



1)

SJF																							
Job	Llegada	CPU	IO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	TR
1	0	8	R1, 2, 2	>1	2	R1	R1																18
2	1	6	R2, 2, 2		>	1	2	R2	R2			3	4	5	6<								11
3	2	4				>		1	2	3	4<												6
			QUEUE	1-	2-	3-	1-	2-															11,6
																							5,6
FCFS																							
Job	Llegada	CPU	IO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	TR
1	0	8	R1, 2, 2	>1	2	R1	R1																14
2	1	6	R2, 2, 2		>	1	2	R2	R2			3	4	5	6	7	8<						17
3	2	4				>		1	2	3	4<							3	4	5	6<		6
			QUEUE	1-	2-	3-	1-	2-															12,3
																							6,3

a y b) El SJF tiene mejor TPR y TPE debido a que favorece a los procesos I/O bound ya que estos pasan mayor parte del tiempo esperando operaciones de E/S que en ejecución.
c) Si, el SJF puede causar inanición en los procesos mas largos ya que favorece a los que tienen menores rafagas de CPU.

2)

Marcos/Paginas	1	2	3	4	1	2	5	1	2	3	4	5	TOTAL
1	1	1	1	4	4	4	5	5	5	3	3	3	10
2		2	2	2	1	1	1	1*	1*	1	4	4	
3			3	3	3	2	2	2	2*	2	2	5	
PF	X	X	X	X	X	X	X			X	X	X	
COLA	1-	2-	3-	4-	1*-	2*-	5-	1-	2-	3	4	5	

3)

a) tamaño maximo del proceso: $2^{32} * 2 = 8589934592$ bytes

b) cuantas paginas puede tener un proceso: $8589934592/512 = 16777216$

c) tamaño maximo que podria alcanzar: $16777216 * 4 = 67108864$

4)

24 marcos disponibles

total paginas= 38

proceso 1 -> $8 * 24 / 38 = 5$

proceso 2 -> $16 * 24 / 38 = 10$

proceso 3 -> $14 * 24 / 38 = 9$

Suponga un sistema que el tamaño de página utilizado es de 512 bytes.

a) ¿Cuál sería el tamaño máximo de un proceso?

b) ¿Cuántas páginas puede tener un proceso?

c) Si la cola entrada en la tabla de páginas es de 4 bytes, ¿cuál sería el tamaño máximo que podría alcanzar la memoria?

5. Suponga un SO con administración de memoria virtual por medio de paginación por demanda. Si la cantidad de marcos disponibles para los procesos es 24, indique cuántos marcos se le asignaron a cada proceso si la cola de asignación fija con reparto proporcional:

Proceso	Páginas del Proc.	Marcos Asignados
1	8	
2	16	
3	14	

6. Suponga un Head con movimiento en 300 pistas (numerados de 0 a 299), que está en la pista 130 tendiendo un requerimiento y anteriormente atendió un requerimiento en la pista 140. Si la cola de requerimientos es: 30, 110, 142, 267, 91, 210, 194PF, 20. Después de 20 movimientos se incorporan los requerimientos de las pistas 95PF, 20, 212. Realice los diagramas e indique la cantidad total de movimientos de head para satisfacer estos requerimientos de acuerdo a los siguientes algoritmos de scheduling de discos:

a) SCAN

b) SSTF

7. Se tiene una unidad de disco con 6 platos, con 2 caras útiles, 1500 pistas por cara y 700 sectores por pista de 256 bytes cada uno. Si el disco gira a 12600 RPM, tiene un tiempo de posicionamiento (seek) de 2 milisegundos y una velocidad de transferencia de 15 Mib/seg (Mebibits por Segundo), calcular e indicar:

a) La capacidad total del disco

b) ¿Cuántas caras en total ocupará un archivo de 513 Mebibytes almacenado de manera contigua a partir del primer sector de la primera pista de una cara determinada

c) ¿Cuántos milisegundos se tardarían en transferir un archivo almacenado de manera contigua de 4500 sectores?

Solo para alumnos que no respondieron las autoevaluaciones o no se inscribieron al parcial
Indique Verdadero o Falso y justifique su respuesta

a) El algoritmo de planificación de CPU First Come First Served, favorece a los procesos ligados a Entrada Salida

b) El algoritmo de planificación de CPU Shortest Job First, al igual que el algoritmo Round Robin puede provocar inanición

c) El algoritmo de planificación de CPU Virtual Round Robin Favorece a los procesos con carga de Entrada Salida

6)

- a) $6 * 2 * 1500 * 700 * 256 = 3225600000$ bytes
b) capacidad de almacenamiento de una cara
 $1500 * 700 * 256 = 268800000$
 $513 \text{ Mib} = (2^{20} * 513) = 537919488$ bytes

$$537919488 / 268800000 = 2,00118 = 3 \text{ caras}$$

c) latencia:

Asumo que es el tiempo que tarda en dar $\frac{1}{2}$ vuelta

12600 vueltas — 1' = 60 segundos = 60000 milisegundos ($60 * 1000$)

0.5 vueltas — x

$$x = 2.3809 \text{ milisegundos}$$

transferencia

$$15 * 8 = 120$$

$$120 * 2^{20} = 125829120$$

125829120 bytes — 1 segundo = 1000 milisegundos

256 bytes — x

$$x = 0.002043274 \text{ milisegundos}$$

datos obtenidos:

seek time: 2 milisegundos

latency time: 2.3809 milisegundos

tiempo de transferencia por bloque: 0.002043274 milisegundos

almacenamiento secuencial:

$$2 + 2.3809 + 0.002043274 * 4500 = 13.575633 \text{ ms}$$

almacenamiento aleatorio:

$$(2 + 2.3809 + 0.002043274) * 4500 = 19723.24473 \text{ ms}$$