

Esercitazione 11

Esercitazione di riepilogo

Agenda

Esercizio 1 - Monitor java

Il grill del ristorante.

Esercizio 2 - Programma C di sistema

Gestione e monitoraggio dell'esecuzione del programma che risolve l'esercizio 1

Esercizio 3 - File comandi bash

Gestione e monitoraggio dell'esecuzione del programma che risolve l'esercizio 1

Esercizio 1

Gestione del barbecue di un
ristorante



Esercizio 1

Si consideri un ristorante specializzato in piatti di carne alla griglia (bistecche, spiedini, ecc.); a questo scopo è presente un unico barbecue che viene utilizzato per la cottura di tutti i piatti. La cottura avviene collocando la carne su una griglia rimovibile.

Il ristorante è **certificato per la preparazione di piatti senza glutine**, adatti alle persone affette da **celiachia**.

A questo scopo il ristorante, per evitare contaminazioni, dispone di **due griglie**:

- una per le preparazioni **senza glutine (PSG)**;
- una per tutte le **altre preparazioni** (preparazioni normali o **PN**).

Il barbecue può essere utilizzato con una griglia alla volta, pertanto:

- quando il barbecue è usato per la cottura di PSG, non è possibile cuocere contemporaneamente PN.
- viceversa, se il barbecue è utilizzato per PN, non possono essere contemporaneamente cotte preparazioni PSG.

Esercizio 1 - Traccia (2/3)

Ognuna delle 2 griglie ha una **capacità limitata a N** che esprime il massimo numero di preparazioni che può contemporaneamente contenere.

Inoltre, prima di iniziare una nuova fase di uso della griglia per PSG, è necessario l'intervento di un **inserviente dedicato alla pulizia** di tutto il barbecue per rimuovere eventuali tracce di contaminazione (cioè, tracce di glutine).

Realizzare un'applicazione concorrente in Java basata sul monitor, nella quale le singole preparazioni e l'inserviente siano rappresentati da thread distinti.

L'applicazione dovrà gestire il barbecue tenendo conto dei vincoli indicati e inoltre, nella cottura delle preparazioni dovrà dare la **priorità alle preparazioni senza glutine** (PSG).

Esercizio 1 - Traccia (3/3)

In particolare, è richiesto che al termine di ogni fase (cottura PSG o cottura PN), il gestore del barbecue scriva sullo **standard** output, il tipo di fase compiuta (“PSG” o “PN”), quante preparazioni sono state completate nella fase e l’istante di tempo in cui si è conclusa la fase.

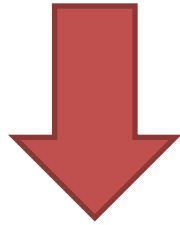
Per esempio:

“PSG -13 piatti - Sun May 31 12:16:12 CEST 2020”

```
Date d=new Date(); //data corrente  
d.toString() //dow mon dd hh:mm:ss zz yyyy
```

Impostazione

- Quali sono i thread?
- Qual è la risorsa condivisa?
- Quali variabili di stato ha?
- Quali sono le condizioni di sospensione?
- Quali vincoli di priorità? => quale ordine di priorità?



☐ Lock

☐ Condition

☐ Contatori dei thread sospesi

Esercizio 2

Programma C di sistema

Gestione e monitoraggio
dell'esecuzione del programma
che risolve l'esercizio 1

Esercizio 2 - Traccia (1/2)

Si realizzi un programma C di sistema che, utilizzando le system call Unix, preveda la seguente sintassi:

lancia fileout timeout fase

Dove:

- **fileout** è il nome di un file scrivibile
- **timeout** è un intero positivo.
- **fase** è una stringa che può assumere soltanto 2 valori: «**PSG**» o «**PN**».

Il processo iniziale **P0** dovrà creare un processo figlio **P1** il cui compito sarà quello di **eseguire**

l'applicazione java realizzata come soluzione dell'esercizio 1, in modo tale che il suo output venga ridirezionato nel file **fileout**.

Esercizio 2 - Traccia (2/2)

Il processo P0, dopo **timeout** secondi dovrà terminare forzatamente l'esecuzione di P1.

Una volta terminato P1, P0 dovrà stampare sullo standard output il contenuto del file **fileout**, filtrato secondo il seguente criterio:

- **Se fase = «PSG»**, verranno stampate le linee contenenti la stringa **“PSG”**.
 - **Se fase = «PN»**, verranno stampate le linee contenenti la stringa **“PN”**.
-

Esercizio 3

File comandi bash

Gestione e monitoraggio
dell'esecuzione del programma
che risolve l'esercizio 1

Esercizio 3 - Traccia

Si realizzi file comandi bash che preveda la seguente sintassi:

lancia.sh fileout timeout fase

Dove:

- **fileout** è il nome di un file scrivibile
- **timeout** è un intero positivo.
- **fase** è una stringa che può assumere soltanto 2 valori: «**PSG**» o «**PN**».

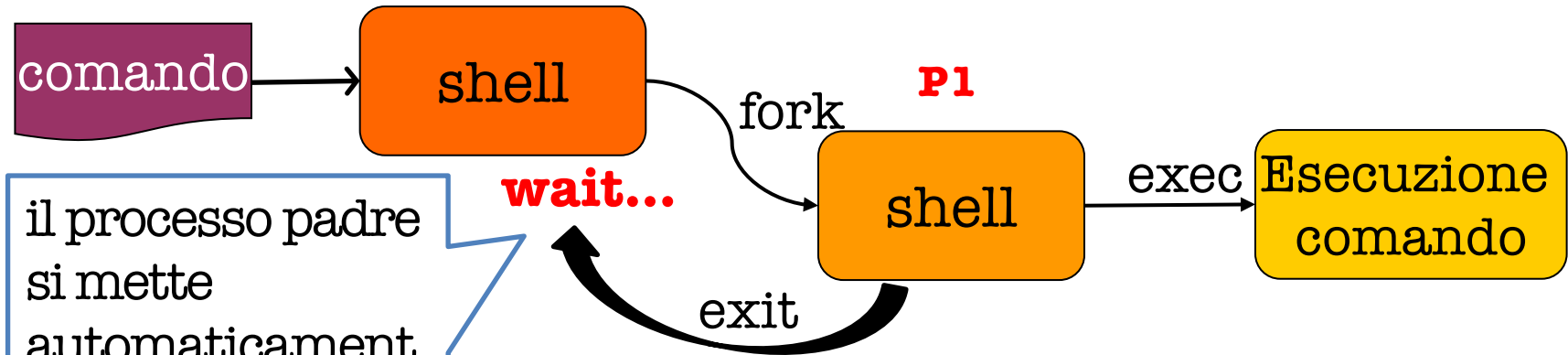
Il file comandi dovrà rispettare le stesse specifiche dell'esercizio 2.

E' consentito usare una **sleep** per far in modo che il padre attenda **timeout** secondi prima di terminare il figlio.

Come genero un figlio P1 in bash?

- Quando la shell riceve un comando da eseguire:

P0



Se al contrario NON voglio che il padre attenda la terminazione del figlio (perchè voglio che faccia qualcosa intanto):

-\$ comando &

esecuzione in background!

Come trovo il pid di un processo?

-\$ **ps**

PID	TTY	TIME	CMD
3203	pts/0	00:00:00	bash
3875	pts/0	00:00:00	ps

stampa a video una selezione di processi in esecuzione sul SO, uno per riga. Ciascuno con le proprie caratteristiche.

- **ps** (senza opzioni): tutti i processi con lo stesso UID dell'utente e lo stesso terminale dell'invoker (i.e., lanciati dalla stessa shell)
- **ps aux** (opzioni senza - davanti): tutti i processi del SO

Serve anche un modo per filtrare solo la prima colonna:

- **awk '{ print \$1 }'** → **NB**: \$1 non viene espanso perchè è tra apici singoli. Viene passata ad **awk** esattamente la stringa **\$1**, che nella sintassi di **awk** vuol dire: 'considera il testo come una tabella e filtra solo la colonna 1'