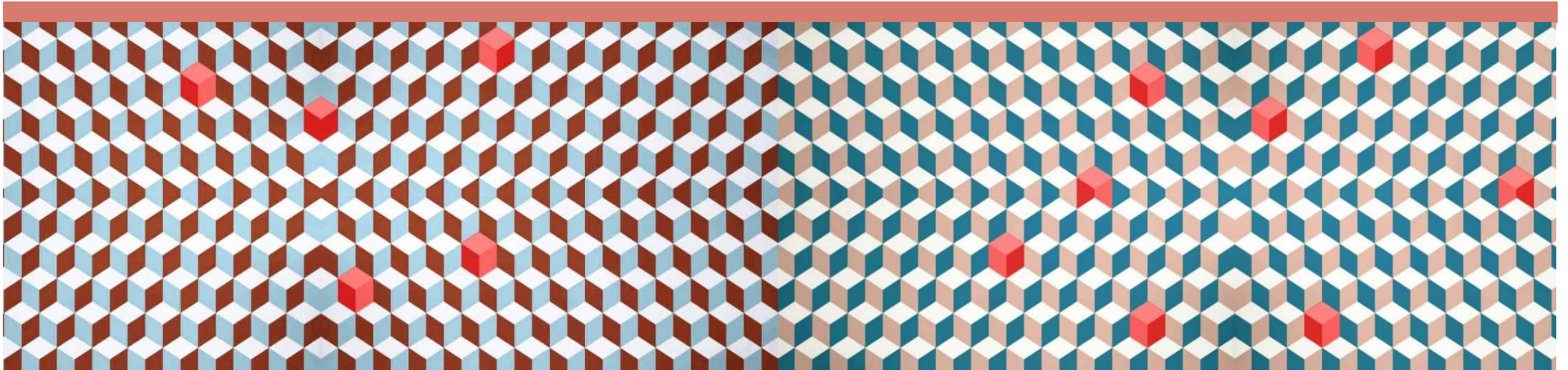


ESERCITAZIONE

Scheduling, Sincronizzazione, Deadlock

Tutor: Giovanni Hauber



L Banchiere

- **Algoritmo del banchiere:** previene la formazione dei **deadlock** assicurando che il sistema si trovi in uno **stato tale da garantire il completamento di un processo**, e che quindi le risorse che un processo richiede siano disponibili (**stato sicuro**).
 - **Risorse_Allocate_{J,K}**: Quantità di risorse (unità) di una classe R_K **allocate** ad un processo P_J
 - **Risorse_Richieste_{J,K}**: Quantità di risorse (unità) di una classe R_K **richieste** dal processo P_J
 - **Risorse_Massime_{J,K}**: Quantità di risorse (unità) di una classe R_K richieste **massime** dal proc. P_J
 - **Risorse_Disponibili_K**: Quantità di risorse (unità) di una classe R_K **attualmente nel S.O.**
 - **Risorse_Totali_Allocate**: Quantità di risorse (unità) totali allocate a tutti i processi nel sistema:
ovvero, $\sum Risorse_Allocate_{JK}$
- **OBIETTIVO:**
Verificare un plausibile stato sicuro «proiettando» il sistema in uno stato di allocazione futuro, verificando la sequenza di risoluzione ottima dell'esecuzione dei processi.

L Banchiere: Risoluzione

1. Definire quali processi si trovano in uno stato sicuro: Per ogni processo, usiamo:

$$\text{Risorse_Massime}_i - \text{Risorse_allocate}_i \leq \text{Risorse_Disponibili}_i$$
 - Se tutti i processi rispettano ciò, allora il sistema è in uno stato sicuro. Si sceglie quindi il processo che prende più risorse a parità di processi ammissibili.
2. Una volta trovato il processo, si soddisfa la sua richiesta e una volta terminato, le risorse da esso possedute vengono rilasciate e sommate alle risorse disponibili:

$$\text{Risorse_Disponibili}_i = \text{Risorse_Allocate}_i + \text{Risorse_Disponibili}_i$$
3. Torna allo step 1 e ripeti fin quando tutti i processi non sono finiti.

NOTE:

- Può capitare che un processo abbia un numero di risorse allocate incognite. In questo caso, non sappiamo con certezza se il processo sia **safe** o **non safe**; se le altre risorse ad esso allocate risultano rispettare lo step 1, allora lo consideriamo come **plausibile safe**.
- Quando questo termina e verrà rilasciato inoltre, si risolverà un semplice sistema di disequazioni lineari al fine di stabilire un range di risorse rilasciate da quel processo per quella specifica classe, risolvendo per l'incognita. In forma:

$$\begin{cases} \text{Ris_massime}_i - \text{Ris_allocate}_i \geq 0 \\ \text{Ris_massime}_i - \text{Ris_allocate}_i \leq \text{Ris_disponibili}_i \end{cases}$$

L Banchiere

	A	B	C	D
P0	4	$x-1$	3	2
P1	8	0	$y-2$	2
P2	4	0	0	0
P3	0	0	3	2
P4	2	1	$z+1$	4

Risorse_Allocate_{J,K}

	A	B	C	D
P0	6	4	5	6
P1	10	7	6	8
P2	6	2	0	8
P3	0	3	4	2
P4	9	1	6	9

Risorse_Massime_{J,K}

A	B	C	D
2	2	10	4

Risorse_Disponibili

Determinare:

1. gli intervalli dei valori X, Y, Z per i quali il sistema si trova in uno stato sicuro
2. la sequenza dei processi
3. Se la richiesta $P2[2,0,0,2]$ può essere soddisfatta.

Banchiere

A	B	C	D
2	2	10	4

Risorse_Disponibili

$$\text{Risorse_Massime}_i - \text{Risorse_allocate}_i \leq \text{Risorse_Disponibili}_i$$

- P0: $6-4 \leq 2$? SI | $4-(x-1) \leq 2$? **NON SO** | $5-3 \leq 10$? SI | $6-2 \leq 4$? SI -> **SAFE (Plausibilmente)**
- P1: $10-8 \leq 2$? SI | $7-0 \leq 2$? **NO!** Mi fermo, processo non sicuro
- P2: $6-4 \leq 2$? SI | $2-0 \leq 2$? SI | $0-0 \leq 10$? SI | $8-0 \leq 4$? **NO!** . . .
- P3: $0-0 \leq 2$? SI | $3-0 \leq 2$? **NO!** . . .
- P4: $9-2 \leq 2$? **NO!** . . .

Quindi di tutti questi scelgo P0! Viene eseguito, termina e rilascia le sue risorse: Essendoci incognita x dobbiamo calcolare il range di risorse rilasciate per la i-sima risorsa dove l'incognita si presenta in forma di...

01 ESERCIZI

L Banchiere

$$\begin{cases} \text{Ris_massime}_i - \text{Ris_allocate}_i \geq 0 \\ \text{Ris_massime}_i - \text{Ris_allocate}_i \leq \text{Ris_disponibili}_i \end{cases}$$



$$\begin{cases} 4 - (x-1) \geq 0 \\ 4 - (x-1) \leq 2 \end{cases}$$

Creo un Sistema di disequazioni lineari!

Banchiere

A	B	C	D
2	2	10	4

Risorse_Disponibili

- P0: $6-4 \leq 2$? SI | $4-(x-1) \leq 2$? **NON SO** | $5-3 \leq 10$? SI | $6-2 \leq 4$? SI -> **SAFE (Plausibilmente)**
- P1: $10-8 \leq 2$? SI | $7-0 \leq 2$? **NO!** Mi fermo, processo non sicuro
- P2: $6-4 \leq 2$? SI | $2-0 \leq 2$? SI | $0-0 \leq 10$? SI | $8-0 \leq 4$? **NO!** . . .
- P3: $0-0 \leq 2$? SI | $3-0 \leq 2$? **NO!** . . .
- P4: $9-2 \leq 2$? **NO!** . . .

Quindi di tutti questi scelgo P0! Viene eseguito, termina e rilascia le sue risorse: Essendoci incognita x dobbiamo calcolare il range di risorse rilasciate per la i -sima risorsa dove l'incognita si presenta in forma di:

$$\begin{cases} 4 - (x-1) \geq 0 \\ 4 - (x-1) \leq 2 \end{cases} \longrightarrow \begin{cases} 5 \geq x \\ x \leq 3 \end{cases} \quad \text{RANGE: } 3 \leq x \leq 5$$

Quindi, rilascio le risorse, sommando quelle allocate a P0 alle risorse disponibili:

$$[2+4, 2+(x-1), 10+3, 4+2] \longrightarrow [6, x+1, 13, 6]$$

L Banchiere

	A	B	C	D
	2	2	10	4
P0	6	X+1	13	6

Risorse_Disponibili

- P1: $10-8 \leq 6$? SI | $7-0 \leq \cancel{x+1} 5+1$? ($3 \leq x \leq 5$, e caso peggiore: $x=5$)? **NO!** . . .
- P2: $6-4 \leq 6$? SI | $2-0 \leq x+1$? SI | $0-0 \leq 13$? SI | $8-0 \leq 6$? **NO!** . . .
- P3: $0-0 \leq 6$? SI | $3-0 \leq x+1$? SI | $4-3 \leq 13$? SI | $2-2 \leq 6$? SI -> **SAFE**
- P4: $9-2 \leq 6$? **NO!** . . .

Scelgo P3 come processo SAFE, lo eseguo e rilascio le risorse: qui non abbiamo incognite, quindi procedo col semplice rilascio:

$$[6+0, x-1+0, 13+3, 6+2] \longrightarrow [6, x+1, 16, 8]$$

L **Banchiere**

	A	B	C	D
	2	2	10	4
P0	6	X+1	13	6
P3	6	X+1	16	8

Risorse_Disponibili

- P1: $10-8 \leq 6$? SI | $7-0 \leq x+1$ 5+1? ($3 \leq x \leq 5$, e caso peggiore: $x=5$)? **NO!** . . .
- P2: $6-4 \leq 6$? SI | $2-0 \leq x+1$? SI | $0-0 \leq 13$? SI | $8-0 \leq 8$? SI -> **SAFE**
- P4: $9-2 \leq 6$? **NO!** . . .

Scelgo P2 come processo SAFE, lo eseguo e rilascio le risorse: qui non abbiamo incognite, quindi procedo col semplice rilascio:

$$[6+4, x-1+0, 16+0, 8+0] \longrightarrow [10, x+1, 16, 8]$$

Banchiere

	A	B	C	D
P0	2	2	10	4
P3	6	X+1	13	6
P2	6	X+1	16	8
P2	10	X+1	16	8

Risorse_Disponibili

- P1: $10-8 \leq 6$? SI | $7-0 \leq x+1$ 5+1? ($3 \leq x \leq 5$, e caso peggiore: $x=5$)? **NO!** . . .
- P4: $9-2 \leq 10$? SI | $1-1 \leq x+1$? SI | $6-(z+1) \leq 16$? **NON SO** | $9-4 \leq 8$? SI -> **SAFE**

Scelgo P4 come processo SAFE, lo eseguo e rilascio le risorse: essendoci incognita z dobbiamo calcolare il range di risorse rilasciate per la i -sima risorsa dove l'incognita si presenta:

$$\begin{cases} 6 - (z+1) \geq 0 \\ 6 - (z+1) \leq 16 \end{cases} \longrightarrow \begin{cases} 5 \geq z \\ z \geq -11 \end{cases} \quad \text{RANGE: } -11 \leq z \leq 5$$

Quindi, rilascio le risorse, sommando quelle allocate a P4 alle risorse disponibili:

$$[10+2, x+1+1, 16+z+1, 8+4] \longrightarrow [12, x+2, z+17, 12]$$

Banchiere

	A	B	C	D
	2	2	10	4
P0	6	X+1	13	6
P3	6	X+1	16	8
P2	10	X+1	16	8
P4	12	X+2	Z+17	12

Risorse_Disponibili

- P1: $10-8 \leq 6$? SI | $7-0 \leq x+2$? SI | $6-(y-2) \leq z+17$? **NON SO** | $8-2 \leq 12$? SI -> **SAFE**

Scelgo P1 come processo SAFE, lo eseguo e rilascio le risorse: essendoci incognita y dobbiamo calcolare il range di risorse rilasciate per la i-sima risorsa dove l'incognita si presenta questa volta con una doppia incognita (y,z):

$$\begin{aligned}
 &\left[\begin{array}{l} 6 - (y-2) \geq 0 \\ 6 - (y-2) \leq z+17 \end{array} \right] \rightarrow \boxed{\begin{array}{l} \text{Sostituisco } z \\ \text{con il suo} \\ \text{valore} \\ \text{peggiore di} \\ \text{allocazione: } 5 \end{array}} \rightarrow \left[\begin{array}{l} 8 \geq y \\ -y \leq 5+17-8 \end{array} \right] \rightarrow \left[\begin{array}{l} 8 \geq y \\ y \geq -14 \end{array} \right] \quad \text{RANGE: } -14 \leq y \leq 8
 \end{aligned}$$

Quindi, rilascio le risorse, sommando quelle allocate a P1 alle risorse disponibili:

$$[12+8, x+2+0, z+17+(y-2), 12+2] \longrightarrow [20, x+2, z+y+15, 14]$$

L Banchiere

	A	B	C	D
	2	2	10	4
P0	6	X+1	13	6
P3	6	X+1	16	8
P2	10	X+1	16	8
P4	12	X+2	Z+17	12
P1	20	X+2	Z+Y+15	14

Risorse_Disponibili

- La sequenza sicura è: $\langle P0, P3, P2, P4, P1 \rangle$
- Range X,Y,Z:
 - X: $3 \leq x \leq 5$
 - Z: $-11 \leq z \leq 5$
 - Y: $-14 \leq y \leq 8$
- La richiesta $P2[2,0,0,2]$ Può essere soddisfatta dopo che P0 e P3 hanno rilasciato le loro risorse. Perché?

Banchiere: Richiesta

	A	B	C	D
P0	4	x-1	3	2
P1	8	0	y-2	2
P2	4	0	0	0
P3	0	0	3	2
P4	2	1	z+1	4

Risorse_Allocate_{J,K}

	A	B	C	D
P0	6	4	5	6
P1	10	7	6	8
P2	6	2	0	8
P3	0	3	4	2
P4	9	1	6	9

Risorse_Massime_{J,K}

	A	B	C	D
	2	2	10	4
P0	6	X+1	13	6
P3	6	X+1	16	8
P2	10	X+1	16	8
P4	12	X+2	Z+17	12
P1	20	X+2	Z+Y+15	14

Risorse_Disponibili

Richiesta: P2[2,0,0,2]

Come faccio a capire?

$\text{Ris_Max} - (\text{Ris_Allocate} + \text{Richiesta}) \leq \text{Ris_disp} - \text{Richiesta}$

ES: $6 - (4 + 2) \leq 2 - 2$? SI

ES: $8 - (0 + 2) \leq 4 - 2$? NO!

Provo dopo P0 -> NO!

Provo dopo P3 -> SI!

L Formule

- **TEMPO DI ATTESA**

- **Normale:** $(T_{\text{completamento}} - T_{\text{arrivo}} - T_{\text{servizio}})$
- **Medio:** $\sum (T_{\text{completamento}} - T_{\text{arrivo}} - T_{\text{servizio}}) / N_{\text{processi}}$

- **TURNAROUND**

- **Normale:** $T_{\text{completamento}} - T_{\text{arrivo}}$
- **Medio:** $\sum (T_{\text{completamento}} - T_{\text{arrivo}}) / N_{\text{processi}}$
- **Normalizzato:** $T_{\text{completamento}} - T_{\text{arrivo}} / \text{Burst}$

- **THROUGHPUT**

Calcolato come:

- $N_{\text{processi}} / \text{Tempo Totale}$

Scheduling

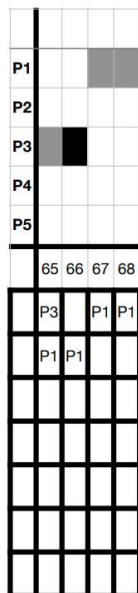
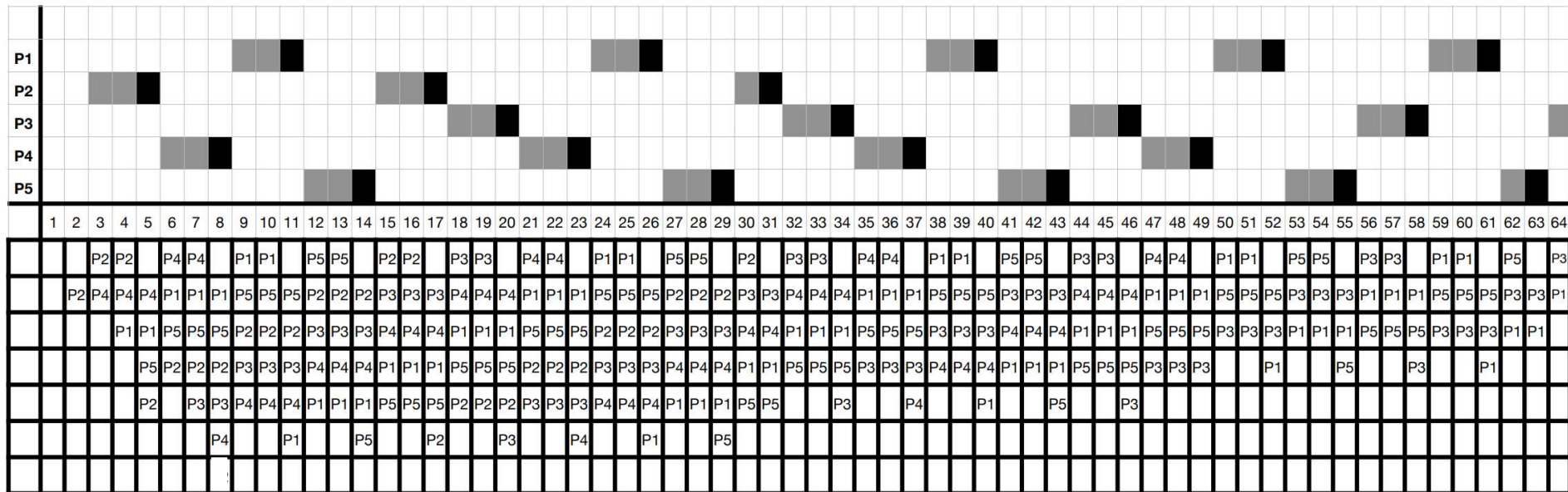
Esercizio 1

- Si considerino i seguenti processi, attivi in un sistema multiprogrammato:

Processo	Tempo di Arrivo	CPU Burst	Priorità
P1	4ms	12ms	2
P2	2ms	5ms	1
P3	7ms	10ms	5
P4	3ms	8ms	4
P5	5ms	9ms	3

- Supponendo che il cambio di contesto sia 1ms, si mostri l'ordine di esecuzione dei processi e quanto vale il tempo di attesa medio, il tempo di turnaround medio ed il tempo di turnaround normalizzato medio per ciascuno dei seguenti algoritmi di scheduling della CPU:
 - Round Robin con quanto $q = 2\text{ms}$

Diagramma di esecuzione(Round Robin, q=2ms)



Processo	Tempo di Arrivo	CPU Burst	Priorità
P1	4ms	12ms	2
P2	2ms	5ms	1
P3	7ms	10ms	5
P4	3ms	8ms	4
P5	5ms	9ms	3

- Tempi di esecuzione: (Round Robin, $q=2\text{ms}$)

- Tempo di Attesa:

$$P1 = 68 - 4 - 12 = 52 \text{ ms}$$

$$P2 = 30 - 2 - 5 = 23 \text{ ms}$$

$$P3 = 65 - 7 - 10 = 48 \text{ ms}$$

$$P4 = 48 - 3 - 8 = 37 \text{ ms}$$

$$P5 = 62 - 5 - 9 = 48 \text{ ms}$$

$$\text{Tempo di attesa medio} = 41,6 \text{ ms}$$

- Turnaround

$$P1 = 68 - 4 = 64 \text{ ms}$$

$$P2 = 30 - 2 = 28 \text{ ms}$$

$$P3 = 65 - 7 = 58 \text{ ms}$$

$$P4 = 48 - 3 = 45 \text{ ms}$$

$$P5 = 62 - 5 = 57 \text{ ms}$$

$$\text{Turnaround medio} = 50,4 \text{ ms}$$

- Tempi di esecuzione: (Round Robin, $q=2\text{ms}$)

- Turnaround normalizzato

$$P1 = (68 - 4)/12 = 64/12 = 5,3 \text{ ms}$$

$$P2 = (30 - 2)/5 = 28/5 = 5,6 \text{ ms}$$

$$P3 = (65 - 7)/10 = 58/10 = 5,8 \text{ ms}$$

$$P4 = (48 - 3)/8 = 45/8 = 5,6 \text{ ms}$$

$$P5 = (62 - 5)/9 = 57/9 = 6,3 \text{ ms}$$

$$\text{Turnaround normalizzato medio} = 5,8 \text{ ms}$$

- Throughput

$$X = \frac{5}{68000} * 1000 = 0,07$$

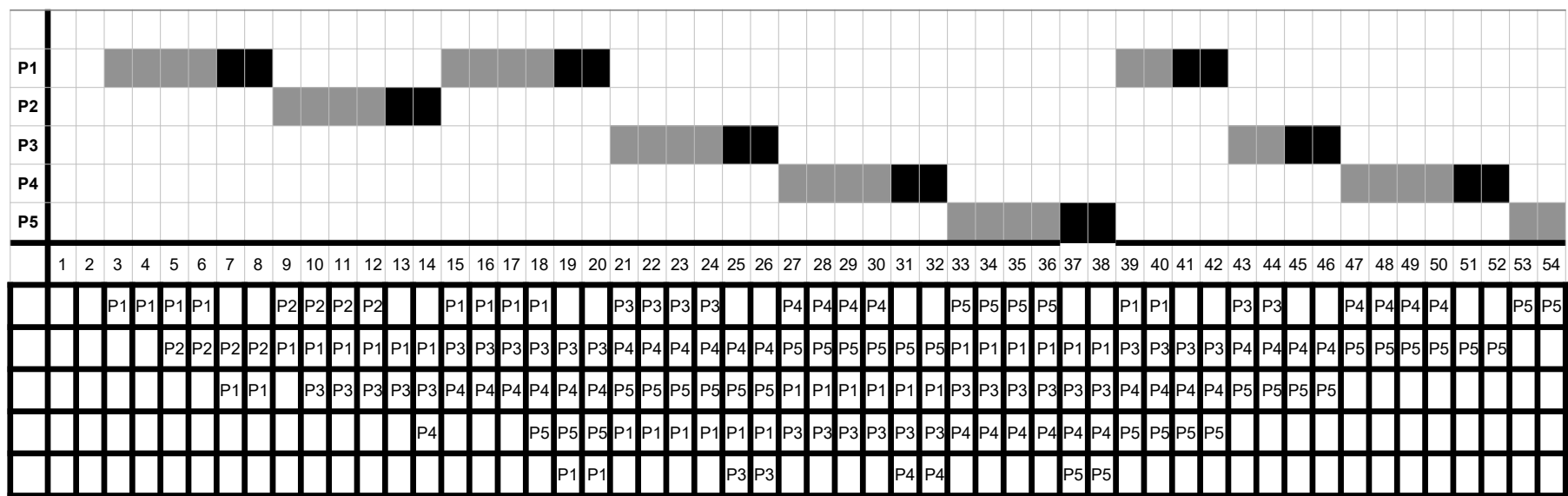
Esercizio 2

- Si considerino i seguenti processi, attivi in un sistema multiprogrammato:

Processo	Tempo di Arrivo	CPU Burst
P1	2ms	10ms
P2	4ms	4ms
P3	10ms	6ms
P4	14ms	8ms
P5	18ms	6ms

- Assumendo che il context switch richieda 2 ms, fornire il diagramma di esecuzione, il tempo medio di attesa e completamento per lo scheduling:
 - Round Robin ($q=4$ ms)

Diagramma di esecuzione(Round Robin, q=4ms)



Processo	Tempo di Arrivo	CPU Burst
P1	2ms	10ms
P2	4ms	4ms
P3	10ms	6ms
P4	14ms	8ms
P5	18ms	6ms

- Tempi di esecuzione: (Round Robin, $q=4\text{ms}$)

- Tempo di Attesa:

$$P1 = 40 - 10 - 2 = 28 \text{ ms}$$

$$P2 = 12 - 4 - 4 = 4 \text{ ms}$$

$$P3 = 44 - 6 - 10 = 28 \text{ ms}$$

$$P4 = 50 - 8 - 14 = 28 \text{ ms}$$

$$P5 = 54 - 6 - 18 = 30 \text{ ms}$$

$$\text{Tempo di attesa medio} = 23,6\text{ms}$$

- Turnaround:

$$P1 = 40 - 2 = 38 \text{ ms}$$

$$P2 = 12 - 4 = 8 \text{ ms}$$

$$P3 = 44 - 10 = 34 \text{ ms}$$

$$P4 = 50 - 14 = 35 \text{ ms}$$

$$P5 = 54 - 18 = 36 \text{ ms}$$

$$\text{Turnaround medio} = ..$$

- Tempi di esecuzione: (Round Robin, $q=4\text{ms}$)
- Turnaround Normalizzato:
Turnaround Normalizzato Medio:
- Througput:

L Banchiere

	A	B	C	D
P0	4	3	Y-1	2
P1	8	Z-1	0	1
P2	4	0	1	1
P3	0	2	1	3
P4	2	X+2	1	4

Risorse_Allocate_{J,K}

	A	B	C	D
P0	6	5	4	5
P1	10	6	1	8
P2	6	0	2	4
P3	2	6	3	3
P4	3	4	2	5

Risorse_Massime_{J,K}

A	B	C	D
3	11	5	8

Risorse_Disponibili

Determinare:

1. gli intervalli dei valori X,Y,Z per i quali il sistema si trova in uno stato sicuro
2. Se la richiesta P0[3,1,0,2] può essere soddisfatta.

Scheduling

Esercizio 1

- Si considerino i seguenti processi, attivi in un sistema multiprogrammato:

Processo	Tempo di Arrivo	CPU Burst	Priorità
P1	2ms	14ms	3
P2	8ms	10ms	4
P3	16ms	16ms	5
P4	20ms	12ms	2
P5	20ms	16ms	1

- Supponendo che il cambio di contesto sia 1ms, si mostri l'ordine di esecuzione dei processi e quanto vale il tempo di attesa medio, il tempo di turnaround medio ed il tempo di turnaround normalizzato medio per ciascuno dei seguenti algoritmi di scheduling della CPU:
 - Priorità con prelazione (la priorità massima è 5)
 - FCFS
 - STG
 - Round Robin con quanto $q = 4ms$

Scheduling

Esercizio 1

- Si considerino i seguenti processi, attivi in un sistema multiprogrammato:

Processo	Tempo di Arrivo	CPU Burst	Priorità
P1	4ms	4ms	3
P2	2ms	10ms	3
P3	4ms	10ms	1
P4	6ms	3ms	2
P5	7ms	6ms	4

- Supponendo che il cambio di contesto sia 3ms, si mostri l'ordine di esecuzione dei processi e quanto vale il tempo di attesa medio, il tempo di turnaround medio ed il tempo di turnaround normalizzato medio per ciascuno dei seguenti algoritmi di scheduling della CPU:
 - Priorità con prelazione (la priorità massima è 1)
 - FCFS
 - STG
 - Round Robin con quanto $q = 3ms$