secondo la tabella di input) e genera la parola successiva;
- nel generare il testo casuale la parola all'inizio di una frase (cioè la parola dopo un punto ., ?, o !) deve avere l'iniziale maiuscola;
- si assuma che una parola sia lunga al massimo 30 caratteri stampabili.

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 la tabella, 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. **Eccezioni**:
    - per generare le parole secondo la tabella potete usare la GNU Scientific Library oppure implementare direttamente (con la funzione rand() **e la costante** RAND_MAX) il metodo descritto qui: https://www.gnu.org/software/gsl/doc/html/randist.html#general-discrete-distributions
    - 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;
- nel leggere il file CSV contenente la tabella della frequenza delle parole controllate che la somma delle frequenze di ogni linea sia 1;
- 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* del codice: assicuratevi con più semplici testi di input che l'output sia quello richiesto. Se avete tempo alla fine, ottimizzate il codice per uso di CPU e/o RAM, mantenendone la correttezza;

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:
https://elearning.uniroma1.it/mod/assign/view.php?id=633768