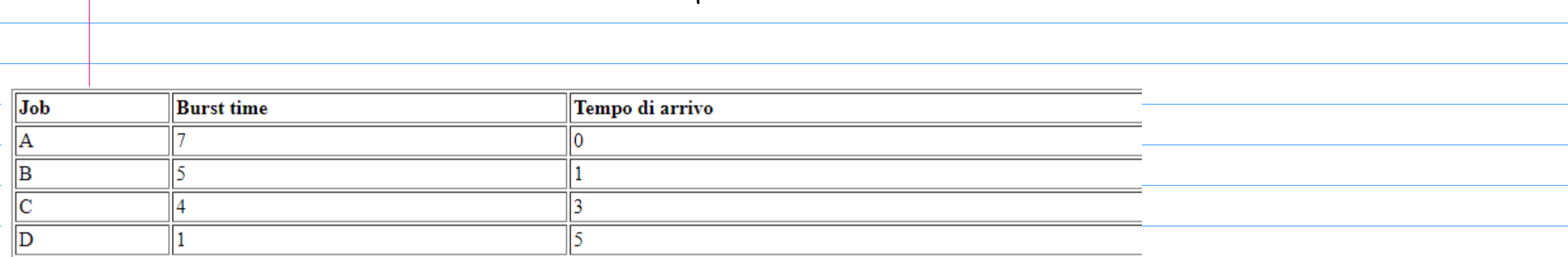


$int N = 2$ turn
 bool $intended [N] = \text{false}$

while-region (process)
 other = false
 intended[process] = true
 turn = process
 while (intended[other] AND turn = process) {do nothing}
 reader() mutex = 1 db = 1
 fc = 0
 while (true)
 | down(db)
 | fc++
 | if (fc == 1) down(db)
 | up(mutex)
 | op-db()
 | down(m)
 | fc--
 | if (fc == 0) up(db)
 | up(mutex)
 writer()
 while (true)
 | down(db)
 | op-db()
 | up(db)

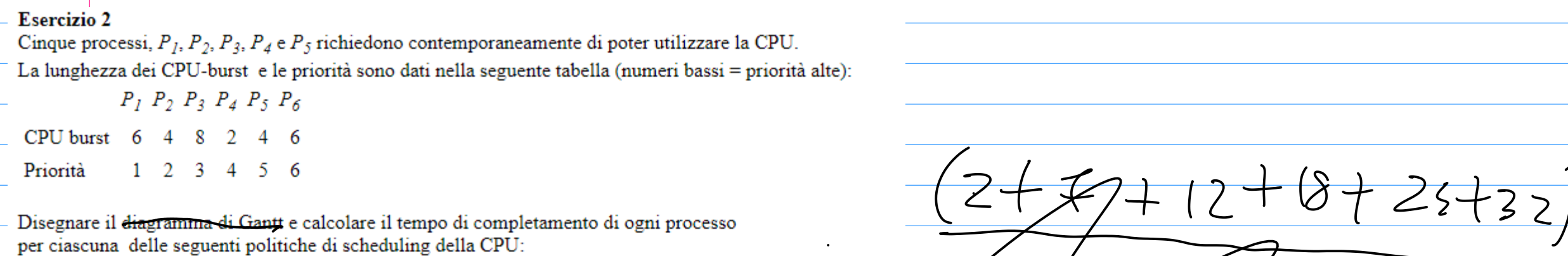
Job	Burst time	Tempo di arrivo
A	7	0
B	5	1
C	4	3
D	1	5



$$(0 + 2 + 5 + 11) / 4 = 4,5$$

$$(7 + 3 + 9 + 16) / 4 = 8,75$$

Job	Burst time	Tempo di arrivo
A	7	0
B	5	1
C	4	3
D	1	5

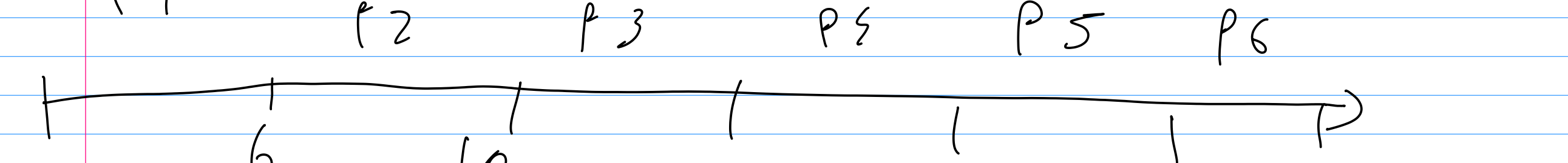


$$t_{m,A} = (7 + 0 + 7 + 3) / 4 = 4$$

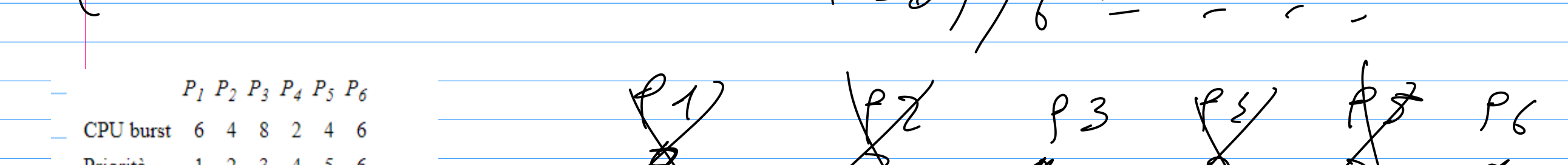
$$t_{m,C} = (17 + 5 + 2 + 8) / 4 = 8$$

Esercizio 2
 Cinque processi, P_1, P_2, P_3, P_4 e P_5 richiedono contemporaneamente di poter utilizzare la CPU.
 La lunghezza dei CPU-burst e le priorità sono dati nella seguente tabella (numeri bassi = priorità alte):
 CPU burst: $P_1: 6, P_2: 4, P_3: 2, P_4: 2, P_5: 6$
 Priorità: $P_1: 1, P_2: 2, P_3: 3, P_4: 4, P_5: 6$

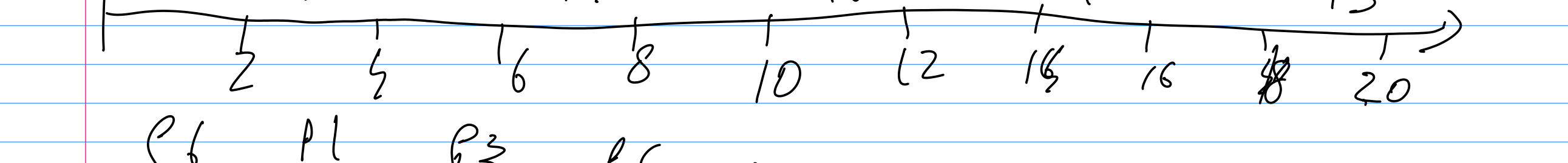
Disegnare il diagramma di Gantt e calcolare il tempo di completamento di ogni processo per ciascuna delle seguenti politiche di scheduling della CPU:
 1. Shortest Job First
 2. Con Priorità
 3. Round Robin (con quanto di tempo $q=2$ e trascurando il tempo di context switch)



$$(2 + 7 + 12 + 18 + 25 + 32) / 6 = 15,83$$



$$(6 + 10 + 18 + 20 + 24 + 30) / 6 = 18$$



$$(2 + 16 + 30 + 8 + 20 + 28) / 6 = 18$$

Esercizio 3
 Considerare la seguente stringa di riferimenti di un processo alla memoria in un sistema con memoria virtuale:
 4 8 3 11 5 7 2 8 3 1 2 6 11 4 5 6 7 11 9
 Illustrare il comportamento dell'algoritmo LRU di sostituzione delle pagine per una memoria fisica di 4 blocchi.
 Calcolare il numero di page fault che si verificano.

$$PF = 5 \dots 5 \dots 6 \dots 7 \dots 8 \dots 9 \dots 10 \dots 11$$

5	5	5	7	7
7	7	7	7	7
3	2	3	3	1
11	11	11	11	11

Esercizio 4
 Calcolare la dimensione minima di una FAT necessaria per indirizzare un disco di capacità 5096 MB con blocchi da 4KB. Considerate ora i seguenti due file:
 1. A di dimensione 13000 byte
 2. B di dimensione 7500 byte

Mostrare un esempio di come vengono memorizzati gli indirizzi dei blocchi dei file A e B in una FAT per le caratteristiche viste sopra (scegliete indirizzi di blocco a piacere per i due file).

$$5096 \cdot 2^{20} B \quad 4KB = 4 \cdot 2^{10} B$$

$$\frac{5096 \cdot 2^{20} B}{4 \cdot 2^{10} B} = 1273 \cdot 2^{10}$$

$$11 \text{ bit} \quad 10 \text{ bit} \quad 21 \text{ bit}$$

$$((1273 \cdot 2^{10}) \cdot 21 \text{ bit}) \quad 29 \text{ bit} = 3 \text{ byte}$$

$$13000 \cdot 8 \text{ bit} \cdot 3 \text{ byte} = 3913728 \text{ byte}$$

$$\frac{3913728 \text{ byte}}{5096 \text{ byte}} = 768$$

$$\text{File A} = 13000 \text{ byte}$$

$$\frac{13000 \text{ byte}}{5096 \text{ byte}} = 2$$

FAT

0	1
1	2
2	3
3	4
4	5

1. byte 6758 del file "pippo" che inizia al blocco 4
 2. byte 4097 del file "pluto" che inizia al blocco 3
 3. byte 2044 del file "paperino" che inizia al blocco 5

Frammento FAT:
 Entry: 0 10 1 2 3 4 5 6 7 8 9 10 11 12
 Contente: 10 1 2 3 4 5 6 7 8 9 10 11 12

$$P_{\text{pippo}}: 5 \rightarrow 1 \rightarrow 2 \rightarrow 0 \rightarrow 10$$

$$6758 \text{ B} / 2048 \text{ B} = 3^{\circ}$$

$$6758 \bmod 2048 = 612$$

$$P_{\text{pluto}}: 3 \rightarrow 6 \rightarrow 7 \rightarrow 11$$

$$4097 \text{ B} / 2048 \text{ B} = 2^{\circ}$$

$$4097 \bmod 2048 = 201$$

$$12 \text{ blocks} \quad 12 \text{ KB}$$

$$12 \text{ blocks} \quad 12 \text{ KB}$$

$$(12 + 256 + 256^2) \text{ blocks}$$

$$12 \text{ blocks} \quad 12 \text{ KB}$$

$$3) 3004 \text{ B} \quad 3004 \text{ B} / 14 \text{ B} = 214$$

Partizione	Dimensione
0	200K
1	500K
2	300K
3	600K

e dato che il sistema operativo deve allocare quattro processi P_1 di 212K, P_2 di 417K, P_3 di 112K e P_4 di 427K (in quest'ordine), scrivere in quale partizione il sistema alloca ciascun processo se viene usato l'algoritmo:
 1. First Fit
 2. Best Fit
 3. Worst Fit

$$\text{FF } P_1: 1 \quad P_2: 3 \quad P_3: 0 \quad P_4: X$$

$$\text{DF } P_1: 2 \quad P_2: 1 \quad P_3: 0 \quad P_4: 3$$

$$\text{WF } P_1: 3 \quad P_2: 1 \quad P_3: 2 \quad P_4: X$$

$$B: 2 \text{ KB} \quad m: 2 \text{ B} = 16 \text{ bit}$$

$$2^{16} \quad 2^{16} \cdot 2048 \text{ B}$$

$$2004 \text{ B} / 2048 \text{ B} = 101 \text{ blocks}$$

$$2048 \text{ B} / 2 \text{ B} = 1024$$

$$1024^2 \cdot 2048 \text{ B} = 2^{20} \cdot 2^{11} = 2^{31} \text{ B}$$

$$100 \text{ blocks}$$

$$TSL \quad r, l \rightarrow \begin{cases} \text{MOVE } r, l \\ \text{MOVE } l, r \end{cases}$$

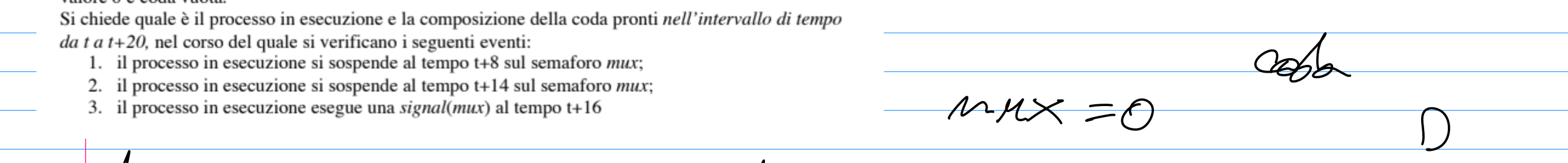
Un sistema gestisce il processore con politica RoundRobin con quanto di tempo di 5 msec. Quando un processo va in esecuzione gli viene assegnato un intero quanto di tempo di 5 ms, indipendentemente dal tempo consumato nel precedente turno di esecuzione.

Nel sistema sono presenti 4 processi (A,B,C,D) e un semaforo di mutua esclusione $mutex$. Al tempo t passa in esecuzione il processo A, la coda pronti contiene i processi B->C->D e il semaforo $mutex$ ha valore 0 e coda vuota.

Si chiede quale il processo in esecuzione e la composizione della coda pronti nell'intervallo di tempo da $t+4$ a $t+20$, nel corso del quale si verificano i seguenti eventi:

- il processo in esecuzione si sospende al tempo $t+8$ sul semaforo $mutex$;
- il processo in esecuzione si sospende al tempo $t+14$ sul semaforo $mutex$;
- il processo in esecuzione esegue una $signal(mutex)$ al tempo $t+16$

$$Coda: \quad B \rightarrow A$$



P1 in esecuzione, P2 e P3 pronti, P4 in attesa di una operazione di I/O.

Le priorità dei processi sono in questa relazione:
 • $P1 > P2 > P4 > P3$

Descrivendo, motivando la risposta, come cambia lo stato del sistema (cioè come cambia lo stato dei processi) se a partire dalla situazione data si verificano nell'ordine tutti e soli i seguenti eventi:

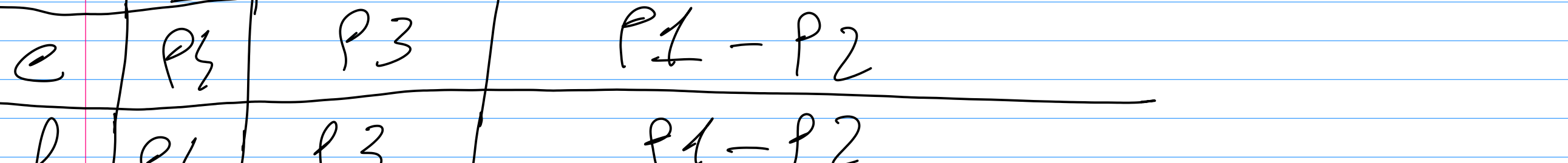
- trascorre un quarto di tempo
- termina l'operazione di I/O del processo in attesa
- il processo in esecuzione chiede una operazione di I/O
- trascorre un quarto di tempo
- il processo in esecuzione chiede una operazione di I/O
- trascorre un quarto di tempo
- il processo in esecuzione termina

	E	RQ	I/O
0	P1	P2-P3	P4
a	P2	P1-P3	P4
b	P2	P1-P3-P4	Ø
c	P1	P3-P4	P2
d	P1	P1-P3	P2
e	P4	P3	P1-P2
f	P4	P3	P1-P2
g	P3	Ø	P1-P2

2. Quattro processi (da P_1 a P_4) entrano nello stato di pronto a distanza di 100 ms l'uno dall'altro e nell'ordine da P_1 a P_4 . I processi hanno stessa priorità. Si supponga che il tempo stimato del prossimo CPU-burst di ogni processo sia rispettivamente di 600, 300, 500 e 200 ms. Dopo questo CPU burst inizia l'operazione di I/O molto lunga. Per le politiche di scheduling elencate di seguito, determinare il tempo di attesa per ciascun processo nello stato di pronto, (considerando solo il CPU burst) e il tempo di attesa medio dei quattro processi:

- Shortest Process Next
- Round-robin con un quanto di tempo di 400 ms
- Round-robin con un quanto di tempo di 200 ms
- First Come First Served

$$Q = 400 \text{ ms}$$



$$(900 + 300 + (500 + 400) + 800) / 4 = 725$$

3. Si consideri un processo che fa riferimento a 5 pagine virtuali nel seguente ordine:

Si consideri una memoria fisica inizialmente vuota di 3 pagine e si illustri il comportamento dei seguenti algoritmi di sostituzione delle pagine, evidenziando a page fault che vengono generati:

- Ottimo
- FIFO (First In First Out)
- LRU (Least Recently Used)

$$5-1-3 \quad \text{Optim PF: } 3 \dots 4 \dots 5 \dots 6 \quad \text{FIFO PF: } 3 \dots 4 \dots 5 \dots 6 \dots 7$$

3	5
5	1
1	3