Tutoraggio di Sistemi Operativi

Matteo Federico

Lezione #7

TOR VERGATA

UNIVERSITÀ DEGLI STUDI DI ROMA

In the last Episode

matching-in-file

È richiesta l'implementazione di un programma per il conteggio delle lettere. Il programma dovrà leggere un file A composto da N righe, ognuna delle quali contiene un numero M di lettere casuali. Il programma dovrà quindi generare un file B, anch'esso composto da N righe, in cui la riga i-esima riporterà il numero di occorrenze della lettera "H" presenti nella riga i-esima del file A. E' raccomandato l'utilizzo di molteplici thread per velocizzare il conteggio. **NON** ignorare problemi di sincronizzazione, ma senza usare meccanismi di sincronizzazione nativi (mutex, semaphores)
NON SONO NECESSARI!

Esercizio - lettere nei file

File

È richiesta l'implementazione di un programma per lo smistamento di lettere.
Il programma deve leggere un file A composto da M righe. Ogni riga è formattata nel modo seguente: "i-v", dove i è un indice numerico compreso tra 0 (incluso) e 5 (escluso), e v è una lettera dell'alfabeto.
Il programma dovrà creare 5 file distinti (B0, B1, ..., B4) e scrivere ciascuna lettera v nel file Bi corrispondente all'indice i.
Ad esempio, se una riga del file contiene la stringa 1-H, la lettera H dovrà essere scritta nel file B1.
Ignorare problemi di sincronizzazione

Esercizio - Comunicazione tra processi

Tx-Rx

È richiesta la creazione di un servizio di "uppercasing".
Il programma è composto da due processi, A e B, e da due pipe, Tx e Rx.
Il processo A deve leggere da stdin stringhe inserite dall'utente; una volta ricevuta una stringa, deve inviarla al processo B tramite la pipe Tx e poi stampare la risposta di B ricevuta tramite la pipe Rx.
Il processo B deve ricevere le stringhe dalla pipe Tx, trasformare tutte le lettere in maiuscolo e rimandarle indietro tramite la pipe Rx.

Esercizio - Pipeline di processamento

pipeline di creazione

Creare un processo che prende in input un file di tipo ".csv": ogni linea rappresenta un determinata persona composta da nome,cognome,eta,altezza!
Sviluppare un flusso di processamento , in cui sono coninvolti 4 processi. Questi processi devono essere creati nel seguente modo:

- il processo padre (aka P0) crea due P1 e P2
- il processo P2 crea il processo P3

Esercizio - Pipeline di processamento

pipeline di processamento

- il processo P0 legge riga per riga il file passato da argv, per ogni riga crea una struttura dati che passa tramite metodi di comunicazione scelti da voi i dati al P1 e P2.
- Il processo P1 deve calcolare l'età media delle persone presenti nel file.
- Il processso P2 deve calcolare l'altezza media delle persone presenti nel file.
- Il processo P3 deve calcolare per l'altezza media per : gli under 20, per coloro compresi tra 20 e 40 e per gli over 40.

Quando il file è terminato il processo P0 deve mandare un messaggio di terminazione a tutti i processi figli, e loro dovranno stampare i valori da loro computati e terminare.
Il processo P0 aspetta che tutti terminano prima di terminare!

TOR VERGATA