

Prova pratica del 18/10/2012
Durata della prova: 150 minuti

Cognome Nome Matr.

Lo studente legga attentamente il testo e produca il programma, il makefile, ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata del programma daranno luogo alla valutazione come **prova non superata**. Ricordarsi di indicare Nome, Cognome e matricola su questo stesso foglio, che dovrà essere in ogni caso consegnato alla Commissione. Al termine della prova lo studente dovrà fare verificare il funzionamento del programma ad un membro della Commissione.

Testo della prova

Si realizzi in linguaggio C/C++ un'applicazione **multiprocesso** per il calcolo distribuito della formula

$$y = (a * b + (c + d) * (e + f) + g * h) .$$

Il calcolo deve essere suddiviso tra più processi sulla base del **grafo delle dipendenze** riportato sotto. I processi si scambiano operandi e risultati utilizzando **code di messaggi UNIX**. Ciascun processo si mette in attesa di ricevere gli operandi di cui ha bisogno da una coda, e ritorna il risultato al processo richiedente attraverso un'ulteriore coda. Il processo P1 genera un vettore casuale di operandi, e richiede delle elaborazioni, da svolgere **in parallelo**, ai processi P2 (sugli operandi **a** e **b**), P3 (operandi **c**, **d**, **e**, **f**) e P4 (operandi **g** e **h**). Il processo P3 a sua volta manda delle richieste di elaborazione, anche esse **in parallelo**, ai processi P5 (operandi **c** e **d**) e P6 (operandi **e** e **f**), e ne attende i risultati prima di produrre un risultato a sua volta. Il processo P1 deve stampare a video gli operandi che esso genera, i risultati che riceve e il valore di **y** ottenuto. Il processo P1 ripete l'intero procedimento per 3 volte. Si richiede inoltre che lo studente sviluppi **6 programmi separati, uno per ogni processo P1...P6, e che vengano generati 6 eseguibili dalla compilazione**. Un ulteriore programma deve generare i processi P1...P6, tramite l'uso delle primitive `fork()` ed `exec()`, ed attenderne la terminazione.

