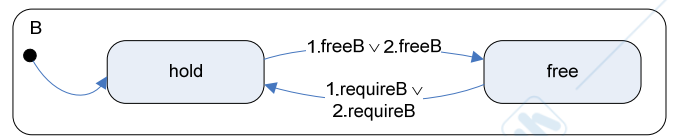
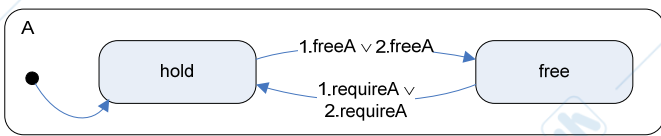
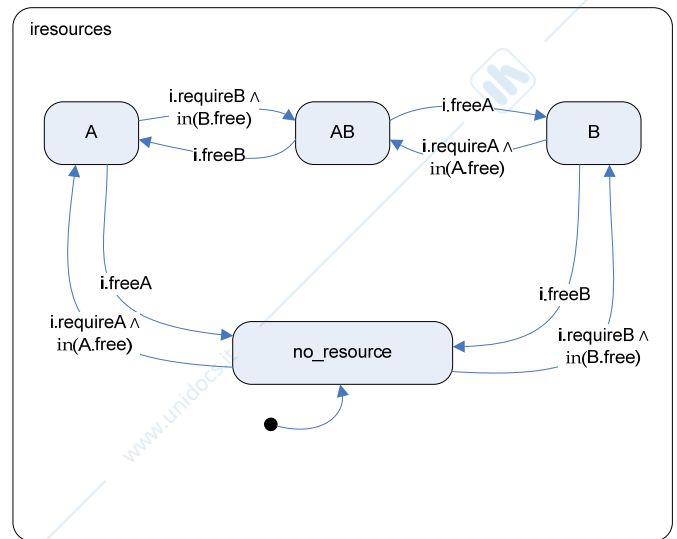
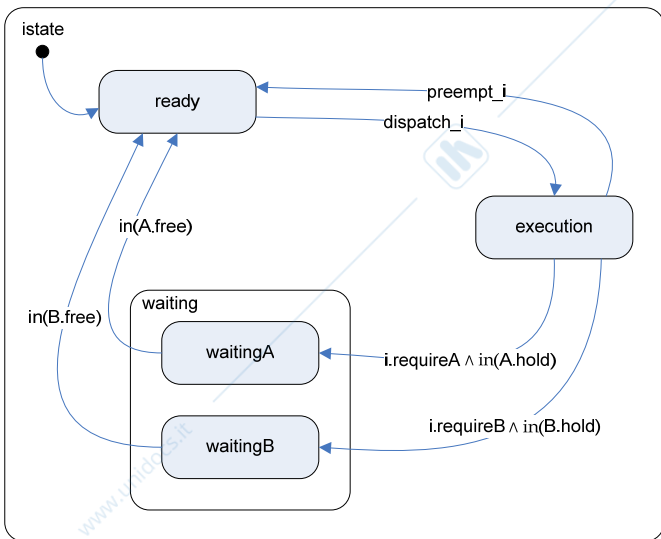


| | | | |
|---|---|----------------|-----------|
|  | Politecnico di Milano Facoltà di Ingegneria dell'Automazione INFORMATICA INDUSTRIALE Appello 8 luglio 2008 | COGNOME E NOME | |
| | RIGA | COLONNA | MATRICOLA |

- Il presente plico pinzato, composto di quattro fogli (fronte/retro), deve essere debitamente compilato con cognome, nome, numero di matricola, posizione durante lo scritto, e deve essere firmato.
- I compiti non compilati, non firmati o con fogli mancanti non saranno considerati validi e quindi non saranno corretti.
- Sarà valutato solo quanto scritto su questi fogli.
- Non è consentito consultare testi né appunti.
- Sul tavolo non devono essere presenti telefoni cellulari, né astucci, né custodie di altro tipo.
- Le risposte devono essere scritte negli appositi riquadri, qualsiasi testo esterno a tali riquadri non verrà preso in considerazione.
- Se lo spazio per la soluzione degli esercizi non fosse sufficiente, si può scrivere sull'ultimo foglio.

FIRMA

Esercizio 1 (10 punti). Si vuole specificare tramite il formalismo degli Statecharts, e senza l'utilizzo di variabili, l'uso di due risorse (A e B) da parte di due processi periodici (1 e 2). Un generico processo i può trovarsi nello stato di *ready*, *execution* o *wait*, in quest'ultimo si riconoscono due sottostati: *waitingA* e *waitingB* (corrispondenti all'attesa della risorsa A e B rispettivamente). Si presuppone che un processo non possa aspettare di accedere contestualmente a più di una risorsa (un processo può però utilizzare entrambe le risorse). Gli eventi che guidano le transizioni da *execution* a *ready* sono generati dallo scheduler e sono: *dispatch_i* e *preempt_i*. Ogni qual volta un processo i è in *execution* e richiede una risorsa X , viene generato un evento *i.requireX*. Qualora la risorsa è occupata (vedi sotto) il processo passa a *waitingX*. È necessario modellizzare lo stato delle risorse e l'uso delle risorse da parte di ogni singolo processo. Una risorsa X può trovarsi o libera (*free*) o occupata (*hold*). Le transizioni da *free* a *hold* sono guidate dagli eventi *i.requireX* e *i.freeX* (quest'ultimo evento viene generato quando un processo i in esecuzione libera la risorsa X). L'uso delle risorse da parte di un processo i viene modellizzata con uno statechart costituito da un numero di stati pari al numero di combinazioni di risorse utilizzate dal processo.



Esercizio 2 (9 punti). Si consideri il problema di analizzare la schedulabilità del problema di schedulazione composto dai processi periodici τ_1 - τ_5 riportati in tabella:

| | τ_1 | τ_2 | τ_3 | τ_4 | τ_5 |
|----------|----------|----------|----------|----------|----------|
| ϕ_i | 0 | 0 | 0 | 0 | 0 |
| C_i | 1 | 1 | 2 | 2 | 1 |
| T_i | 4 | 5 | 10 | 11 | 12 |

Si dimostri la schedulabilità del problema qualora si utilizzi rate monotonic, discutendo l'applicazione, nell'ordine, di: condizione di Lyu&Layland, condizione di Bini&Buttazzo, e, infine, response time analysis. Se ne riporti la schedulazione

Si consideri successivamente un problema di schedulazione derivato da quello sopra, dove i processi τ_4 e τ_5 vengono eliminati. Dati i processi aperiodici soft real-time riportati nella tabella qui sotto, si riporti la schedulazione prodotta tramite Slack Stealing dove i processi aperiodici sono ordinati rispetto al tempo di arrivo.

| | J_1 | J_2 | J_3 | J_4 |
|-------|-------|-------|-------|-------|
| a_i | 1 | 6 | 16 | 17 |
| C_i | 3 | 2 | 1 | 1 |

Applico la condizione di Lyu&Layland:

$$U_{lub} = 5(2^{0.2} - 1) = 0.7425$$

$$U = 1/4 + 2/7 + 2/9 + 2/13 = 0.9152$$

$$U > U_{lub}$$

Non è possibile stabilire se il problema è schedulabile o meno tramite la condizione di Lyu&Layland.

Applico la condizione di Bini&Buttazzo:

$$(1/4 + 1) * (1/5 + 1) * (2/10 + 1) * (2/11 + 1) * (1/12 + 1) > 2$$

Non è possibile stabilire se il problema è schedulabile o meno tramite la condizione di Bini&Buttazzo.

Applico la RTA:

$$R_1^0 = 1, I_1^0 = 0,$$

$$R_1^1 = 1, \text{ quindi } \tau_1 \text{ è schedulabile.}$$

$$R_2^0 = 1, I_2^0 = 1,$$

$$R_2^1 = 2, I_2^1 = 1,$$

$$R_2^2 = 2, \text{ quindi } \tau_2 \text{ è schedulabile.}$$

$$R_3^0 = 2, I_3^0 = 2,$$

$$R_3^1 = 4, I_3^1 = 2,$$

$$R_3^2 = 4, \text{ quindi } \tau_3 \text{ è schedulabile.}$$

$$R_4^0 = 2, I_4^0 = 4,$$

$$R_4^1 = 6, I_4^1 = 6,$$

$$R_4^2 = 8, I_4^2 = 6,$$

$$R_4^3 = 8, \text{ quindi } \tau_4 \text{ è schedulabile.}$$

$$R_5^0 = 1, I_5^0 = 6,$$

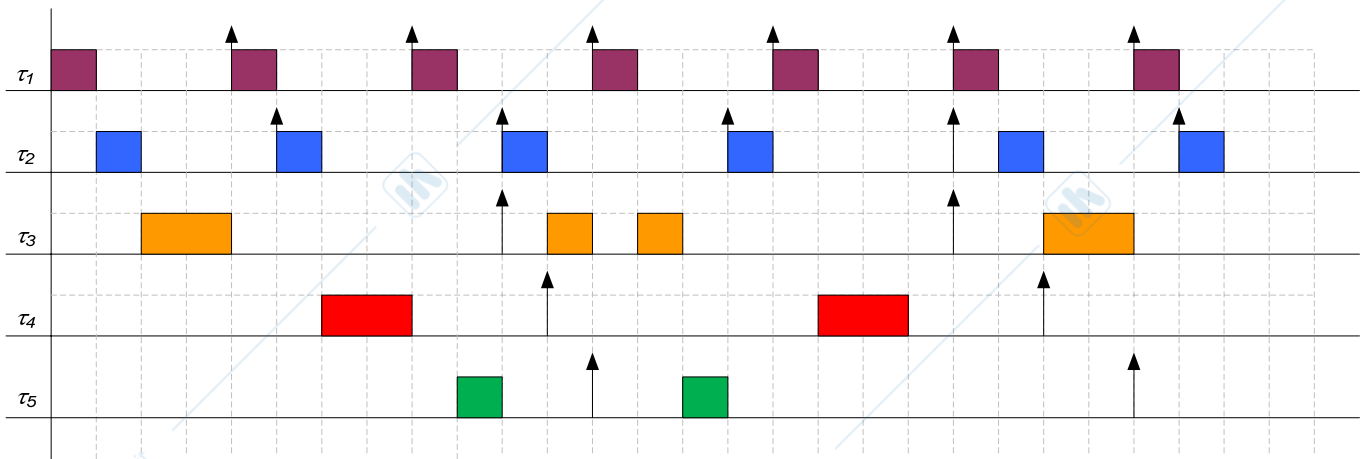
$$R_5^1 = 7, I_5^1 = 8,$$

$$R_5^2 = 9, I_5^2 = 8,$$

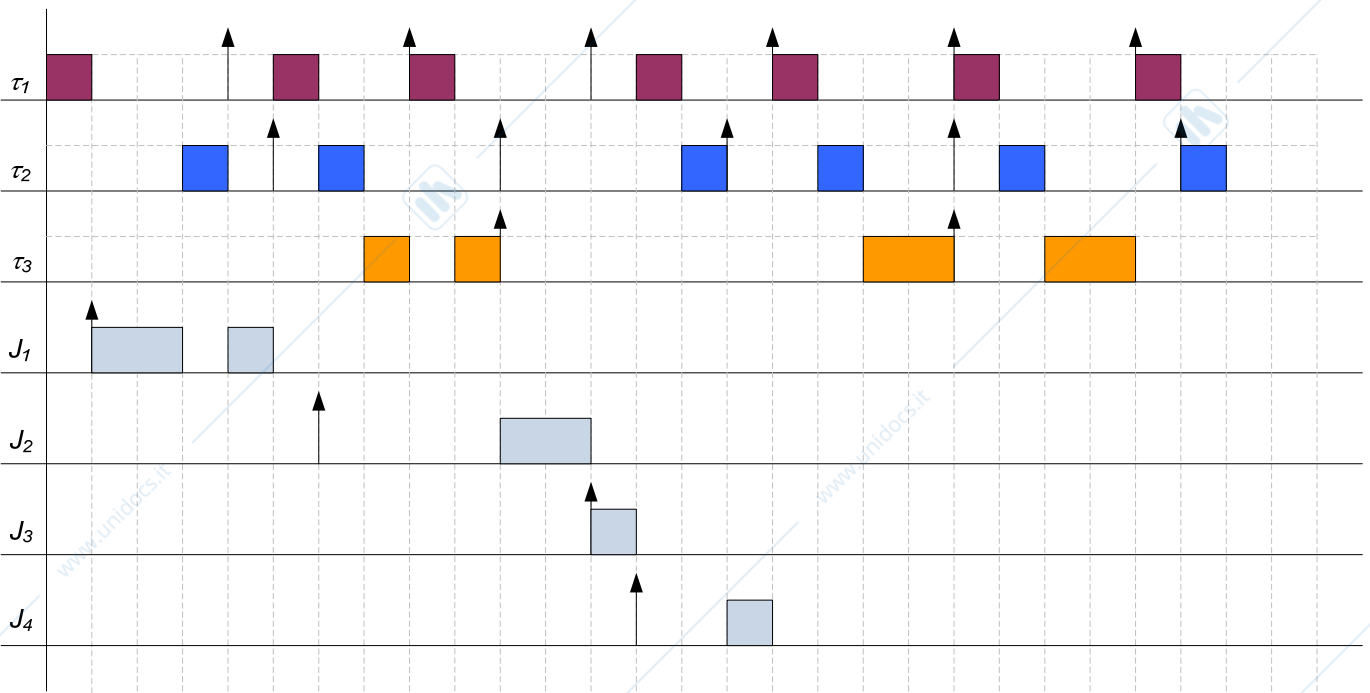
$$R_5^3 = 9, \text{ quindi } \tau_5 \text{ è schedulabile.}$$

Il problema è quindi schedulabile.

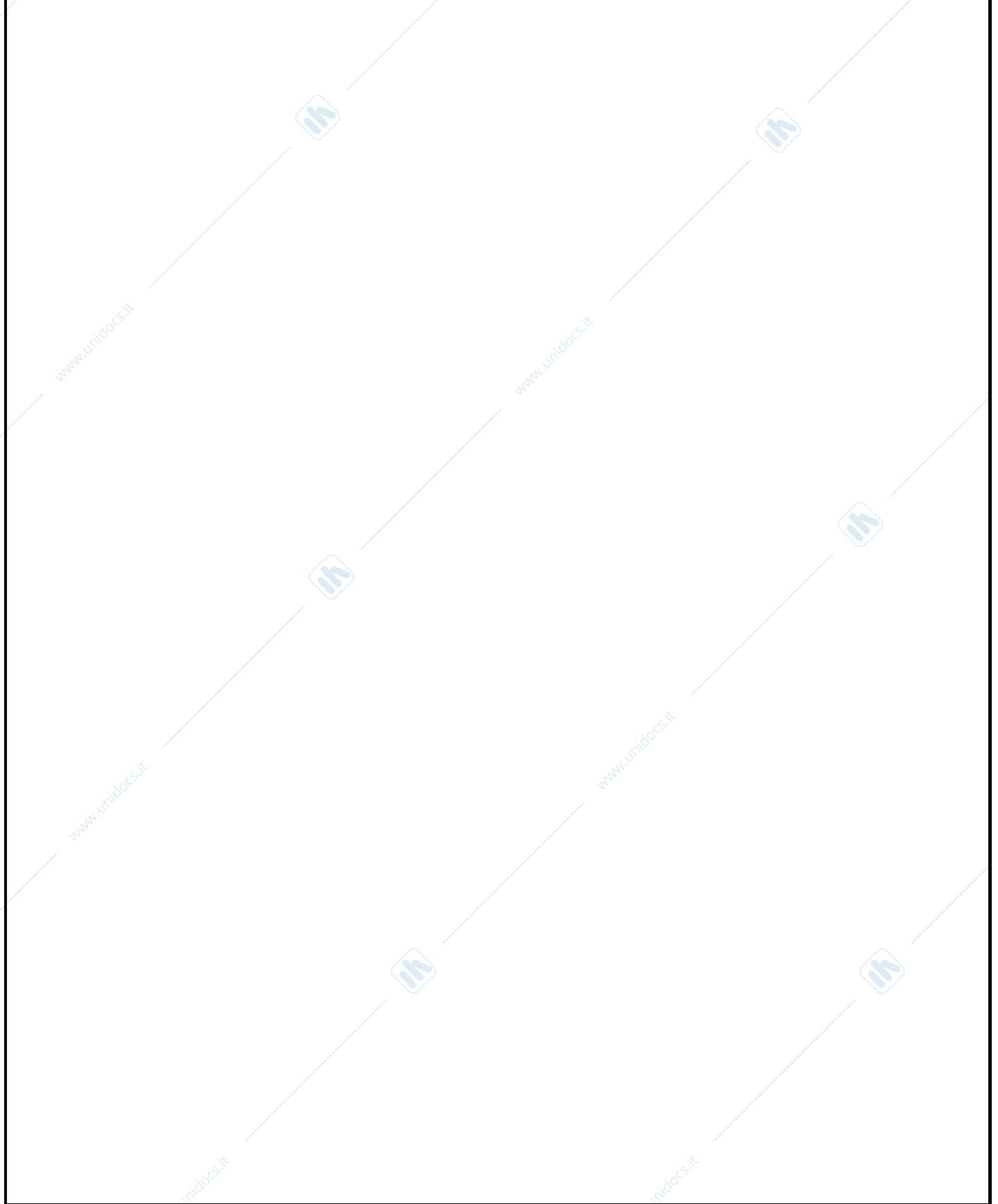
La schedulazione prodotta da rate monotonic è la seguente.



Riporto ora la schedulazione prodotta dall'algoritmo di Slack Stealing applicato al problema di schedulazione misto soft real-time.



Esercizio 3 (6 punti). Si dica per quali problemi di schedulazione l'algoritmo EDF è ottimale e che forma di ottimalità garantisce. Si dica inoltre per quali problemi è applicabile l'algoritmo TBS* e lo si descriva.



Esercizio 4 (6 punti). Rispondere alle seguenti domande inerenti le reti di computer.

- Quali sono le strategie di commutazione e come si differenziano?
- Quale strategia di commutazione è quella adoperata nel protocollo TCP/IP?
- Quale è la relazione tra pila protocollare ISO/OSI e la struttura del singolo pacchetto scambiato sulla rete?
- Quale è la struttura del pacchetto su protocollo IEEE 802.3 (Ethernet)?

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari