

1. Considere la siguiente referencia de página utilizando tres marcos que inicialmente están vacíos. Encuentre las fallas de página usando los algoritmos vistos en clase, donde la secuencia de referencia de página: 0, 2, 1, 6, 4, 0, 1, 0, 3, 1, 2, 1 ¿qué algoritmo tiene mejor rendimiento?

a) LIFO – TF: 0.75

Referencia	0	2	1	6	4	0	1	0	3	1	2	1	Total fallos
	0	0	0	6	6	6	1	1	1	1	1	1	
		2	2	2	4	4	4	4	3	3	3	3	
			1	1	1	0	0	0	0	0	2	2	
Fallo	F	F	F	F	F	F	F		F		F		9

a) LRU – TF: 0.75

Referencia	0	2	1	6	4	0	1	0	3	1	2	1	Total fallos
	0	0	0	6	6	6	1	1	1	1	1	1	
		2	2	2	4	4	4	4	3	3	3	3	
			1	1	1	0	0	0	0	0	2	2	
Fallo	F	F	F	F	F	F	F		F		F		9

b) OPT – TF: 0.58

Referencia	0	2	1	6	4	0	1	0	3	1	2	1	Total fallos
	0	0	0	0	0	0	0	0	3	3	2	2	
		2	2	6	4	4	4	4	4	4	4	4	
			1	1	1	1	1	1	1	1	1	1	
Fallo	F	F	F	F	F				F		F		7

c) RELOJ – TF: 0.75

R																												T																			
0	Bit de uso Apuntador			2	Bit de uso Apuntador			1	Bit de uso Apuntador			6	Bit de uso Apuntador			4	Bit de uso Apuntador			0	Bit de uso Apuntador			1	Bit de uso Apuntador			0	Bit de uso Apuntador			3	Bit de uso Apuntador			1	Bit de uso Apuntador			2	Bit de uso Apuntador			1	Bit de uso Apuntador		
	0	1	*	0	1	*	0	1	*	6	1		6	1		6	1	*	1	1		1	1	*	1	1		1	1		1	1		1	0	*	1	1									
				2	1		2	1		2	0	*	4	1		4	1		4	0	*	4	0		3	1		3	1		3	0		3	0	*											
							1	1		1	0		1	0	*	0	1		0	0		0	1		0	1	*	0	1	*	2	1		2	1												
F	F			F			F			F			F			F			F						F							F								9							

2. Sea la siguiente secuencia de números de página referenciados: 4, 7, 6, 1, 7, 6, 1, 2, 7, 2, 7, 1. Calcule el número de fallas de página que se producen utilizando los algoritmos vistos en clase y considerando que el número de marcos de página de que disfruta nuestro proceso es de 4. ¿qué algoritmo tiene mejor rendimiento?

a) LIFO – TF: 0.41

Referencia	4	7	6	1	7	6	1	2	7	2	7	1	Total fallos
	4	4	4	4	4	4	4	2	2	2	2	2	
		7	7	7	7	7	7	7	7	7	7	7	
			6	6	6	6	6	6	6	6	6	6	
				1	1	1	1	1	1	1	1	1	
Fallo	F	F	F	F				F					5

b) LRU – TF: 0.41

Referencia	4	7	6	1	7	6	1	2	7	2	7	1	Total fallos
	4	4	4	4	4	4	4	2	2	2	2	2	
		7	7	7	7	7	7	7	7	7	7	7	
			6	6	6	6	6	6	6	6	6	6	
				1	1	1	1	1	1	1	1	1	
Fallo	F	F	F	F				F					5

c) OPT – TF: 0.41

Referencia	4	7	6	1	7	6	1	2	7	2	7	1	Total fallos
	4	4	4	4	4	4	4	2	2	2	2	2	
		7	7	7	7	7	7	7	7	7	7	7	
			6	6	6	6	6	6	6	6	6	6	
				1	1	1	1	1	1	1	1	1	
Fallo	F	F	F	F				F					5

d) RELOJ – TF: 0.41

R	4	7	6	1	7	6	1	2	7	2	7	1	T
	Bit de uso	Apuntador	Bit de uso	Apuntador	Bit de uso	Apuntador	Bit de uso	Apuntador	Bit de uso	Apuntador	Bit de uso	Apuntador	
	4	1 *	4	1 *	4	1 *	4	1 *	4	1 *	2	1	
			7	1	7	1	7	1	7	1	7	1	
					6	1	6	1	6	1	6	1	
							1	1	1	1	1	1	
F	F		F		F				F				5

3. Suponiendo un tamaño de página de 1 KB, ¿cuáles son los números de página y los desplazamientos para las siguientes referencias de direcciones (proporcionadas como números decimales)? a. 21105 b. 162450 c. 121753 d. 16947315 e. 25273187

$$2^n = 1000 \rightarrow n = 10 \text{ bits} \quad 2^{10} = 1024$$

a) 21105

$$p = 21105 / 1024 = 20$$

$$d = 21105 - (1024 * 20) = 625$$

[ 20 | 625 ]

b) 162450

$$p = 162450 / 1024 = 158$$

$$d = 162450 - (1024 * 158) = 658$$

[ 158 | 658 ]

c) 121753

$$p = 121753 / 1024 = 118$$

$$d = 121753 - (1024 * 118) = 921$$

[ 118 | 921 ]

d) 16947315

$$p = 16947315 / 1024 = 16550$$

$$d = 16947315 - (1024 * 16550) = 115$$

[ 16550 | 115 ]

e) 25273187

$$p = 25273187 / 1024 = 24680$$

$$d = 25273187 - (1024 * 24680) = 867$$

[ 24680 | 867 ]

4. Considerando un sistema con MF de 64Kb, se apunta a la dirección física 27421 como respuesta a una DL emitida por el proceso P de tamaño 15635 bytes. Obtener la(s) dirección(es) lógica emitida(s) suponiendo una memoria paginada, con páginas de 4Kb

$$2^n = 4000 \rightarrow n = 12 \quad 2^{12} = 4096$$

$$m = 64/4 = 16$$

$$p = 27421 / 4096 = 6 \rightarrow \text{en 4 bits (m-n)} \quad 0110$$

$$d = 27421 - (4096 * 6) = 2845 \rightarrow \text{en 12 bits (n)} \quad 101100011101$$

DL: [ 0110 | 101100011101 ] o [ 4 | 12 ]

$$\# \text{ páginas Proceso P} = 15635/4096 = 3 \quad (6,7,8)$$

[6 | 2845] [7 | 6941] [8 | 11037]

5. Considere un sistema con una DL de 32 bits y tamaño de página de 8Kb, además el tamaño de MF es de 1GB. Obtener:

$$2^n = 8000 \rightarrow n = 13 \quad 2^{13} = 8192$$

$$2^m = 1000000000 \rightarrow m = 30 \quad 2^{30} = 1073741824$$

$$m - n = 17$$

a) El formato de la dirección virtual

Bits 31-13: # de página - Bits 12-0: desplazamiento

$$[ \quad m - n = 17 \quad | \quad n = 13 \quad ]$$

b) Máximo número de páginas

$$2^{30} / 2^{13} = 2^{17} = 131072$$

6. En un sistema de memoria paginada, la máxima capacidad de DV es de 4GB y la MP es de 256 MB. El desplazamiento dentro de una página y de un marco se realiza con 12 bits.

$$2^{m1} = 4000000000 \rightarrow m1 = 32 \quad 2^{32} = 4294967296$$

$$2^{m2} = 256000000 \rightarrow m2 = 28 \quad 2^{28} = 268435456$$

$$m1 - n = 12 \rightarrow n1 = 32 - 12 = 20$$

$$m2 - n = 12 \rightarrow n2 = 28 - 12 = 16$$

a. ¿Cuál es el tamaño de una página? ¿Y de un marco?

$$\text{Tamaño de página: } 2^{20} = 1048576 \text{ (1MB)}$$

$$\text{Tamaño de marco: } 2^{16} = 65536 \text{ (65KB)}$$

b. Define el formato de una dirección de MV y de MP

$$\text{MV: } [ \quad 20 \text{ bits} \quad | \quad 12 \text{ bits} \quad ]$$

$$\text{MP: } [ \quad 16 \text{ bits} \quad | \quad 12 \text{ bits} \quad ]$$

7. Considere un espacio de DL de 2048 páginas con un tamaño de página de 4 KB, asignado a una MF de 512 marcos.

$$2^n = 4000 \rightarrow n = 12 \quad 2^{12} = 4096$$

$$2^{m_1-n} = 2048 \rightarrow \log_2(2048) = (m_1-n) = 11$$

$$2^{m_2-n} = 512 \rightarrow \log_2(512) = (m_2-n) = 9$$

a. ¿Cuántos bits se requieren en la dirección lógica?

$$m_1 = 12 + 11 = 23 \text{ bits}$$

b. ¿Cuántos bits se requieren en la dirección física?

$$m_2 = 12 + 9 = 21 \text{ bits}$$

8. Teniendo un procesador con DV de 32 bits y páginas de 2 KB. Defina el formato de la DV, así como el número máximo de páginas que puede tener.

$$m = 32$$

$$2^{m-n} = 2000 \rightarrow (m-n) = 11 \quad 2^{11} = 2048$$

$$n = 32 - 11 = 21$$

DV: [ 11 bits | 21 bits ]

$$\# \text{ máximo de paginas} = 2^{11} = 2048$$

9. Considerando 18 bits en MV, con páginas de 4 KB y MF de 512 KB:

$$m = 18$$

$$2^n = 4000 \rightarrow n = 12 \quad 2^{12} = 4096$$

a. Indique los campos y el número de bits, para indicar la DV

DV: [ 6 bits | 12 bits ]

b. ¿Cuál es el número máximo de entradas de la tabla de páginas?

$$2^6 = 64$$

c. ¿Cuántos marcos de página tiene la MP?

$$512/4 = 128$$

10. Suponga que un proceso emite la dirección lógica (2,18004) utilizando un modelo de gestión de memoria basado en segmentación y el espacio de memoria física es de 64K bytes.

$$2^m = 64000 \rightarrow m = 16 \quad 2^{16} = 65536$$

$$\# \text{ de segmentos} = 3 = 16 - n \rightarrow n = 13$$

$$DL < DF$$

a. ¿A qué direcciones físicas de las siguientes (11084, 33270 y 22112), sería posible traducir dicha dirección lógica?

A las de 33270 y 22112 ya que 18004 es menor a ambas, si fuera mayor no sería posible hacer el desplazamiento.

b. ¿Cuál sería el resultado de traducir la dirección lógica (0,65536) en dicho sistema?

Un fallo ya que el tamaño del desplazamiento es mayor que el de la memoria física.

11. Considere un programa de usuario de dirección lógica de tamaño 6 páginas y el tamaño de página es de 4 bytes. La dirección física contiene 300 marcos. El programa de usuario consta de 22 instrucciones a, b, c, . . . t, u, v. Cada instrucción ocupa 1 byte. Suponga que en ese momento los marcos libres son 7, 26, 52, 20, 55, 6, 18, 21, 70 y 90. Encuentre lo siguiente:

a. Dibujar los mapas lógicos y físicos y las tablas de páginas

Página	Instrucciones
0	a, b, c, d
1	e, f, g, h
2	i, j, k, l
3	m, n, o, p
4	q, r, s, t
5	u, v

1 = ocupado - 0 = libre

Marco	Bit de validez
0	1
1	1
...	1

6	0
7	0
18	0
20	0
21	0
26	0
52	0
55	0
70	0
90	0
...	1
300	1

Página	Marco
0	-
1	-
2	-
3	-
4	-
5	-

- Aún no se le ha asignado un marco a cada página

b. Asignar cada página en el marco correspondiente

Página	Marco
0	7
1	26
2	52
3	20
4	55
5	6



Quedan 4 marcos libres: 18, 21, 70 y 90

c. Encuentra las direcciones físicas de las instrucciones m, d, v, r

bytes por marco = # de páginas \* bytes por página =  $4 * 4 = 16$

m -> byte 12

página 3 -> marco 20

DF =  $20 * 16 + 12 = 332$

d -> byte 3

página 0 -> marco 7

DF =  $7 * 16 + 3 = 115$

v -> byte 21

página 5 -> marco 6

DF =  $6 * 16 + 21 = 117$

r -> byte 17

página 4 -> marco 55

DF =  $55 * 16 + 17 = 897$

12. Considere un programa que consta de cinco segmentos: S0 = 600, S1 = 14 KB, S2 = 100 KB, S3 = 580 KB y S4 = 96 KB. Supongamos que, en ese momento, las particiones de memoria de espacio libre disponibles son 1200–1805, 50–150, 220-234 y 2500-3180. Encuentre lo siguiente:

a. Dibuje mapas lógicos a físicos y tabla de segmentos

Segmento	Tamaño
S0	600 KB
S1	14 KB
S2	100 KB
S3	580 KB
S4	96 KB

1 = ocupado - 0 = libre

Espacio de memoria	Bit de validez
0-49	1
50-150	0
151-219	1
220-234	0
235-1199	1
1200-1805	0
1806-2499	1
2500-3180	0

Segmento	Espacio de memoria
S0	-
S1	-
S2	-
S3	-
S4	-

- Aún no se le ha asignado un espacio a cada segmento

b. Asignar espacio para cada segmento en la memoria

Segmento	Espacio de memoria
S0	1200-1800 (5 libres)
S1	220-234 (0 libres)
S2	2500-2600 (580 libres)
S3	2600-3180 (0 libres)
S4	50-146 (4 libres)

c. ¿Cuáles son las direcciones en la memoria física para las siguientes direcciones lógicas:

i. 0.580      ->       $580 + 1200 = 1780$

ii. 1.17      ->       $17 + 220 = 237 > 234$  (Fallo)

- iii. 2.66       $66 + 2500 = 2566$   
 iv. 3.82       $82 + 2600 = 2682$   
 v. 4.20       $20 + 50 = 70$

13. Dadas particiones de memoria de 150K, 750K, 300K, 400K y 800K (en orden): ¿Cómo se colocarían procesos de 216K, 428K, 212K, y 246K (en orden) en memoria utilizando los algoritmos de First Fit, Best Fit y Worst Fit? ¿Cuál algoritmo aprovecha la memoria de forma más eficiente?

**First Fit**

Memoria libre	150 *	750	300	400	800
Proceso		<b>216</b>			

Memoria libre	150	534 *	300	400	800
Proceso		216 <b>428</b>			

Memoria libre	150	106	300 *	400	800
Proceso		216 428	<b>212</b>		

Memoria libre	150	106	88	400 *	800
Proceso		216 428	212	<b>246</b>	

Memoria libre	150	106	300	154	800 *
Proceso		216 428	212	246	

Pasos =  $2 + 1 + 1 + 1 = 5$

### Best Fit

Memoria libre	150	300	400	750	800
Proceso		<b>216</b>			

Memoria libre	84	300	400	750	800
Proceso	216			<b>428</b>	

Memoria libre	84	300	322	400	800
Proceso	216	<b>212</b>	428		

Memoria libre	84	88	322	400	800
Proceso	216	212	428 <b>246</b>		

Memoria libre	76	84	88	400	800
Proceso	428 246	216	212		

Pasos = 2 + 4 + 2 + 3 = 11

### Worst Fit

Memoria libre	800	750	400	300	150
Proceso	<b>216</b>				

Memoria libre	750	584	400	300	150
Proceso	<b>428</b>	216			

Memoria libre	584	400	322	300	150
Proceso	216 <b>212</b>		428		

Memoria libre	400	372	322	300	150
Proceso	<b>246</b>	216 212	428		

Memoria libre	372	322	300	154	150
Proceso	216 212	428		246	

Pasos = 1 + 1 + 1 + 1 = 4

Comparando los bloques finales de memoria libre:

Algoritmo						Pasos
First Fit	150	106	300	154	800	5
Best Fit	76	84	88	400	800	11
Worst Fit	372	322	300	154	150	4

La eficiencia del First Fit depende completamente del orden inicial de las particiones de memoria, en este caso, en cuestión de tiempo es muy rápido y hace un buen trabajo minimizando un poco la fragmentación.

En el caso del Worst Fit, este es el más rápido, siempre con el tiempo de búsqueda mínimo, pero con la consecuencia de correr el riesgo de la fragmentación.

Por último, el Best Fit, es el que más evita la fragmentación, pero con la consecuencia de que forzosamente será el que más pasos de búsqueda tendrá que realizar, sacrificando tiempo por la seguridad de la memoria, por lo que es el algoritmo que mejor la aprovecha.