

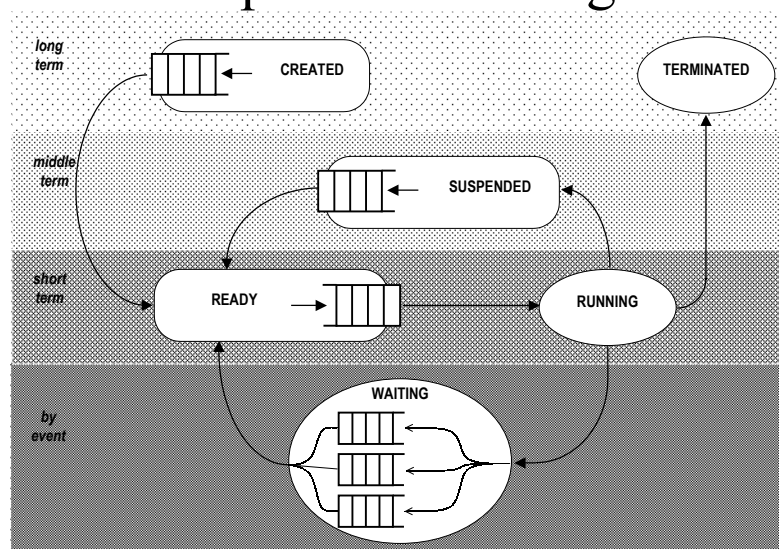
Architettura degli Elaboratori 2

Esercitazioni 1

- Scheduling della CPU

A. Memo - 2004

tipi di scheduling



scheduling

- criteri di valutazione delle politiche di scheduling
 - efficienza di utilizzo (*tempo utile/tempo di scheduling*)
 - throughput (*processi completati per unità di tempo*)
 - tempo di turn-around (*tempo di completamento*)
 - tempo di attesa
 - tempo di risposta

scheduling della CPU (1)

- consiste nel selezionare un processo dalla *ready list* e attribuirgli la CPU
- l'operazione viene effettuata dal **dispatcher** (un modulo del kernel)
- possibili algoritmi di scheduling:
 - First Come First Served [FCFS]
 - Round Robin [RR]
 - Shortest Job First [SJF]
 - con priorità esterna

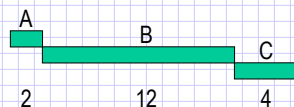
First Come First Served

- la CPU viene assegnata al processo che la richiede per primo (viene alimentata con una coda FIFO)

esempio processo A: tempo di esecuzione = 2 [u.t.]
 processo B: tempo di esecuzione = 12 [u.t.]
 processo C: tempo di esecuzione = 4 [u.t.]

N.B. = trascuriamo per semplicità i tempi di scambio di contesto

First Come First Served



TEMPO DI ATTESA

$$T_{att}(A) = 0$$

$$T_{att}(B) = 2$$

$$T_{att}(C) = 2 + 12 = 14$$

$$T_{att}(\text{medio}) = (0 + 2 + 14) / 3 = 5,3 \text{ [u.t.]}$$

TEMPO DI TURN AROUND

$$T_{ta}(A) = 2$$

$$T_{ta}(B) = 2 + 12 = 14$$

$$T_{ta}(C) = 2 + 12 + 4 = 18$$

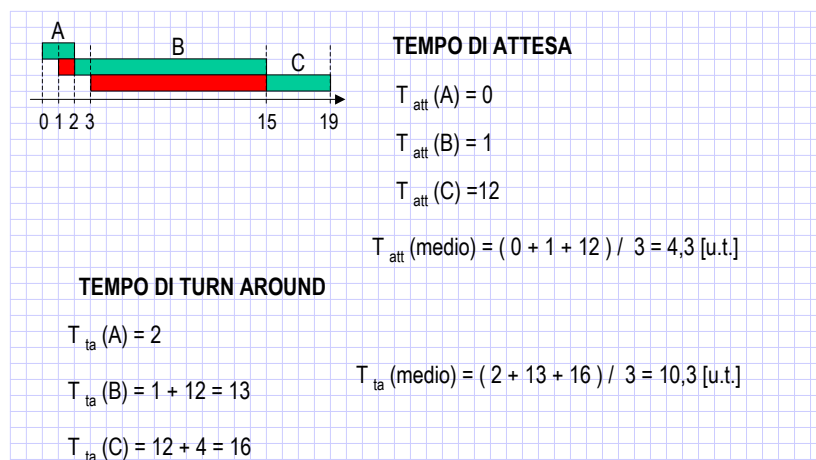
$$T_{ta}(\text{medio}) = (2 + 14 + 18) / 3 = 11,3 \text{ [u.t.]}$$

e il tempo di risposta ?

First Come First Served (2)

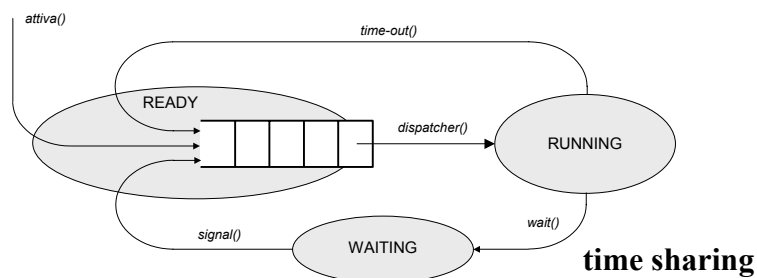
esempio 2 processo A: tempo di arrivo = 0
 tempo di esecuzione = 2 [u.t.]
 processo B: tempo di arrivo = 1
 tempo di esecuzione = 12 [u.t.]
 processo C: tempo di arrivo = 3
 tempo di esecuzione = 4 [u.t.]

First Come First Served (2)

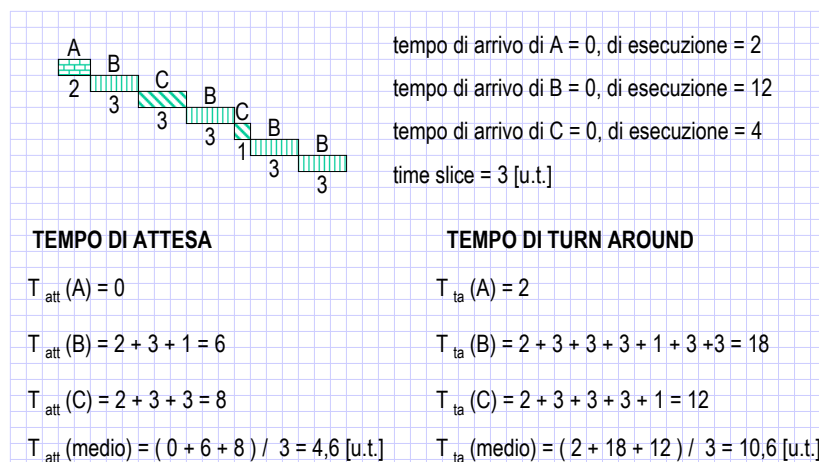


Round Robin

- come l'FCFS, ma con prelazione per esaurimento del quanto di tempo (la *ready list* viene trattata come una coda circolare)



Round Robin



Round Robin

Provate a calcolare i tempi di attesa e di turnaround medi con un valore di time slice pari a 1e 5 [u.t.]. Cambierà qualcosa?

time slice = 1 [u.t.]

$$T_{att}(\text{medio}) = (2 + 6 + 6) / 3 = 4,6 \text{ [u.t.]}$$

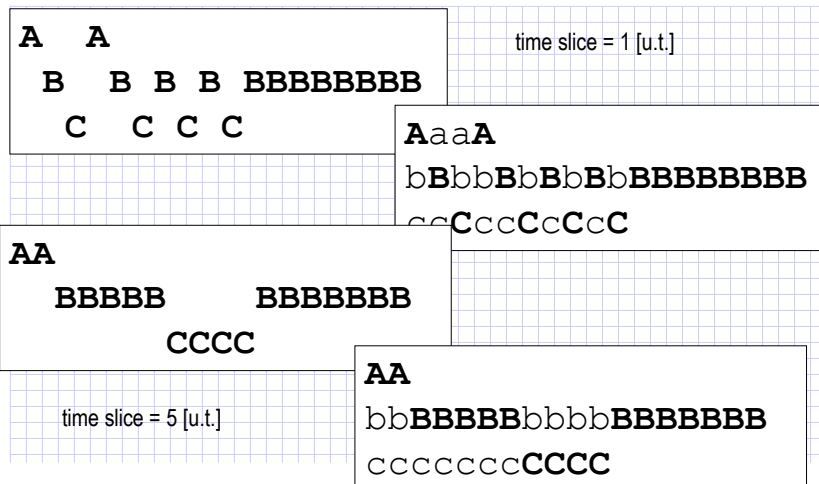
$$T_{ta}(\text{medio}) = (4 + 18 + 10) / 3 = 10,6 \text{ [u.t.]}$$

time slice = 5 [u.t.]

$$T_{att}(\text{medio}) = (0 + 2 + 7) / 3 = 3 \text{ [u.t.]}$$

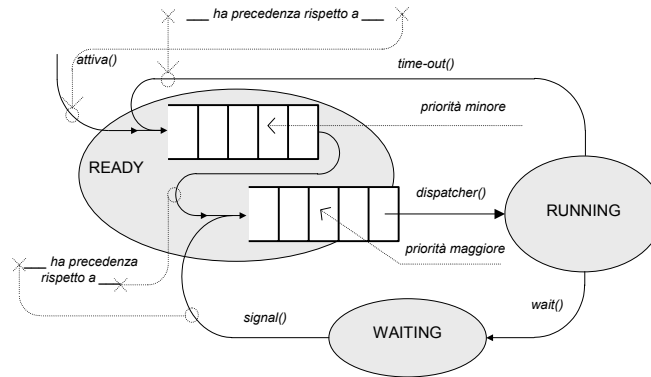
$$T_{ta}(\text{medio}) = (2 + 18 + 11) / 3 = 10,3 \text{ [u.t.]}$$

Round Robin

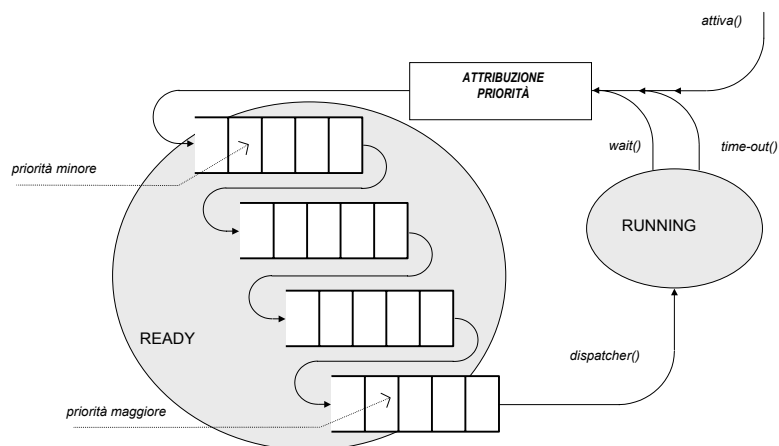


Round Robin con priorità all'I/O

- processi I/O bound e CPU bound



Round Robin con priorità multipla



Round Robin con priorità multipla

TEMPO DI ATTESA = 16,50 [u.t.]

TEMPO DI TURN AROUND = 22,33 [u.t.]

proc.	arrivo	esec.	priorità
A	0	7	2
B	0	6	4
C	0	10	3
D	0	2	3
E	0	7	1
F	0	3	5

slot time = 10,6 [u.t.]

EEEE .EEE

aaaa .aaa .AAAA .AAA

cccc .ccc .cccc .ccc .CCCC .cc .CCCC .CC

dddd .ddd .dddd .ddd .dddd .DD

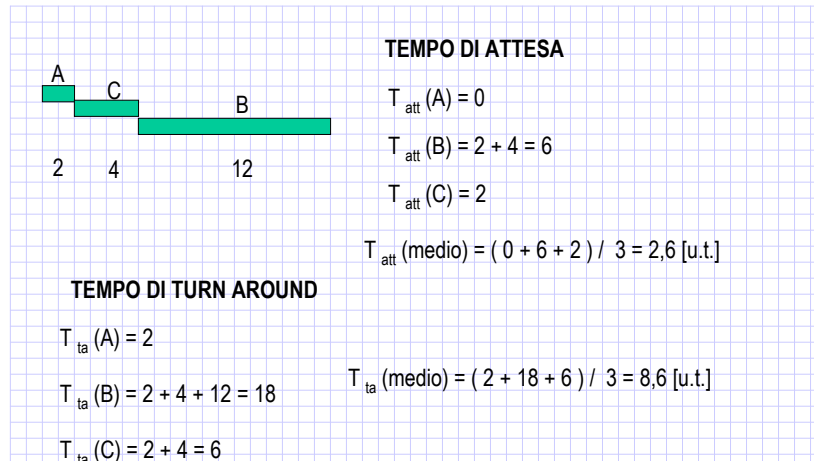
bbbb .bbb .bbbb .bbb .bbbb .bb .bbbb .bb .BBBB .BB

ffff .fff .ffff .fff .ffff .ff .ffff .ff .ffff .ff .FFF

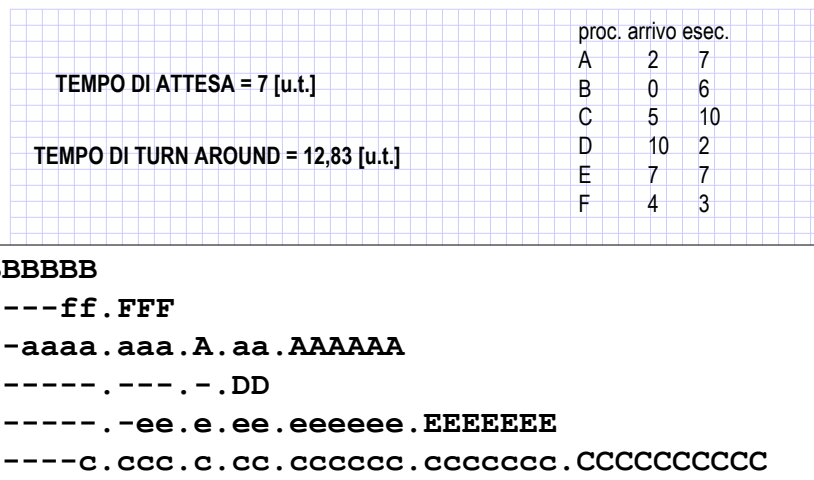
Shortest Job First

- o meglio *Shortest next-CPU-burst First*
- la CPU viene assegnata al processo che ha il *CPU-burst* successivo più breve
- può essere *non preemptive*
- o *preemptive* (SRTF)

Shortest Job First (*senza prerilascio*)



Shortest Job First (*con prerilascio*)



Shortest Job First

- non è implementabile

Esercizio con soluzioni (1)

Cinque processi batch, identificati dalle lettere da A a E, arrivano al calcolatore approssimativamente allo stesso istante. I processi hanno un tempo di esecuzione stimato di 8, 10, 2, 4 e 8 unità di tempo rispettivamente, mentre le loro priorità (determinate esternamente) sono rispettivamente 2, 4, 5, 1 e 3 (dove 5 è la priorità massima). Per ognuno dei seguenti algoritmi di schedulazione determinare il *tempo medio di turnaround* e il *tempo medio di attesa*, trascurando i tempi dovuti allo scambio di contesto.

- Round Robin (time slice = 2) in un sistema multiprogrammato
- con priorità (un processo per volta, fino al completamento)
- FCFS (un processo per volta, fino al completamento)
- SJF (un processo per volta, fino al completamento)

soluzioni

RR

$$t_{att}(\text{medio}) = 15,6 \text{ [u.t.]}$$

$$t_{ta}(\text{medio}) = 22 \text{ [u.t.]}$$

priorità

$$t_{att}(\text{medio}) = 12,4 \text{ [u.t.]}$$

$$t_{ta}(\text{medio}) = 18,8 \text{ [u.t.]}$$

FCFS

$$t_{att}(\text{medio}) = 14 \text{ [u.t.]}$$

$$t_{ta}(\text{medio}) = 20,4 \text{ [u.t.]}$$

SJF

$$t_{att}(\text{medio}) = 8,8 \text{ [u.t.]}$$

$$t_{ta}(\text{medio}) = 15,2 \text{ [u.t.]}$$

Esercizio con soluzioni (2)

Cinque processi batch, identificati dalle lettere da A a E, arrivano al calcolatore rispettivamente agli istanti di tempo 0, 2, 5, 8 e 11. I processi hanno un tempo di esecuzione stimato di 9, 1, 7, 3 e 5 unità di tempo rispettivamente, mentre le loro priorità (mantenute staticamente) sono rispettivamente 3, 2, 4, 5 e 1 (dove 5 è la priorità massima). Per ognuno dei seguenti algoritmi di schedulazione determinare il *tempo medio di risposta*, il *tempo medio di turnaround* e il *tempo medio di attesa*, trascurando i tempi dovuti allo scambio di contesto. I tempi sono espressi in unità di tempo.

- FCFS (un processo per volta, fino al completamento)
- Round Robin (time slice = 3) senza priorità
- Round Robin con time sharing (time slice = 3) e priorità ma senza prerilascio; nel caso di arrivo di un processo contemporanea ad un uscita per time_out(), si dia precedenza al processo prerilasciato per time_out()
- SJF senza prerilascio (un processo per volta, fino al completamento)
- SJF con prerilascio

FCFS

(un processo per volta, fino al completamento)

	proc.	arrivo	esec.	priorità
TEMPO DI RISPOSTA = 5,0 [u.t.]	A	0	9	2
TEMPO DI ATTESA = 5,0 [u.t.]	B	2	1	4
TEMPO DI TURN AROUND = 11,0 [u.t.]	C	5	7	3
	D	8	3	3
	E	11	5	1

```

AAAAAAAA
--bbbbbbbB
-----ccccCCCCCCC
-----dDDDDDDDDDDD
-----eEEEEEEEEEEEE
  
```

Round Robin con time sharing, time slice = 3

```

AAAAAAAA
--bbbbbbbB
-----ccccCCCCCCC
-----dDDDDDDDDDDD
-----eEEEEEEEEEEEE
  
```

TEMPO DI RISPOSTA = 3,2 [u.t.]
 TEMPO DI ATTESA = 6,0 [u.t.]
 TEMPO DI TURN AROUND = 11,0 [u.t.]

Round Robin con time sharing e priorità, senza prerilascio

TEMPO DI RISPOSTA = 5,6 [u.t.]
 TEMPO DI ATTESA = 8,2 [u.t.]
 TEMPO DI TURN AROUND = 13,2 [u.t.]

```

AAAAAAaAAAAAAAAAAAAA
--bbbbbbbbbbbbbbbbbbB
-----cCCCcccCCCC
-----dDDD
-----eEEEEEEEEEEEE
  
```

Round Robin con time sharing e priorità, con prerilascio

```

AAAAAaaaaaaaaaaaaAAAA
--bbbbbbbbbbbbbbbbbbB
-----CCCcccCCCC
-----DDD
-----eeeeeeeeeeEEEEEE
    
```

TEMPO DI RISPOSTA = 5,2 [u.t.]

TEMPO DI ATTESA = 7,8 [u.t.]

TEMPO DI TURN AROUND = 12,8 [u.t.]

SJF senza time sharing e senza prerilascio

TEMPO DI RISPOSTA = 4,8 [u.t.]

TEMPO DI ATTESA = 4,8 [u.t.]

TEMPO DI TURN AROUND = 9,8 [u.t.]

```

AAAAAAAAA
--bbbbbbbB
-----ccccccccccccCCCCCCC
-----ddDDD
-----eeEEEEEE
    
```

SJF senza time sharing e con prerilascio

TEMPO DI RISPOSTA = 3,4 [u.t.]

TEMPO DI ATTESA = 3,6 [u.t.]

TEMPO DI TURN AROUND = 8,6 [u.t.]

```

AAaAAAAAAA
--B
-----ccccccccccccCCCCCCC
-----ddDDD
-----eeEEEEEE
    
```

Esercizio

Si supponga che tre clienti arrivino ad una stazione di servizio per fare il pieno di benzina, e che ognuna impieghi il seguente tempo (noto a priori)

auto	arrivo	servizio (in minuti)
A	8:00	8
B	8:06	5
C	8:07	2

Nell'ipotesi che alle 8:00 l'unica pompa di benzina della stazione sia libera, calcolare il tempo medio di attesa ed il tempo medio di turnaround nel caso di politiche di schedulazione FIFO, SJF non preemptive ed SJF preemptive.

Esercizio

Cinque processi batch, identificati dalle lettere da A a E, arrivano al calcolatore approssimativamente allo stesso istante. I processi hanno un tempo di esecuzione stimato di 10, 6, 2, 4 e 8 unità di tempo rispettivamente, mentre le loro priorità (determinate esternamente) sono rispettivamente 3, 5, 2, 1 e 4 (dove 5 è la priorità massima). Per ognuno dei seguenti algoritmi di schedulazione determinare il *tempo medio di turnaround* e il *tempo medio di attesa*, trascurando i tempi dovuti allo scambio di contesto.

- Round Robin (*time slice* = 2) in un sistema multiprogrammato
- con priorità esterna (un processo per volta, fino al completamento)
- FCFS (un processo per volta, fino al completamento)
- SJF (un processo per volta, fino al completamento)