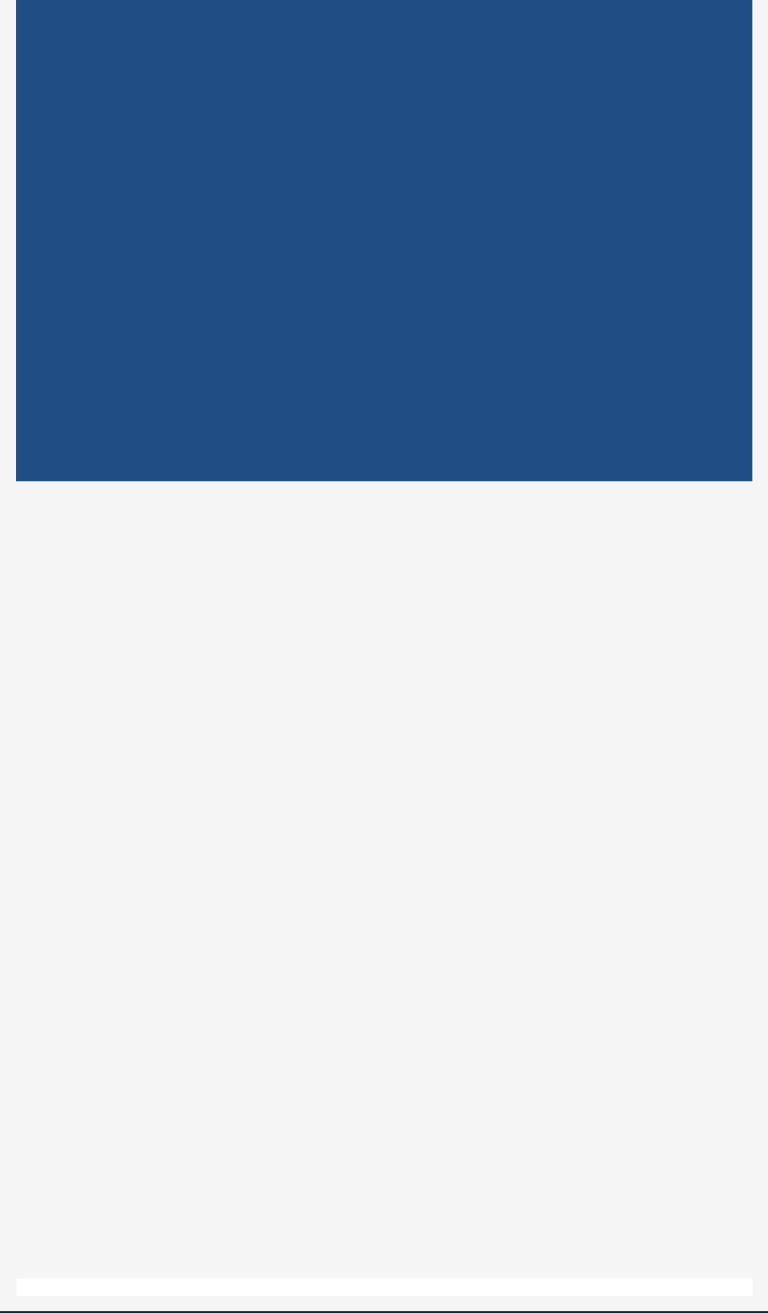
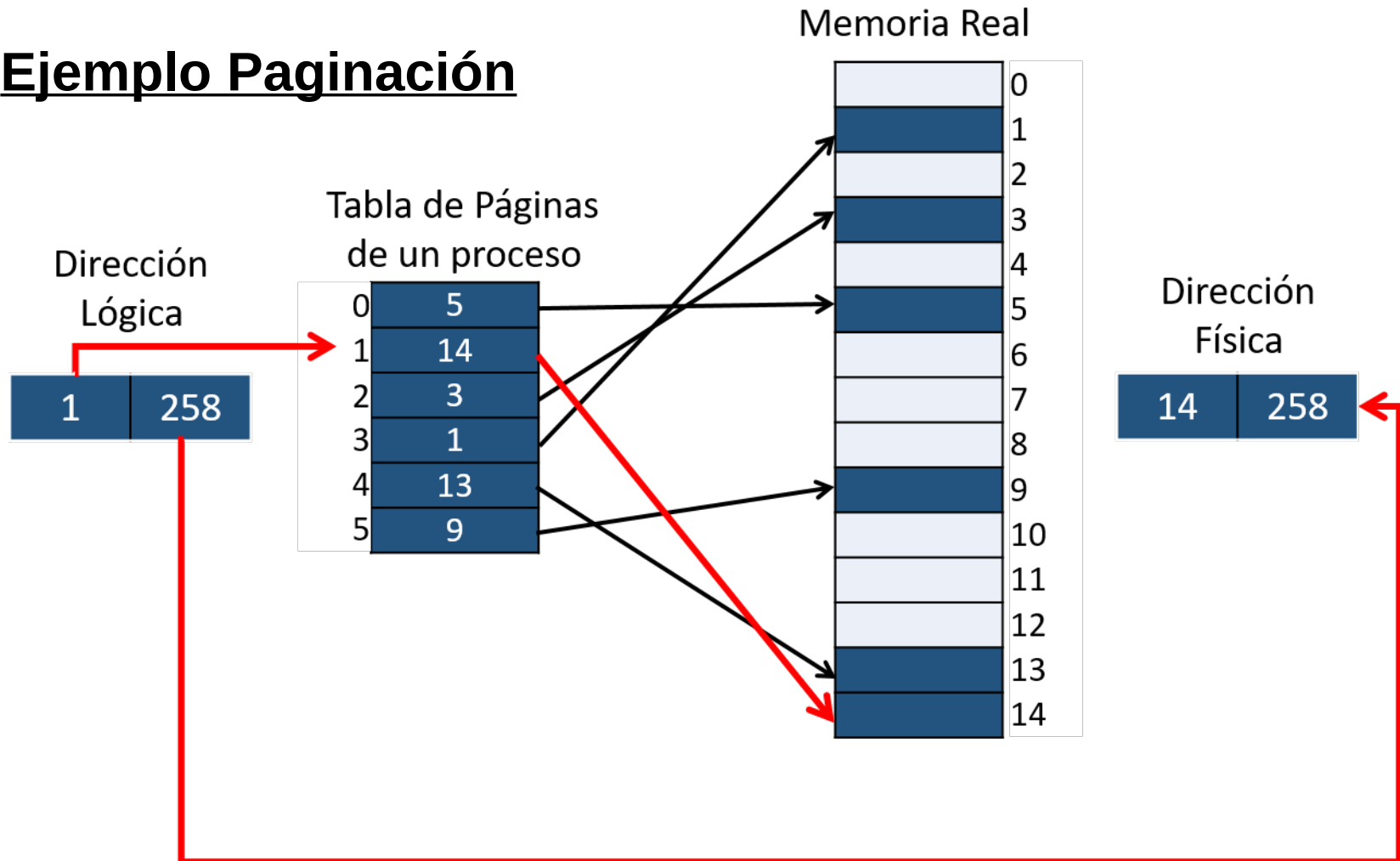


# Memoria Virtual



# Memoria Virtual

## Ejemplo Paginación



# Memoria Virtual

## Características que permiten Memoria Virtual

- Traducción de direcciones
- Proceso dividido en partes (páginas o segmentos)

# Memoria Virtual

## Motivaciones

- Durante la ejecución del proceso no se usan todas sus partes (páginas o segmentos).
- Espacio reservado sin usar.

¿Es necesario ocupar la memoria con partes que no se usan?

# Memoria Virtual

Tabla de páginas

0	5
1	14
2	3
3	1
4	13
5	9

Memoria Real

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	.
	.
	N

Memoria Virtual

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	.
	.
	.
	.
	M

# Memoria Virtual

## Ventajas

- Procesos más grandes que la memoria real.

Tabla de páginas

0	5
1	14
2	3
3	1
4	13
5	9

Memoria Real

0
1
2
3
4
5
6
7
8
9
10
11
.
.
N

0
1
2
3
4
5
6
7
8
9
10
11
12
13
.
.
.
.
M

# Memoria Virtual

## Ventajas

- Más procesos en memoria: Permite aumentar el grado de multiprogramación.

Tabla de páginas  
Proceso 1

0	5
1	10
2	3
3	1
4	7
5	9

Memoria Real

P2-pag0	0
P2-pag1	1
P6-pag2	2
P1-pag2	3
P6-pag1	4
P1-pag0	5
P4-pag1	6
P3-pag0	7
P5-pag1	8
P1-pag5	9
Libre	10

Memoria Virtual

P1-pag0	0
P1-pag1	1
P1-pag2	2
P1-pag3	3
P1-pag4	4
P1-pag5	5
P3-pag0	6
P3-pag1	7
P4-pag0	8
P4-pag1	9
P2-pag0	10
P2-pag1	11
P2-pag2	12
P2-pag3	13
P6-pag0	14
P6-pag1	15
P6-pag2	16
P5-pag0	17
P5-pag1	18

# Memoria Virtual

## Otras Ventajas

- Menos restricciones para el programador

Tabla de páginas

0	5
1	14
2	3
3	1
4	13
5	9

Memoria Real

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	.
	.
	N

Memoria Virtual

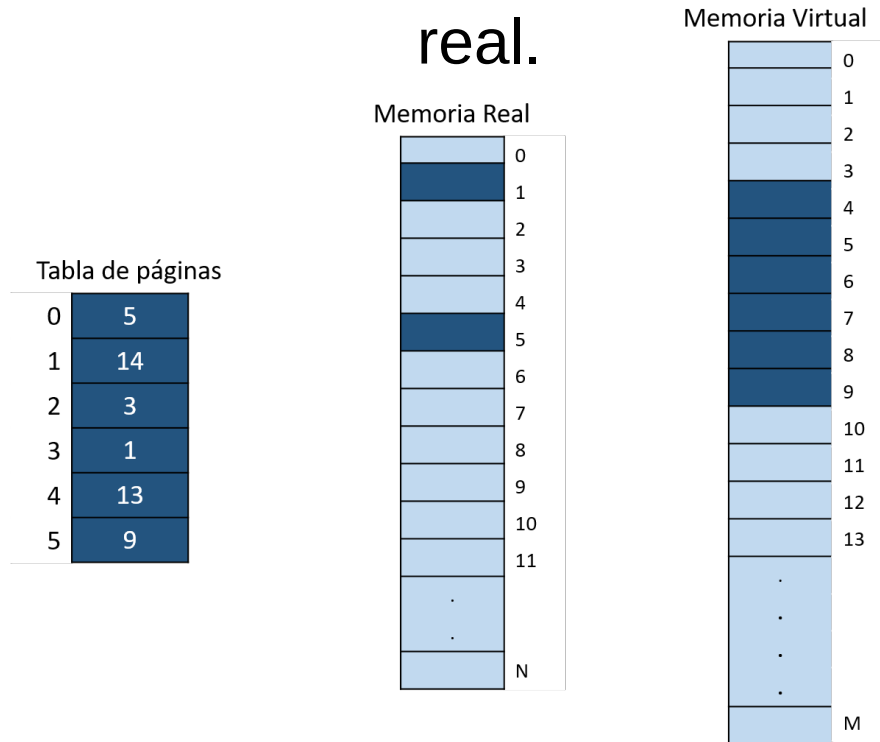
	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	.
	.
	.
	.
	M



# Memoria Virtual

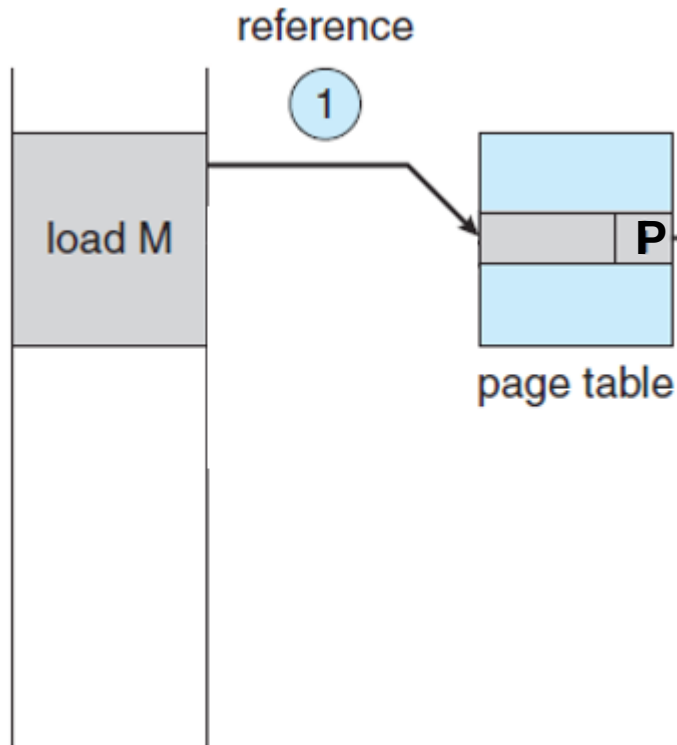
## Definición

Espacio de memoria secundaria (el disco) que puede ser direccionado como si fuese memoria real.



# Memoria Virtual

## Mecanismo de búsqueda de páginas

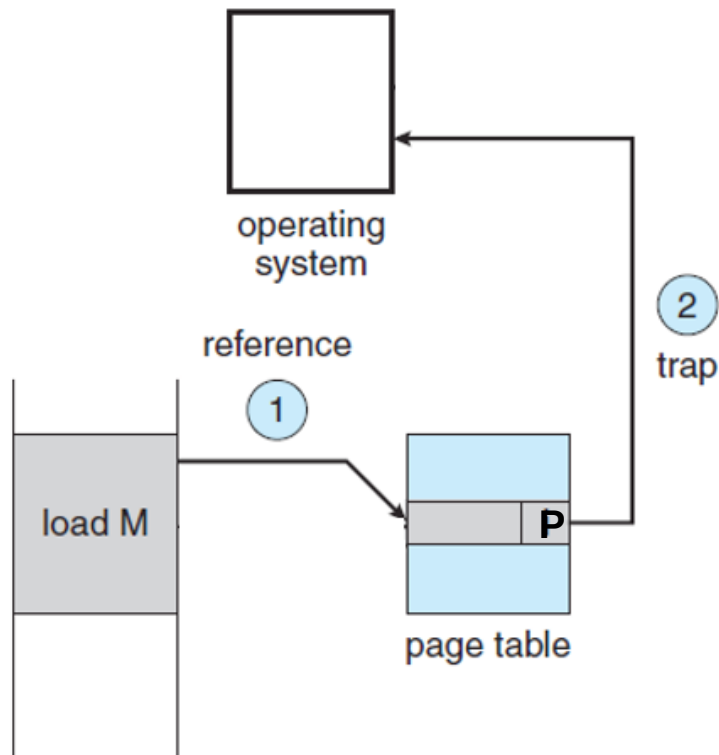


1- Se realiza una referencia a memoria:

- Verificar la **P**resencia de la página en memoria.
- Fallo de Página (Page Fault).

# Memoria Virtual

## Mecanismo de búsqueda de páginas

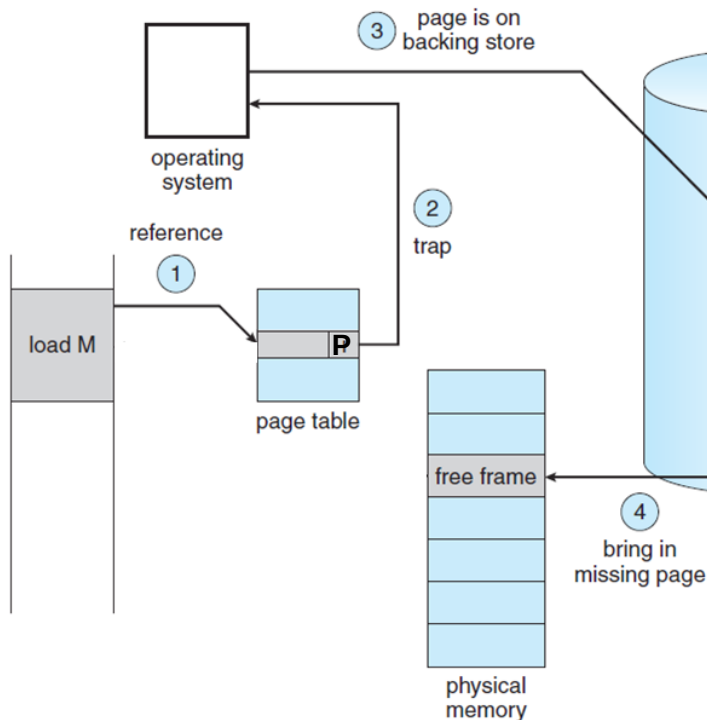


2- Interrupción por fallo de página:

- Se bloquea el proceso.
- Mientras tanto otro proceso puede utilizar el CPU.

# Memoria Virtual

## Mecanismo de búsqueda de páginas



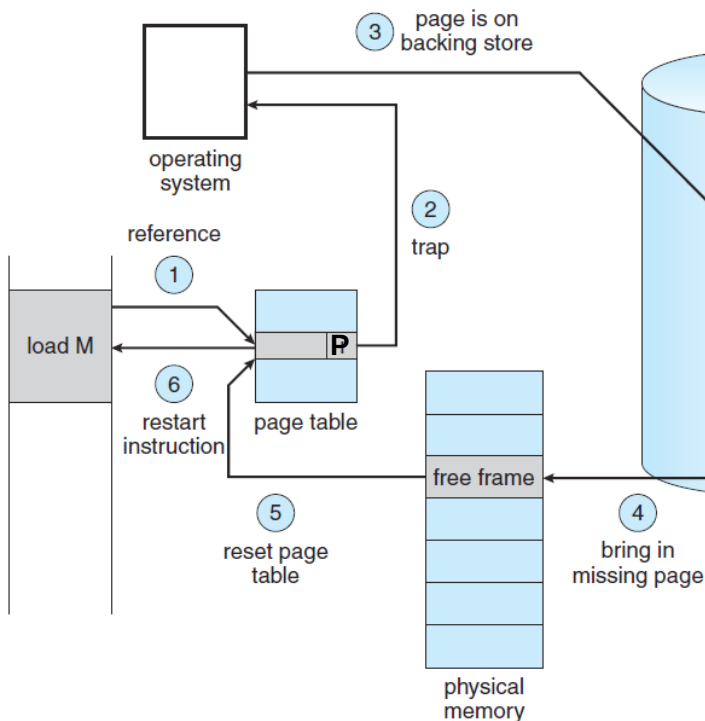
### 3- Solicitud de página

- Se determina en qué marco ubicar la página faltante.
- Se solicita la página faltante (E/S).

4- Se carga la página en memoria y se produce una interrupción para que el SO tome el control.

# Memoria Virtual

## Mecanismo de búsqueda de páginas



5- SO atiende la interrupción:

- Actualiza la tabla de páginas.
- Desbloquea el proceso (cambia su estado a Listo).

6- Se ejecuta nuevamente la instrucción que provocó el fallo de página.

# Memoria Virtual

## Estructura de Tabla de Páginas

Dir. Lógica

3	214
---	-----

Dir. Física

1	214
---	-----

Dos accesos a  
Memoria

	Número de Marco	Bit de Presencia
0	5	1
1	14	0
2	3	0
3	1	1
4	13	0
5	9	0

Memoria Real

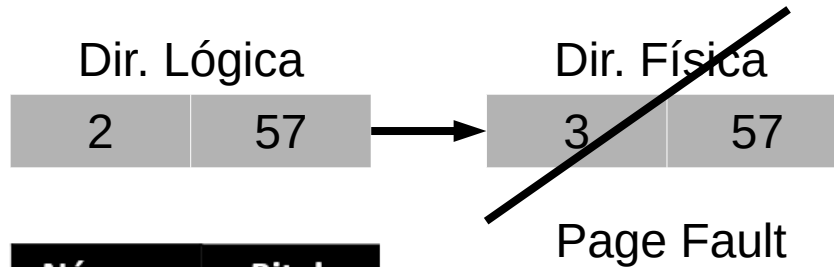
0
1
2
3
4
5
6
7
8
9
10
11
.
.
N

Memoria Virtual

0
1
2
3
4
5
6
7
8
9
10
11
12
13
.
.
.
.
M

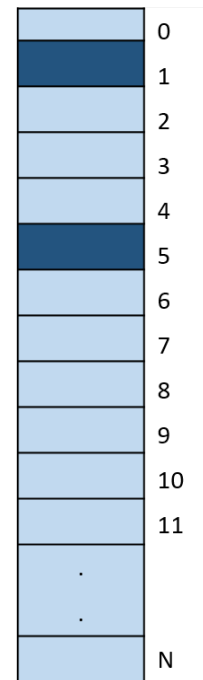
# Memoria Virtual

## Estructura de Tabla de Páginas

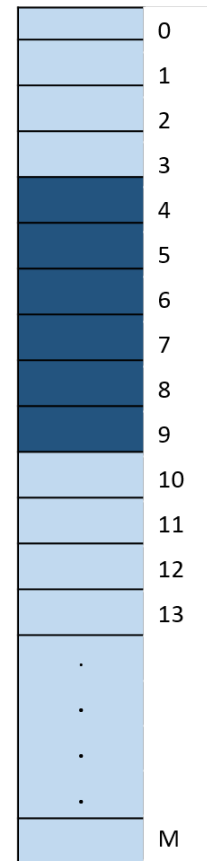


	Número de Marco	Bit de Presencia
0	5	1
1	14	0
2	3	0
3	1	1
4	13	0
5	9	0

Memoria Real

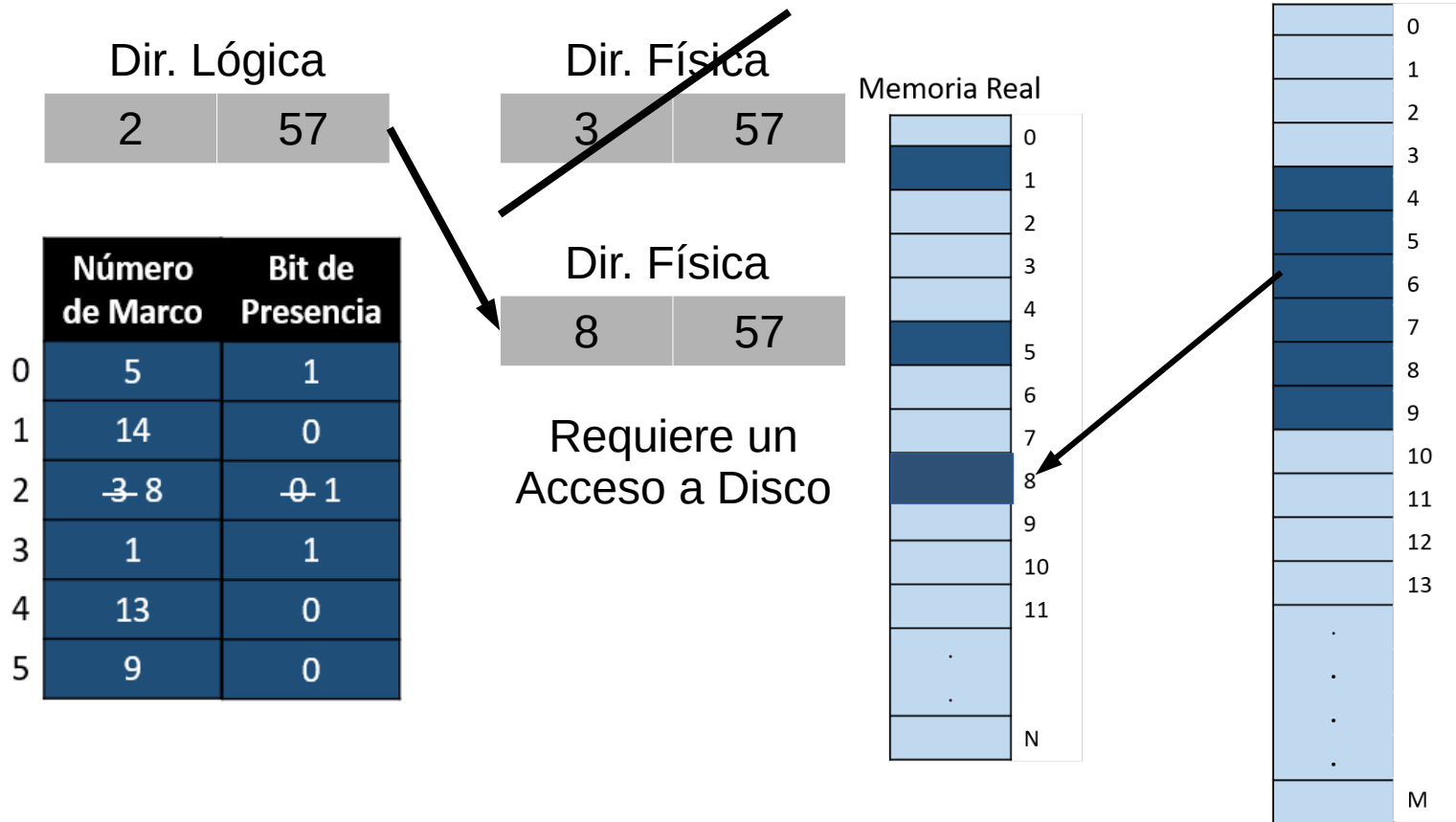


Memoria Virtual



# Memoria Virtual

## Estructura de Tabla de Páginas





# Memoria Virtual

## Estructura de Tabla de Páginas

Dir. Lógica

2

57

Sin espacio en memoria

	Número de Marco	Bit de Presencia	Bit de Modificado
0	5	1	1
1	14	0	0
2	3	0	0
3	1	1	1
4	13	0	0
5	9	0	0

Memoria Real

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	.
	.
	N

Memoria Virtual

	0
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	.
	.
	.
	.
	M

# Memoria Virtual

## Estructura de Tabla de Páginas

Dir. Lógica

2

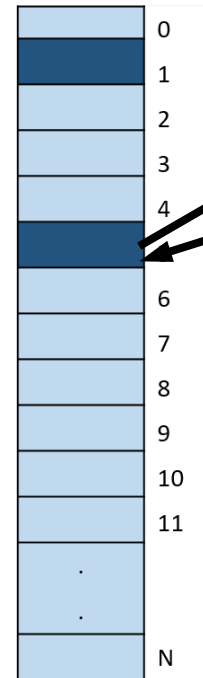
57

Sin espacio en memoria

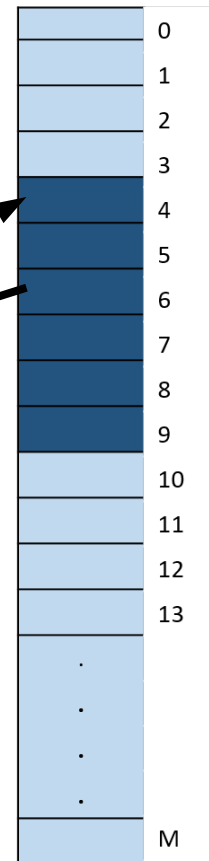
	Número de Marco	Bit de Presencia	Bit de Modificado
0	5	1 0	1
1	14	0	0
2	3 5	0 1	0
3	1	1	1
4	13	0	0
5	9	0	0

Requiere un acceso a disco para escritura y otro para lectura

Memoria Real



Memoria Virtual



# Memoria Virtual

## Eficiencia

- Requiere más accesos a memorias.
- Requiere más accesos a disco.
- No mejora el rendimiento para la ejecución del proceso.

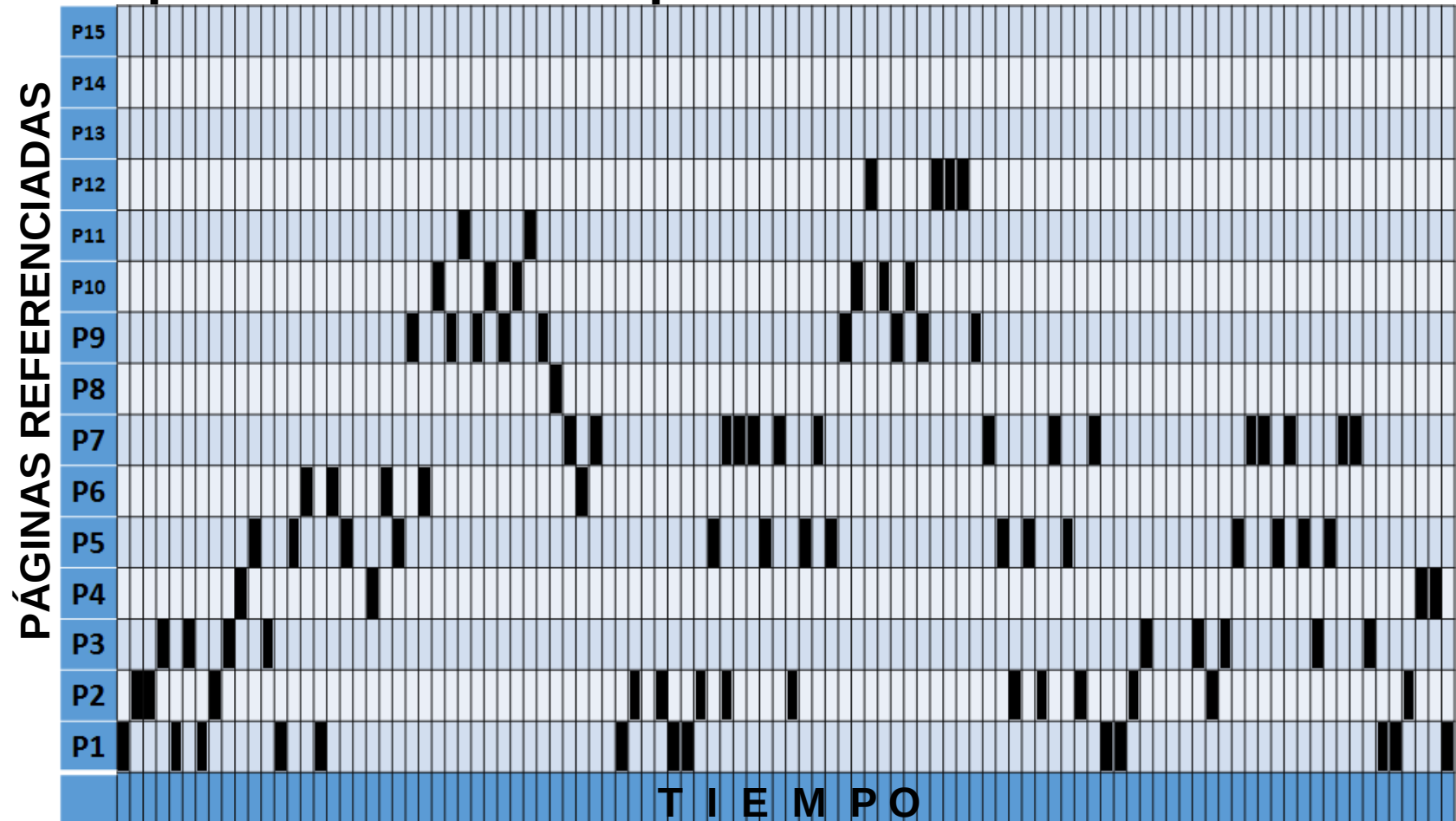
# Memoria Virtual

## Principio de Localidad o proximidad

- Permite que la memoria virtual funcione y que no genere bajo rendimiento.
- Durante un intervalo de tiempo sólo se usan unas páginas de forma activa. A este conjunto se lo denomina localidad.

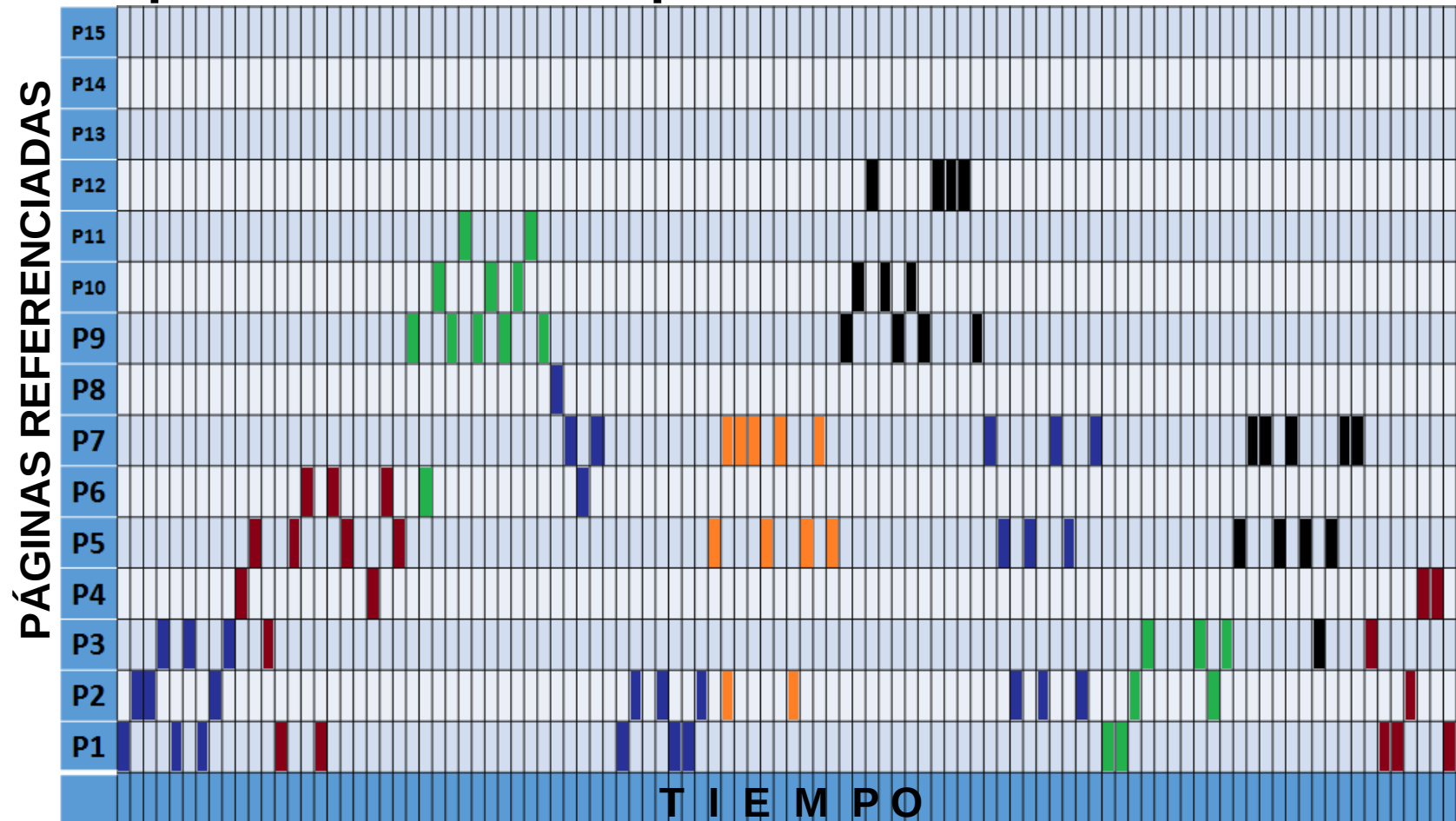
# Memoria Virtual

## Principio de Localidad o proximidad



# Memoria Virtual

## Principio de Localidad o proximidad



# Memoria Virtual

## Otras Estructuras para Tabla de Páginas

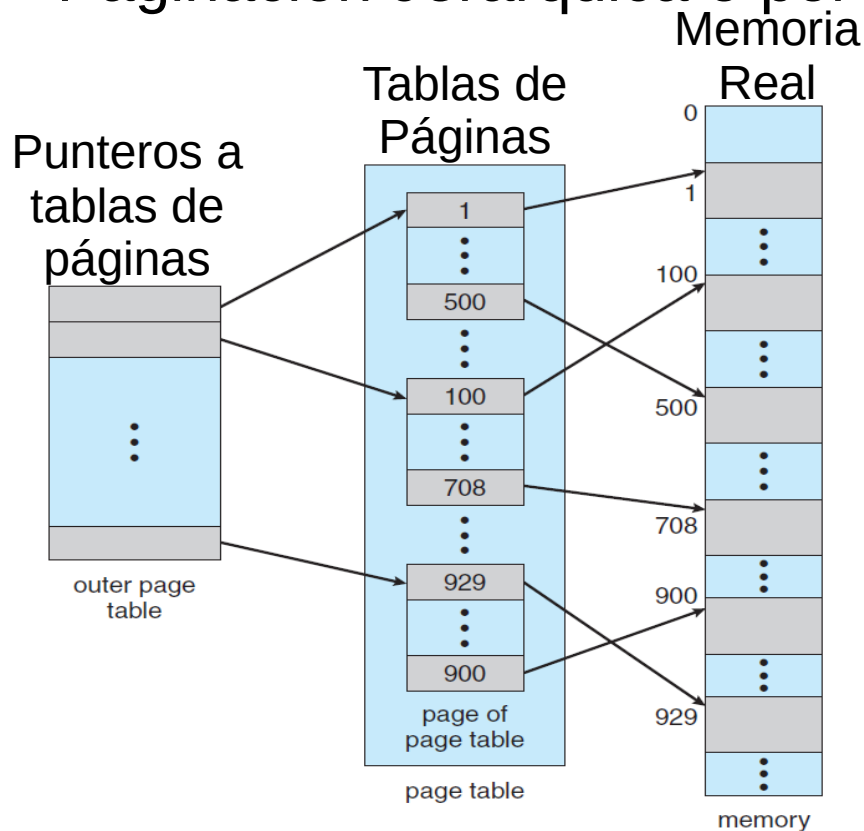
- Las páginas suelen tener tamaño  $2^n$ .
- Ej: Dirección con páginas de 4KiB de tamaño.

52 bits	12 bits
Número de Página	Desplazamiento

# Memoria Virtual

## Otras Estructuras para Tabla de Páginas

- Paginación Jerárquica o por niveles.

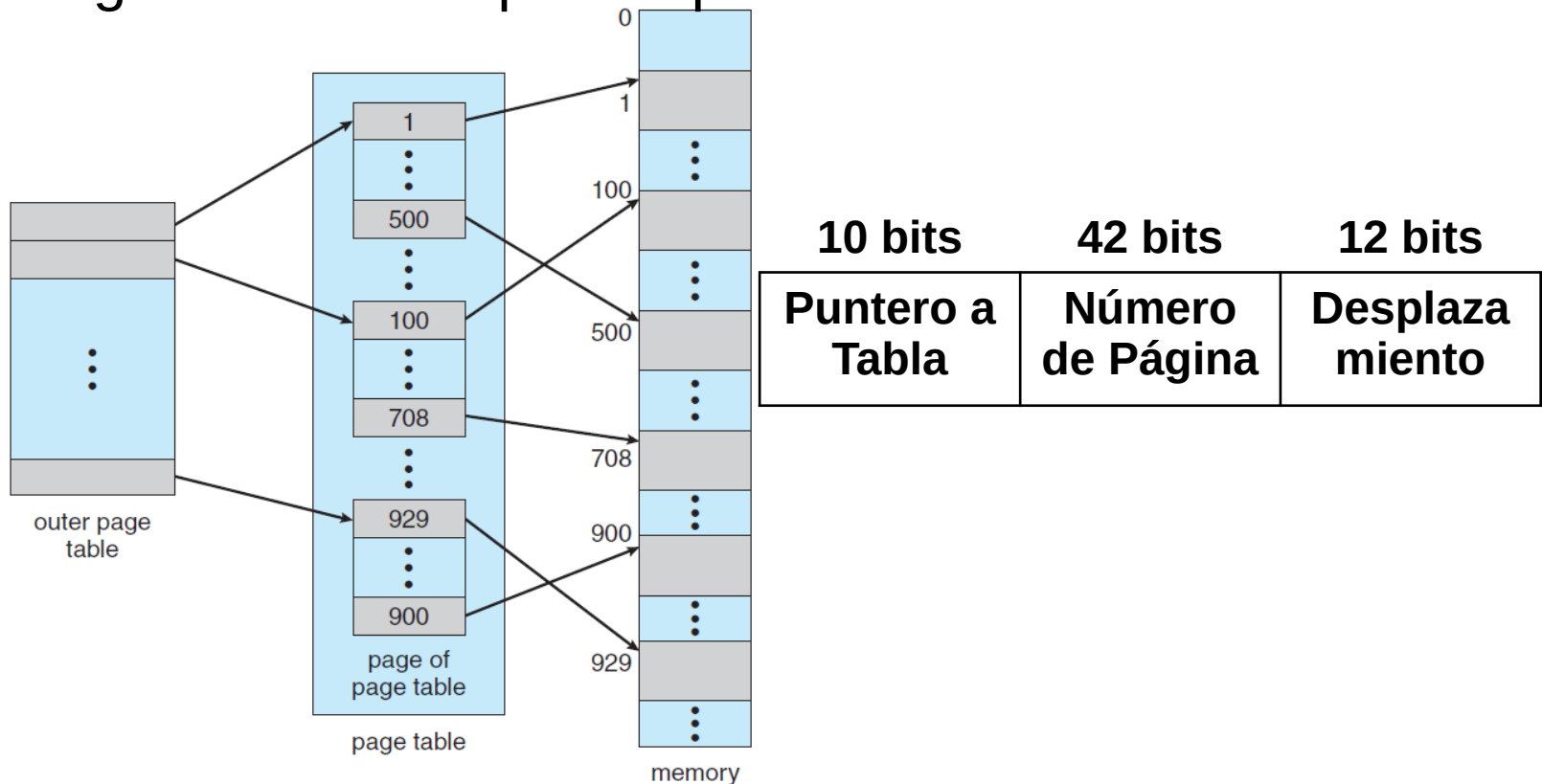




# Memoria Virtual

## Otras Estructuras para Tabla de Páginas

- Paginación Jerárquica o por niveles.



# Memoria Virtual

## Otras Estructuras para Tabla de Páginas

- Tabla de páginas invertida:
  - Hay una tabla de páginas para todos los procesos.
  - La cantidad de entradas de la tabla es igual a la cantidad de marcos en la memoria.

# Memoria Virtual

## Otras Estructuras para Tabla de Páginas

- Tabla de páginas invertida:

N Pág	Desp
-------	------

Tabla de páginas invertida  
(N entradas)

	N Pág	PID
1		
2		
3		
i		
N		

Búsqueda  
Secuencial



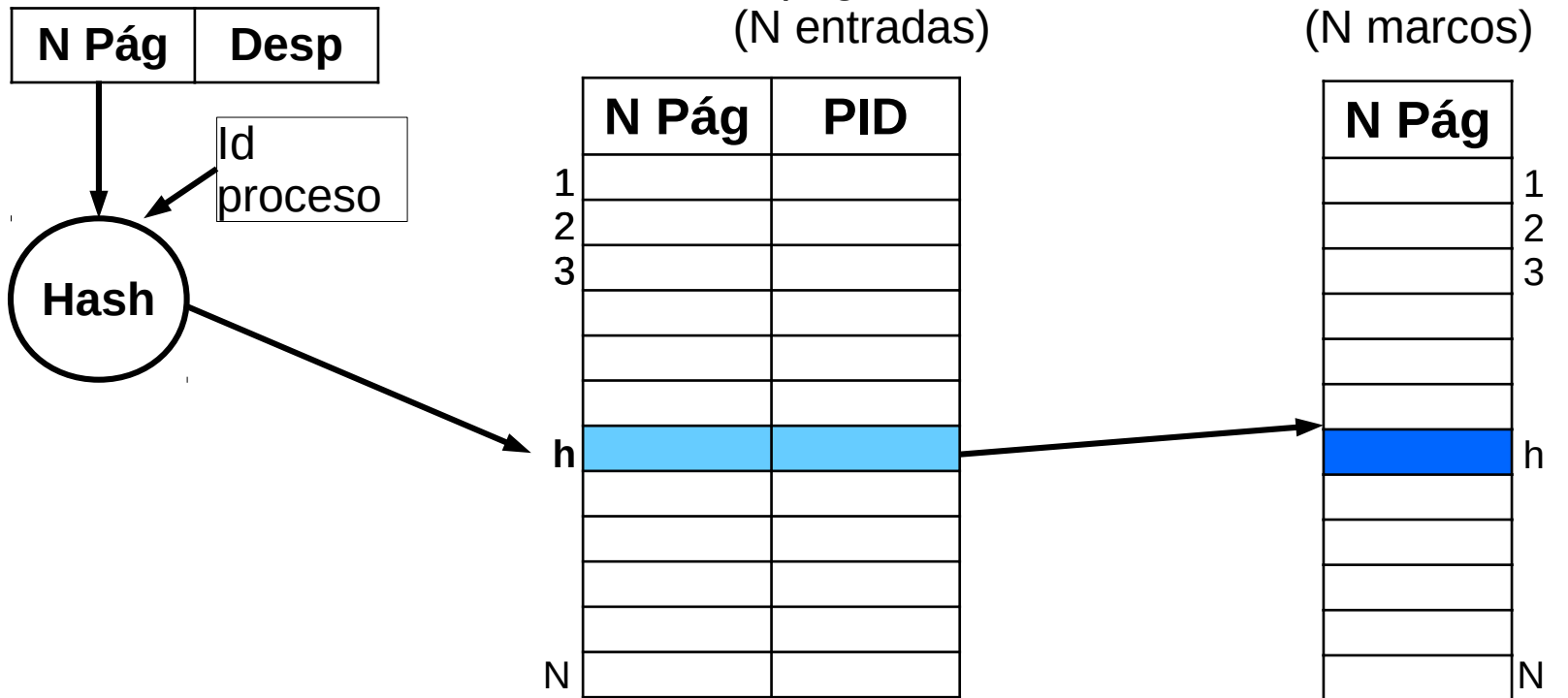
Memoria Real  
(N marcos)

N Pág
i
N



## Otras Estructuras para Tabla de Páginas

- Tabla de páginas invertida:



## Otras Estructuras para Tabla de Páginas

- Tabla de páginas invertida:

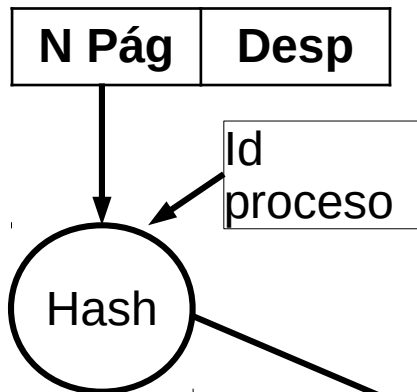


Tabla de páginas invertida  
(N entradas)

	N Pág	PID	PTR
1			
2			
3			
h	<b>COLISION</b>		—
N			

Memoria Real  
(N marcos)

N Pąg	
	1
	2
	3
	h
	N

# Memoria Virtual

## **TLB**: Buffer de Traducción Anticipada

- Soporte de HW.
- Mejora el rendimiento de la traducción de direcciones.
- Puede reducir la cantidad de accesos a memoria.

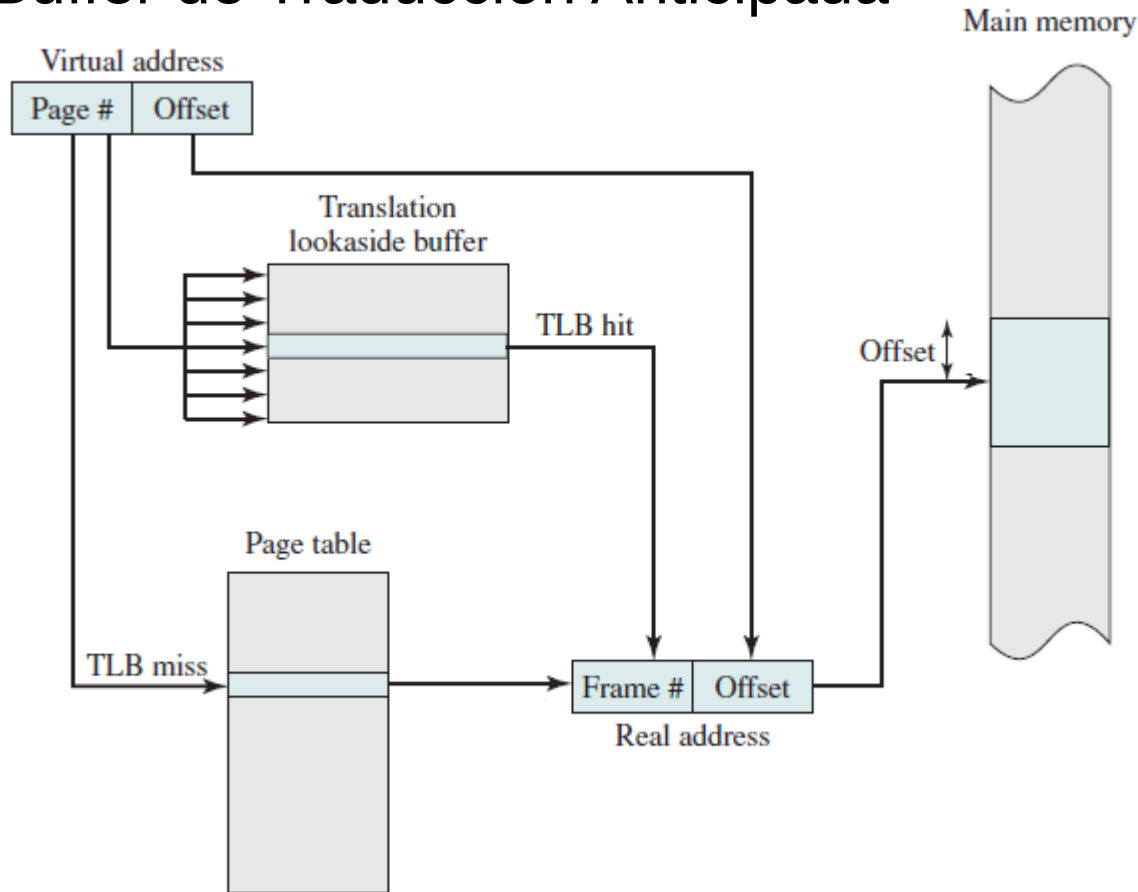
# Memoria Virtual

**TLB**: Buffer de Traducción Anticipada

- Soporte de HW.
- Cache de la tabla de páginas.

# Memoria Virtual

## TLB: Buffer de Traducción Anticipada

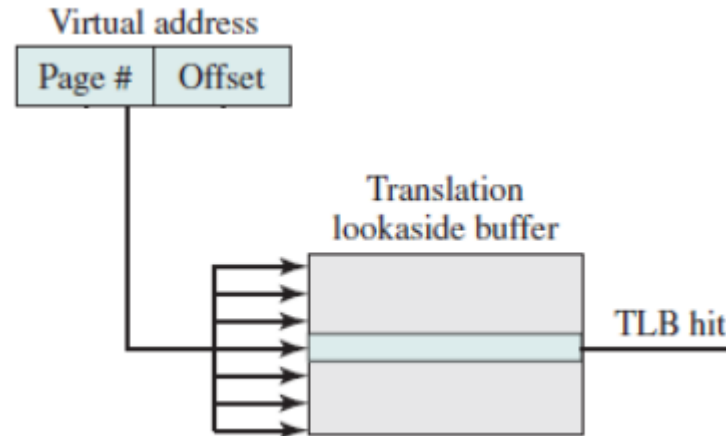




# Memoria Virtual

**TLB**: Buffer de Traducción Anticipada

- Características de la TLB:
  - Muy rápida pero con pocos entradas.
- Entradas de la TLB.
  - ASID (identificador de espacio de direcciones).



# Memoria Virtual

## Diseño del Sistema Operativo:

- Hardware:
  - Soporta o no memoria virtual.
  - Soporta o no **paginación**, segmentación o ambas.
- Software: Políticas para la gestión de memoria virtual.
  - Objetivo: Mejorar el rendimiento.

# Memoria Virtual

## Diseño del Sistema Operativo:

- Políticas:
  - a) de Recuperación.
  - b) de Ubicación.
  - c) de Reemplazo o Sustitución.
  - d) del Conjunto Residente.
  - e) de Limpieza.

## Diseño del Sistema Operativo:

- 
- The Gantt chart displays the schedule for 15 projects (P1 to P15) over 100 days. The chart is organized into a grid where each row represents a project and each column represents a day. The projects are listed on the left side of the chart, and the days are listed on the top. The chart uses a color-coded system to represent different tasks or activities:
- P15 (Blue):** Starts at day 0 and ends at day 100.
  - P14 (Light Blue):** Starts at day 0 and ends at day 100.
  - P13 (Light Blue):** Starts at day 0 and ends at day 100.
  - P12 (Green):** Starts at day 0 and ends at day 100.
  - P11 (Green):** Starts at day 0 and ends at day 100.
  - P10 (Green):** Starts at day 0 and ends at day 100.
  - P9 (Green):** Starts at day 0 and ends at day 100.
  - P8 (Blue):** Starts at day 0 and ends at day 100.
  - P7 (Orange):** Starts at day 0 and ends at day 100.
  - P6 (Red):** Starts at day 0 and ends at day 100.
  - P5 (Red):** Starts at day 0 and ends at day 100.
  - P4 (Red):** Starts at day 0 and ends at day 100.
  - P3 (Blue):** Starts at day 0 and ends at day 100.
  - P2 (Blue):** Starts at day 0 and ends at day 100.
  - P1 (Blue):** Starts at day 0 and ends at day 100.
- The chart shows a complex sequence of tasks, with some projects starting and ending at various points throughout the 100-day period. The tasks are represented by colored bars, and the chart is organized into a grid where each row represents a project and each column represents a day.

# Memoria Virtual

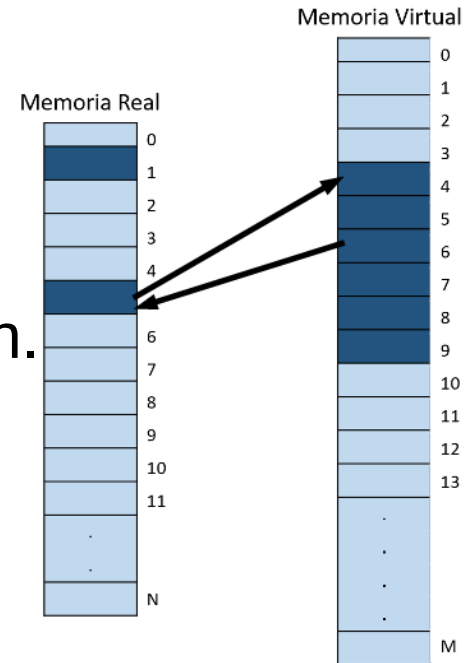
## Diseño del Sistema Operativo:

- b) Política de Ubicación:
  - Con Segmentación Pura.
    - Algoritmos Best fit, First fit, Next fit y Worst fit.

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Bloqueo de marcos.
- Algoritmos:
  - Óptimo, FIFO, LRU, Clock, Clock Mejorado.



# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo Óptimo
  - Selección de la víctima: se elige a la página a la que se realizará una referencia en el futuro más lejano.

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
  - Algoritmo FIFO
    - Selección de la víctima: se elige a la página que hace más tiempo está en memoria.



# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo LRU (Least Recently Used)
  - Selección de la víctima: se elige a la página que hace más tiempo no es referenciada.

# Memoria Virtual

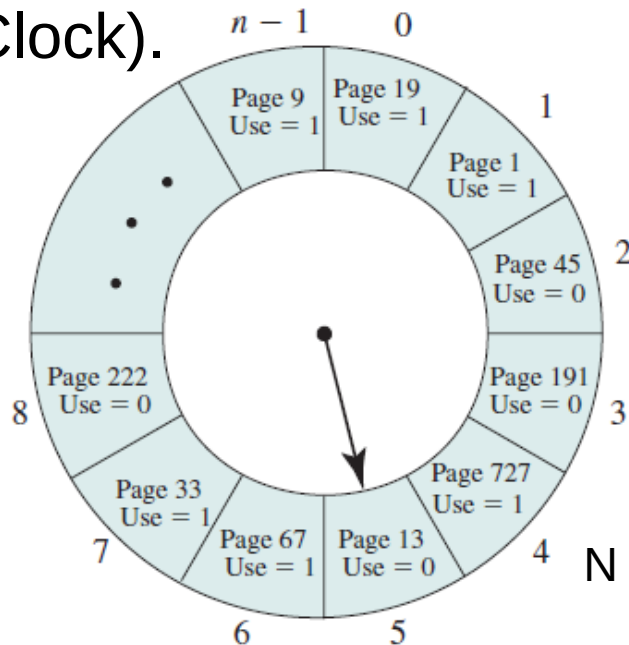
## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
  - Algoritmo del Reloj o de Segunda Oportunidad (Clock).
  - Se apunta al próximo marco a reemplazar, pero:
    - Si (Bit de uso == 0) ----> se reemplaza el marco.
    - Si (Bit de uso == 1) ----> Bit de Uso = 0 y se apunta al siguiente marco y se vuelve a preguntar.

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo del Reloj o de Segunda Oportunidad (Clock).



N marcos de un proceso

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo – Tabla de Páginas:

FIFO: No requiere un formato diferente.  
Requiere actualizar un puntero a la próxima página que será reemplazada.

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo – Tabla de Páginas:

LRU: Por cada página se requiere almacenar el momento de su última referencia.

#Marco	Bit de Presencia	Bit de Modificado	Instante de referencia
--------	------------------	-------------------	------------------------

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo – Tabla de Páginas:

Clock: Requiere que se agregue un “Bit de Uso” a la tabla de página.

#Marco	Bit de Presencia	Bit de Modificado	Bit de Uso
--------	---------------------	----------------------	---------------

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
  - Algoritmo Clock Mejorado.
    - Requiere:
      - Un puntero al siguiente marco a analizar.
      - Bit de uso.
      - Bit de modificado.

# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo Clock Mejorado.
- (u=0; m=0): No accedido recientemente, no modificado.
- (u=1; m=0): Accedido recientemente, no modificado.
- (u=0; m=1): No accedido recientemente, modificado.
- (u=1; m=1): Accedido recientemente, modificado.



# Memoria Virtual

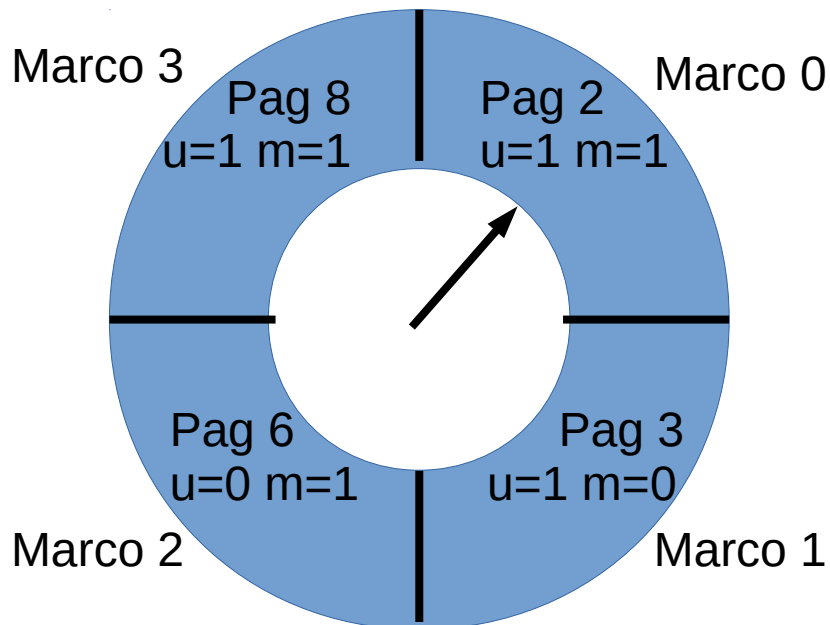
## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
  - Algoritmo Clock Mejorado.
    - 1) Recorre los marcos y selecciona el primero con  $(u=0; m=0)$ .
    - 2) Si no encuentra, recorre los marcos y selecciona el primero con  $(u=0; m=1)$ . A medida que recorre, modifica el bit de uso de cada marca de 1 a 0.
    - 3) Repite el 1er paso y si es necesario el 2do paso.

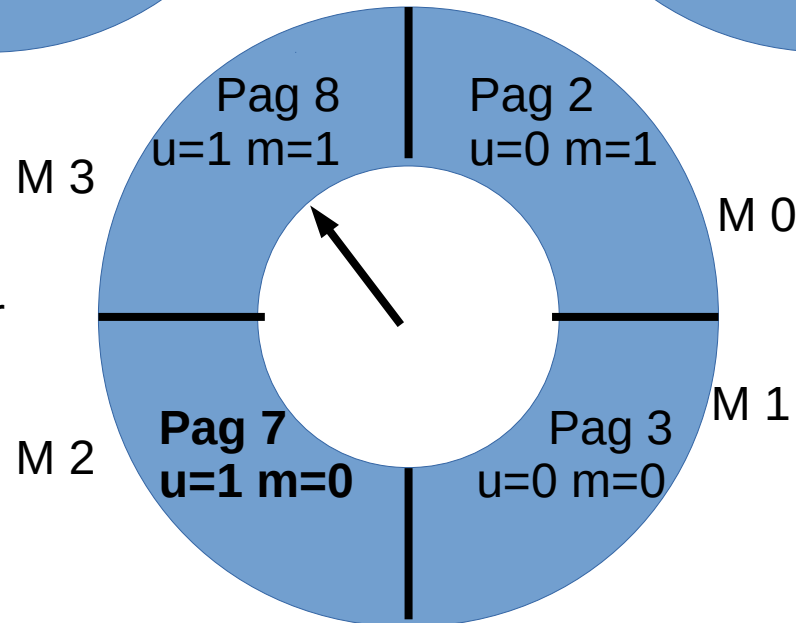
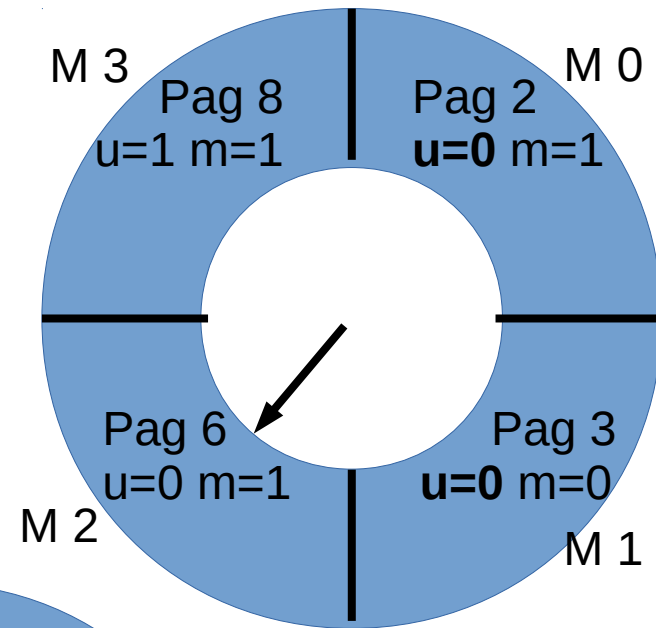
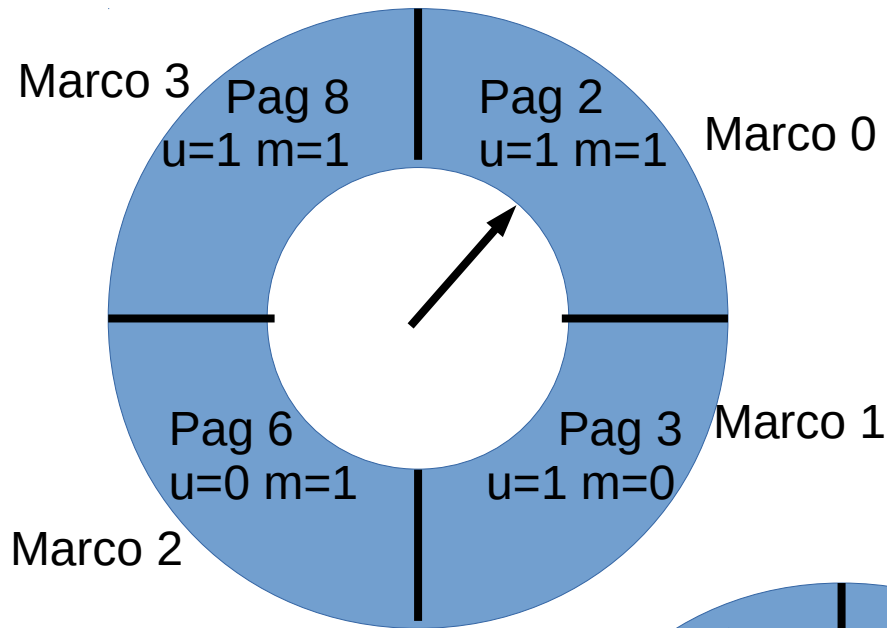
# Memoria Virtual

## Diseño del Sistema Operativo:

- c) Políticas de Reemplazo o Sustitución.
- Algoritmo Clock Mejorado.



Se referencia a la página nº 7 para leer una variable.



Se referencia a la  
página nº 7 para leer  
una variable.

# Memoria Virtual

## Diseño del Sistema Operativo:

- d) Gestión del conjunto residente.

	Reemplazo Local	Reemplazo Global
Asignación Fija	<ul style="list-style-type: none"><li>• Número de marcos asignados a un proceso no varía.</li><li>• Páginas a cambiar son del mismo proceso.</li></ul>	<ul style="list-style-type: none"><li>• No es posible.</li></ul>
Asignación Variable	<ul style="list-style-type: none"><li>• Número de marcos asignados a un proceso puede cambiar.</li><li>• Páginas a cambiar son del mismo proceso.</li></ul>	<ul style="list-style-type: none"><li>• Las páginas a reemplazar se eligen entre todos los marcos.</li><li>• Cada reemplazo puede afectar a otro proceso.</li></ul>

# Memoria Virtual

## Diseño del Sistema Operativo:

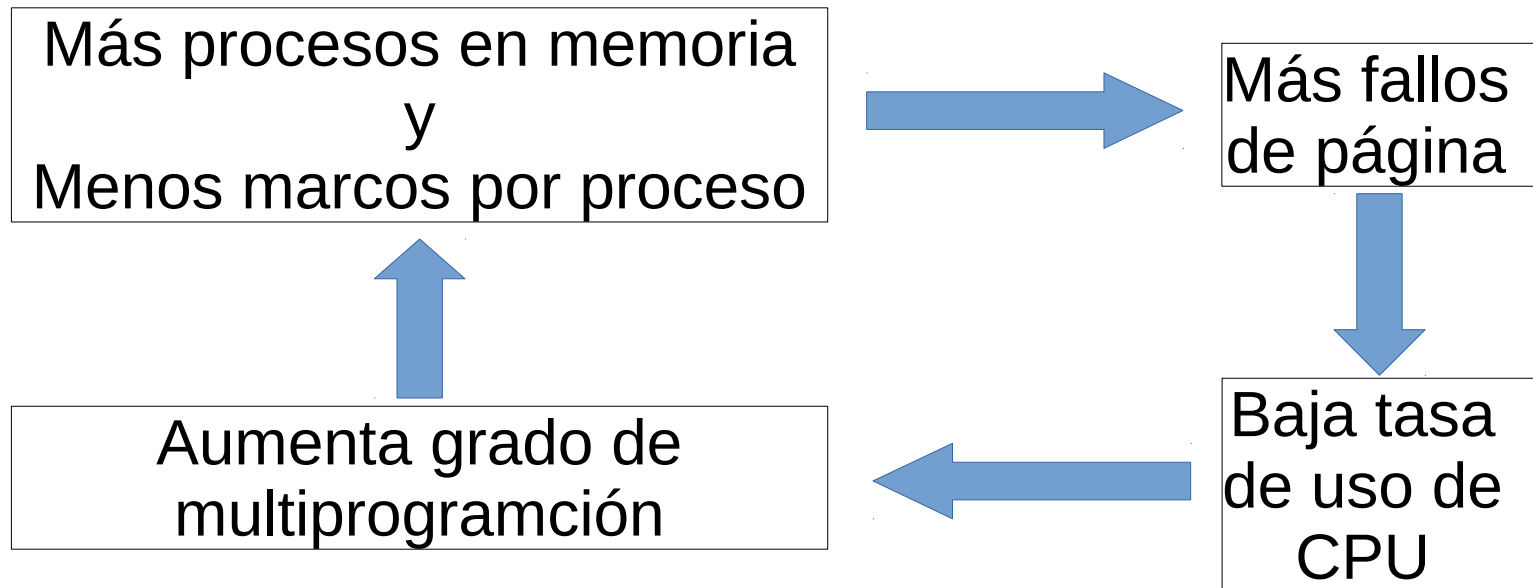
- e) Políticas de Limpieza.
  - Limpieza bajo demanda
  - Limpieza adelantada

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Sobrepaginación / Trashing (Trasiego)



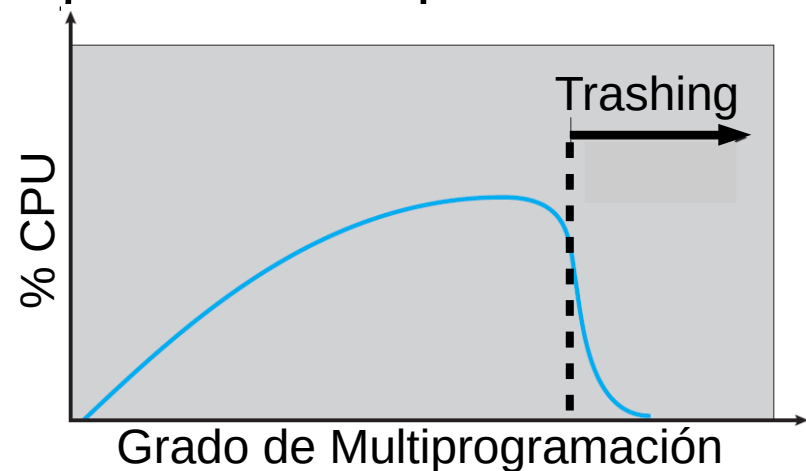
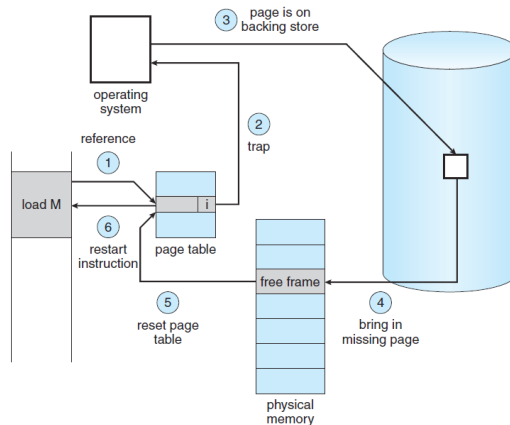
# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Sobrepaginación / Trashing (Trasiego)

Se invierte más tiempo en el mecanismo de paginación que en la ejecución del proceso.



# Memoria Virtual

## Diseño del Sistema Operativo:

- **Consideraciones**
- **Tamaño de la página**

	<b>Chica</b>	<b>Grande</b>
<b>Tabla de páginas</b>	<b>Muchas Entradas</b>	<b>Pocas Entradas</b>
<b>Fallos de página</b>	Más	Menos
<b>TLB</b>	Más fallos de TLB	Más aciertos de TLB
<b>Fragmentación</b>	Menos	Más
<b>Localidad</b>	Más preciso	Menos preciso
<b>Transferencia E/S</b>	Menos	Más



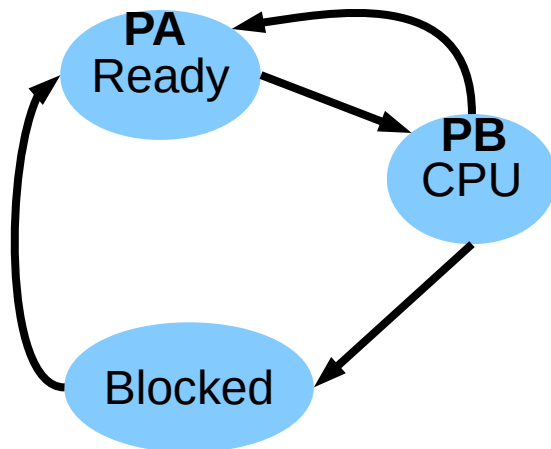
# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Bloqueo de páginas

1- PB provoca un fallo de página.



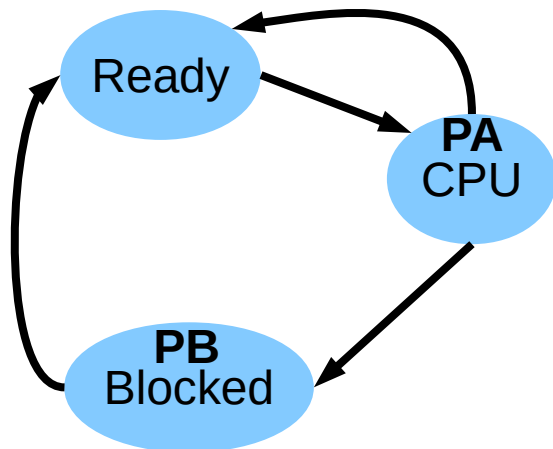
PA: Proceso alta prioridad  
PB: Proceso baja prioridad

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Bloqueo de páginas



- 1- PB provoca un fallo de página.
- 2- PB es bloqueado hasta tener la página solicitada.
- 3- PA es asignado al CPU.

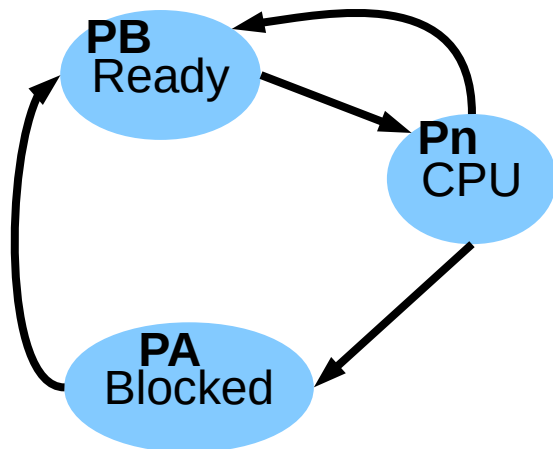
PA: Proceso alta prioridad  
PB: Proceso baja prioridad

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Bloqueo de páginas



- 1- PB provoca un fallo de página.
- 2- PB es bloqueado hasta tener la página solicitada.
- 3- PA es asignado al CPU.
- 4- PA provoca un fallo de página.
- 5- La página solicitada por PB se carga en memoria y PB se desbloquea.

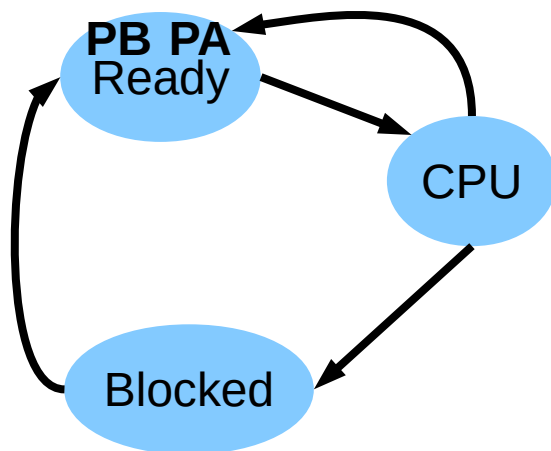
PA: Proceso alta prioridad  
PB: Proceso baja prioridad

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Bloqueo de páginas



PA: Proceso alta prioridad  
PB: Proceso baja prioridad

- 1- PB provoca un fallo de página.
- 2- PB es bloqueado hasta tener la página solicitada.
- 3- PA es asignado al CPU.
- 4- PA provoca un fallo de página.
- 5- La página solicitada por PB se carga en memoria y PB se desbloquea.
- 6- PB puede sufrir inanición y la página que solicitó no fue modificada y permanece mucho tiempo en memoria.**
- 7-La página solicitada por PA sustituye a la de PB.**

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Bloqueo de páginas
  - Se utiliza un bit de bloqueo de marcos.
  - Evita que una página que aún no fue utilizada sea sustituida.
  - Se utiliza para marcos de memoria pertenecientes al SO.

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

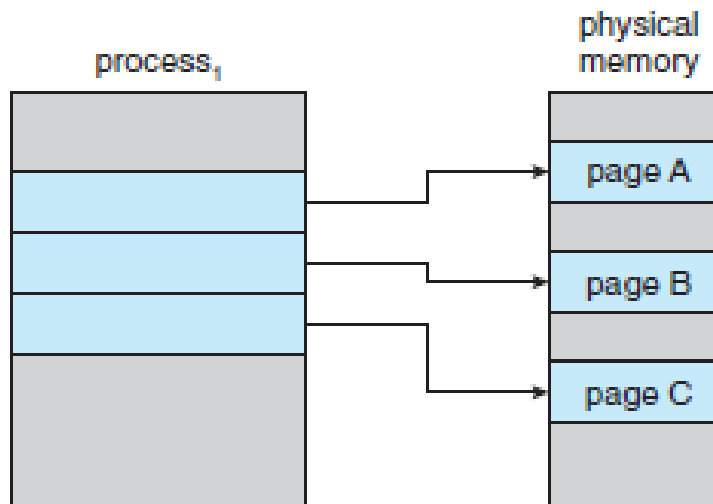
- Compartición de páginas: Copia durante escritura
- Paginación permite compartir memoria.
- `fork()` crea un proceso hijo duplicado.
  - Para mejor el rendimiento padre e hijo comparten memoria.

# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Compartición de páginas: Copia durante escritura

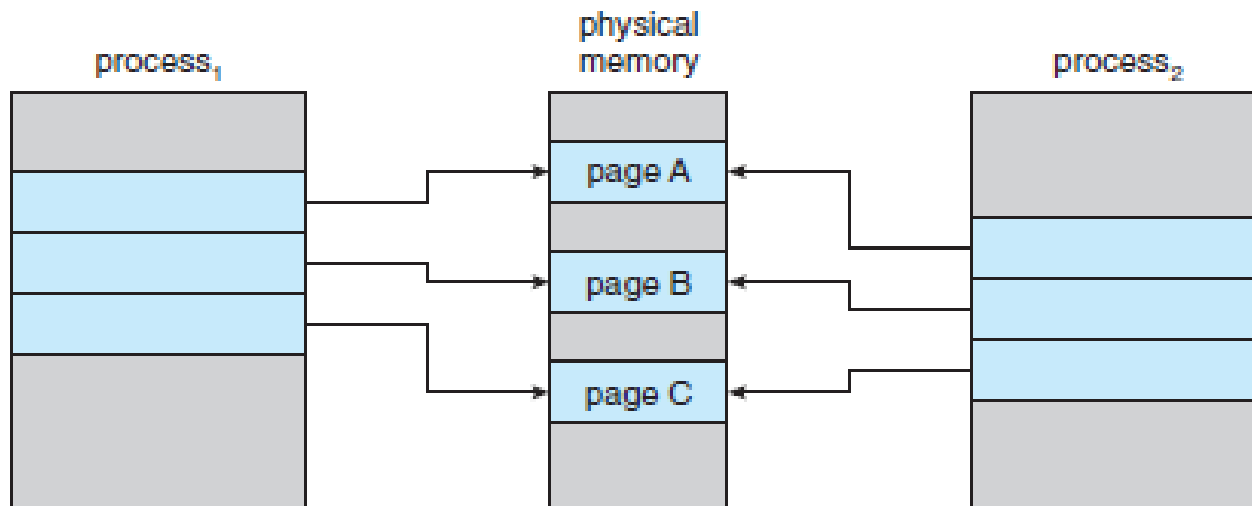


# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Compartición de páginas: Copia durante escritura



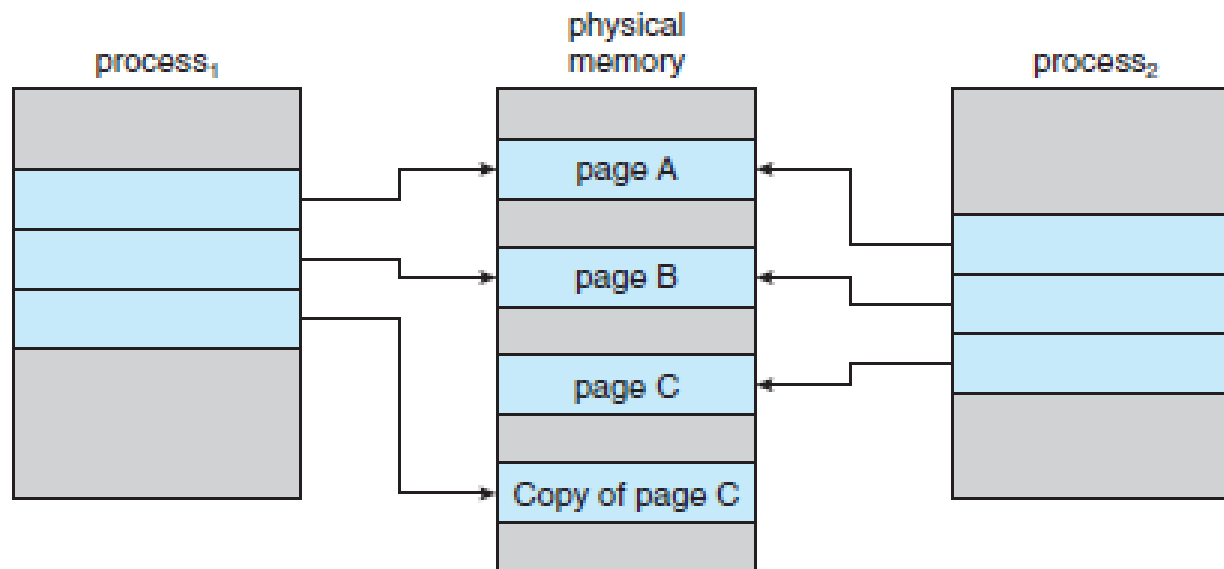


# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Compartición de páginas: Copia durante escritura



# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Estructura de Programas

```
for (I = 0; I < 128; I++)  
  for (J = 0; J < 128; J++)  
    matriz[I][J] = 0;
```

```
for (J = 0; J < 128; J++)  
  for (I = 0; I < 128; I++)  
    matriz[I][J] = 0;
```

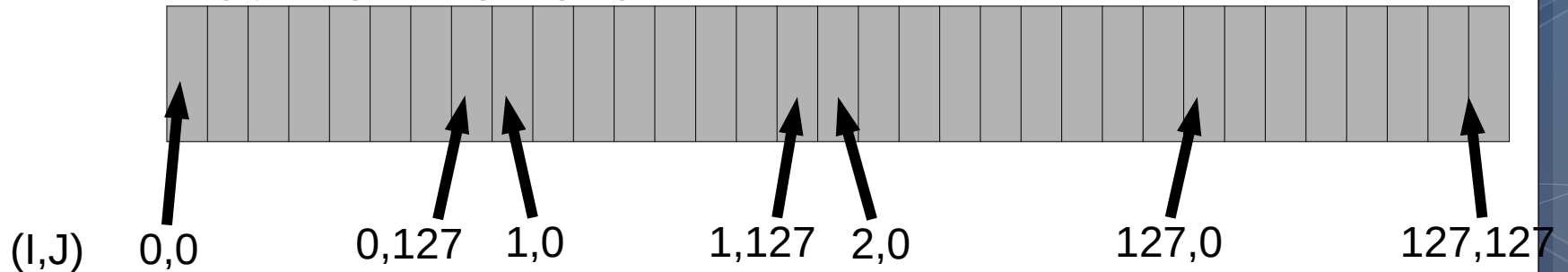
# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Estructura de Programas

Matriz en memoria



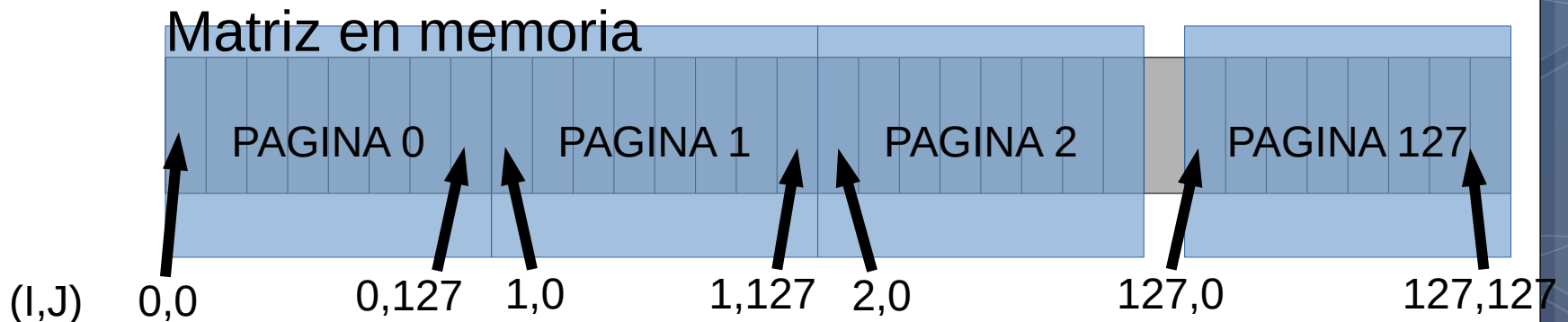
# Memoria Virtual

## Diseño del Sistema Operativo:

### Consideraciones

- Estructura de Programas

Páginas de 512 Bytes



# Memoria Virtual

## Algoritmos de reemplazo:

	2	3	2	1	5	2	4	5	3	2	5	2	
OPTIMO	2	2	2	2	2	2	4	4	4	2	2	2	
	-	3	3	3	3	3	3	3	3	3	3	3	
	-	-	-	1	5	5	5	5	5	5	5	5	
	PF	PF		PF	PF		PF			PF			
FIFO	2	3	2	1	5	2	4	5	3	2	5	2	
	2	2	2	2	5	5	5	5	3	3	3	3	
	-	3	3	3	3	2	2	2	2	2	5	5	
	-	-	-	1	1	1	4	4	4	4	4	2	
	PF	PF		PF	PF	PF	PF		PF		PF	PF	

3PF + 3PF  
Total= 6PF

6PF + 3PF  
Total= 9PF

# Memoria Virtual

## Algoritmos de reemplazo:

LRU

2	3	2	1	5	2	4	5	3	2	5	2
2	2	2	2	2	2	2	2	3	3	3	3
-	3	3	3	5	5	5	5	5	5	5	5
-	-	-	1	1	1	4	4	4	2	2	2
PF	PF		PF	PF		PF		PF	PF		

4PF + 3PF  
Total= 7PF

CLOCK →

2	3	2	1	5	2	4	5	3	2	5	2
2*	2*	2* → 2*	5*	5* → 5*	5*	5*	3*	3* → 3*	3*	3*	3*
-	3*	3*	3* → 3	3	2*	2*	2* → 2	2	2*	2	2*
-	- → -	-	1*	1 → 1	1	4*	4*	4	4	5*	5*
PF	PF		PF	PF	PF	PF		PF		PF	

5PF + 3PF  
Total= 8PF

# Memoria Virtual

## Algoritmos de reemplazo:

OPTIMO      **3PF + 3PF**  
                 **Total= 6PF**

FIFO          **6PF + 3PF**  
                 **Total= 9PF**

LRU          **4PF + 3PF**  
                 **Total= 7PF**

CLOCK       **5PF + 3PF**  
                 **Total= 8PF**