

Università di Ferrara
Laurea Triennale in Informatica
A.A. 2021-2022
Sistemi Operativi e Laboratorio

Lab-11. Programmazione di Sistema
Prova d'esame

Prof. Carlo Giannelli

`http://www.unife.it/scienze/informatica/insegnamenti/
sistemi-operativi-laboratorio`
`http://docente.unife.it/carlo.giannelli`
`https://ds.unife.it/people/carlo.giannelli`

11 giugno 2014 - conta VM - 1/3

Un'applicazione concorrente deve aiutare un amministratore di sistema a controllare il consumo di risorse delle varie applicazioni che compongono l'infrastruttura informatica della propria azienda. Poiché l'azienda utilizza delle applicazioni ospitate in "Cloud", le risorse utilizzate sono delle macchine virtuali (VM) fornite da vari fornitori di servizi Cloud, come Google, Amazon, SalesForce, etc.

L'applicazione, realizzata in C, deve presentare la seguente interfaccia:

conta_vm dir

dove dir è un nome assoluto di directory.

Le informazioni sulle VM utilizzate da ciascuna applicazione (si noti che ogni applicazione tipicamente potrà fare uso di più di una VM) sono presenti in un file con estensione "txt" per ognuno dei fornitori di servizi Cloud dell'azienda all'interno della directory dir passata come parametro. Quindi, il nome del file contenente i dati per le VM ospitate presso Amazon sarà "Amazon.txt", ecc.

11 giugno 2014 - conta VM - 2/3

In ciascun file saranno riportati i dati di monitoraggio per le VM ospitate dal rispettivo fornitore, una per riga. Ogni riga del file conterrà quindi:

- il nome della VM (es. “VM-126”);
- il nome dell’applicazione a cui essa fa riferimento (es. “contabilità”);
- il tipo di VM (es. “x1.large”);
- lo stato della VM (es. “operativa”, “inattiva”...).

L’applicazione deve essere composta da un processo iniziale P0 che si interfaccia con l’amministratore, da cui riceve (via terminale):

- la stringa alfanumerica nome_fornitore che rappresenta il nome di uno dei fornitori di servizi utilizzati (es. “Amazon”, “SalesForce”, ecc.);
- la stringa alfanumerica nome_applicazione che rappresenta il nome della specifica applicazione (tra le numerose applicazioni che formano l’infrastruttura informatica aziendale).

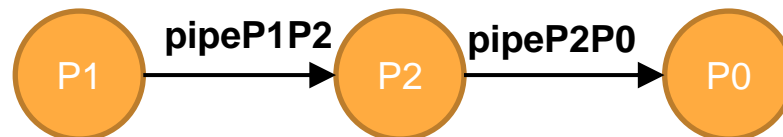
11 giugno 2014 - conta VM - 3/3

P0 deve quindi creare due processi figli P1 e P2 che devono collaborare per fornire all'amministratore **il numero di VM attualmente operative per l'applicazione indicata e ospitata presso il fornitore selezionato**. P0 riceve da P2 il numero di VM che rispettano i requisiti e stampa tale valore sullo standard output.

P0 continua a rispondere alle richieste dell'amministratore fino all'inserimento della stringa "fine" o alla pressione del tasto Ctrl-C. In entrambi i casi, il processo P0 deve visualizzare il numero di richieste servite prima di terminare.

Suggerimento 1: ricordarsi di eliminare eventuali processi zombie.

Suggerimento 2: un processo figlio P1 per identificare le VM relative ad una specifica applicazione, un processo figlio P2 per contare le VM effettivamente operative, il processo padre P0 → in totale 2 pipe.



8 giugno 2010 - trova vini - 1/3

Si progetti un'applicazione concorrente multi processo che permetta al sommelier di un ristorante di verificare la disponibilità di alcuni vini pregiati presso diverse cantine fornitrici.

L'applicazione dovrà essere realizzata in C e presentare la seguente interfaccia:

trova_vini cantina-1 ... cantina-N

dove cantina-1 ... cantina-N sono stringhe alfanumeriche che rappresentano nomi relativi di file contenenti i dati sui vini disponibili in ciascuna cantina.

L'applicazione deve essere composta da un processo iniziale P0 che prima di tutto si interfaccia con l'utente, da cui riceve (via terminale) il tipo di vino a cui è interessato. Per ogni richiesta, P0 genera N figli, P1 ... PN, uno per ogni file passato come parametro. Ciascun processo Pi a sua volta dovrà restituire a P0 le informazioni sui vini del tipo a cui l'utente è interessato, disponibili nel file cantina-i.

8 giugno 2010 - trova vini - 2/3

In particolare, i processi Pi devono aver cura di **selezionare solo i vini attualmente disponibili** in magazzino e di **restituire a P0 le informazioni richieste in ordine di prezzo crescente**. A questo fine, si supponga che i file cantina-i contengano una riga differente per ciascuno dei vini attualmente in vendita. Ogni riga è nel seguente formato: prezzo, tipo di vino, nome del vino, annata, e la stringa “disponibile” per i vini attualmente in magazzino.

P0 deve quindi stampare a video le informazioni sui vini disponibili in ciascuna cantina. Al termine di ogni richiesta dell'utente, P0 rimane in attesa della prossima richiesta. L'applicazione termina quando l'utente inserisce la stringa “fine”, ma **non deve terminare nel caso in cui venga premuto il Ctrl-C**.

8 giugno 2010 - trova vini - 3/3

Attenzione: per prima cosa realizzare la soluzione considerando una sola cantina, poi estendere ad n cantine.

Suggerimento: processo X per identificare il vino col nome giusto, processo Y per selezionare solo quelli disponibili, processo Z per ordinare in relazione al prezzo, processo padre P che stampa a video.