

# Scheduling dei Processi

Sistemi Operativi

Antonino Staiano

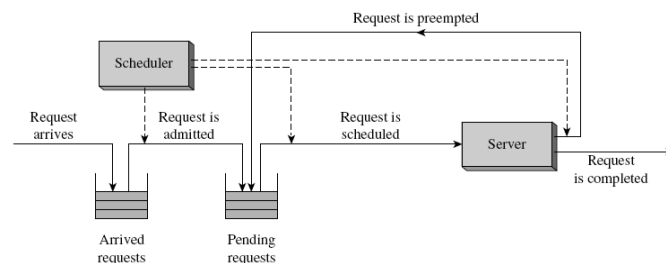
Email: antonino.staiano@uniparthenope.it

## Introduzione

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

## 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
<b>Request related</b>	
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.)

<b>User service related: individual request</b>	
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 $\frac{\text{time since arrival} + \text{service time of a job or process}}{\text{service time of the job or process}}$
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.
Turnaround time ( $ta$ )	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.
<b>User service related: average service</b>	
Mean response time ( $\bar{rt}$ )	Average of the response times of all subrequests serviced by the system.
Mean turnaround time ( $\bar{ta}$ )	Average of the turnaround times of all jobs or processes serviced by the system.
<b>Performance related</b>	
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.

## 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 del server
  - 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

- Un server serve sempre una richiesta schedulata fino al completamento
- Intrigante 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

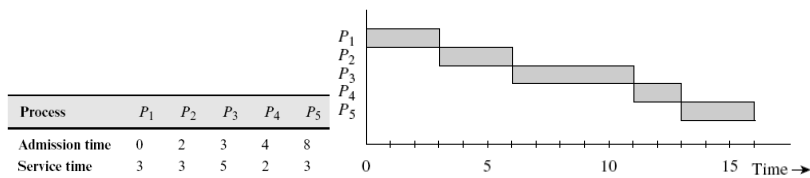
## Scheduling First Come, First Served (FCFS)

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

Time	Completed process			Processes in system (in FCFS order)	Scheduled process
	id	ta	w		
0	—	—	—	P <sub>1</sub>	P <sub>1</sub>
3	P <sub>1</sub>	3	1.00	P <sub>2</sub> , P <sub>3</sub>	P <sub>2</sub>
6	P <sub>2</sub>	4	1.33	P <sub>3</sub> , P <sub>4</sub>	P <sub>3</sub>
11	P <sub>3</sub>	8	1.60	P <sub>4</sub> , P <sub>5</sub>	P <sub>4</sub>
13	P <sub>4</sub>	9	4.50	P <sub>5</sub>	P <sub>5</sub>
16	P <sub>5</sub>	8	2.67	—	—

Turnaround  
Turnaround  
pesato (weighted)

$\overline{ta} = 6.40$  seconds  
 $\overline{w} = 2.22$



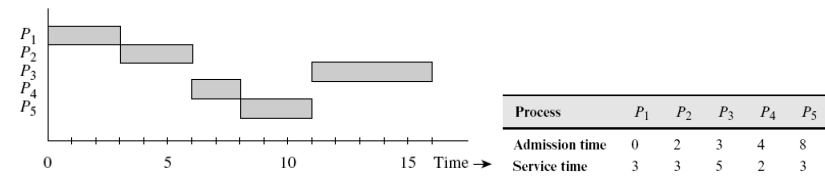
## Scheduling Shortest Job First (SJF)

- Scheda sempre la richiesta con il minimo tempo di servizio

Time	Completed process			Processes in system	Scheduled process
	id	ta	w		
0	—	—	—	{P <sub>1</sub> }	P <sub>1</sub>
3	P <sub>1</sub>	3	1.00	{P <sub>2</sub> , P <sub>3</sub> }	P <sub>2</sub>
6	P <sub>2</sub>	4	1.33	{P <sub>3</sub> , P <sub>4</sub> }	P <sub>4</sub>
8	P <sub>4</sub>	4	2.00	{P <sub>3</sub> , P <sub>5</sub> }	P <sub>5</sub>
11	P <sub>5</sub>	3	1.00	{P <sub>3</sub> }	P <sub>3</sub>
16	P <sub>3</sub>	13	2.60	{}	—

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

Può causare starvation dei processi più lunghi



## Scheduling High Response Ratio Next (HRN)

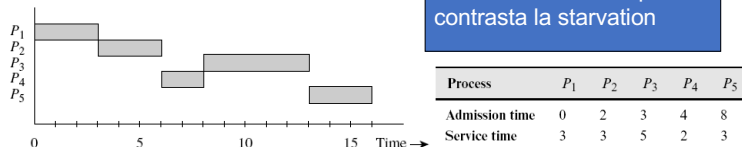
- Calcola i rapporti di risposta di tutti i processi nel sistema e seleziona il processo con il rapporto più elevato

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

Time	Completed process			Response ratios of processes					Scheduled process
	id	ta	w	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	
0	—	—	—	1.00	—	—	—	—	P <sub>1</sub>
3	P <sub>1</sub>	3	1.00	—	1.33	—	—	—	P <sub>2</sub>
6	P <sub>2</sub>	4	1.33	—	—	1.60	2.00	—	P <sub>4</sub>
8	P <sub>4</sub>	4	2.00	—	—	2.00	—	1.00	P <sub>3</sub>
13	P <sub>3</sub>	10	2.00	—	—	—	2.67	—	P <sub>5</sub>
16	P <sub>5</sub>	8	2.67	—	—	—	—	—	—

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

L'uso del tasso di risposta contrasta la starvation



## 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 schedata
- Una richiesta può essere schedata molte volte prima che sia completata
  - Overhead maggiore rispetto allo scheduling senza prelazione
- Usato nei SO multi-programmati e time-sharing

## Scheduling Round-Robin

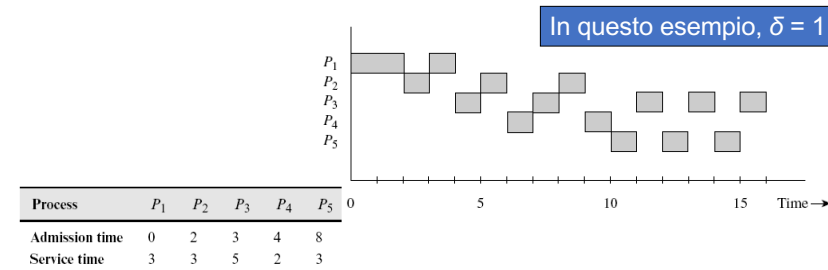
- Obiettivo: fornire buoni tempi di risposta a tutte le richieste
- Il time slice, o quanto di tempo,  $\delta$ , è la massima quantità di tempo di CPU che una richiesta schedulata può usare
- Al termine del quanto di tempo, la richiesta viene sospesa
  - Il kernel genera un interrupt allo scadere del 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

13

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

Time of scheduling	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	c	$t_d$	w
Position of $P_1$	1	1	2	1													4	4	1.33
Position of $P_2$			1	3	2	1	3	2	1								9	7	2.33
Position of $P_3$				2	1	3	2	1	4	3	2	1	2	1	2	1	16	13	2.60
Position of $P_4$					3	2	1	3	2	1							10	6	3.00
Position of $P_5$								3	2	1	2	1	2	1			15	7	2.33
Process scheduled	$P_1$	$P_1$	$P_2$	$P_1$	$P_3$	$P_2$	$P_4$	$P_3$	$P_2$	$P_4$	$P_5$	$P_3$	$P_5$	$P_3$	$P_5$	$P_3$			

$\bar{t}_d = 7.4$  seconds,  $\bar{w} = 2.32$   
c: completion time of a process



14

## Variazione del Tempo di Risposta nello Scheduling RR

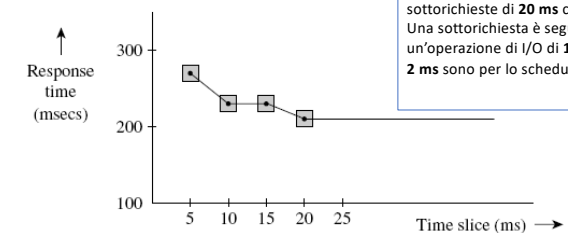
- Ricordiamo che per un processo il tempo di risposta  $rt = n \times (\delta + \sigma)$ , dove  $n$  è il numero di processi nel sistema
- In realtà la relazione tra  $rt$  e  $\delta$  è più complessa
  - Alcuni processi saranno bloccati per operazioni di I/O o per l'attesa di azioni degli utenti
    - Quindi  $rt$  è governato dal numero di processi attivi piuttosto che da  $n$
  - Se una richiesta ha bisogno di più di  $\delta$  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

15

## Esempio: 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



16

## Least Completed Next (LCN)

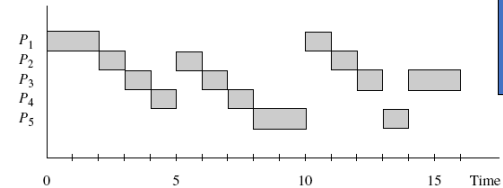
- Schedula il processo che ha consumato fino a quel momento la minor quantità di tempo di CPU
- La natura del processo (CPU o I/O bound) e il suo requisito di CPU non influenzano il suo progresso nel sistema
- Tutti i processi progrediscono in modo approssimativamente uguale in termini di tempo di CPU 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

17

## Least Completed Next (LCN)

Time of scheduling		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$c$	$t_d$	$w$
CPU time consumed by processes	$P_1$	0	1	2	2	2	2	2	2	2	2	2						11	11	3.67
	$P_2$			0	1	1	1	2	2	2	2	2	2					12	10	3.33
	$P_3$				0	1	1	1	2	2	2	2	2	2	3	4	5	16	13	2.60
	$P_4$					0	1	1	1									8	4	2.00
	$P_5$										0	1	2	2	2	2		14	6	2.00
Process scheduled		$P_1$	$P_1$	$P_2$	$P_3$	$P_4$	$P_2$	$P_3$	$P_4$	$P_5$	$P_5$	$P_1$	$P_2$	$P_3$	$P_5$	$P_3$	$P_3$			

$\bar{t}_d = 8.8$  seconds,  $\bar{w} = 2.72$   
 $c$ : 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

Problemi:

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

18

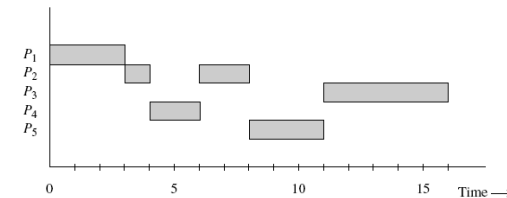
## Shortest Time to Go (STG)

- 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 scheduling		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	$c$	$t_a$	$w$
Remaining	$P_1$	3	2	1														3	3	1.00
CPU time requirement of a process	$P_2$			3	3	2	2	2	1									8	6	2.00
	$P_3$				5	5	5	5	5	5	5	5	5	4	3	2	1	16	13	2.60
	$P_4$					2	1											6	2	1.00
	$P_5$										3	2	1					11	3	1.00
Process scheduled	$P_1$	$P_1$	$P_1$	$P_2$	$P_4$	$P_4$	$P_2$	$P_5$	$P_5$	$P_5$	$P_3$	$P_3$	$P_3$	$P_3$	$P_3$	$P_3$				

$\bar{t}_d = 5.4$  seconds,  $\bar{w} = 1.52$   
 $c$ : completion time of a process



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

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

19

20