

Sistemi Operativi L-A
Compito di Martedì 13 Gennaio 2009
CdS in Ingegneria Informatica - Prof. Paolo Bellavista

Compito A - Parte di Programmazione di Sistema (16 punti)

Si scriva un programma C che, utilizzando le System Call del sistema operativo UNIX, abbia un'interfaccia del tipo:

senzaFili car1 car2 car3 fileIn

dove **car1**, **car2** e **car3** sono singoli caratteri e **fileIn** un nome assoluto di file esistente nel file system. Dopo aver effettuato gli opportuni controlli sui parametri di invocazione, il processo iniziale **P0** deve generare due processi **P1** e **P2**, fratelli fra loro.

fileIn contiene una sequenza di caratteri organizzati in righe (numero di righe non noto a priori, lunghezza massima di una riga pari a 80 caratteri). Il processo padre deve leggere **fileIn** riga per riga: per ogni riga, deve eliminare tutte le occorrenze di **car1**, comunicare a **P1** la riga privata di tali occorrenze di **car1** e scrivere su **fileIn** la medesima riga se la riga è di posizione multipla di 3 (terza, sesta, nona, ...), partendo da inizio file e avendo cura di non sovrascrivere parti del file ancora non esaminate. Analogamente, **P1** deve leggere riga per riga i caratteri comunicatigli dal padre **P0**, eliminare tutte le occorrenze di **car2**, comunicare al fratello **P2** la riga privata di tali occorrenze di **car2** e scrivere su **fileIn** la medesima riga se la riga è in posizione 1, 4, 7, ..., ovviamente senza sovrascrivere ciò che **P0** sta scrivendo e il contenuto del file ancora da esaminare. A sua volta **P2** deve leggere riga per riga i caratteri comunicatigli dal fratello **P1**, eliminare tutte le occorrenze di **car3**, comunicare al padre **P0** la riga privata di tali occorrenze di **car3** e scrivere su **fileIn** la medesima riga se e solo se di posizione 2, 5, 8, ... Oltre ai vincoli già menzionati, non è richiesto alcun ordinamento particolare delle righe scritte su **fileIn** dai tre processi.

In ogni istante, deve essere inoltre possibile per l'utente forzare la modifica del comportamento del programma concorrente premendo la combinazione di tasti <CTRL-C>. In particolare, in seguito alla pressione di <CTRL-C>, si dovrà invertire il verso del telefono senza fili, con **P0** che comunica caratteri, riga per riga, a **P2** e questo che li gira, dopo il suo processamento, a **P1**, fino a chiudere la catena come al solito su **P0**.

Si facciano le ipotesi semplificative desiderate in termini di modello affidabile dei segnali e di innestamento dell'esecuzione dei gestori associati. Come sempre, si facciano le scelte di sincronizzazione dei processi ritenute più opportune, cercando di sequenzializzare il meno possibile le varie operazioni richieste.

Sistemi Operativi L-A
Compito di Martedì 13 Gennaio 2009
CdS in Ingegneria Informatica - Prof. Paolo Bellavista

Compito A - Parte di Programmazione Java Thread (8 punti)

Si scriva un programma concorrente che utilizzi i Java Thread per simulare lo scenario di una cucina di ristorante.

Uno chef (Chef.java) gestisce e controlla il corretto dosaggio dei 4 ingredienti per la preparazione del piatto. Lo chef riceve ingredienti dagli **aiuto cuochi** (AiutoCuoco.java), ciascun aiuto cuoco porta un solo tipo di ingrediente e in quantità **non necessariamente uguale** al dosaggio richiesto. Una volta ottenute le quantità di ingredienti necessarie, un **addetto al forno** (AddettoForno.java) può prendere la preparazione così ottenuta e infornarla per terminare il piatto.

Un aiuto cuoco che tenti di portare un ingrediente la cui quantità necessaria è già stata raggiunta deve **attendere** che il piatto venga terminato (cotto) prima di poter consegnare il proprio ingrediente.

Inoltre, onde evitare rallentamenti nel processo di preparazione dei piatti, qualora ci siano aiuto cuochi **in attesa** da più di N secondi e **manchino ancora ingredienti necessari al completamento**, l'**ultimo cuoco ad essersi messo in attesa** deve occuparsi di **portare tutti gli ingredienti residui** (e tralasciare il proprio).

I file allegati Chef.txt, AiutoCuoco.txt, AddettoForno.txt e Launcher.txt forniscono uno scheletro parziale di una possibile traccia di soluzione.

Sistemi Operativi L-A
Compito di Martedì 13 Gennaio 2009
CdS in Ingegneria Informatica - Prof. Paolo Bellavista

Compito A – Domande di Teoria a Risposta Aperta Sintetica (8 punti)

- 1) Si illustri nel dettaglio che cosa si intenda per **segmentazione della memoria virtuale**, descrivendone con cura le motivazioni fondamentali e i meccanismi necessari per l'implementazione.

- 2) Si descriva l'**algoritmo di short-term scheduling MLFQ** e se ne illustrino i vantaggi, che ne giustificano l'impiego diffuso nei moderni sistemi operativi general-purpose, rispetto ad algoritmi ottimali quali SJF. Inoltre, si presenti sinteticamente che cosa si intende per long-term scheduling.

- 3) Si mostri con precisione come la **shell di Linux effettui i passi di sostituzione** dei "caratteri speciali" in un qualunque comando. A tal fine si presenti un caso esemplificativo di codice che metta insieme metacaratteri su file, ridirezione, variabili e caratteri speciali modificatori come singoli/doppi apici e back quote.

Si rammenta ai candidati che durante la prova scritta non può essere consultato nessun tipo di materiale (slide del corso, appunti, libri, ...); l'accesso al sito Web del corso è stato disabilitato.