

# SRTF

## SRTF (Shortest Remaining Time First)

Este algoritmo chamado **SRTF (Shortest Remaining Time First)** ten en conta a quenda de chegada sendo expropiativo, é dicir, ven sendo o algoritmo **SJF pero expropiativo**, así determina que o proceso a entrar na CPU de todos os posibles será aquel que teña menos duración de execución na mesma, isto é, entrará o proceso con menor ciclos de CPU a executar, tendo en conta que se quere entrar un proceso con menor duración na CPU co proceso que se está executando expulsará ao proceso que se está executando e entrará na CPU para executarse -pois o algoritmo é expropiativo-.

Imos ver un exemplo para explicar como traballa o algoritmo *SRTF*:

- Supoñemos a situación seguinte:

◊ **Tempo de chegada:** P1-->0, P2-->5, P3-->4, P4-->2

◊ **Cola:** P1, P4, P3, P2

◊ **Duración Proceso:** P1-->4 ciclos de CPU, P2-->7 ciclos de CPU, P3-->4 ciclos de CPU. P4-->1 ciclo de CPU.

sendo,

$te|_{P_i}$  O tempo de espera do Proceso  $P_i$

$tr|_{P_i}$  O tempo de retorno do Proceso  $P_i$

Imos calcular o tempo medio de espera para este algoritmo, así como o Diagrama de Gantt correspondente,

|                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| P1               | P1 | P1 | E  | P1 | P1 |    |    |    |    |    |    |    |    |    |    |    |
| P2               |    |    |    |    |    | E  | E  | E  | E  | P2 | P2 | P2 | P2 | P2 | P2 | P2 |
| P3               |    |    |    |    | E  | P3 | P3 | P3 | P3 |    |    |    |    |    |    |    |
| P4               |    |    | P4 |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Ciclos CPU       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Tempo de chegada | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |

↑                    ↑                    ↑    ↑  
 P1                  P4                  P3   P2

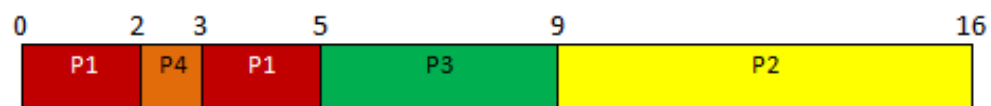
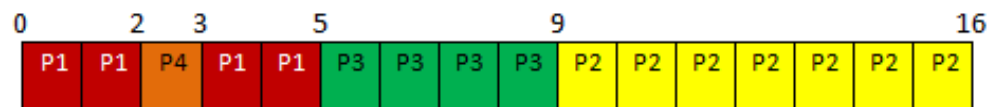
$te|_{p_1}=1$

$te|_{p_2}=4$

$te|_{p_3}=1$

$te|_{p_4}=0$

$$\overline{te} = [(te|_{p_1} + te|_{p_2} + te|_{p_3} + te|_{p_4})/4] = [(1+4+1+0)/4] = 1.5$$



Como podemos ver na imaxe o primeiro en entrar na CPU é o proceso P1 pois na orde de chegada é o primeiro da cola de procesos. O algoritmo **SRTF** determina que ao entrar un proceso esté ocupará a CPU ata que outro proceso teña máis preferencia de uso, neste caso, ata que queira entrar na CPU outro proceso con menor número de ciclos de CPU para a súa execución, xa que o algoritmo é expropiativo, así:

1. **Ciclo de CPU 1-Tempo de Chegada 0:** Entra o proceso P1 na CPU e acapara os ciclos da mesma ata a entrada do proceso P4 -paso seguinte no cal estudarase o que acontecerá-, co cal -de momento- acapara a CPU 2 ciclos da mesma.
2. **Ciclo de CPU 3-Tempo de Chegada 2:** A continuación entra o proceso P4, xa que o algoritmo **SRTF** determina que o proceso a entrar na CPU sexa aquel que ocupe menos ciclos da mesma. Así o proceso P4 soamente ocuparía a CPU 1 ciclo a diferenza do P1 que ocuparía, aínda, a CPU 2 ciclos máis da mesma. Entón entra P4 que acapararía 1 ciclo de CPU expulsando o proceso P1 da súa execución.
3. **Ciclo de CPU 4-Tempo de Chegada 3:** Nesta situación temos soamente 1 proceso en cola, o proceso P1, co cal execútase P1 durante 1 ciclo de execución -o ciclo de CPU 4-, xa que no ciclo de CPU 5 quere entrar P3, co cal debemos mirar segundo o algoritmo **SRTF** quen ocupa a CPU na súa execución.
4. **Ciclo de CPU 5-Tempo de Chegada 4:** Agora vemos que o proceso P3 ocuparía 4 ciclos de CPU e o proceso P1 ocuparía 1 ciclo de CPU, logo segue executándose o proceso P1 durante un ciclo da mesma, finalizando así a súa execución.
5. **Ciclo de CPU 6-Tempo de Chegada 5:** O algoritmo **SRTF** determina que o seguinte en entrar na CPU é o que menos duración ocupe na mesma, entón entrará P3 ata que remate.
6. **Ciclo de CPU 10-Tempo de Chegada 9:** Entra P2 ata que remate.