

temps d'attente = temps de Séjour - BT

t=13 t=15

~~D~~  
E

~~E~~



t=0  
~~A~~

t=1  
B

t=3  
~~B~~

t=4  
C

t=6  
C  
D

t=7  
C  
D  
E

t=9  
~~D~~  
E

File d'attente FIFO

# Exercice 1

Algorithme non préemptif: FCFS: First Come First Served

Processus	temps d'exécution Burst Time (BT)	Temps d'arrivée	temps de notation temps de Séjour	temps d'attente	Début
A	3	0	3-0=3	3-3=0	
B	6	1	9-1=8	8-6=2	5/16
C	4	4	13-4=9	9-4=5	
D	2	6	15-6=9	9-2=7	
E	1	7	16-7=9	9-1=8	

temps de Séjour

temps de notation

$\overline{TA}$

= Turn Around Time

= temps de  
temps d'a

## Exercice 2: SJF

Processus	temps d'exécution BT	temps d'arrivée	temps de rotation TAT = temps de séjour	temps d'attente	Débit nb de processus / temps total exécuté
A	3	0	$3 - 0 = 3$	$3 - 3 = 0$	5/16
B	6	1	$9 - 1 = 8$	$8 - 6 = 2$	
C	4	4	$16 - 4 = 12$	$12 - 4 = 8$	
D	2	6	$12 - 6 = 6$	$6 - 2 = 4$	
E	1	7	$10 - 7 = 3$	$3 - 1 = 2$	

## Les algorithmes d'ordonnement

A	B	E	D	C
---	---	---	---	---

t=0	t=1	t=3	t=4	t=6	t=7	t=9	t=10	t=12
A	B	B	C	D(2)	E(1)	<del>E(1)</del>	<del>D(2)</del>	<del>C(4)</del>
				C(4)	D(2)	D(2)	C(4)	
					C(4)	C(4)		

temps de séjour = temps de terminaison - temps d'arrivée  
 temps d'attente = temps de séjour - temps d'exécution (BT)

## Les algorithmes d'ordonnement

Algorithme: RR (Round Robin) = en tourniquet  $\Rightarrow$  la file d'attente est considérée circulaire, on utilise le principe du quantum.

- Si le quantum est faible  $\Rightarrow$  le nombre de commutation de contexte sera très grand  $\Rightarrow$  une dégradation de la performance du processeur.
- Si le quantum est grand  $\Rightarrow$  on revient vers l'algorithme FCFS, problème de la famine.

Exercice 3: Quantum = 3 ms

TAI  
temps de séjour

nb de  
processus / temps total  
écoulé

Processus	temps d'exécution BT	temps d'arrivée	temps de rotation	temps d'attente	débit
A	5	0	$15 - 0 = 15$	$15 - 5 = 10$	5/19
B	3	1	$6 - 1 = 5$	$5 - 3 = 2$	
C	6	3	$18 - 3 = 15$	$15 - 6 = 9$	
D	1	5	$10 - 5 = 5$	$5 - 1 = 4$	
E	4	6	$19 - 6 = 13$	$13 - 4 = 9$	

$$\frac{19}{5} = \frac{15 + 5 + 15 + 5 + 13}{5} = \frac{53}{5}$$

Les algorithmes d'ordonnement

A	B	C	D	E	A	C	E	
0	3	6	9	10	13	15	18	19

$t=0$  A  
 $t=1$  B  
 $t=3$  C  
 $t=5$  D  
 $t=6$  E  
 $t=9$  A(2)  
 $t=10$  C(3)  
 $t=13$  E(1)  
 $t=15$  A(3)  
 $t=18$  C(3)  
 $t=19$  E(1)

### Exercice 3:

Processus	temps d'exécution ET	temps d'arrivée	Priorité	temps de séjour	temps d'attente	Etat
A	5	0	4	$7 - 0 = 7$	$7 - 5 = 2$	4/15
B	4	2	2	$15 - 2 = 13$	$13 - 4 = 9$	
C	2	2	6	$4 - 2 = 2$	$2 - 2 = 0$	
D	4	4	3	$11 - 4 = 7$	$7 - 4 = 3$	

Hypothèse: le processus ayant la priorité la (+) élevée est considéré (+) prioritaire  
 $\text{prior}(C) > \text{prior}(A) > \text{prior}(D) > \text{prior}(B)$

### Les algorithmes d'ordonnement

Algorithme avec priorité: On considère la priorité attribuée aux processus.

A	C	A	D	B
t=0	t=2	t=4	t=7	t=11
A	C	A(3)	D	B
	A(3)	D	B	
	B	B		

Algo  
SJF avec  
priorité

Exercice 5. Hypothèse: celui qui détient la priorité la (+) faible est le (+) prioritaire.

Processus	temps d'exécution BT	temps d'arrivée	Priorité	temps de Séjour	temps d'attente	Débit
A	3	0	4			
B	6	1	2			
C	4	4	1			
D	2	4	1			
E	1	7	3			

Algo  
SJF avec  
priorité

Exercice 5: Hypothèse: Ce lui qui détient la priorité la plus faible est le + prioritaire.

Processus	temps d'exécution	temps d'arrivée	Priorité	temps de séjour	temps d'attente	Retard
A	3	0	4	$16 - 0 = 16$	$16 - 3 = 13$	5/16
B	6	1	2	$13 - 1 = 12$	$12 - 6 = 6$	
C	4	4	1	$10 - 4 = 6$	$6 - 4 = 2$	
D	2	4	1	$5 - 4 = 1$	$2 - 1 = 1$	
E	1	7	3	$14 - 7 = 7$	$7 - 1 = 6$	

### Les algorithmes d'ordonnement

A	B	D	C	B	E	A
0	1	4	6	10	13	14

$t=0$  A  
 $t=1$  ~~B~~ A(2)  
 $t=4$  ~~D~~ C B(3) A(2)  
 $t=6$  ~~C~~ B(2) A(2)  
 $t=7$  B(3) E A(2)  
 $t=10$  ~~B(3)~~ E A(2)  
 $t=13$  ~~E~~ A(2)  
 $t=14$  ~~A(2)~~