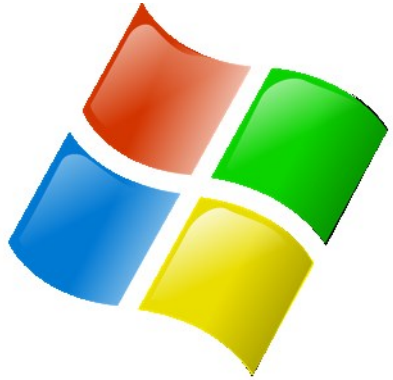
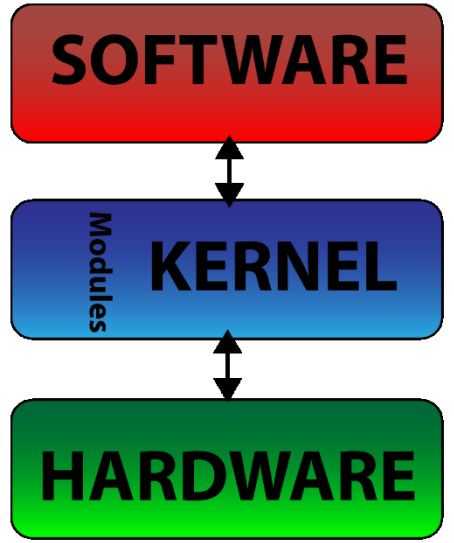
UNIVERSITÀ DEGLI STUDI DELLA BASILICATA

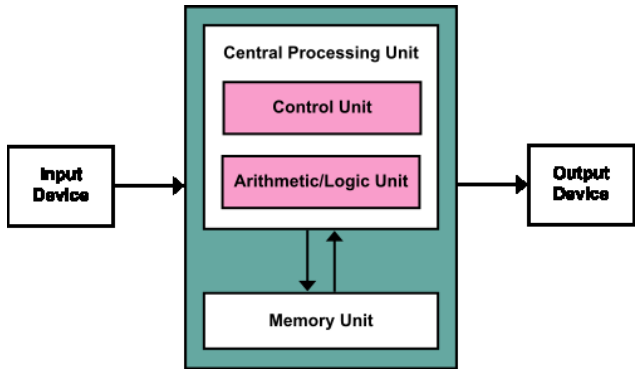


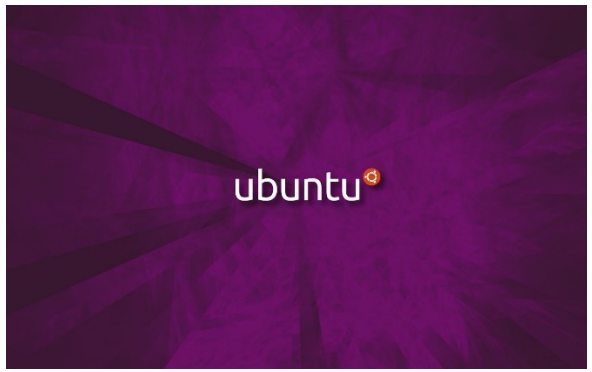
Corso di Sistemi Operativi

Esercitazione
Scheduling della CPU

Docente:
Domenico Daniele Bloisi



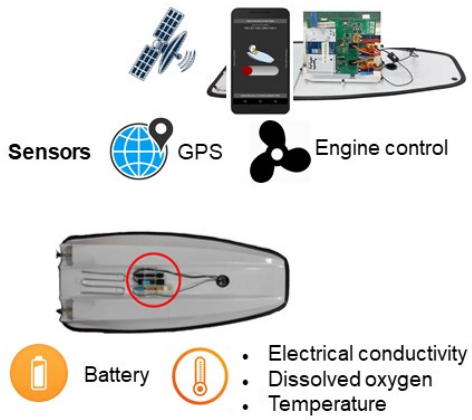


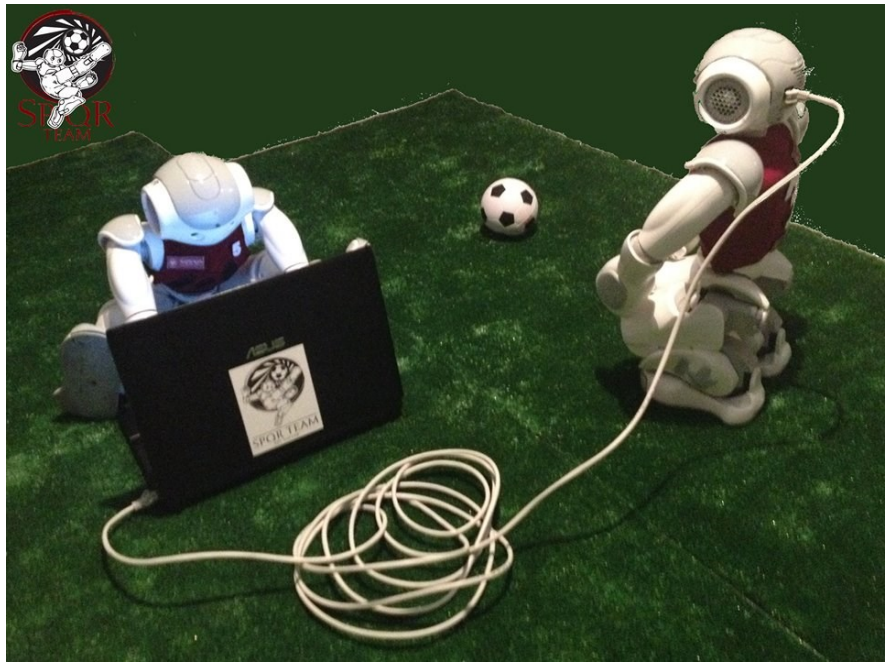


Domenico Daniele Bloisi

- Ricercatore RTD B
  Dipartimento di Matematica, Informatica ed Economia
  Università degli studi della Basilicata
  http://web.unibas.it/bloisi

- SPQR Robot Soccer Team
  Dipartimento di Informatica, Automatica e Gestionale Università degli studi di Roma “La Sapienza”
  http://spqr.diag.uniroma1.it





Informazioni sul corso

- Home page del corso:
  http://web.unibas.it/bloisi/corsi/sistemi-operativi.html
- Docente: Domenico Daniele Bloisi
- Periodo: I semestre ottobre 2020 – febbraio 2021

  - Lunedì 15:00-17:00
  - Martedì 9:30-11:30

Le lezioni saranno erogate in modalità esclusivamente on-line

Codice corso Google Classroom:
https://classroom.google.com/c/MTQ2ODE2NTk3ODIz?cjc=67646ik

Ricevimento

- Su appuntamento tramite Google Meet

Per prenotare un appuntamento inviare una email a domenico.bloisi@unibas.it

Credits

Alcuni esercizi derivano dai contenuti del corso

“Sistemi Operativi”
del Prof. Giorgio Grisetti
https://sites.google.com/diag.uniroma1.it/sistemi-operativi-1819

Scheduling

- Lo scheduling è una funzione fondamentale dei sistemi operativi.
- Si sottopongono a scheduling quasi tutte le risorse di un calcolatore.
- La CPU è una delle risorse principali e il suo scheduling è alla base della progettazione dei sistemi operativi.

Algoritmi di scheduling

Esistono differenti **algoritmi di scheduling della CPU**

- Scheduling in ordine d'arrivo (first-come, first-served, FCFS)
- Scheduling per brevità (shortest-job-first, SJF)
- Scheduling circolare (round-robin, RR)
- Scheduling con priorità
- Scheduling a code multilivello
- Scheduling a code multilivello con retroazione

Esercizio 1

Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Durata della sequenza
P_1	24
P_2	3
P_3	3

- Assumendo di usare un algoritmo di scheduling FCFS, qual è il tempo di attesa medio nel caso i processi arrivino nell'ordine P_1 , P_2 , P_3 ?
- Qual è, invece, il tempo di attesa medio nel caso in cui i processi arrivino nell'ordine P_2 , P_3 , P_1 ?

FCFS

Il più semplice algoritmo di scheduling della CPU è l'algoritmo di **scheduling in ordine d'arrivo (*scheduling first-come, first-served o FCFS)***

- La CPU si assegna al processo che la richiede per primo.
- Senza prelazione
- Tempo medio di attesa spesso abbastanza lungo
- Effetto convoglio

Soluzione Esercizio 1

- Il tempo di attesa è la somma degli intervalli di attesa passati nella ready queue

- Il tempo di attesa medio è dato dalla somma dei tempi di attesa per processo diviso per il numero di processi considerati

Soluzione Esercizio 1

Se i processi arrivano nell'ordine P_1 , P_2 , P_3 e sono serviti con un algoritmo di scheduling FCFS si ottiene la seguente situazione

Tempo di attesa per P_1 = 0
Tempo di attesa per P_2 = 24
Tempo di attesa per P_3 = 27

Tempo di attesa medio = (0 + 24 + 27) / 3 = 17

Soluzione Esercizio 1

Se i processi arrivano nell'ordine P_2 , P_3 , P_1 e sono serviti con un algoritmo di scheduling FCFS si ottiene la seguente situazione:

Tempo di attesa per P_1 = 6
Tempo di attesa per P_2 = 0
Tempo di attesa per P_3 = 3

Tempo di attesa medio = (6 + 0 + 3) / 3 = 3

Soluzione Esercizio 1

Dai risultati ottenuti si evince che il tempo medio di attesa in condizioni FCFS:

- non è in generale minimo
- può variare molto all'aumentare della variabilità dei CPU burst dei processi da schedulare

Esercizio 2

Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Durata della sequenza
P_1	6
P_2	8
P_3	7
P_4	3

Ipotizzando

1. che tutti i processi siano in ready queue
2. l'uso di un algoritmo di scheduling SJF

- Quale sarà l'ordine di esecuzione dei processi in tabella?
- Quale sarà il tempo di attesa medio generato?

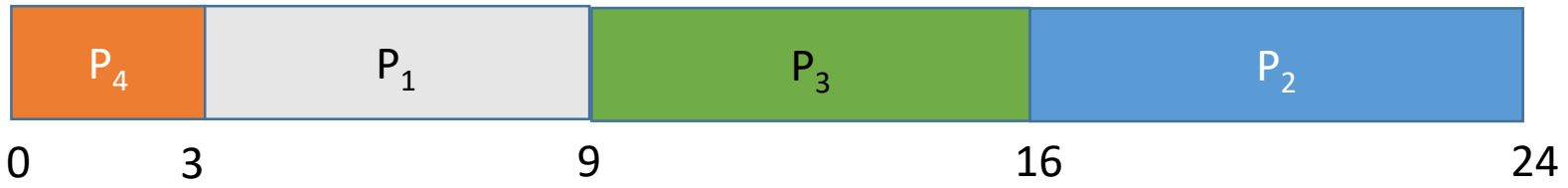
SJF

L'**algoritmo di scheduling per brevità** (*shortest-job-first,* ***SJF***) assegna la CPU al processo che ha la più breve lunghezza della successiva sequenza di operazioni della CPU.

L'algoritmo SJF è ottimale rispetto al tempo di attesa medio per un dato insieme di processi.

Soluzione Esercizio 2

Con uno scheduling SJF, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Tempo di attesa per P_1 = 3
Tempo di attesa per P_2 = 16
Tempo di attesa per P_3 = 9
Tempo di attesa per P_4 = 0

Tempo di attesa medio = (3 + 16 + 9 + 0) / 4 = 7

Esercizio 3

Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Istante di arrivo	Durata della sequenza
P_1	0	8
P_2	1	4
P_3	2	9
P_4	3	5

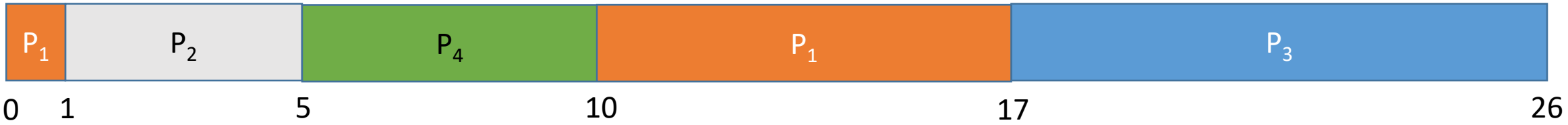
Ipotizzando che i processi

1. arrivino nella ready queue nei momenti mostrati in tabella
2. richiedano i tempi di CPU indicati in tabella

- Quale sarà il tempo di attesa medio generato dall'algoritmo di scheduling shortest remaining time first (SJF con prelazione) ?

Soluzione Esercizio 3

Con uno scheduling shortest remaining time first i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Istante di arrivo
0
1
2
3

Tempo di attesa per P_1 = 10 - 1
Tempo di attesa per P_2 = 1 - 1
Tempo di attesa per P_3 = 17 - 2
Tempo di attesa per P_4 = 5 - 3

Tempo di attesa medio = (9 + 0 + 15 + 2) / 4 = 6,5

Esercizio 4

Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Durata della sequenza
P_1	24
P_2	3
P_3	3

Ipotizzando

1. che tutti i processi siano in ready queue
2. di settare il quanto di tempo a disposizione di ogni processo in esecuzione al valore 4

- Quale sarà il tempo di attesa medio generato dall'algoritmo di scheduling circolare (round robin) se i processi vengano schedulati nell'ordine P_1 , P_2 , P_3 ?

Round Robin (RR)

L'algoritmo di scheduling circolare (*round-robin, RR*) è simile allo scheduling FCFS (in ordine di arrivo), ma aggiunge la capacità di prelazione in modo che il sistema possa commutare fra i vari processi.

Un **quanto di tempo** o **porzione di tempo (*time slice*)** è la quantità – fissata – di tempo CPU che riceverà ogni processo

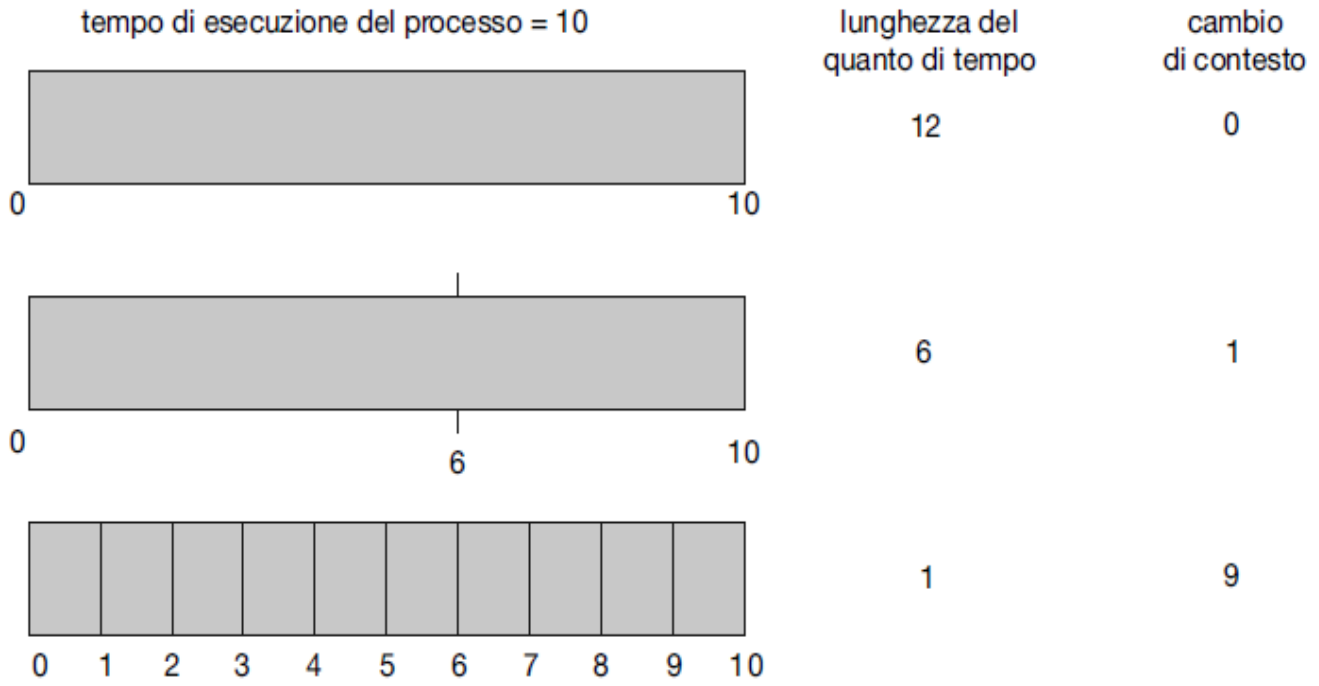
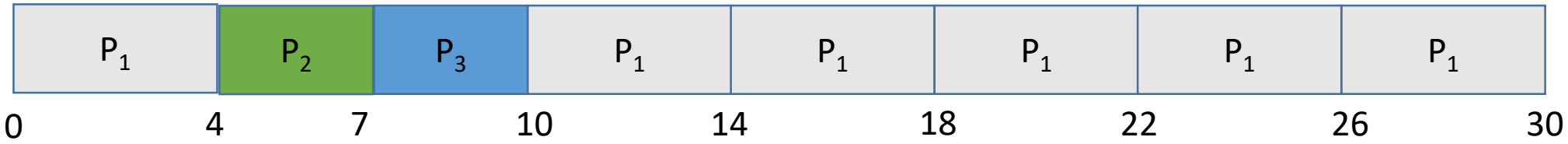


Figura 5.5 Aumento del numero dei cambi di contesto al diminuire del quanto di tempo.

Soluzione Esercizio 4

Se i processi arrivano nell'ordine P_1 , P_2 , P_3 e sono serviti con un algoritmo di scheduling RR si ottiene la seguente situazione



Tempo di attesa per P_1 = 10 - 4
Tempo di attesa per P_2 = 4
Tempo di attesa per P_3 = 7

Tempo di attesa medio = (6 + 4 + 7) / 3 = 5,66

Esercizio 5

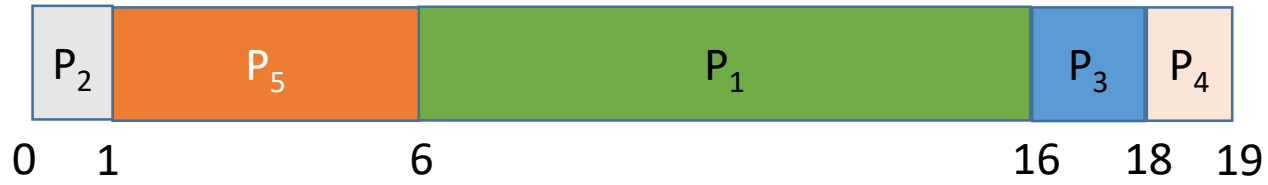
Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Durata della sequenza	Priorità
P_1	10	3
P_2	1	1
P_3	2	4
P_4	1	5
P_5	5	2

Ipotizzando che tutti i processi siano in ready queue, qual è il tempo di attesa medio generato utilizzando l'algoritmo di scheduling con priorità?

Soluzione Esercizio 5

Con uno scheduling con priorità, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Tempo di attesa per P_1 = (6 – 0) = 6
Tempo di attesa per P_2 = (0 – 0) = 0
Tempo di attesa per P_3 = (16 – 0) = 16
Tempo di attesa per P_4 = (18 – 0) = 18
Tempo di attesa per P_5 = (1 – 0) = 1

Tempo di attesa medio = (6 + 0 + 16 + 18 + 1) / 5 = 8,2

Esercizio 6

Sia data la seguente tabella che descrive il comportamento di un insieme di processi

Processo	Durata della sequenza	Priorità
P_1	4	3
P_2	5	2
P_3	8	2
P_4	7	1
P_5	3	3

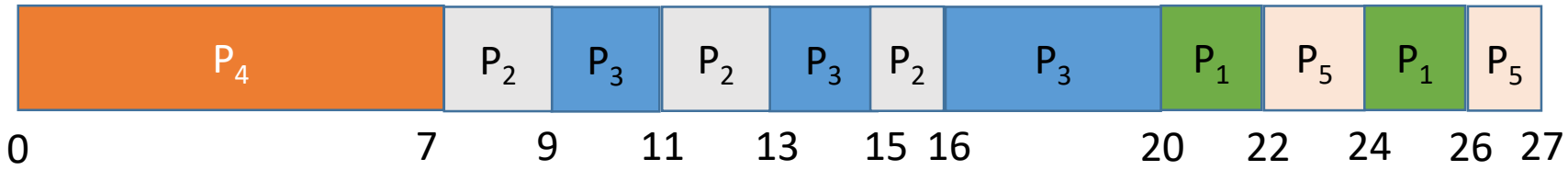
Ipotizzando

1. che tutti i processi siano in ready queue
2. di settare il quanto di tempo a disposizione di ogni processo in esecuzione al valore 2

- Quale sarà il tempo di attesa medio generato utilizzando l'algoritmo di scheduling con priorità insieme all'algoritmo di scheduling circolare (round robin) nel caso di processi a pari priorità?

Soluzione Esercizio 6

Con uno scheduling con priorità insieme all'algoritmo di scheduling circolare (round robin) nel caso di processi a pari priorità, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Tempo di attesa per P_1 = 20 + (24 – 22) = 22
Tempo di attesa per P_2 = 7 + (11 – 9) + (15 – 13) = 11
Tempo di attesa per P_3 = 9 + (13 – 11) + (16 – 15) = 12
Tempo di attesa per P_4 = 0
Tempo di attesa per P_5 = 22 + (26 – 24) = 24

Tempo di attesa medio = (22 + 11 + 12 + 0 + 24) / 5 = 13,8

Esercizio 7

Sia data la seguente tabella che descrive il comportamento di un insieme di processi in un sistema hard real-time

Processo	Durata del periodo	Tempo di esecuzione
P_1	50	25
P_2	80	35

Ipotizzando

1. di utilizzare l'algoritmo di scheduling EDF (earliest-deadline-first)
2. che la scadenza di ogni processo imponga di terminare l'esecuzione entro il periodo seguente

- Quale sarebbe il diagramma di scheduling per P_1 e P_2?

Scheduling EDF

Lo **scheduling EDF** *(earliest-deadline-first, ossia "per prima la scadenza più ravvicinata")*, attribuisce le priorità dinamicamente, sulla base delle scadenze.

Più vicina è la scadenza, maggiore è la priorità

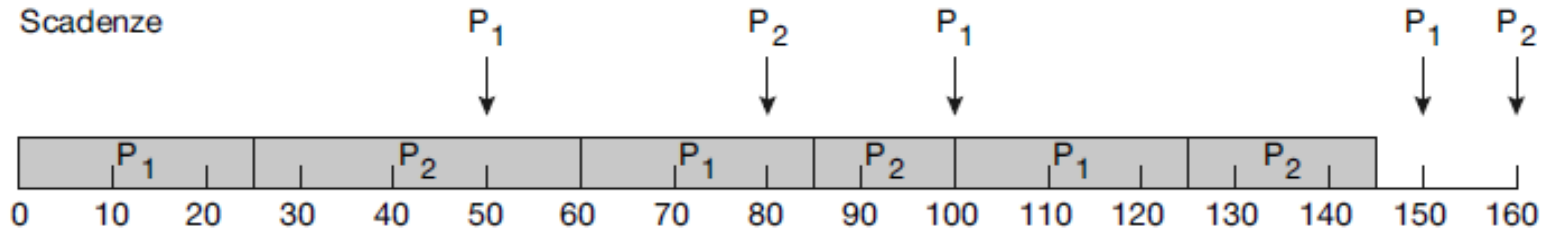


Figura 5.24 Scheduling EDF.

A differenza dell'algoritmo con priorità proporzionale alla frequenza, lo scheduling EDF non postula la periodicità dei processi, e non prevede neanche di impiegare sempre la stesso tempo della CPU per ogni burst.

Soluzione Esercizio 7

Poiché i processi devono essere eseguiti in real-time, per prima cosa dobbiamo verificare se sia possibile rispettare le deadline con una sola CPU. Calcoliamo, quindi, la percentuale di utilizzo della CPU come

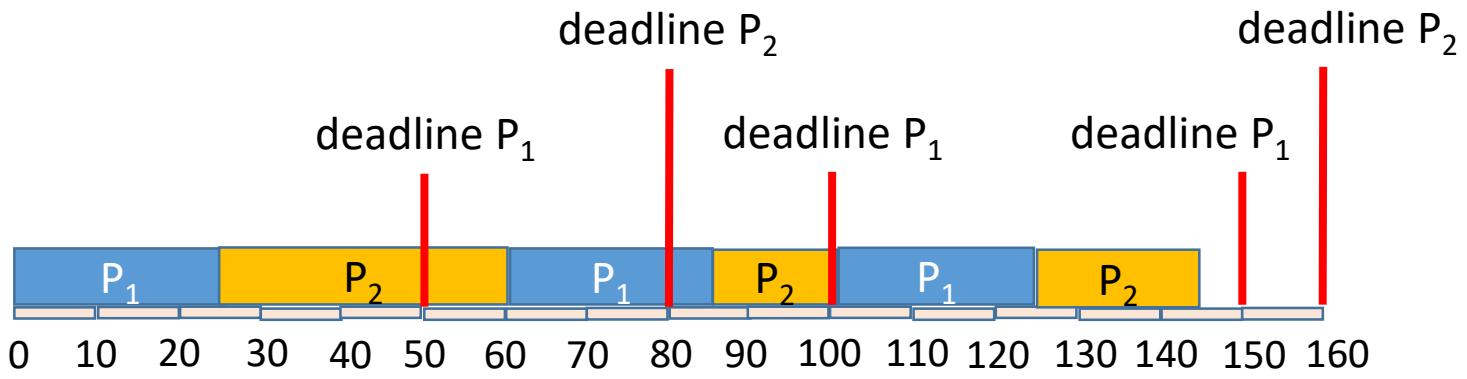
$$U_{CPU} = \sum_{p} \frac{t_p}{d_p} = \frac{25}{50} + \frac{35}{80} = 0{,}94$$

Processo	Durata del periodo	Tempo di esecuzione
P_1	50	25
P_2	80	35

Poiché $U_{CPU} \leq 1$ è possibile effettuare lo scheduling rispettando tutte le deadline.

Soluzione Esercizio 7

Con uno scheduling EDF, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Esercizio 8

Sia data la seguente tabella che descrive le caratteristiche di un insieme di processi periodici real-time.

Processo	Tempo di inizio	Deadline	Periodo	Tempo di esecuzione
P_1	0	4	6	1
P_2	0	8	12	4
P_3	0	12	14	3

Descrivere il comportamento di uno scheduler preemptive Earliest Deadline First (EDF) per l'insieme di processi in tabella, assumendo che:

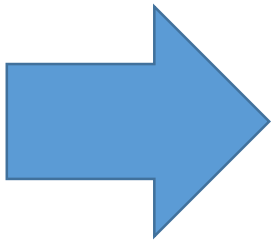
1. nessuno dei processi debba attendere il rilascio di una risorsa posseduta da un altro processo;
2. i processi in entrata alla CPU dichiarino il tempo di esecuzione necessario al proprio completamento;
3. l'operazione di avvio di un processo lo porti nella coda di ready, ma non necessariamente in esecuzione.

Soluzione Esercizio 8

Poiché i processi devono essere eseguiti in real-time, per prima cosa dobbiamo verificare se sia possibile rispettare le deadline con una sola CPU. Calcoliamo quindi la percentuale di utilizzo della CPU come

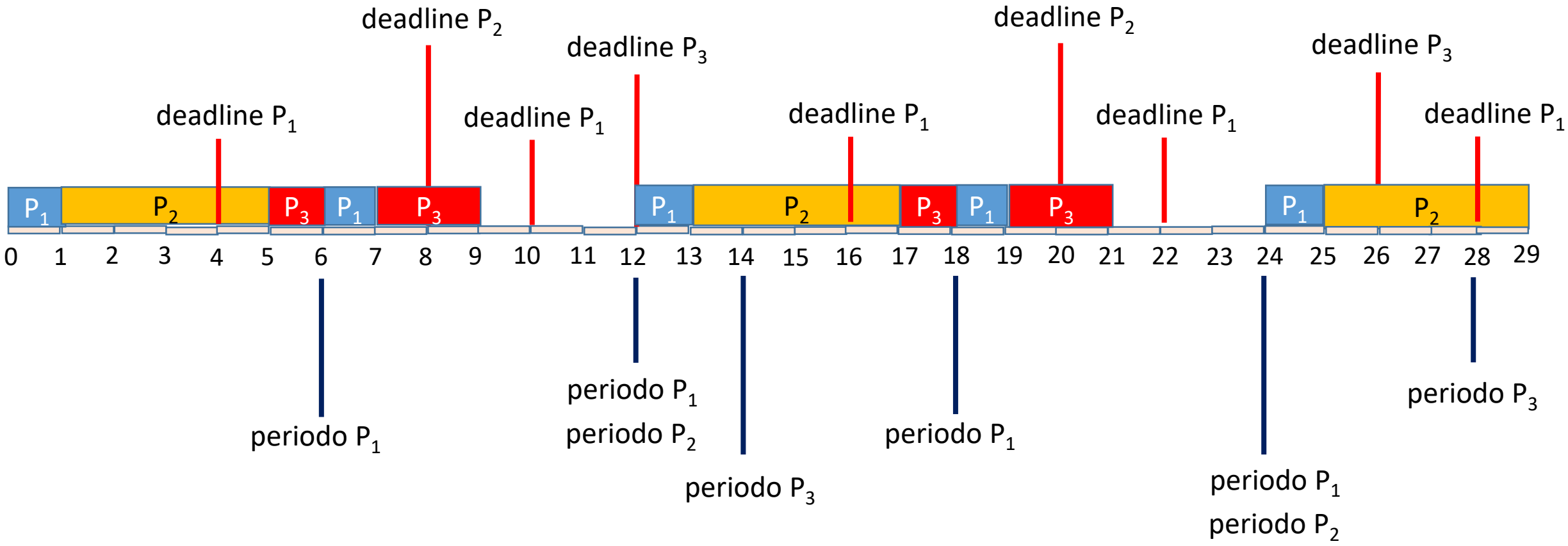
$$\mathrm{U}_{CPU} = \sum_{p} \frac{t_p}{d_p} = \frac{1}{4} + \frac{4}{8} + \frac{3}{12} = 1$$

Poiché $U_{CPU} \leq 1$ è possibile effettuare lo scheduling dei processi rispettando tutte le deadline.



Soluzione Esercizio 8

Con uno scheduling EDF, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto



Esercizio 9

Sia data la seguente tabella che descrive le caratteristiche di un insieme di processi periodici real-time.

Processo	Tempo di inizio	Periodo	CPU burst
P_1	0	3	1
P_2	0	5	2
P_3	0	6	1

Descrivere il comportamento di uno scheduler preemptive Earliest Deadline First (EDF) per l'insieme di processi in tabella, assumendo che:

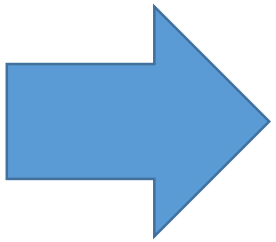
- la deadline di ogni processo coincida con il suo periodo;
- nessuno dei processi debba attendere il rilascio di una risorsa posseduta da un altro processo;
- i processi in entrata alla CPU dichiarino il tempo di esecuzione necessario al proprio completamento;
- l'operazione di avvio di un processo lo porti nella coda di ready, ma non necessariamente in esecuzione.

Soluzione Esercizio 9

Dato il requisito di esecuzione dei processi in real-time, per prima cosa bisogna verificare che lo scheduler in questione possa garantire l'esecuzione di ogni ciclo di CPU burst entro la deadline specificata. Ricordando che in questo caso la deadline coincide con il periodo, calcoliamo quindi la percentuale di utilizzo della CPU come

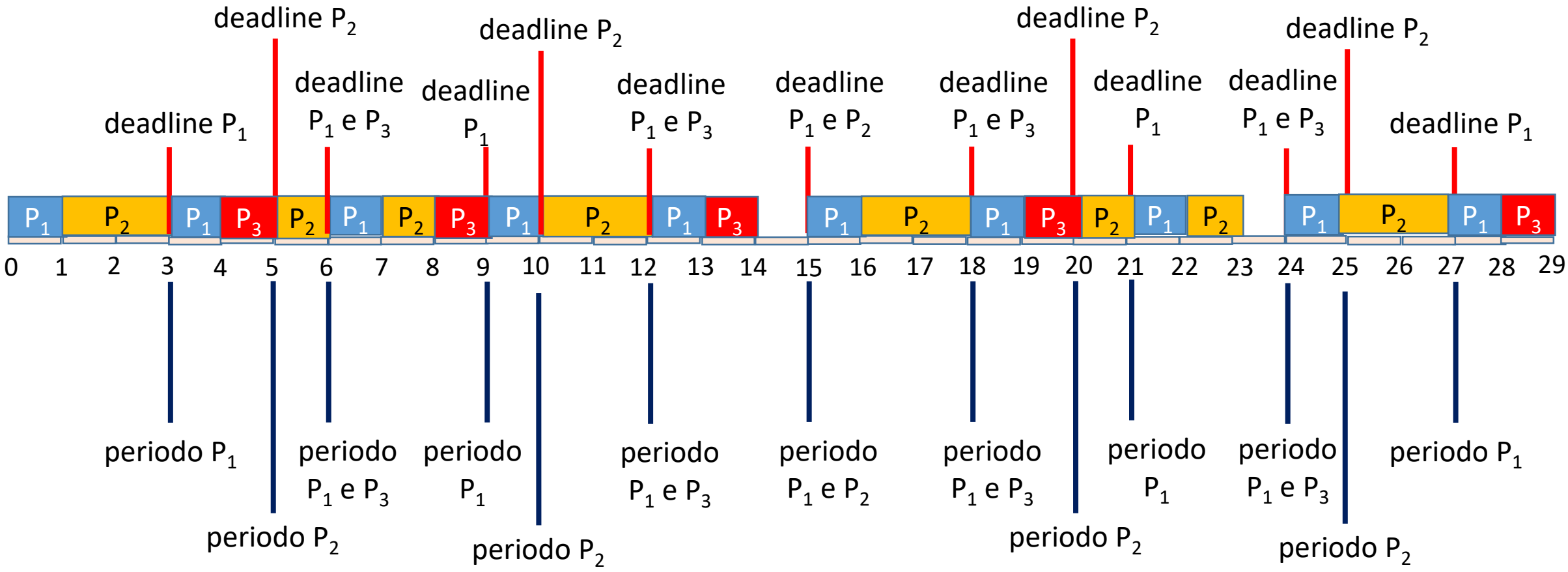
$$U_{CPU} = \sum_{p} \frac{t_p}{d_p} = \frac{1}{3} + \frac{2}{5} + \frac{1}{6} = 0.9$$

Poiché $U_{CPU} \leq 1$ sarà possibile effettuare lo scheduling tramite EDF rispettando il vincolo di esecuzione real-time.



Soluzione Esercizio 9

Con uno scheduling EDF, i processi in tabella si ordinerebbero secondo il diagramma riportato sotto. Quando due processi si trovano nelle stesse condizioni, si è scelto di privilegiare il processo con PID minore.



Esercizio 10

Sia data la seguente tabella che descrive le caratteristiche di un insieme di processi.

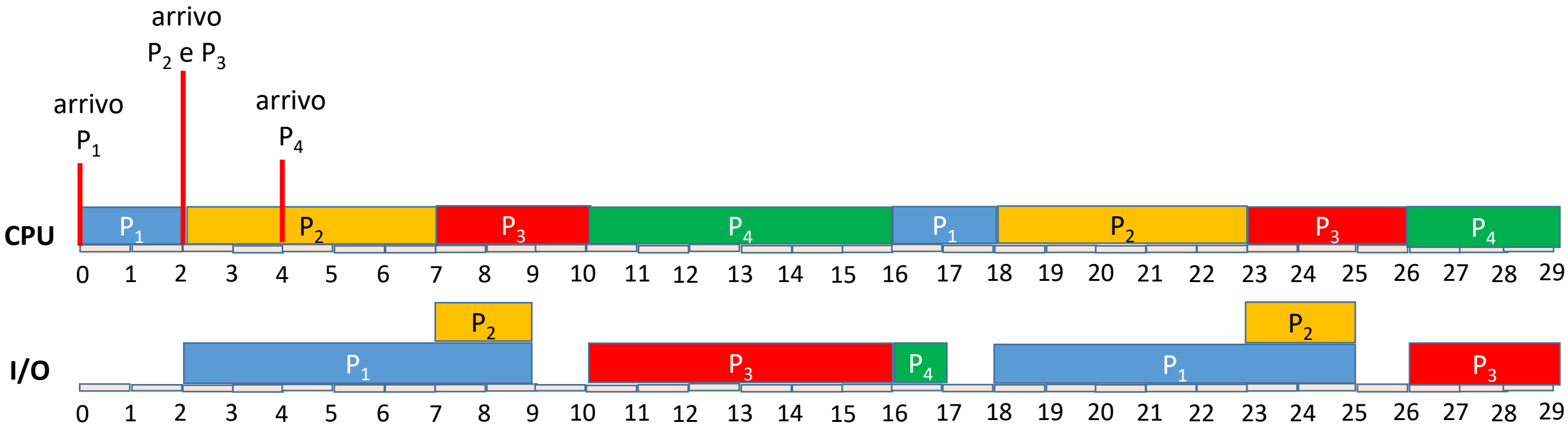
Processo	Tempo di inizio	CPU burst	I/O burst
P_1	0	2	7
P_2	2	5	2
P_3	2	3	6
P_4	4	6	1

Descrivere il comportamento di uno scheduler preemptive Round Robin (RR) con quanto di tempo pari a q = 10 applicato ai processi in tabella. Si assuma inoltre che:

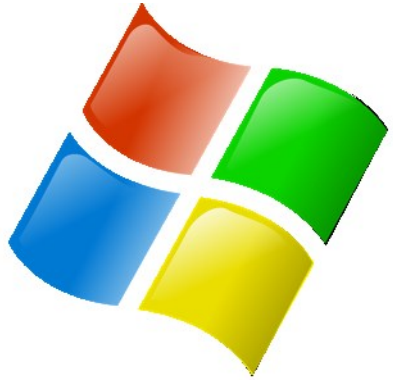
1. i processi in entrata alla CPU dichiarino il numero di burst necessari al proprio completamento;
2. l'operazione di avvio di un processo lo porti nella coda di ready, ma non necessariamente in esecuzione.
3. il termine di un I/O porti il processo che termina nella coda di ready, ma non necessariamente in esecuzione.
4. i processi si ripresentino con le stesse specifiche una volta completato l'I/O.

Soluzione Esercizio 10

In base alle specifiche e supponendo che quando arrivino due processi in contemporanea venga scelto quello con PID minore, la traccia di esecuzione sarà quella riportata in basso. Si noti che nessuno dei processi supera il quanto di tempo, quindi lo scheduler si comporterà come un First Come, First Served (FCFS).



UNIVERSITÀ DEGLI STUDI DELLA BASILICATA



Corso di Sistemi Operativi

Esercitazione
Scheduling della CPU

Docente:
Domenico Daniele Bloisi