

Ex. 1	
Ex. 2	
Ex. 3	
Ex. 4	
Ex. 5	
Ex. 6	
Tot.	

# Sistemi Operativi

## Compito d'esame

15 Febbraio 2013

Matricola \_\_\_\_\_ Cognome \_\_\_\_\_ Nome \_\_\_\_\_

Docente:       Laface       Quer

**Non si possono consultare testi, appunti o calcolatrici. Riportare i passaggi principali. L'ordine sarà oggetto di valutazione.**

**Durata della prova: 60 minuti.**

1. Si supponga di avere a disposizione l'implementazione Pthread dei semafori binari attraverso le funzioni:

- `int pthread_mutex_init (pthread_mutex_t *mutex, const pthread_mutexattr_t *attr);`
- `int pthread_mutex_lock (pthread_mutex_t *mutex);`
- `int pthread_mutex_unlock (pthread_mutex_t *mutex);`

Si mostri come sia possibile implementare un semaforo con conteggio, ovvero non binario, il cui valore iniziale sia uguale a count.

*Suggerimento:* si ricordino le implementazioni "originali" delle primitive `signal` e `wait`.

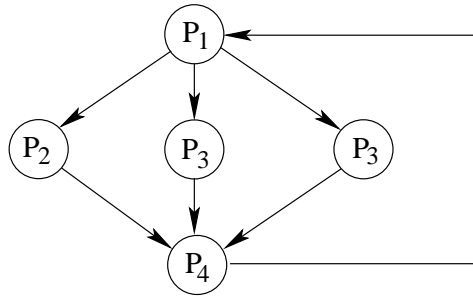
2. Si descrivano il significato e l'utilizzo delle system call `signal` e `kill`.

Che cosa è una corsa critica (race condition)?

Quali sono le funzioni principali della system call `signal(SIGCHLD, SIG_IGN)`? Che relazione ha con la system call `wait`?

3. Con riferimento alle soluzioni hardware al problema della sincronizzazione mediante le procedure di `testAndSet`, se ne illustrino le funzionalità, riportandone il codice e il relativo protocollo di utilizzo.

4. Dato il seguente grafo di precedenza, realizzarlo utilizzando il **minimo** numero possibile di semafori. Si utilizzino le primitive semaforiche implementate tramite pipe. Riportare il corpo dei processi ( $P_1, \dots, P_4$ ), nonché delle primitive `init`, `signal` e `wait` realizzate mediante pipe, e l'inizializzazione dei semafori. I processi rappresentati devono essere processi ciclici (con corpo del tipo `while(1)`).



Si osservi che il grafo prevede l'esecuzione di 4 processi tra i quali ci sono *due istanze identiche* dello stesso processo  $P_3$ .

5. Implementare uno script BASH che riceva sulla linea di comando quattro argomenti: `dir`, `dim1`, `dim2` e `cancella`.

Lo script deve

- controllare il corretto passaggio dei parametri
- cercare tutti i file
  - nell'albero di direttori con radice `dir` e profondità compresa tra 2 e 4
  - con estensione `txt`
  - con dimensione maggiore di `dim1+dim2`
- cancellare ogni riga che inizia oppure termina con la stringa contenuta nella variabile `cancella`.

Non si ricorra né all'utilizzo di SED né a quello di AWK.

6. Realizzare uno script AWK che legga da un file dati.txt, che contiene una linea con sequenze di numeri, separati da uno spazio, come nel seguente esempio:

```
1 1 1 1 1 2 2 2 2 2 2 3 3 3 4 4 4 4 4 1 1 1 2 2 3 3 3 3 3 1 1 1 1 1 1 2 3 3 3
```

Si può osservare che i numeri appaiono in serie con ripetizioni, ciascuna serie è in ordine crescente. Lo script deve creare un file differente per ciascuna serie, che riporti il numero di ripetizioni di ciascun numero che appare nella serie secondo il formato seguente (corrispondente all'elaborazione del file dati.txt):

File out.1	File out.2	File out.3
1 - 5	1 - 3	1 - 6
2 - 6	2 - 2	2 - 1
3 - 3	3 - 5	3 - 3
4 - 5		

Non si ricorra né all'utilizzo di BASH né a quello di SED.