

Shortest-Job-First (SJF)

Primero el trabajo más corto

Adaptación (ver referencias)

Shortest-Job-First (SJF)

- Se asocia a cada proceso la longitud de la siguiente ráfaga de CPU
 - En lugar de su longitud total como si sucede en FCFS
- Cuando la CPU está disponible, se le entrega al proceso con la ráfaga de CPU más corta
 - Situaciones de empate se resuelven mediante FCFS

| Procesos | Tiempo siguiente ráfaga de CPU |
|----------|--------------------------------|
| P_1 | 6 ms |
| P_2 | 8 ms |
| P_3 | 7 ms |
| P_4 | 3 ms |

Tiempo espera $P_1 = 3$

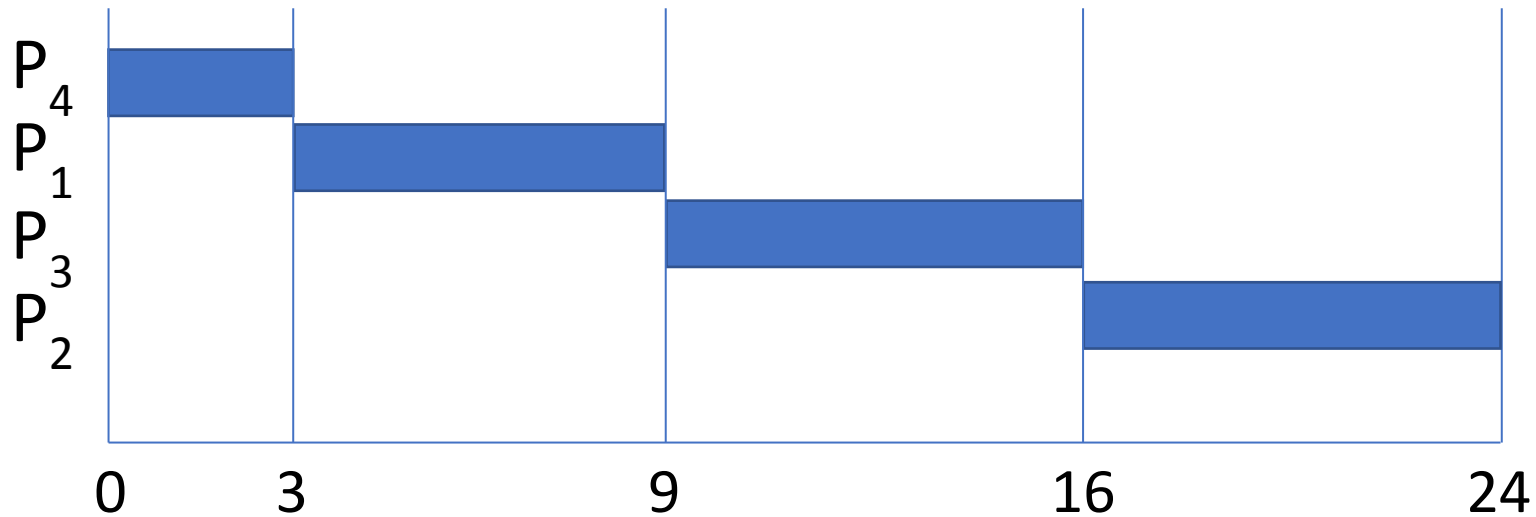
Tiempo espera $P_2 = 16$

Tiempo espera $P_3 = 9$

Tiempo espera $P_4 = 0$

Tiempo espera promedio = 7 ms

Con FCFS
promedio = 10.25 ms



Shortest-Job-First (SJF)

- En la práctica resulta difícil saber (a priori) las necesidades de ráfagas de CPU de un proceso.
- Se puede predecir la duración de la siguiente ráfaga de CPU asumiendo que son similares a las previas.
 - Se usa promedio exponencial para predecir la duración de la siguiente ráfaga de CPU de un proceso.
- Puede ser apropiativo y no apropiativo
 - Decisión tiene lugar cuando llega un nuevo proceso con una ráfaga de CPU más corta de lo que le queda al proceso que actualmente se está ejecutando.

| Procesos | Tiempo siguiente ráfaga de CPU | Tiempo de llegada |
|----------------|-----------------------------------|----------------------|
| P ₁ | 8 ms | 0 |
| P ₂ | 4 ms | 1 |
| P ₃ | 9 ms | 2 |
| P ₄ | 5 ms | 3 |

Tiempo espera P₁ = 10 - 1

Tiempo espera P₂ = 1 - 1

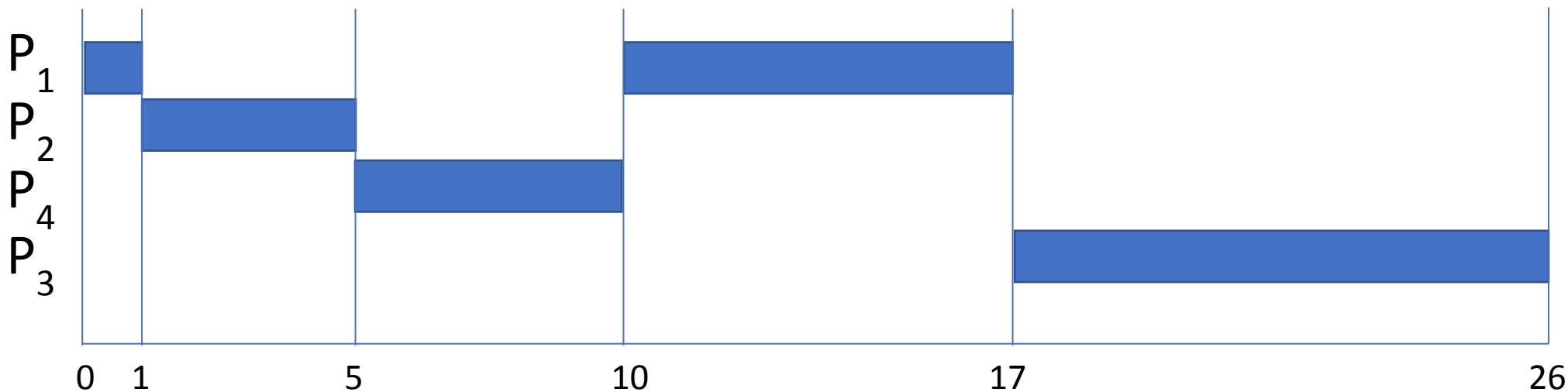
Tiempo espera P₃ = 17 - 2

Tiempo espera P₄ = 5 - 3

Tiempo espera promedio = 6.5 ms

Ejecución no apropiativa
promedio = 7.75 ms

SJF ejecución apropiativa



SJF ejecución apropiativa

- t_0 : P_1 es el único proceso en cola, se ejecuta
- t_1 : Llega P_2 con menos tiempo de CPU
 - Sale P_1 y le quedan 7ms
 - Se ejecuta P_2
- t_2 : Llega P_3 pero requiere 9ms, espera
 - Se sigue ejecutando P_2 , sigue siendo el trabajo más corto
- t_3 : Llega P_4 pero requiere 5 ms, espera
 - P_2 está ejecutando y seguirá siendo el trabajo más corto
- t_5 : Termina P_2 , sale de CPU
 - Entra a ejecutar P_4
 - Aún esperan P_1 con 7ms y P_3 con 9ms
- t_{10} : Se ejecuta todo P_4 y
 - Sale de CPU
 - Entra P_1
- t_{17} : Termina P_1
 - Sale de CPU
 - Entra P_3 y se ejecuta
- t_{26} : Termina P_3

Referencias

- Silberschatz, A., Baer Galvin, P., & Gagne, G. (2018). CPU Scheduling. In *Operating Systems Concepts* (10th ed., pp. 207–209). John Wiley & Sons, Inc.