

Sistemi Operativi

# **Scheduling**

**LEZIONE 15** 

prof. Antonino Staiano

Corso di Laurea in Informatica – Università di Napoli Parthenope
antonino.staiano@uniparthenope.it

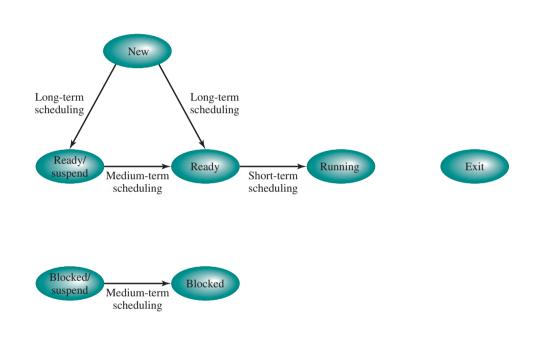
### Introduzione

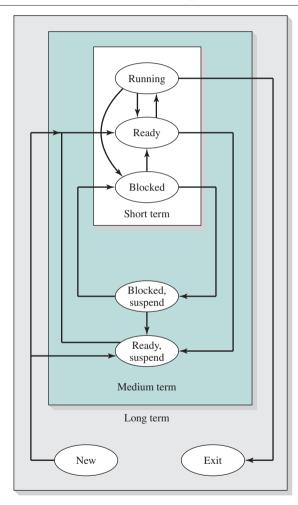
- Terminologia e concetti dello scheduling
- Politiche di scheduling senza prelazione
- Politiche di scheduling con prelazione
- Scheduling in pratica
- Scheduling real-time

### Introduzione

- Tipi di scheduling
  - Lungo termine
    - Ammette un processo all'insieme dei processi da eseguire
  - Medio termine
    - Ammette un processo all'insieme dei processi presenti parzialmente o totalmente in memoria
  - Breve termine
    - Quale dei processi pronti è assegnato alla CPU
  - |/0
    - Quale richiesta di I/O pendente di un processo deve essere gestita da un dispositivo di I/O disponibile

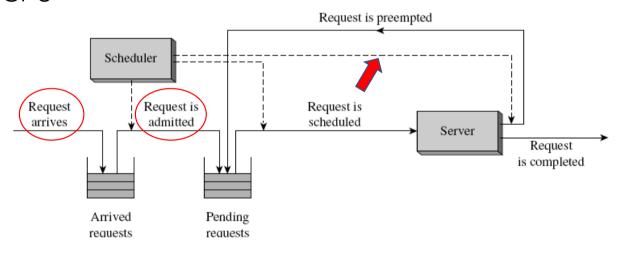
## Scheduling e transizioni di stato dei processi





### Terminologia e Concetti di Scheduling

- Lo scheduling è l'attività che consiste nel selezionare la prossima richiesta da servire per un server
  - In un SO, una richiesta è l'esecuzione di un job o di un processo ed il server è la CPU



Schematizzazione dello scheduling

### Terminologia e Concetti di Scheduling (cont.)

• Termini e concetti di scheduling

Term or concept	Definition or description
Request related	
Arrival time	Time when a user submits a job or process.
Admission time	Time when the system starts considering a job or process for scheduling.
Completion time	Time when a job or process is completed.
Deadline	Time by which a job or process must be completed to meet the response requirement of a real-time application.
Service time	The total of CPU time and I/O time required by a job, process or subrequest to complete its operation.
Preemption	Forced deallocation of CPU from a job or process.
Priority	A tie-breaking rule used to select a job or process when many jobs or processes await service.

## Terminologia e Concetti di Scheduling (cont.)

#### User service related: individual request

Deadline overrun The amount of time by which the completion time of

a job or process exceeds its deadline. Deadline

overruns can be both positive or negative.

Fair share A specified share of CPU time that should be devoted

to execution of a process or a group of processes.

Response ratio The ratio

time since arrival + service time of a job or process

service time of the job or process

Response time (rt) Time between the submission of a subrequest for

processing to the time its result becomes available. This concept is applicable to interactive processes.

Time between the submission of a job or process and

its completion by the system. This concept is

meaningful for noninteractive jobs or processes only.

Weighted turnaround (w) Ratio of the turnaround time of a job or process to its

own service time.

#### User service related: average service

Mean response time (rt) Average of the response times of all subrequests

serviced by the system.

Mean turnaround Average of the turnaround times of all jobs or

time  $(\overline{ta})$  processes serviced by the system.

Performance related

Schedule length The time taken to complete a specific set of jobs or

processes.

Throughput The average number of jobs, processes, or

subrequests completed by a system in one unit

of time.

Turnaround time (ta)

### Tecniche di Scheduling

- Gli scheduler usano tre tecniche fondamentali per ottenere un buon servizio utente ed elevate prestazioni di sistema
  - Scheduling basato su priorità
    - Fornisce un elevato throughput del sistema
  - Riordino delle richieste
    - Implicito nella prelazione
      - Migliora il servizio utente (in un sistema time-sharing) e/o il throughput (in un sistema multiprogrammato)
  - Variazione dello slot temporale
    - Ricordiamo  $\eta = \delta / (\delta + \sigma)$ , con  $\eta$  efficienza di CPU,  $\sigma$  overhead dello scheduling,  $\delta$  slot temporale
    - Valori più piccoli degli slot temporali forniscono migliori tempi di risposta, ma una minore efficienza della CPU
    - Impiega slot temporali più grandi per processi CPU-bound

### Il Ruolo delle Priorità

- Priorità: regola adottata dallo scheduler quando ci sono molte richieste in attesa della CPU
  - Può essere statica o dinamica
- E' possibile fare un riordino sulla base delle priorità
  - Ad esempio, i processi brevi serviti prima di quelli più lunghi
  - Alcuni riordini necessitano di complesse funzioni di priorità
- Cosa accade se i processi hanno la stessa priorità?
  - Si usa uno scheduling round-robin
- Può portare alla starvation di richieste a bassa priorità
  - Soluzione: aging delle richieste

### Scheduling senza Prelazione

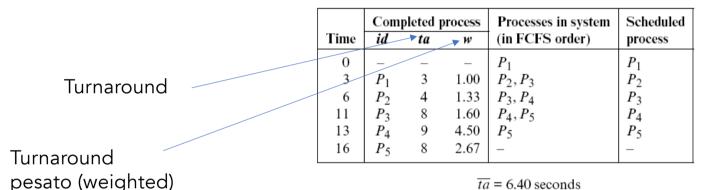
- La CPU serve sempre una richiesta schedulata fino al completamento
- Conveniente in virtù della sua semplicità
  - · Lo scheduler non deve distinguere tra richieste non elaborate e parzialmente elaborate
  - Lo scheduler deve solo eseguire la funzione di riordino per migliorare servizio utente o prestazioni del sistema
- Alcune politiche di scheduling senza prelazione:
  - Scheduling First-Come, First-Served (FCFS)
  - Scheduling Shortest Job First (SJF)
  - Scheduling High Response Ratio (HRN)

Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

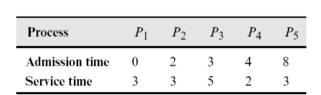
Processi per lo Scheduling

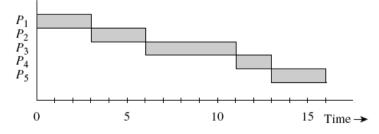
### Scheduling First Come, First Served (FCFS)

Le richieste sono schedulate sempre nell'ordine in cui giungono al sistema



 $\overline{w} = 6.40 \text{ second}$  $\overline{w} = 2.22$ 



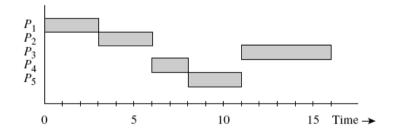


### Scheduling Shortest Job First (SJF)

Schedula sempre la richiesta con il minimo tempo di servizio

	Comp	oleted p	rocess	Processes	Scheduled
Time	id	ta	w	in system	process
0	_	_	_	$\{P_1\}$	$P_1$
3	$P_1$	3	1.00	$\{P_2, P_3\}$	$P_2$
6	$P_2$	4	1.33	$\{P_3, P_4\}$	$P_4$
8	$P_4$	4	2.00	$\{P_3, P_5\}$	$P_5$
11	$P_5$	3	1.00	$\{P_3\}$	$P_3$
16	$P_3$	13	2.60	{}	_

$$\overline{ta} = 5.40 \text{ seconds}$$
  
 $\overline{w} = 1.59$ 



#### Problematiche:

- Può causare starvation dei processi più lunghi
- I tempi di servizio non sono noti a priori

Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

### Scheduling High Response Ratio Next (HRN)

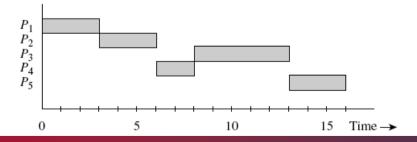
 Calcola i rapporti di risposta di tutti i processi nel sistema e seleziona il processo con il rapporto più elevato

Response ratio = 
$$\frac{\text{time since arrival + service time of the process}}{\text{service time of the process}}$$

	Comp	oleted p	rocess	Res	sponse i	ratios o										
Time	id	ta	w	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	Scheduled process							
0	_	_	_	1.00					$P_1$							
3	$P_1$	3	1.00		1.33	1.00			$P_2$							
6	$P_2$	4	1.33			1.60	2.00		$P_4$							
8	$P_4$	4	2.00			2.00		1.00	$P_3$							
13	$P_3$	10	2.00					2.67	$P_5$							
16	$P_5$	8	2.67						_							

L'uso del tasso di risposta contrasta la starvation

$$\overline{ta} = 5.8 \text{ seconds}$$
  
 $\overline{w} = 1.80$ 



Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

### Politiche di Scheduling con Prelazione

- Nello scheduling con prelazione, il server può commutare alla prossima richiesta prima di completare quella corrente
  - · La richiesta prelazionata è messa in una lista di richieste pendenti
  - Il suo servizio è ripristinato quando è nuovamente schedulata
- Una richiesta può essere schedulata molte volte prima che sia completata
  - Overhead maggiore rispetto allo scheduling senza prelazione
- Usato nei SO multi-programmati e time-sharing

### **Scheduling Round-Robin**

- Obiettivo: fornire buoni tempi di risposta a tutte le richieste
- Il time slice, o quanto di tempo,  $\delta$ , è la massima quantità di tempo di CPU che una richiesta schedulata può usare
- Al termine del quanto di tempo, la richiesta viene sospesa
  - Il kernel genera un interrupt allo scadere del quanto di tempo
- Fornisce servizi comparabili a tutti i processi CPU-bound
  - Valori approssimativamente uguali dei turnaround pesati
  - L'effettivo valore del turnaround pesato di un processo dipende dal numero di processi nel sistema
  - I turnaround pesati di processi che fanno operazioni di I/O dipendono dalla durata di tali operazioni
- Lo scheduling RR non è una misura adeguata per le prestazioni del sistema poiché non favorisce i processi brevi

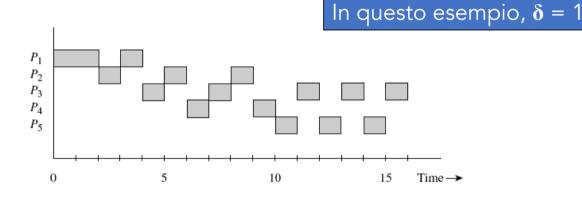
## Scheduling Round-Robin (RR) con Time-Slice

Time of schedul	ing	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	с	$t_a$	w
Position of	$P_1$	1	1	2	1													4	4	1.33
Processes in	$P_2$			1	3	2	1	3	2	1								9	7	2.33
ready queue	$P_3$				2	1	3	2	1	4	3	2	1	2	1	2	1	16	13	2.60
(1 implies	$P_4$					3	2	1	3	2	1							10	6	3.00
head of queue)	$P_5$									3	2	1	2	1	2	1		15	7	2.33
Process schedule	ed	$P_1$	$P_1$	$P_2$	$P_1$	$P_3$	$P_2$	$P_4$	$P_3$	$P_2$	$P_4$	$P_5$	$P_3$	$P_5$	$P_3$	$P_5$	$P_3$			

 $\overline{ta}$  = 7.4 seconds,  $\overline{w}$  = 2.32 c: completion time of a process

#### Oss.:

ta e w peggiori rispetto alle politiche senza prelazione



Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

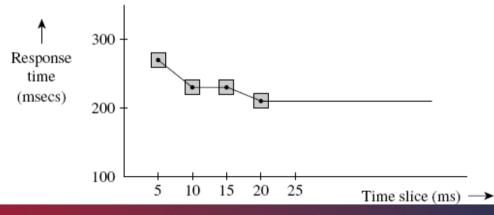
### Variazione del Tempo di Risposta nello Scheduling RR

- Ricordiamo che per un processo il tempo di risposta  $\mathbf{rt} = \mathbf{n} \times (\mathbf{\delta} + \boldsymbol{\sigma})$ , dove n è il numero di processi nel sistema
- In realtà la relazione tra  ${\sf rt}$  e  ${\sf \delta}$  è più complessa
  - Alcuni processi saranno bloccati per operazioni di I/O o per l'attesa di azioni degli utenti
    - · Quindi rt è governato dal numero di processi attivi piuttosto che da n
  - Se una richiesta ha bisogno di più di δ secondi di tempo di CPU, sarà schedulata più di una volta prima di produrre una risposta
    - Per cui per piccoli valori di  $\delta$ , i valori dei tempi di risposta possono essere più elevati

### Variazione del Tempo di Risposta nello Scheduling RR

• Per piccoli valori di  $\delta$ , l'rt di una richiesta può essere maggiore

		•		•
Time slice	5 ms	10 ms	15 ms	20 ms
Average rt for first subrequest (ms)	248.5	186	208.5	121
Average rt for subsequent subrequest (ms)	270	230	230	210
Number of scheduling decisions	600	300	300	150
Schedule length (ms)	4200	3600	3600	3300
Overhead (percent)	29	17	17	9



- 10 processi, ognuno riceve 15 sottorichieste di 20 ms ciascuna
- Una sottorichiesta è seguita da un'operazione di I/O di 10 ms
- 2 ms sono per lo scheduling

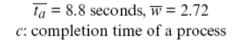
### **Least Completed Next (LCN)**

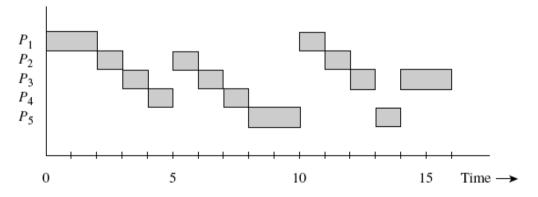
- Schedula il processo che ha usato fino a quel momento la minor quantità di tempo di CPU
- La natura del processo (CPU o I/O bound) e il suo requisito di CPU non influenzano il suo progresso nel sistema
- Tutti i processi progrediscono in modo approssimativamente uguale in termini di tempo di CPU consumato
  - Quindi, è garantito che i processi brevi finiscono prima dei processi lunghi
- L'inconveniente è che i processi lunghi possono soffrire di starvation
  - Ma anche processi non molto lunghi possono soffrire di starvation o elevati tempi di turnaround poiché LCN trascura i processi esistenti a vantaggio dei processi appena arrivati

### **Least Completed Next (LCN)**

Time of schedu	ıling	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	С	$t_a$	w
	$P_1$	0	1	2	2	2	2	2	2	2	2	2						11	11	3.67
CPU time	$P_2$			0	1	1	1	2	2	2	2	2	2					12	10	3.33
consumed by	$P_3$				0	1	1	1	2	2	2	2	2	2	3	4	5	16	13	2.60
processes	$P_4$					0	1	1	1									8	4	2.00
	$P_5$									0	1	2	2	2	2			14	6	2.00
Process schedu	ıled	$P_1$	$P_1$	$P_2$	$P_3$	$P_4$	$\overline{P_2}$	$P_3$	$P_4$	$P_5$	$P_5$	$P_1$	$\overline{P}_2$	$P_3$	$P_5$	$P_3$	$P_3$			

A parità di completamento, è preferito il processo che non è servito da più tempo





#### Problemi:

- Priva i processi lunghi dell'attenzione della CPU
- Trascura i processi esistenti se arrivano nuovi processi

Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

### **Shortest Time to Go (STG)**

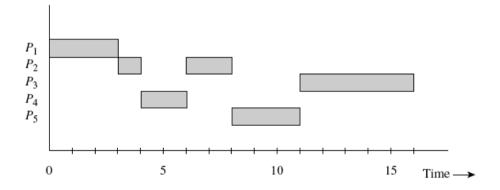
- Schedula un processo che ha il minor requisito di tempo di CPU restante
- Favorisce i processi brevi e fornisce un buon throughput
- Favorisce i processi in prossimità di completamento rispetto a nuovi processi brevi che arrivano nel sistema
  - Migliora i tempi di turnaround e di turnaround pesato
- Processi lunghi possono soffrire di starvation

### **Shortest Time to Go (STG)**

Time of scheo	luling	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	с	$t_a$	w
Remaining	$P_1$	3	2	1														3	3	1.00
CPU time	$P_2$			3	3	2	2	2	1									8	6	2.00
requirement	$P_3$				5	5	5	5	5	5	5	5	5	4	3	2	1	16	13	2.60
of a process	$P_4$					2	1											6	2	1.00
	$P_5$									3	2	1						11	3	1.00
Process sched	luled	$P_1$	$P_1$	$P_1$	$\overline{P}_2$	$P_4$	$P_4$	$\overline{P_2}$	$\overline{P}_2$	$P_5$	$P_5$	$P_5$	$P_3$	$P_3$	$P_3$	$P_3$	$P_3$			

A parità di completamento, è preferito il processo che non è servito da più tempo

 $\overline{t_a}$  = 5.4 seconds,  $\overline{w}$  = 1.52 c: completion time of a process



Process	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
Admission time	0	2	3	4	8
Service time	3	3	5	2	3

Analogo alla potitica SJF -> processi più lunghi possono andare in starvation