



Universidad de
Huelva



Escuela Técnica
Superior de
Ingeniería
ETSI
Grado en Informática

Diseño y Estructura de los Sistemas Operativos

Problemas de Gestión de Memoria Soluciones

Miguel Ángel Vélez Vélez

José Ponce González

Huelva, Diciembre de 2020

Diseño y Estructura de los Sistemas Operativos

Problemas de Gestión de Memoria Soluciones

Autores:

MIGUEL ÁNGEL VÉLEZ VÉLEZ

JOSÉ PONCE GONZÁLEZ

ISBN: XXXXXXXXX

Usted es libre de:



copiar, distribuir y comunicar públicamente la obra



Hacer obras derivadas

Bajo las condiciones siguientes:



Reconocimiento. Debe reconocer los créditos de la obra de la manera especificada por el autor o el licenciador (pero no de una manera que sugiera que tiene su apoyo o apoyan el uso que hace de su obra).



No comercial. No puede utilizar esta obra para fines comerciales.

- Al reutilizar o distribuir la obra, tiene que dejar bien claro los términos de la licencia de esta obra.
- alguna de estas condiciones puede no aplicarse si se obtiene el permiso del titular de los derechos de autor
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

Esto es un resumen del texto legal (la licencia completa) disponible en <http://creativecommons.org/licenses/by-nc/2.5/es/legalcode.es>

TABLA DE CONTENIDOS

1. Seven Quest.....	2
1.1. Solución.....	3
2. El Quisquilloso.....	5
2.1. Solución.....	6
3. Tabla grande, ande o no ande.....	8
3.1. Solución.....	9
4. Corona Memo.....	12
4.1. Solución.....	13

1. Seven Quest

En un sistema de segmentación paginada con memoria virtual las direcciones lógicas son de 45 bits. El ancho de la memoria es de 16 bits y disponemos de 64 Mbytes de memoria RAM. Un proceso no puede tener más de 64 segmentos.

Si la memoria RAM está descompuesta en 8192 tramas y se usa como algoritmo de reemplazo el algoritmo de reloj:

1. Diseñe la MMU de este sistema rellenando el tamaño de todos los campos
2. ¿Qué tamaño en bytes tiene la tabla de segmentos?
3. ¿Qué tamaño en bytes tienen las páginas?
4. ¿Cuál es el principal problema que se presenta en esta MMU? ¿Qué dos soluciones se han visto en clase para tratar de solucionar este problema?

Se muestran a continuación las Tablas de segmentos y de páginas de un proceso actualmente cargado en memoria cuyo STLR vale 0 y su STBR vale 0:

	@Base	Límite
0	1500	16383
4		
8		
12		
16		
20		

	t	V	P	R
1500	1705	1	0	0
1501	8	1	1	0
1502	1101	1	1	0
1503	3	1	0	0
1504	2	0	0	0
1505	0	0	0	0

Conteste a las siguientes preguntas:

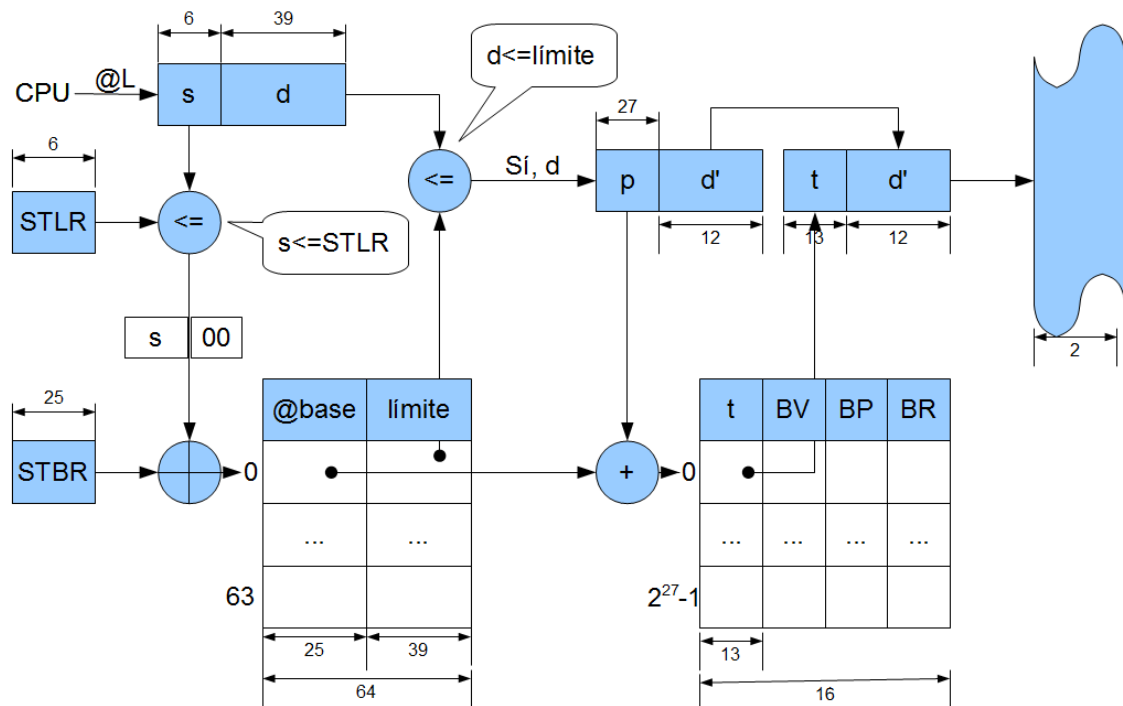
5. ¿Qué tamaño en bytes ocupa en total el proceso?
6. ¿Cuántos segmentos tiene este proceso?
7. ¿Cuántas páginas tiene este proceso?
8. ¿Cuántas páginas tiene cargadas en memoria este proceso?
9. Si llega la dirección lógica 1, ¿qué ocurriría?

1.1. Solución

En un sistema de segmentación paginada con memoria virtual las direcciones lógicas son de 45 bits. El ancho de la memoria es de 16 bits y disponemos de 64 Mbytes de memoria RAM. Un proceso no puede tener más de 64 segmentos.

Si la memoria RAM está descompuesta en 8192 tramas y se usa como algoritmo de reemplazo el algoritmo de reloj:

1. Diseñe la MMU de este sistema rellenando el tamaño de todos los campos



2. ¿Qué tamaño en bytes tiene la tabla de segmentos?

512 bytes

3. ¿Qué tamaño en bytes tienen las páginas?

8192 bytes

4. ¿Cuál es el principal problema que se presenta en esta MMU? ¿Qué dos soluciones se han visto en clase para tratar de solucionar este problema?

La tabla de páginas es enorme y no cabe en memoria. Hay que disminuir el tamaño de la tabla de páginas. Tabla de páginas multinivel y Tabla de páginas invertida.

Se muestran a continuación las Tablas de segmentos y de páginas de un proceso actualmente cargado en memoria cuyo STLR vale 0 y su STBR vale 0:

	@Base	Límite
0	1500	16383
4		
8		
12		
16		
20		

	t	V	P	R
1500	1705	1	0	0
1501	8	1	1	0
1502	1101	1	1	0
1503	3	1	0	0
1504	2	0	0	0
1505	0	0	0	0

Conteste a las siguientes preguntas:

5. ¿Qué tamaño en bytes ocupa en total el proceso?

32768 bytes (32 Kbytes)

6. ¿Cuántos segmentos tiene este proceso?

1 segmento

7. ¿Cuántas páginas tiene este proceso?

4 páginas

8. ¿Cuántas páginas tiene cargadas en memoria este proceso?

2 páginas

9. Si llega la dirección lógica 1, ¿qué ocurriría?

Se produce un fallo de página ya que la página 0 del segmento 0 tiene su bit de presencia a 0 y no se encuentra cargada en memoria.

2. El Quisquilloso

En un sistema gestionado mediante memoria virtual paginada los procesos no pueden tener más de 16384 páginas. La memoria física es de 32 Mbytes y tiene 8192 tramas. El ancho de la memoria física es de 2 bytes.

1. Dibuje el esquema completo de la MMU del sistema, suponiendo que la tabla de páginas está en memoria, no tiene bit de validez y se usa un algoritmo de reemplazo NRU.
 - (a) ¿Qué tamaño total en bytes tiene una tabla de páginas? Mencione dos técnicas para evitar que la tabla de páginas sea tan grande.
 - (b) ¿Cuál sería el tamaño más grande de proceso que se podría ejecutar en este sistema?
 - (c) ¿Cuántas posiciones de memoria ocupa una página?

Inicialmente a cada proceso que llega se le asignan 2 tramas de memoria. Este número de tramas variará según un Conjunto de Trabajo de $S=5$. Todo proceso cuando llega carga sus dos primeras páginas en las dos primeras tramas que encuentra libre. El S.O. está cargado en las tramas 0, 1 y 2 y llegan los procesos P2 y P1 por ese orden.

Si se produce la siguiente secuencia de referencias:

Proceso	P2	P2	P2	P2	P2	P2	P1	P1	P1	P1	P1	P1	P1	P2	P2	P2	P2	P2	P1	P1	P1	P1
Página	0	4	0	2	3	4	1	2	1	0	2	1	0	0	1	3	4	2	3	0	2	3

2. Dibuje las tablas de páginas de los dos procesos al final de la secuencia de referencias, con todos los campos posibles rellenos. Se sabe que el RBTP del proceso P2 vale 35100 y el de P1 15600.
3. ¿Cuántos fallos de página, no fallos y reemplazos ha sufrido cada proceso en su ejecución?

NOTAS:

- El bit de referenciada se pondrá a cero cada 5 referencias.
- En negrita las referencias que suponen modificación de la página.
- **Si una página abandona la memoria y su bit de modificada está a 1, cuando vuelva a cargarse ese bit estará a 0 (a no ser que esa referencia suponga modificación)**

		**		**		**		*		*		**	
P2		0	4	0	2	3	4	0	1	3	4	2	
T3	0	0	0	0	0	0	0	0	0	0	0	0	
T4	1	1	4	4	2	3	3	3	3	3	3	3	
T7							4	4	4	4	4	4	
T8									1	1	1	2	
WS(5,5)=4							WS(10,5)=4						

$$WS(10,5)=4$$

Proceso P1		T.P.				Proceso P2		T.P.			
15600	9	1	0	1	35100	3	1	0	1		
15601	6	1	0	1	35101		0	0	0		
15602	5	1	0	0	35102	8	1	1	1		
15603	10	1	1	1	35103	4	1	0	1		
15604		0	0	0	35104	7	1	0	1		
:	:	:	:	:	:	:	:	:	:		
:	:	:	:	:	:	:	:	:	:		
31983					51483						

3. ¿Cuántos fallos de página, no fallos y reemplazos ha sufrido cada proceso en su ejecución?

7 fallos, 5 reemplazos, 4 no fallos

6 fallos, 4 reemplazos, 5 no fallos

3. Tabla grande, ande o no ande

Tenemos una MMU gestionada mediante segmentación paginada con memoria virtual. La memoria física, cuyo ancho hay que calcular adecuadamente, es de 512 MBytes particionada en 8192 tramas y el desplazamiento en cada una de ellas se realiza mediante 15 bits. Las tablas de páginas tienen 2097152 entradas y no tienen bit de validez. En dicho sistema sólo tendremos procesos con un máximo de 8 segmentos y como algoritmo de reemplazo de páginas usará el NRU. . En base a esto:

1. Diseñe la MMU rellenando adecuadamente todos los campos
2. ¿Cuál es el tamaño en KBytes de una página?
3. ¿Cuánto ocupa una tabla de páginas? ¿y la tabla de segmentos?
4. Las tablas de página parecen demasiado grandes ¿sabría indicar alguna técnica para mejorar este aspecto?

Teniendo en cuenta la MMU diseñada en el apartado anterior, a nuestro sistema llega un proceso A compuesto de 2 segmentos donde el tamaño de cada uno de ellos es 192 y 300 KBytes respectivamente. Sabemos además que el STBR vale 65536 y que tiene las primeras cuatro páginas cargadas en memoria en las tramas 131, 132, 133 y 134 respectivamente, no habiendo sido ni referenciada ni modificada en ningún momento ninguna de ellas. Se han perdido datos y lo único que queda legible, es lo representado a continuación:

STLR		@Base	#		t	#	...	#		t	#	...	#
#	65536	98304	#	#	#	#	...	#	2195456	#	#	...	#
	#	#	#	#	#	#	...	#	#	#	#	...	#
	#		#	#	#	#	...	#	#	#	#	...	#
STBR	#
65536	#			2195455	#	#	...	#	#	#	#	...	#
		T. Segmentos				T. Página Segmento 0					T. Página Segmento 1		
					PTLR-S0	#				PTLR-S1	#		

5. Reescriba todos los datos necesarios de las tablas y registros límites que sean necesarios, para restablecer el estado inicial descrito en el párrafo anterior.

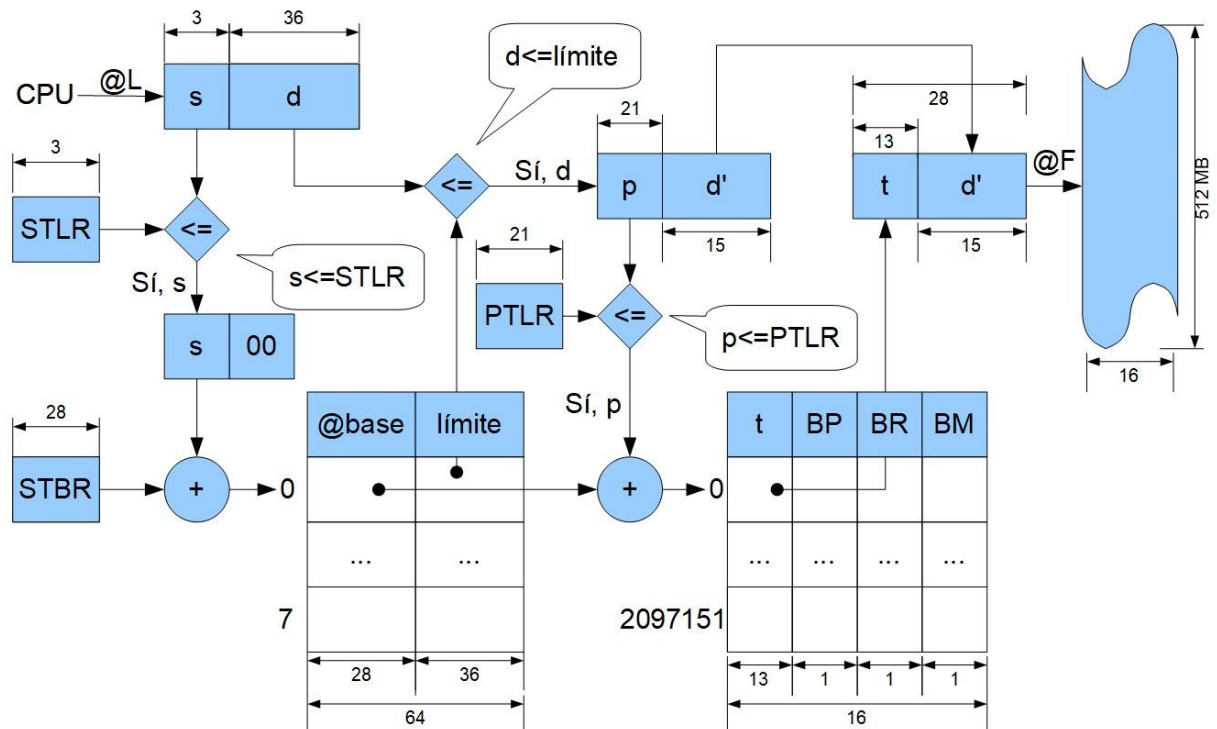
Si tras reestructurar dicha información este proceso, que tiene asignada 4 tramas de inicio, genera la siguiente secuencia de referencias a páginas, teniendo en cuenta los datos del apartado anterior:

S1P0	S0P0	S0P2	S1P3	S1P2	S1P1	S0P0	S1P1	S0P0	S1P4
------	-------------	-------------	------	-------------	-------------	------	------	-------------	------

6. Represente el diagrama de ejecución para el reemplazo de páginas e indique los fallos, no fallos y reemplazos que se producen si usamos el algoritmo indicado en el primer punto y el *Working Set* es de $S=4$. Tenga en cuenta para ello las siguientes notas:
 - Si hay que quitar una trama al proceso se eliminará la que tenga el valor más bajo
 - Aquellas referencias **subrayadas y en negrita** implican modificación
 - Los bits de referenciada pasan a cero, justo después de calcular el Working Set.
 - En caso de igualdad al decidir la eliminación de una página en el NRU, se eliminará aquella que lleve más tiempo sin referenciarse (al igual que haría un LRU)

3.1. Solución

1. Diseñe la MMU rellenando adecuadamente todos los campos



Nota: En todas las comparaciones para el STLR, Límite y PTLR, falta en el gráfico indicar la rama en la que no se cumple la condición, en la cuál derivaría en un error para cada concepto valorado.

2. ¿Cuál es el tamaño en KBytes de una página?

Como tenemos 2^{15} de desplazamiento y cada una se corresponde a 2 Bytes, $2^{15} * 2 = 2^{16}$, por tanto **64 KBytes**

3. ¿Cuánto ocupa una tabla de páginas? ¿y la tabla de segmentos?

Tabla de Páginas : tenemos 16 bits de ancho (2 Bytes) por el alto que serían 2097152 filas, por tanto : $2097152 * 2 \text{ Bytes} = 4194304 \text{ Bytes} \rightarrow$ **4 MBytes**

Tabla de Segmentos : tenemos 64 bits de ancho (8 Bytes) por el alto que serían 8 filas, por tanto : $8 * 8 \text{ Bytes} =$ **64 Bytes**

- Las tablas de página parecen demasiado grandes ¿sabría indicar alguna técnica para mejorar este aspecto?

Se podría usar por ejemplo, **tablas de páginas multinivel** ó **Tabla de páginas invertida (inversa)**

- Reescriba todos los datos necesarios de las tablas y registros límites que sean necesarios, para restablecer el estado inicial descrito en el párrafo anterior.

Teniendo en cuenta los datos, el proceso A tiene dos segmentos. Si observamos el tamaño de cada segmento en función al tamaño de página obtenemos el número de páginas para cada segmento:

Segmento 0: 192 Kbytes / 64 (Kbytes/pág) = **3 páginas**

Segmento 1: 300 Kbytes / 64 (Kbytes/pág) = 4,6875 -> **5 páginas**

Por tanto el proceso A está compuesto por dos segmentos y 8 páginas

Segmento 0 → Página 0, 1 y 2

Segmento 1 → Página 0, 1, 2, 3 y 4

Los **PTLR** para cada uno de los segmentos valdrían **2 y 4** respectivamente

STLR		@Base	Límite		t	BP	BR	BM		t	BP	BR	BM
1	65536	98304	98303	98304	131	1	0	0	2195456	134	1	0	0
	65540	2195456	153599	98305	132	1	0	0	2195457		0	0	0
	65544			98306	133	1	0	0	2195458		0	0	0
	...			98307					2195459		0	0	0
					2195460		0	0	0
					2195461				
STBR				
65536	65564			2195455					4292607				

T. Segmentos T. Página Segmento 0 T. Página Segmento 1

6. Algoritmo de Reemplazo NRU

Trama	Prea- sig.				**	**	**			**	**
131	S0P0 (0,0)	S0P0 (0,0)	S0P0 (1,1)	S0P0 (1,1)	S0P0 (1,1)	S0P0 (0,1)	S0P0 (0,1)	S0P0 (1,1)	S0P0 (1,1)		
132	S0P1 (0,0)	S0P1 (0,0)	S0P1 (0,0)	S0P1 (0,0)	S1P3 (1,0)	S1P3 (0,0)	S1P1 (1,1)	S1P1 (1,1)	S1P1 (1,1)	S1P1 (0,1)	S1P1 (0,1)
133	S0P2 (0,0)	S0P2 (0,0)	S0P2 (0,0)	S0P2 (1,1)	S0P2 (1,1)	S0P2 (0,1)	S0P2 (0,1)	S0P2 (0,1)	S0P2 (0,1)	S0P0 (1,1)	S0P0 (1,1)
134	S1P0 (0,0)	S1P0 (1,0)	S1P0 (1,0)	S1P0 (1,0)	S1P0 (1,0)	S1P2 (1,1)	S1P2 (1,1)	S1P2 (1,1)	S1P2 (1,1)	S1P2 (0,1)	S1P4 (1,0)
		S1P0	S0P0	S0P2	S1P3	S1P2	S1P1	S0P0	S1P1	S0P0	S1P4

Notaciones: (BR,BM) Ejemplo (1,0) -> BR=1 y BM=0 Grupo 10

* : Fallo de Página

** : Fallo de Página y Reemplazo

10 peticiones : 5 fallos en total (5 reemplazos) y 5 no fallos.

4. Corona Memo

Un diseñador de Sistemas Operativos propone una MMU, que gestiona mediante segmentación paginada con memoria virtual, una memoria de 2 bytes de ancho y un tamaño de 2 Mbytes. Las páginas son de 512 bytes. El tamaño máximo de un proceso es de 8 Mbytes y el número máximo de segmentos que puede tener es de 1024.

1. Dibuje la posible MMU de este sistema con toda la información posible, sabiendo que la tabla de páginas tiene Bit de Referenciada y Bit de Modificada.
2. En dicho sistema hay un proceso cuyo STBR vale 600, y tiene tres segmentos
 - S0 de 514 bytes
 - S1 de 1032 bytes
 - S2 de 1702 bytes

Dicho proceso tiene cargadas las siguientes páginas:

- Página 1 del Segmento 0 en la trama 7.
- Página 1 de Segmento 2 en la trama 8.
- Página 2 de Segmento 2 en la trama 9.

Sabiendo que las tablas de página comienzan justo a continuación de la tabla de segmentos. Dibuje las tablas de páginas y la tabla de segmentos de dicho proceso con toda la información posible.

3. Con la situación descrita en el apartado 2, se producen las siguientes referencias a memoria por parte del proceso.

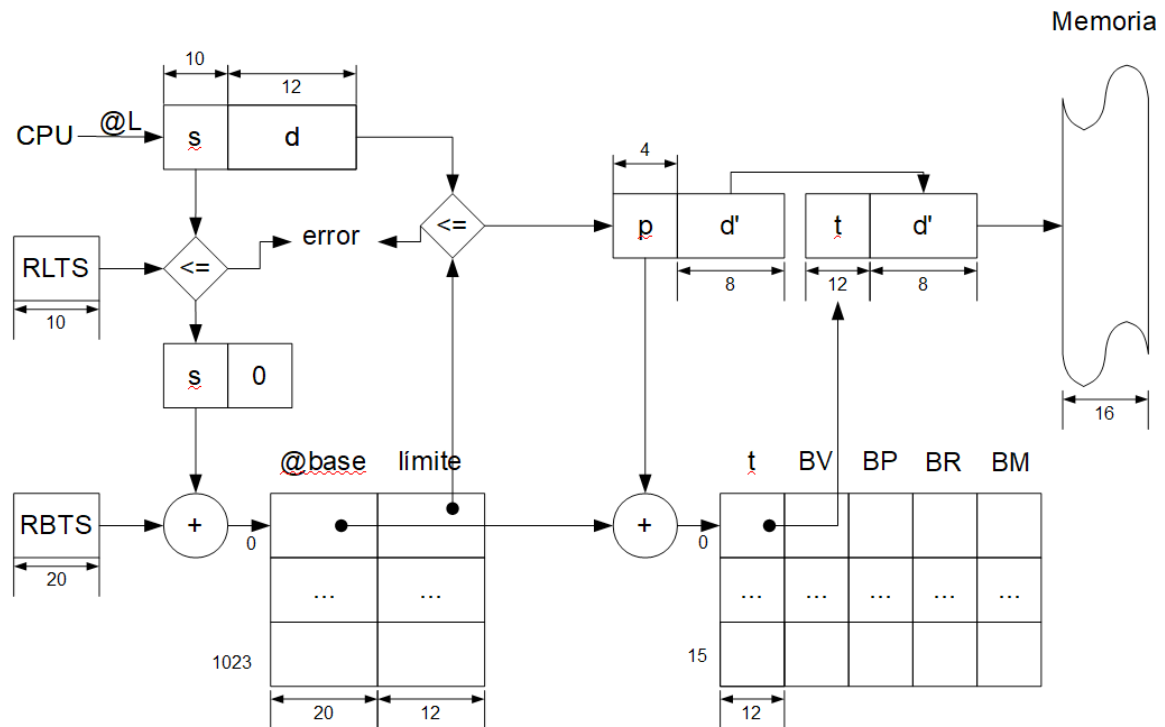
S2P1	S1P0	S0P1	S2P1	S1P0	S1P1	S2P0	S2P3	S1P0	S2P2	S0P0
------	------	------	------	------	------	------	------	------	------	------

Se está usando como algoritmo de asignación de marcos, Working Set con $s=4$. Inicialmente se asignan tantos marcos como páginas tiene el proceso cargadas en memoria. Indicar los fallos, no fallos y reemplazos que producen los siguientes algoritmos:

1. Algoritmo óptimo. (En caso de empate, se reemplazará la que hace más tiempo que no se referencia, igual que el LRU).
2. Algoritmo del reloj. Asuma que los bits de referenciada de todas las páginas están a 0, y que las páginas más antiguas son las que están en marcos más bajos. Los bits de referenciada se resetean justo después del cálculo del Working Set.

4.1. Solución

1.



2.

TS	@BASE	LIMITE
600	2648	256
602	2664	515
604	2680	850
2646		

TP S0						TP S1						TP S2					
	t	V	P	R	M		t	V	P	R	M		t	V	P	R	M
2648		1	0			2664		1	0			2680		1	0		
2649	7	1	1			2665		1	0			2681	8	1	1		
		0						1	0			2682	9	1	1		
		0						0				2683		1	0		
		0										2684		0			
2663		0				2679		0				2685		0			

3.

3. Óptimo

	S2P1	S1P0	S0P1	S2P1	S1P0	S1P1	S2P0	S2P3	S1P0	S2P2	S0P0
T7	S0P1			S0P1			S1P1		S2P3		
T8	S2P1	S2P1			S2P1			S2P0			S0P0
T9	S2P2		S1P0			S1P0				S1P0	
T10										S2P2	

4. Reloj

	S2P1	S1P0	S0P1	S2P1	S1P0	S1P1	S2P0	S2P3	S1P0	S2P2	S0P0
T7	S0P1		S1P0			S1P0			S2P3		
T8	S2P1	S2P1			S2P1			S2P0			S0P0
T9	S2P2			S0P1			S1P1			S2P2	
T10									S1P0		