

SCS  $\rightarrow$  scheduling statico

Proprietà di RCT e mct ed il loro  
diversamento):  
Major  $\rightarrow$  minor

(1, 14, 14)

$\downarrow$   
denote  $\rightarrow$  deadline

- 1) RCT divisibile per mct
- 2)  $mct \leq D_i$ , cioè la deadline
- 3)  $mct + (mct - RCT(mct, T_i)) < D_i$   
per ogni processo

E' statico perché ci sono sempre gli stessi  
processi conosciuti da gestire  
E' vulnerabile ai cambiamenti, è ritenuto  
troppo a basso livello.  
Esercizi pag. 19



→ Scheduling Dinamico

se il processo è in Idle perché il processo che era previsto per quel momento non è pronto viene sfruttato per l'esecuzione di un altro processo, è differenza dello scheduling statico

$D_i \leq T_i \Rightarrow D_i = T_i$  semplificazione

Lo scheduling dinamico è più efficiente

- Fattibilità di uno scheduling

- Ottimalità di uno scheduling

(Lo scheduling è l'algoritmo di schedulazione utilizzato dallo scheduler dei processi)

- Differenza tra scheduling pre-emptivo

(può essere interrotto, per salvaguardare il rispetto della deadline, il processo in esecuzione) oppure non pre-emptivo

(non può essere interrotto)



Tempo di risposta: quando il processo è  
avuto

Tempo di invocazione: quando il processo  
è pronto per essere eseguito (ready)

Tempo di rilascio: quando il processo è  
terminato

Priorità statiche:

- RATE MONITORING

la priorità viene assegnata al processo  
più breve e si realizza la pre-emption

Disegno di Gantt per la riepilogazione  
della schedulazione dei processi

Priorità dinamiche:

la priorità viene assegnata dinamicamente,  
ovvero chi ha la deadline più  
imminente

- EARLIEST DEADLINE FIRST

pag. 23 (b) c'è un context switch in  
meno rispetto ad (a) 3



# • LEAST SLACK TIME

- Deadline relativo

- Deadline assoluta

In ambiente hard real-time RM è ottimale se esiste uno scheduling fattibile con priorità fisse

Fattore di utilizzazione dello CPU

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \quad \text{dove essere } U \leq 1 \quad \text{per la fattibilità}$$

(feasibility) dello scheduling

RM non riesce a schedulare i processi se sono più del 70% secondo il bound LL perché  $LL(n, n \rightarrow \infty) \rightarrow 0.69$  e cioè se il processore lavora al massimo del 70% della sua capacità.