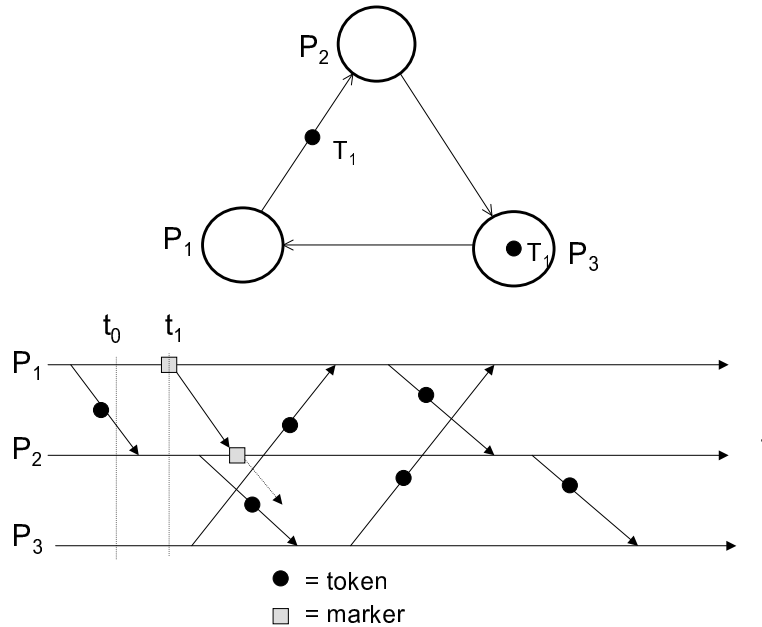


Università degli studi di Verona
Corso di Laurea Specialistica in Informatica
Sistemi Operativi Avanzati
13 Dicembre 2007

1. Si considerino i tre processi P_1 , P_2 , P_3 indicati in figura.



I tre processi comunicano tramite canali unidirezionali, e la comunicazione consiste nello scambio di due *token* T_1 e T_2 , inizialmente (al tempo t_0) nella situazione indicata nella figura (cioè T_1 è in possesso di P_3 e T_2 è in transito tra P_1 e P_2).

Si supponga poi che l'esecuzione del sistema evolva come indicato dal diagramma temporale, in cui i pallini neri indicano trasmissione di token.

Si mostri l'esecuzione dell'algoritmo dei *distributed snapshot*, indicando il risultato da esso fornito, assumendo che l'algoritmo sia lanciato da P_1 al tempo t_1 (vedi figura).

Giustificare la risposta

[9 punti]

2. Si consideri il seguente insieme di task a, b, c, d , che devono accedere a due risorse condivise (semafori) P e Q . I task hanno le seguenti caratteristiche:

Task	Priorità	Tempo di rilascio	Sequenza di operazioni
a	4	4	EEQVE
b	3	2	EVVQQE
c	2	2	EEE
d	1	0	EQQQE

Le operazioni possibili sono 3: E (esecuzione), P (accesso a P) e Q (accesso a Q). Ogni istanza di operazione indica un'unità di tempo. Per esempio, EVVE = (E per un'unità di tempo, V per due unità di tempo, etc.).

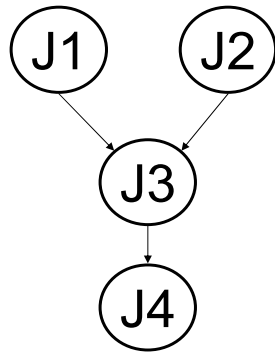
Mostrare l'esecuzione dei 4 processi nei tre casi:

- Utilizzo di un normale algoritmo a priorità (4=max, 1=min);
- Utilizzo di un normale algoritmo a priorità e gestione dei semafori usando la *priority inheritance*;
- Utilizzo di un normale algoritmo a priorità e gestione dei semafori usando il *priority ceiling*;

Si mostrino i diagrammi temporali, e si calcoli per ogni task il tempo di risposta.

[9 punti]

3. Sia dato il se



<i>Task</i>	C_i	d_i
T_1	3	5
T_2	2	6
T_3	1	6
T_4	3	9

Fornire una schedulazione valida dei task utilizzando l'algoritmo EDF*, modificando opportunamente i tempi di rilascio e le deadline. **[7 punti]**

4. Si consideri il seguente insieme di task periodici:

<i>Task</i>	C_i	T_i
τ_1	2	6
τ_2	2	10
τ_3	1	12

Si consideri poi la seguente sequenza di richieste aperiodiche:

$$(1, 2), (6, 1), (10, 1), (15, 2), (22, 1)$$

dove (i, j) indicano rispettivamente il tempo di arrivo a_i e il burst c_i . Si assuma che la deadline di ogni richiesta aperiodica corrisponda al tempo di arrivo della richiesta successiva. Per esempio, la richiesta $(1, 2)$ ha come deadline il tempo $t = 6$.

Si mostri l'andamento temporale dei vari processi e delle richieste aperiodiche usando una soluzione di tipo *background scheduling*, in cui la coda dei processi periodici è gestita secondo RM, e quella delle richieste aperiodiche in modo FIFO.

Si indichi chiaramente se tutte le deadline vengono rispettate.

[8 punti]