

Esercizio S3-L1

Tabella Monotasking

Secondi	P1	P2	P3	P4
0-1	■	■	■	■
1-2	■	■	■	■
2-3	■	■	■	■
3-4	■	■	■	■
4-5	■	■	■	■
5-6	■	■	■	■
6-7	■	■	■	■
7-8	■	■	■	■
8-9	■	■	■	■
9-10	■	■	■	■
10-11	■	■	■	■
11-12	■	■	■	■
12-13	■	■	■	■
13-14	■	■	■	■
14-15	■	■	■	■

Legenda:

















- Esecuzione
- Attesa (I/O)
- Pronto
- Terminato

Spiegazione:

1. Secondi 0-3:

- **P1** è in esecuzione (■) per i primi 3 secondi, mentre P2, P3 e P4 sono pronti (■) in attesa di essere eseguiti.





























2. Secondi 3-5:

- **P1** entra in uno stato di attesa () per operazioni di I/O nei secondi 3-5. Durante questo tempo, P2, P3 e P4 rimangono pronti (), ma nessuno di loro viene ancora eseguito.
3. **Secondi 5-6:**
- **P1** torna in esecuzione () al secondo 5, e P2, P3 e P4 rimangono pronti ().
4. **Secondi 6-8:**
- **P1** termina l'esecuzione () al secondo 6. **P2** inizia a essere eseguito () dal secondo 6 all'8. Durante questo periodo, P3 e P4 rimangono pronti ().
5. **Secondi 8-9:**
- **P2** passa in attesa () per operazioni di I/O al secondo 8, e **P3** inizia l'esecuzione () mentre P4 rimane pronto ().
6. **Secondi 9-10:**
- **P2** torna dall'attesa e termina l'esecuzione (). **P3** continua la sua esecuzione () e P4 rimane pronto ().
7. **Secondi 10-15:**
- **P3** termina l'esecuzione () al secondo 10, e **P4** inizia la sua esecuzione () dal secondo 10 al 14. Al secondo 14, **P4** termina (), completando l'esecuzione di tutti i processi.

Riassunto

- **P1:** Inizia in esecuzione, entra in attesa per I/O, poi riprende e termina.
- **P2:** Inizia dopo P1, entra in attesa per I/O, poi riprende e termina.
- **P3:** Inizia dopo P2 e termina al secondo 10.
- **P4:** Esegue dopo P3 e termina al secondo 14.

Schedulazione Multitasking

Secondi	P1	P2	P3	P4
0-1				
1-2				
2-3				
3-4				
4-5				
5-6				
6-7				

Secondi	P1	P2	P3	P4
7-8	■	■	■	■
8-9	■	■	■	■
9-10	■	■	■	■
10-11	■	■	■	■
11-12	■	■	■	■

Legenda:

- Esecuzione
- Attesa (I/O)
- Pronto
- Terminato

Spiegazione:

1. Secondi 0-3:

- **P1** è in esecuzione (■) per i primi 3 secondi, mentre **P2**, **P3** e **P4** sono pronti (■) ma non ancora eseguiti.

2. Secondi 3-5:

- **P1** entra in attesa per operazioni di I/O (■) al secondo 3, e **P2** inizia la sua esecuzione (■) dal secondo 3 al secondo 5. **P3** e **P4** restano pronti (■).

3. Secondi 5-6:

- **P1** torna dall'attesa e riprende l'esecuzione (■) al secondo 5, mentre **P2** entra in attesa per operazioni di I/O (■). **P3** e **P4** rimangono pronti (■).

4. Secondi 6-7:

- **P1** e **P2** terminano l'esecuzione (■) al secondo 6. **P3** inizia la sua esecuzione (■) mentre **P4** rimane pronto (■).

5. Secondi 7-12:

- **P3** termina l'esecuzione (■) al secondo 7. **P4** inizia a essere eseguito (■) dal secondo 7 e continua fino al secondo 11.
- Al secondo 12, **P4** entra in attesa per operazioni di I/O (■).

Riassunto

- **P1**: Inizia in esecuzione, entra in attesa per I/O, riprende e termina al secondo 6.
- **P2**: Inizia dopo P1, entra in attesa per I/O al secondo 5 e termina al secondo 6.

- **P3**: Inizia dopo P2 e termina al secondo 7.
- **P4**: Esegue dopo P3 e termina la sua esecuzione al secondo 11, entrando poi in attesa al secondo 12.

Scheduling Time Sharing

Secondi	P1	P2	P3	P4
0-1	■	■	■	■
1-2	■	■	■	■
2-3	■	■	■	■
3-4	■	■	■	■
4-5	■	■	■	■
5-6	■	■	■	■
6-7	■	■	■	■
7-8	■	■	■	■
8-9	■	■	■	■
9-10	■	■	■	■
10-11	■	■	■	■
11-12	■	■	■	■
12-13	■	■	■	■
13-14	■	■	■	■
15-16	■	■	■	■
17-18	■	■	■	■
19-20	■	■	■	■
20-21	■	■	■	■
21-22	■	■	■	■
23-24	■	■	■	■

Legenda:

- ■ Esecuzione
- ■ Pronto

Spiegazione:

1. **Secondi 0-3:**

- **P1** è in esecuzione (■) per i primi 3 secondi, mentre **P2**, **P3**, e **P4** sono pronti (■) ma non vengono ancora eseguiti.
2. **Secondi 3-5:**
 - **P1** diventa pronto (■) e **P2** prende il controllo, entrando in esecuzione (■) per i successivi 2 secondi. **P3** e **P4** rimangono pronti (■).
 3. **Secondi 5-6:**
 - **P2** torna nello stato pronto (■) e **P3** inizia la sua esecuzione (■), con **P1**, **P2**, e **P4** ancora pronti.
 4. **Secondi 6-10:**
 - **P3** termina la sua esecuzione (■) e **P4** prende il controllo, entrando in esecuzione (■) per i prossimi 4 secondi (fino a 10 secondi), mentre **P1**, **P2**, e **P3** rimangono pronti (■).
 5. **Secondi 10-13:**
 - Dopo il ciclo di **P4**, **P1** ritorna in esecuzione (■) per altri 3 secondi, con **P2**, **P3**, e **P4** nuovamente pronti (■).
 6. **Secondi 13-16:**
 - **P1** diventa pronto (■) e **P2** riprende l'esecuzione (■) per altri 3 secondi. **P1**, **P3**, e **P4** rimangono pronti (■).
 7. **Secondi 17-18:**
 - **P2** torna pronto (■) e **P3** riprende l'esecuzione (■) per altri 2 secondi. **P1**, **P2**, e **P4** rimangono pronti (■).
 8. **Secondi 19-24:**
 - Infine, **P4** riprende e rimane in esecuzione (■) per gli ultimi 6 secondi del periodo (fino a 24 secondi), mentre **P1**, **P2**, e **P3** rimangono pronti (■).

Riassunto

- **P1** e **P2** si alternano per 3 secondi di esecuzione seguiti da periodi di attesa, riprendendo ciclicamente.
- **P3** segue uno schema simile, con periodi di esecuzione più brevi rispetto a **P1** e **P2**.
- **P4** ha due fasi di esecuzione, una delle quali è più lunga (6 secondi consecutivi).

Time-Sharing - The Best

- **Distribuzione equa:** Ogni processo ha la possibilità di eseguire per un periodo prima di cedere il controllo ad altri.

- **Riduzione dei tempi di attesa:** Questo schema riduce i tempi di attesa percepiti dagli utenti, poiché i processi sembrano eseguire "contemporaneamente" grazie alla rapida alternanza tra di essi.
- **Efficienza:** Il sistema mantiene tutti i processi attivi, alternando l'esecuzione per massimizzare l'utilizzo della CPU.
-

Considerazioni ulteriori

Tutto dipende dalla CPU, dalla frequenza, dai core e quindi dai thread.

- Un computer con un **i7** esegue **1600 processi al secondo**.
- Un computer con un **i9** esegue **2400 processi al secondo**.
- Un **computer quantistico** esegue **un triliardo di processi al secondo**.