



Gestión de Memoria en Procesadores Modernos

Enfoque teórico general

Alejandro Furfaro

22 de agosto de 2023

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Temario

1 Introducción

- **Enfoque preliminar**
- Organización y Administración

2 Abstracción de memoria

- Espacio de abstracción
- Gestión de la Memoria

3 Administración de memoria

- La Memory Management Unit
- Gestión del espacio libre
- Políticas de Administración de memoria

4 Memoria Virtual

- Antecedentes
- Swapping
- Organización en bloques
- Políticas de Administración de memoria

5 Relación con el hardware

- MMU
- Page faults

6 Paginación en un Sistema Operativo Real: Linux

- Lineamientos generales de administración de memoria
- Copy-on-write

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

- cantidad ilimitada

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

- cantidad ilimitada
- accesible de manera instantánea,

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

- cantidad ilimitada
- accesible de manera instantánea,
- costo irrelevante,

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

- cantidad ilimitada
- accesible de manera instantánea,
- costo irrelevante,
- consumo de energía insignificante. . .

La memoria en un computador

Aspiraciones de los programadores de aplicaciones respecto de la memoria

- cantidad ilimitada
- accesible de manera instantánea,
- costo irrelevante,
- consumo de energía insignificante. . .
- y si no es mucho pedir. . . no volátil.

La memoria en un computador

- Como arquitectos de computadores, tratamos de aproximar el comportamiento del sistema a ese ideal

La memoria en un computador

- Como arquitectos de computadores, tratamos de aproximar el comportamiento del sistema a ese ideal
- Sin embargo, los requerimientos anteriores en algunos casos se contraponen.

La memoria en un computador

- Como arquitectos de computadores, tratamos de aproximar el comportamiento del sistema a ese ideal
- Sin embargo, los requerimientos anteriores en algunos casos se contraponen.
- Trade offs permanentes

Temario

1 Introducción

- Enfoque preliminar
- **Organización y Administración**

2 Abstracción de memoria

- Espacio de abstracción
- Gestión de la Memoria

3 Administración de memoria

- La Memory Management Unit
- Gestión del espacio libre
- Políticas de Administración de memoria

4 Memoria Virtual

- Antecedentes
- Swapping
- Organización en bloques
- Políticas de Administración de memoria

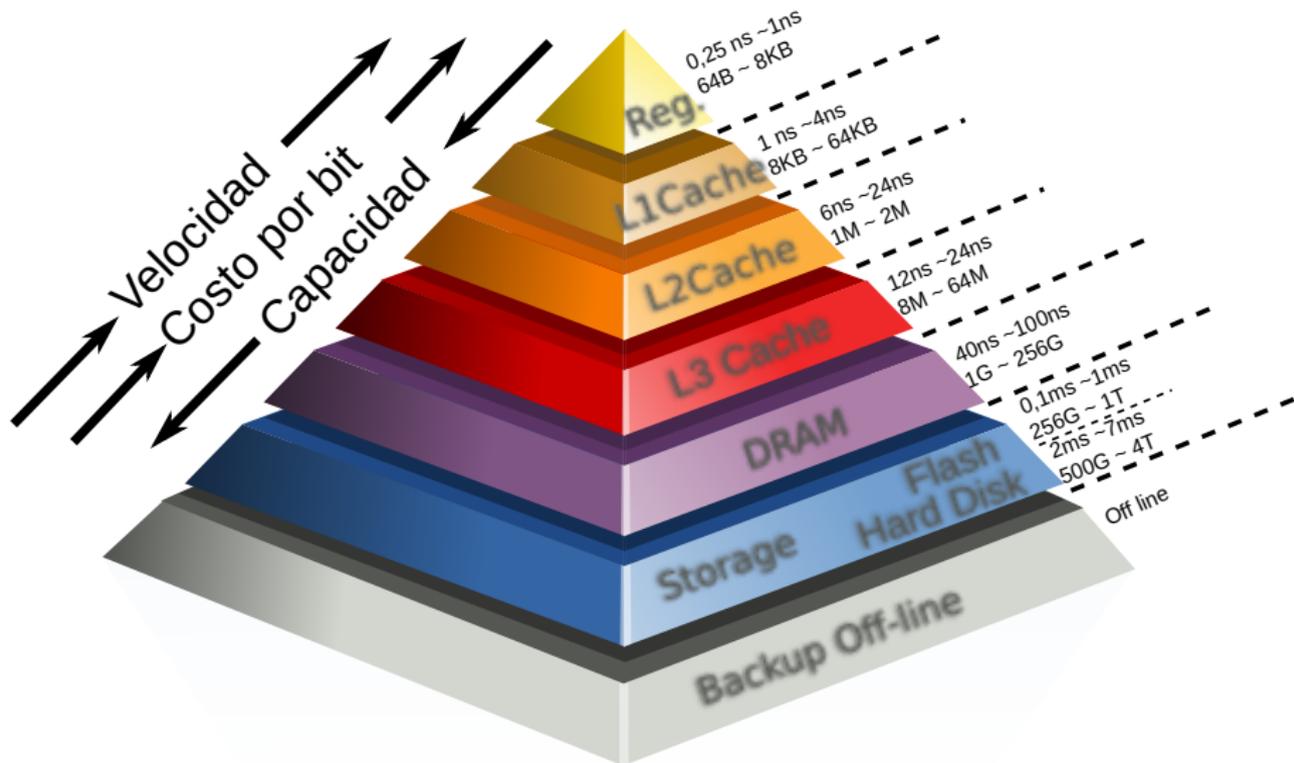
5 Relación con el hardware

- MMU
- Page faults

6 Paginación en un Sistema Operativo Real: Linux

- Lineamientos generales de administración de memoria
- Copy-on-write

Organización Jerárquica



Administración: El Memory Manager

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física
- Llevar el control de los bloques de memoria libres y ocupados

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física
- Llevar el control de los bloques de memoria libres y ocupados
- Asignar a demanda a cada proceso bloques de memoria

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física
- Llevar el control de los bloques de memoria libres y ocupados
- Asignar a demanda a cada proceso bloques de memoria
- Liberar bloques de memoria cuando el proceso los devuelve

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física
- Llevar el control de los bloques de memoria libres y ocupados
- Asignar a demanda a cada proceso bloques de memoria
- Liberar bloques de memoria cuando el proceso los devuelve
- Protección de los bloques de cada proceso

Administración: El Memory Manager

- Es uno de los componentes esenciales de un Sistema Operativo.
- Manejar Abstracción respecto de la memoria Física
- Llevar el control de los bloques de memoria libres y ocupados
- Asignar a demanda a cada proceso bloques de memoria
- Liberar bloques de memoria cuando el proceso los devuelve
- Protección de los bloques de cada proceso
- Swapping

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - **Espacio de abstracción**
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Abstracción: ¿Que significa?

Abstracción: ¿Que significa?

- Un programa ejecuta `mov R8, [10000]`.

Abstracción: ¿Que significa?

- Un programa ejecuta `mov R8, [10000]`.
- ¿Que dirección de memoria se copia en el registro `R8`?

Abstracción: ¿Que significa?

- Un programa ejecuta `mov R8, [10000]`.
- ¿Que dirección de memoria se copia en el registro `R8`?

Rta1: La dirección 10000,
siempre

Abstracción: ¿Que significa?

- Un programa ejecuta `mov R8, [10000]`.
- ¿Que dirección de memoria se copia en el registro `R8`?

Rta1: La dirección 10000,
siempre

Rta2: cualquier dirección
libre *y siempre diferente*

Abstracción: ¿Que significa?

- Un programa ejecuta `mov R8, [10000]`.
- ¿Que dirección de memoria se copia en el registro `R8`?

Rta1: La dirección 10000,
siempre

Rta2: cualquier dirección
libre *y siempre diferente*

La Rta 1 es correcta para un conjunto CPU + SO sin abstracción.
La Rta 2 es correcta para un conjunto CPU + SO con abstracción.

Abstracción y Multitasking

Abstracción y Multitasking

Sin capacidad de Abstracción implementar multitasking es imposible (o al menos se implementaría de forma muy limitada), ya que cualquier proceso tiene la visión física de toda la memoria (incluso al área de memoria del sistema operativo).

Dirección Lógica, Virtual, Física....

Dirección Lógica, Virtual, Física....

- ¿Como es que hay diferentes definiciones de direcciones?. ¿No son todas lo mismo?

Dirección Lógica, Virtual, Física....

- ¿Como es que hay diferentes definiciones de direcciones?. ¿No son todas lo mismo?
- No siempre. Cuando un procesador maneja Abstracción de memoria, definitivamente son direcciones diferentes.

Dirección Lógica, Virtual, Física....

- ¿Como es que hay diferentes definiciones de direcciones?. ¿No son todas lo mismo?
- No siempre. Cuando un procesador maneja Abstracción de memoria, definitivamente son direcciones diferentes.
- En los microcontroladores sencillos o de arquitecturas muy primitivas, estos conceptos significan lo mismo. No hay Abstracción.

Dirección Lógica, Virtual, Física....

- ¿Como es que hay diferentes definiciones de direcciones?. ¿No son todas lo mismo?
- No siempre. Cuando un procesador maneja Abstracción de memoria, definitivamente son direcciones diferentes.
- En los microcontroladores sencillos o de arquitecturas muy primitivas, estos conceptos significan lo mismo. No hay Abstracción.
- A medida que los procesadores se diseñan para soportar Sistemas Operativos con ejecución dinámica de tareas, estos tres valores suelen ser diferentes, a costa de mayor complejidad en su arquitectura, que requiere Abstracción de la memoria física.

Dirección Física

Dirección Física

- Cuando cualquier procesador necesita acceder a memoria, deberá escribir en el registro de direcciones el valor correspondiente a la dirección de memoria externa que debe acceder.

Dirección Física

- Cuando cualquier procesador necesita acceder a memoria, deberá escribir en el registro de direcciones el valor correspondiente a la dirección de memoria externa que debe acceder.
- Este valor se conoce como Dirección Física, ya que corresponde a la dirección que será decodificada por el hardware externo para acceder a la memoria RAM o ROM según corresponda. Es decir a la memoria física.

Dirección Física

- Cuando cualquier procesador necesita acceder a memoria, deberá escribir en el registro de direcciones el valor correspondiente a la dirección de memoria externa que debe acceder.
- Este valor se conoce como Dirección Física, ya que corresponde a la dirección que será decodificada por el hardware externo para acceder a la memoria RAM o ROM según corresponda. Es decir a la memoria física.
- Siempre el procesador pone la Dirección Física en los pines de address cuando su unidad de control habilita la salida del bus de direcciones

¿Que dirección manejamos en los programas?

¿Que dirección manejamos en los programas?

- El texto de nuestros programas fuente provee implícitamente una visión mas o menos abstracta de la memoria.

¿Que dirección manejamos en los programas?

- El texto de nuestros programas fuente provee implícitamente una visión mas o menos abstracta de la memoria.
- En mayor o menor medida intentamos abstraernos del valor numérico de la dirección, aun cuando programamos los procesadores mas rudimentarios

¿Que dirección manejamos en los programas?

- El texto de nuestros programas fuente provee implícitamente una visión mas o menos abstracta de la memoria.
- En mayor o menor medida intentamos abstraernos del valor numérico de la dirección, aun cuando programamos los procesadores mas rudimentarios

```
1 CANT_ELEMENTS .equ 1024
2 SIZE_ELEMENTS .equ 4
3 /*define 1024 words de memoria en 0*/
4 buffer: .skip SIZE_ELEMENTS * CANT_ELEMENTS , 0
5 .....
6     LDR r1,buffer    /*R1 apunta a buffer*/
7     .....
```

¿Que dirección manejamos en los programas?

- El texto de nuestros programas fuente provee implícitamente una visión mas o menos abstracta de la memoria.
- En mayor o menor medida intentamos abstraernos del valor numérico de la dirección, aun cuando programamos los procesadores mas rudimentarios

```
1 CANT_ELEMENTS .equ 1024
2 SIZE_ELEMENTS .equ 4
3 /*define 1024 words de memoria en 0*/
4 buffer: .skip SIZE_ELEMENTS * CANT_ELEMENTS , 0
5 .....
6     LDR r1,buffer    /*R1 apunta a buffer*/
7 .....
```

- En este código se define un buffer de 1024 bytes.

¿Que dirección manejamos en los programas?

- El texto de nuestros programas fuente provee implícitamente una visión mas o menos abstracta de la memoria.
- En mayor o menor medida intentamos abstraernos del valor numérico de la dirección, aun cuando programamos los procesadores mas rudimentarios

```
1 CANT_ELEMENTS .equ 1024
2 SIZE_ELEMENTS .equ 4
3 /*define 1024 words de memoria en 0*/
4 buffer: .skip SIZE_ELEMENTS * CANT_ELEMENTS , 0
5 .....
6     LDR r1,buffer /*R1 apunta a buffer*/
7 .....
```

- En este código se define un buffer de 1024 bytes.
- ¿Cual es su dirección de Memoria?

Dirección Lógica

Definición

Dirección Lógica

Definición

- Es la dirección de memoria expresada en términos abstractos por el programador en su código fuente.

Dirección Lógica

Definición

- Es la dirección de memoria expresada en términos abstractos por el programador en su código fuente.
- Es una forma muy simple de trabajar en forma transparente a la memoria.

Dirección Lógica

Definición

- Es la dirección de memoria expresada en términos abstractos por el programador en su código fuente.
- Es una forma muy simple de trabajar en forma transparente a la memoria.
- En lugar de referirse al valor numérico de la dirección de hardware, reemplazamos este valor por una etiqueta que incluso resulte ilustrativa de lo que representa esa dirección, y dejamos que el toolchain resuelva su valor numérico.

¿Que es la dirección virtual?

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).
- Sin embargo, en algunos procesadores las direcciones lógica y la virtual son diferentes.

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).
- Sin embargo, en algunos procesadores las direcciones lógica y la virtual son diferentes.
- Obviamente me refiero a los procesadores x86 de Intel, cuya gestión de memoria es sumamente particular y un poco mas compleja.

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).
- Sin embargo, en algunos procesadores las direcciones lógica y la virtual son diferentes.
- Obviamente me refiero a los procesadores x86 de Intel, cuya gestión de memoria es sumamente particular y un poco mas compleja.
- Intel refiere a la dirección virtual como dirección lineal.

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).
- Sin embargo, en algunos procesadores las direcciones lógica y la virtual son diferentes.
- Obviamente me refiero a los procesadores x86 de Intel, cuya gestión de memoria es sumamente particular y un poco mas compleja.
- Intel refiere a la dirección virtual como dirección lineal.
- Lo importante: En un procesador con Abstracción, la dirección virtual no coincide (generalmente) con la dirección Física.

¿Que es la dirección virtual?

- Generalmente coincide con la dirección lógica (Aun en procesadores de alta gama).
- Sin embargo, en algunos procesadores las direcciones lógica y la virtual son diferentes.
- Obviamente me refiero a los procesadores x86 de Intel, cuya gestión de memoria es sumamente particular y un poco mas compleja.
- Intel refiere a la dirección virtual como dirección lineal.
- Lo importante: En un procesador con Abstracción, la dirección virtual no coincide (generalmente) con la dirección Física.
- En ocasiones ambas coinciden. Se debe a que programador del sistema establece de manera deliberada y por lo general, transitoria, una organización que denominamos *identity mapping*.

Soporte de la CPU para implementar la Abstracción

Soporte de la CPU para implementar la Abstracción

- La traducción de la *Memoria Virtual* a la *Memoria Física*, es la base del modelo de Abstracción.

Soporte de la CPU para implementar la Abstracción

- La traducción de la **Memoria Virtual** a la **Memoria Física**, es la base del modelo de Abstracción.
- Puede pensarse como una tabla a la que se ingresa el valor de la **Dirección Virtual**, y se obtiene un número que termina siendo el marco del bloque de **la Memoria Física**

Soporte de la CPU para implementar la Abstracción

- La traducción de la **Memoria Virtual** a la **Memoria Física**, es la base del modelo de Abstracción.
- Puede pensarse como una tabla a la que se ingresa el valor de la **Dirección Virtual**, y se obtiene un número que termina siendo el marco del bloque de **la Memoria Física**
- El desplazamiento a partir del inicio del marco obtenido son los bits menos significativos de la **Dirección Virtual**.

Soporte de la CPU para implementar la Abstracción

- La traducción de la **Memoria Virtual** a la **Memoria Física**, es la base del modelo de Abstracción.
- Puede pensarse como una tabla a la que se ingresa el valor de la **Dirección Virtual**, y se obtiene un número que termina siendo el marco del bloque de **la Memoria Física**
- El desplazamiento a partir del inicio del marco obtenido son los bits menos significativos de la **Dirección Virtual**.

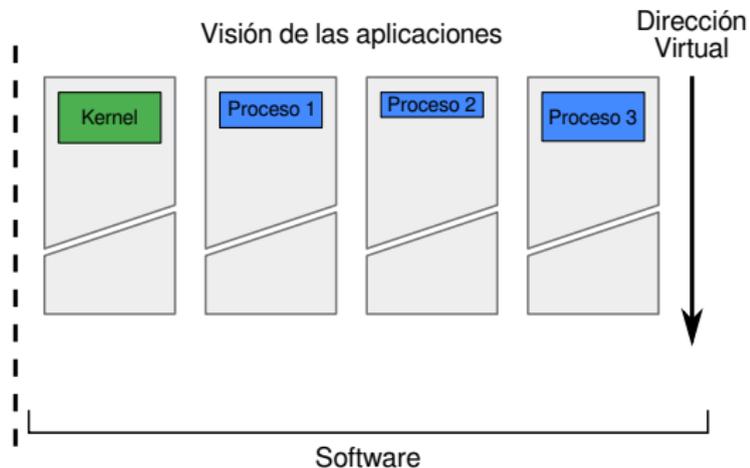
Podemos hacerlo por software... pero si logramos hacerlo por hardware, ¡sería mucho mas veloz!. Por eso la MMU es esencial en un procesador.

Temario

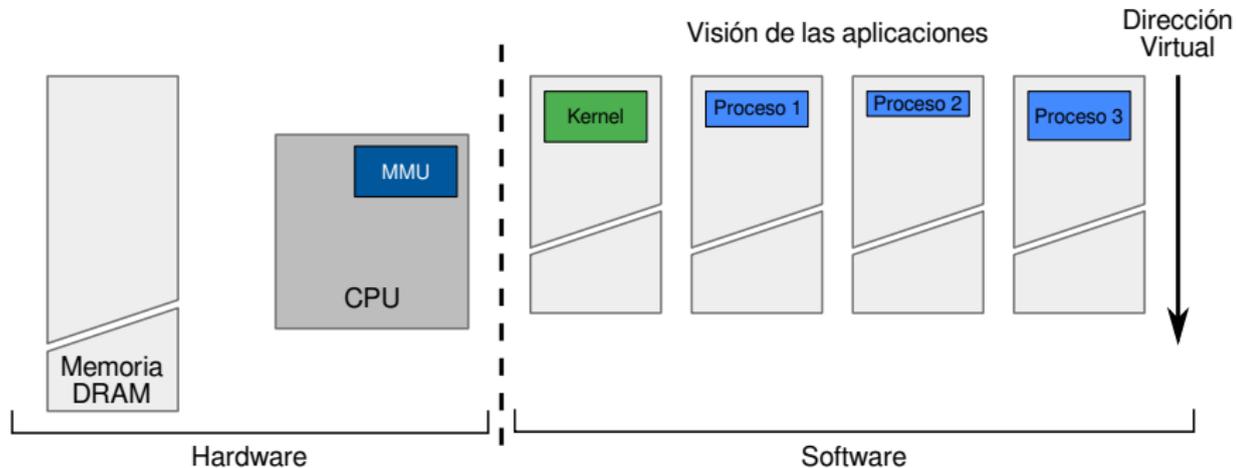
- 1 Introducción
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - **Gestión de la Memoria**
- 3 Administración de memoria
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 Memoria Virtual
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 Relación con el hardware
 - MMU
 - Page faults
- 6 Paginación en un Sistema Operativo Real: Linux
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Abstracción y Multitasking

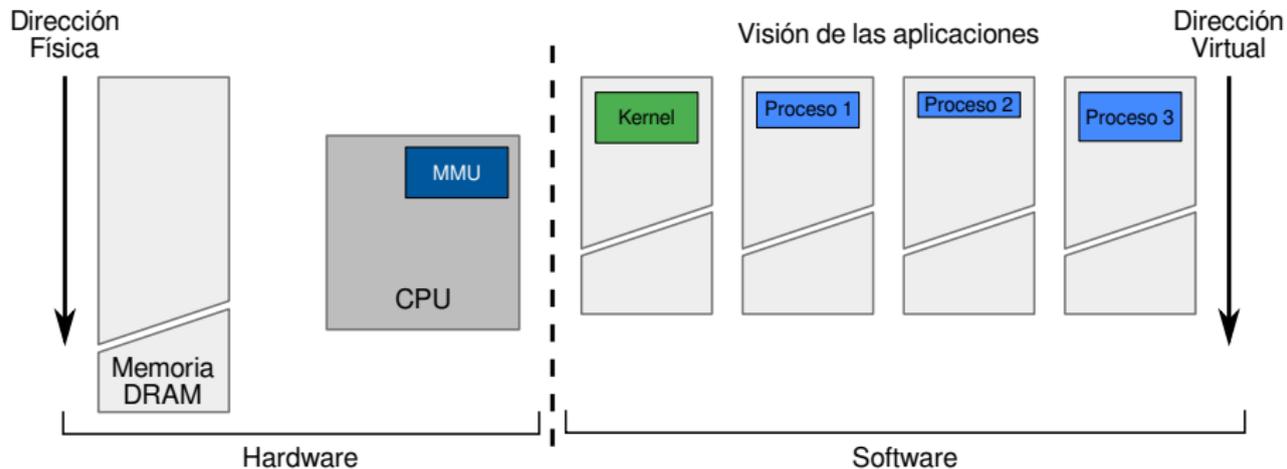
Abstracción y Multitasking



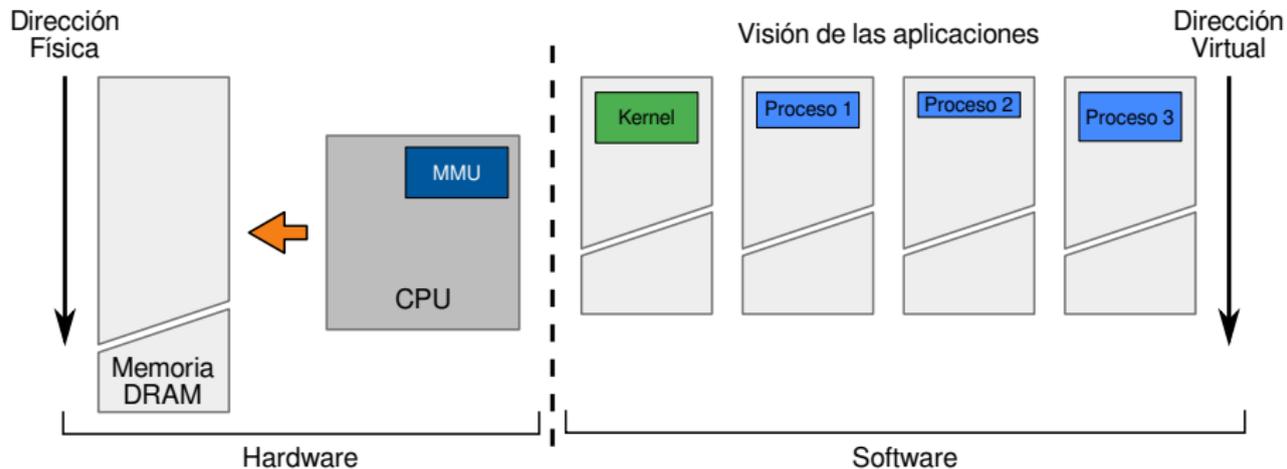
Abstracción y Multitasking



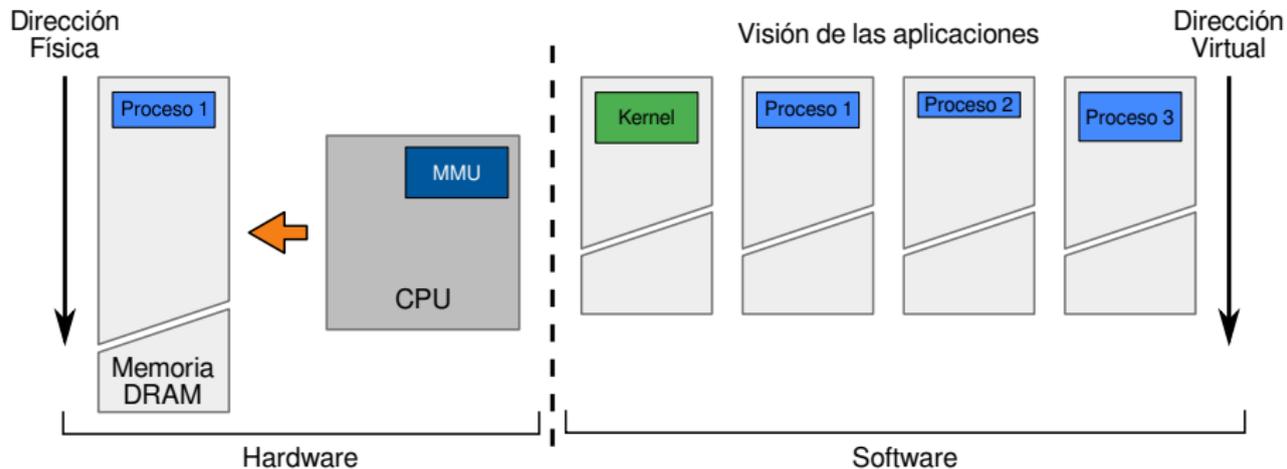
Abstracción y Multitasking



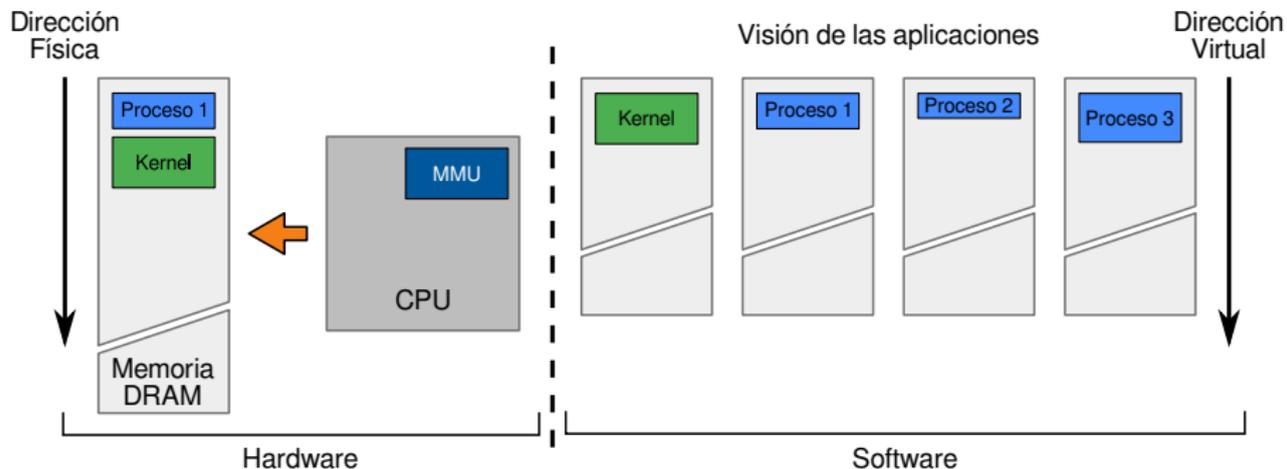
Abstracción y Multitasking



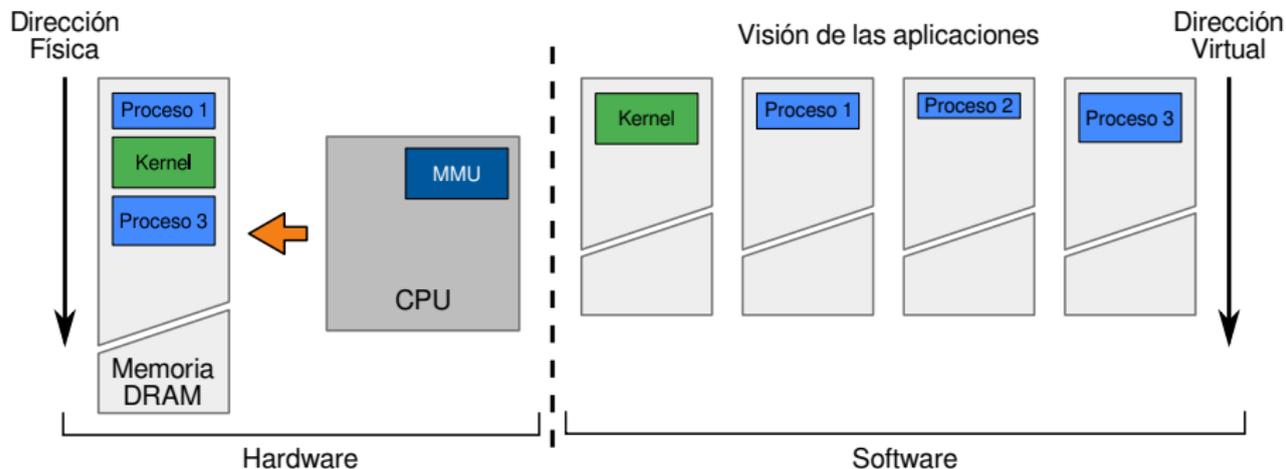
Abstracción y Multitasking



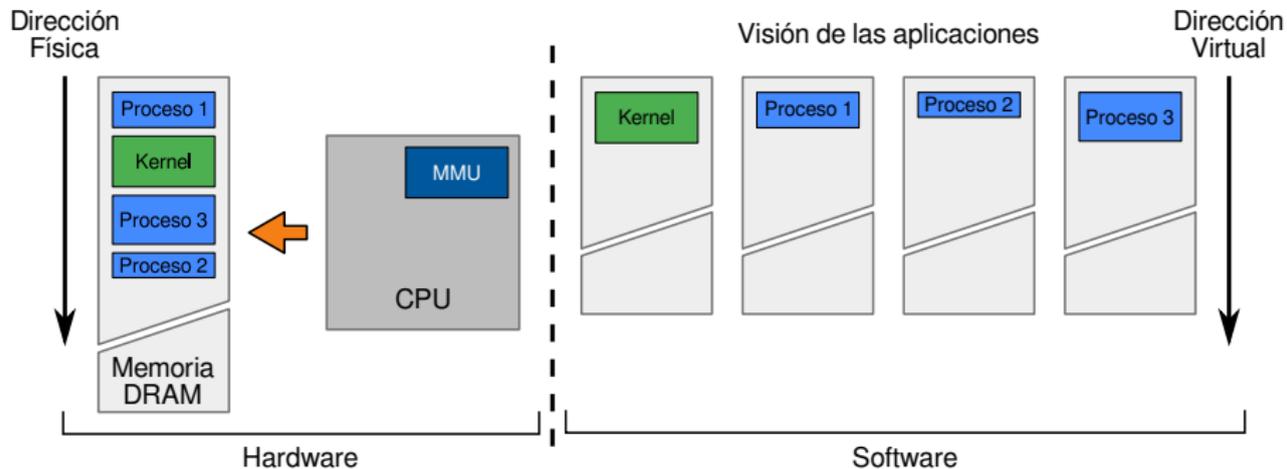
Abstracción y Multitasking



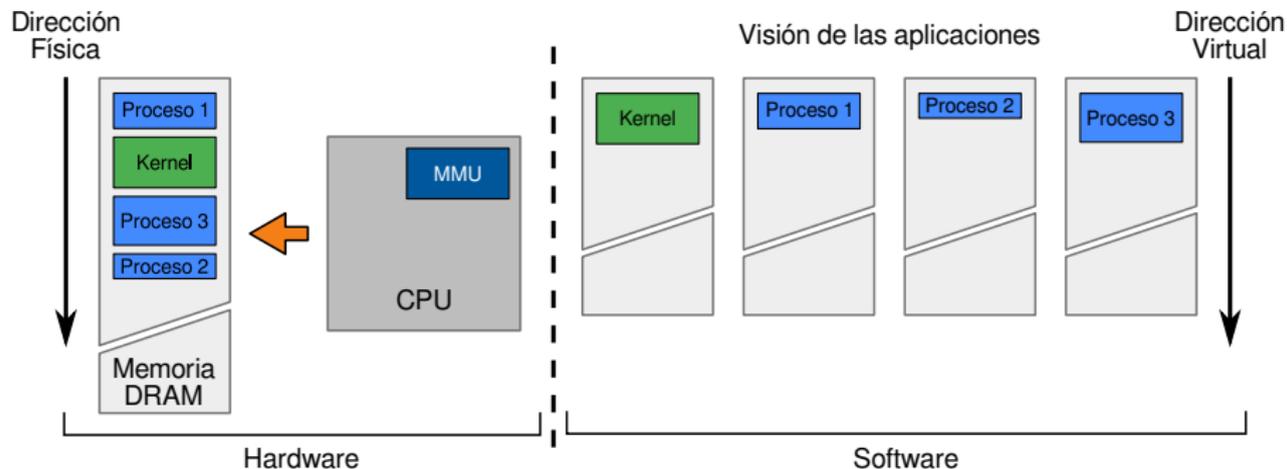
Abstracción y Multitasking



Abstracción y Multitasking



Abstracción y Multitasking



La abstracción se lleva a cabo por medio del Memory Manager del S.O. utilizando recursos de hardware (la MMU) provistos por la CPU (Interfaz Hardware - Software).

Memory Management Unit

Memory Management Unit

- La MMU establece el mapeo entre la Dirección Lógica de un elemento y la Dirección Física que ocupará realmente en la memoria.

Memory Management Unit

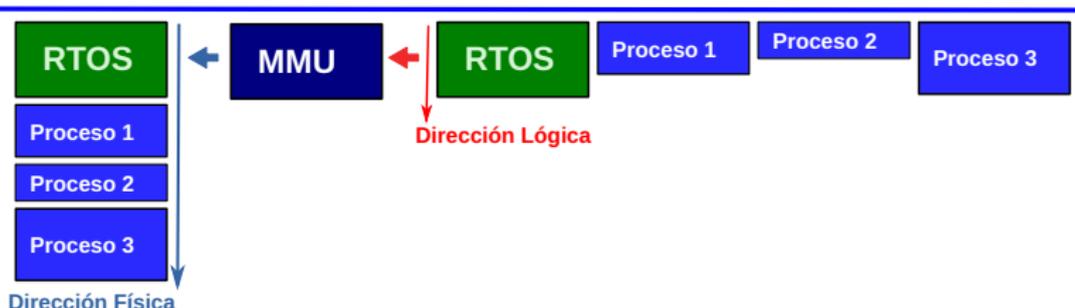
- La MMU establece el mapeo entre la Dirección Lógica de un elemento y la Dirección Física que ocupará realmente en la memoria.
- Organiza la memoria física en bloques para su mejor administración.

Memory Management Unit

- La MMU establece el mapeo entre la Dirección Lógica de un elemento y la Dirección Física que ocupará realmente en la memoria.
- Organiza la memoria física en bloques para su mejor administración.
- Permite al S.O. asignar eventualmente el mismo espacio de direcciones lógicas a los procesos, y separarlos una vez distribuidos en la memoria física, evitando además que se interfieran entre sí.

Memory Management Unit

- La MMU establece el mapeo entre la Dirección Lógica de un elemento y la Dirección Física que ocupará realmente en la memoria.
- Organiza la memoria física en bloques para su mejor administración.
- Permite al S.O. asignar eventualmente el mismo espacio de direcciones lógicas a los procesos, y separarlos una vez distribuidos en la memoria física, evitando además que se interfieran entre sí.



Conclusión. Espacio de direcciones

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).
- El espacio de direccionamiento de los procesos es entonces un espacio virtual (abstracto).

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).
- El espacio de direccionamiento de los procesos es entonces un espacio virtual (abstracto).
- Las direcciones virtuales pueden ser las mismas para los procesos pero las direcciones físicas son únicas y diferentes

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).
- El espacio de direccionamiento de los procesos es entonces un espacio virtual (abstracto).
- Las direcciones virtuales pueden ser las mismas para los procesos pero las direcciones físicas son únicas y diferentes
- Se requiere:

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).
- El espacio de direccionamiento de los procesos es entonces un espacio virtual (abstracto).
- Las direcciones virtuales pueden ser las mismas para los procesos pero las direcciones físicas son únicas y diferentes
- Se requiere:
 - Protección

Conclusión. Espacio de direcciones

- El espacio de direccionamiento físico no se expone a los procesos. (Los inconvenientes son muchos y muy graves).
- El espacio de direccionamiento de los procesos es entonces un espacio virtual (abstracto).
- Las direcciones virtuales pueden ser las mismas para los procesos pero las direcciones físicas son únicas y diferentes
- Se requiere:
 - Protección
 - Reubicación (Relocation)

Una MMU hace la diferencia

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).
- En este caso cada tarea tiene una o mas áreas de memoria para su código y datos.

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).
- En este caso cada tarea tiene una o mas áreas de memoria para su código y datos.
- Cuando el Scheduler retoma la ejecución de una tarea, la MMU mapea el espacio de direccionamiento lógico de ésta (que comienza en 0), en un área de memoria física exclusiva y diferente de la de las demás tareas y de la del kernel.

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).
- En este caso cada tarea tiene una o mas áreas de memoria para su código y datos.
- Cuando el Scheduler retoma la ejecución de una tarea, la MMU mapea el espacio de direccionamiento lógico de ésta (que comienza en 0), en un área de memoria física exclusiva y diferente de la de las demás tareas y de la del kernel.
- De este modo cada tarea (proceso) gana su espacio de memoria físico exclusivo y protegido el del resto de las tareas (procesos).

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).
- En este caso cada tarea tiene una o mas áreas de memoria para su código y datos.
- Cuando el Scheduler retoma la ejecución de una tarea, la MMU mapea el espacio de direccionamiento lógico de ésta (que comienza en 0), en un área de memoria física exclusiva y diferente de la de las demás tareas y de la del kernel.
- De este modo cada tarea (proceso) gana su espacio de memoria físico exclusivo y protegido el del resto de las tareas (procesos).
- A cambio, se genera un overhead al tener que remapear la memoria cada vez que se conmuta de un proceso al siguiente.

Una MMU hace la diferencia

- La MMU es fundamental para implementar el modelo de proceso en un S.O. (como es el caso de Linux).
- En este caso cada tarea tiene una o mas áreas de memoria para su código y datos.
- Cuando el Scheduler retoma la ejecución de una tarea, la MMU mapea el espacio de direccionamiento lógico de ésta (que comienza en 0), en un área de memoria física exclusiva y diferente de la de las demás tareas y de la del kernel.
- De este modo cada tarea (proceso) gana su espacio de memoria físico exclusivo y protegido el del resto de las tareas (procesos).
- A cambio, se genera un overhead al tener que remapear la memoria cada vez que se conmuta de un proceso al siguiente.
- La MMU asegura a cada tarea (proceso), la visibilidad de su propia área de memoria, y de las partes relevantes del S.O.

Visión de la Memoria y la MMU

Visión de la Memoria y la MMU

- Como usuarios nuestra visión de la memoria es por bloques. El tamaño de éstos no nos preocupa demasiado.

Visión de la Memoria y la MMU

- Como usuarios nuestra visión de la memoria es por bloques. El tamaño de éstos no nos preocupa demasiado.
- La organización lógica de nuestros programas se basa en bloques de código (organizados en funciones), y bloques de datos. Las direcciones dentro de cada bloque se representan mediante etiquetas y las herramientas de desarrollo las calculan en forma relativa al inicio del bloque.

Visión de la Memoria y la MMU

- Como usuarios nuestra visión de la memoria es por bloques. El tamaño de éstos no nos preocupa demasiado.
- La organización lógica de nuestros programas se basa en bloques de código (organizados en funciones), y bloques de datos. Las direcciones dentro de cada bloque se representan mediante etiquetas y las herramientas de desarrollo las calculan en forma relativa al inicio del bloque.
- De este modo nos referimos a “el buffer de memoria en donde se reciben los datos”, “el programa principal”, “la función tal o cual”...

Visión de la Memoria y la MMU

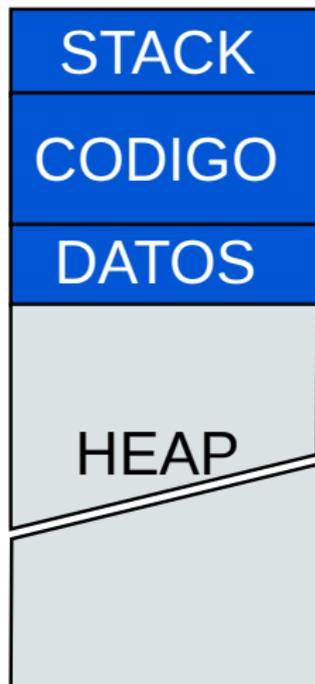
- Como usuarios nuestra visión de la memoria es por bloques. El tamaño de éstos no nos preocupa demasiado.
- La organización lógica de nuestros programas se basa en bloques de código (organizados en funciones), y bloques de datos. Las direcciones dentro de cada bloque se representan mediante etiquetas y las herramientas de desarrollo las calculan en forma relativa al inicio del bloque.
- De este modo nos referimos a “el buffer de memoria en donde se reciben los datos”, “el programa principal”, “la función tal o cual”...
- La dirección de memoria en la que se encuentran, no está clara.

Visión de la Memoria y la MMU

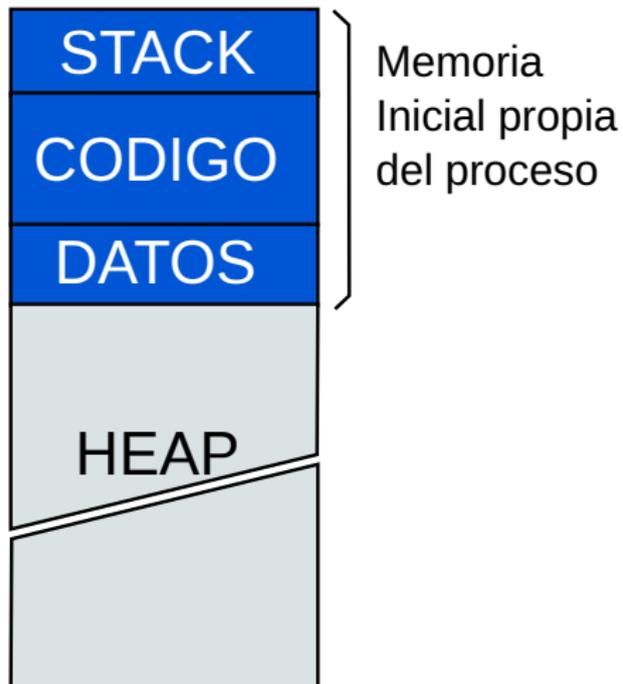
- Como usuarios nuestra visión de la memoria es por bloques. El tamaño de éstos no nos preocupa demasiado.
- La organización lógica de nuestros programas se basa en bloques de código (organizados en funciones), y bloques de datos. Las direcciones dentro de cada bloque se representan mediante etiquetas y las herramientas de desarrollo las calculan en forma relativa al inicio del bloque.
- De este modo nos referimos a “el buffer de memoria en donde se reciben los datos”, “el programa principal”, “la función tal o cual”...
- La dirección de memoria en la que se encuentran, no está clara.
- De hecho cuando programamos en Linux el formato de los programas (ELF) organiza a éstos en secciones: `.text` para el código, `.data` para variables inicializadas, `.rodata` para constantes, `.symbol` para tablas de símbolos, etc

Visión de la memoria para un proceso

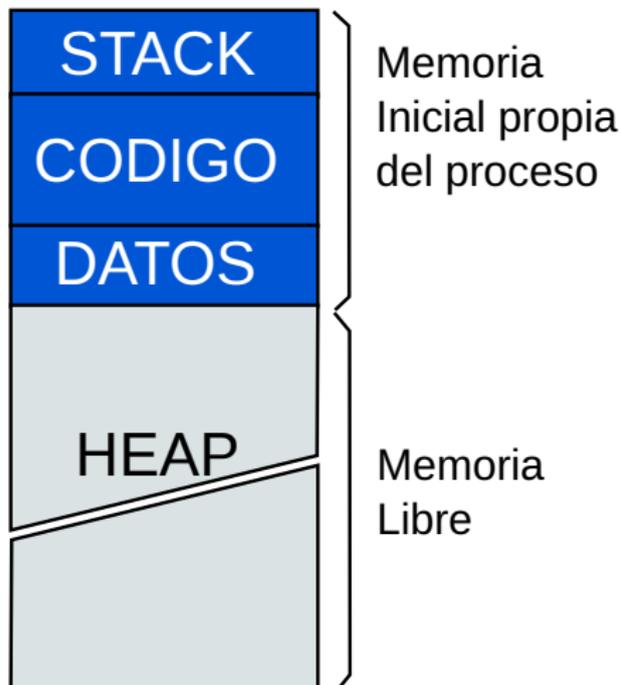
Visión de la memoria para un proceso



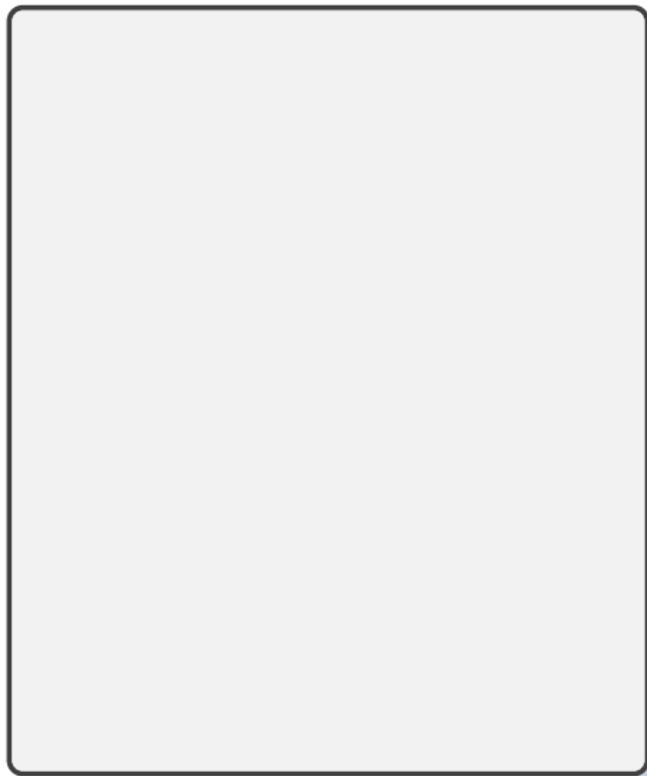
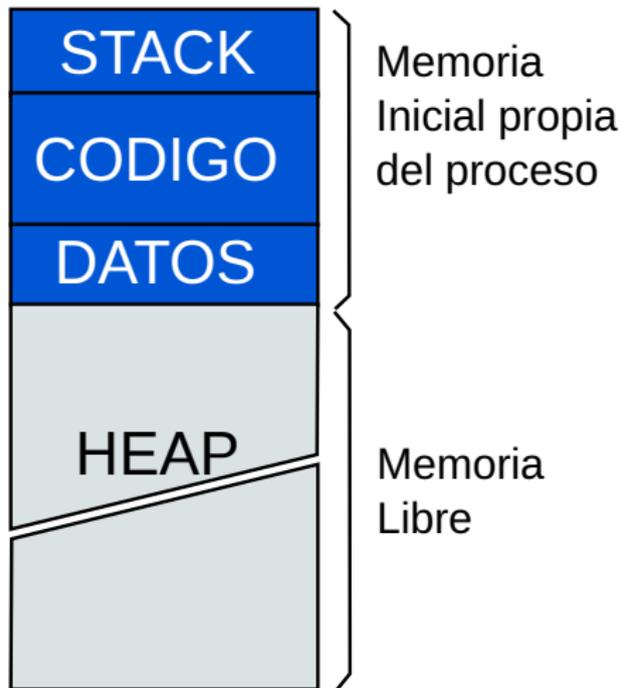
Visión de la memoria para un proceso



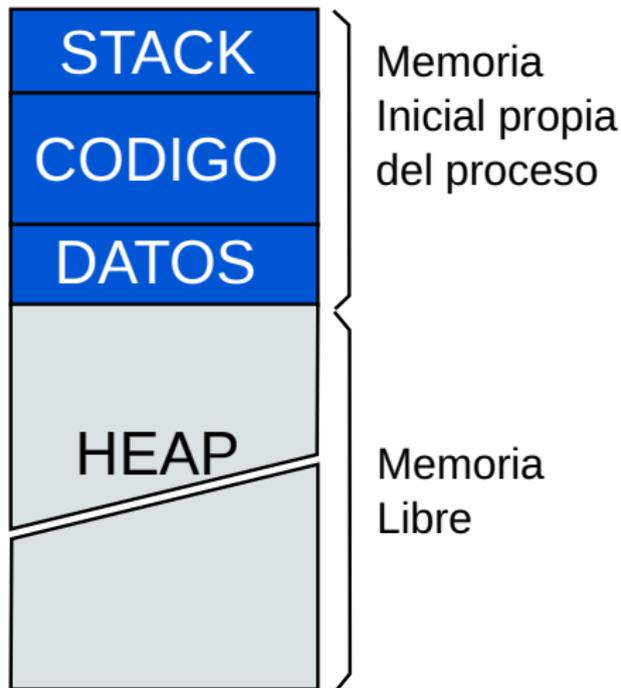
Visión de la memoria para un proceso



Visión de la memoria para un proceso

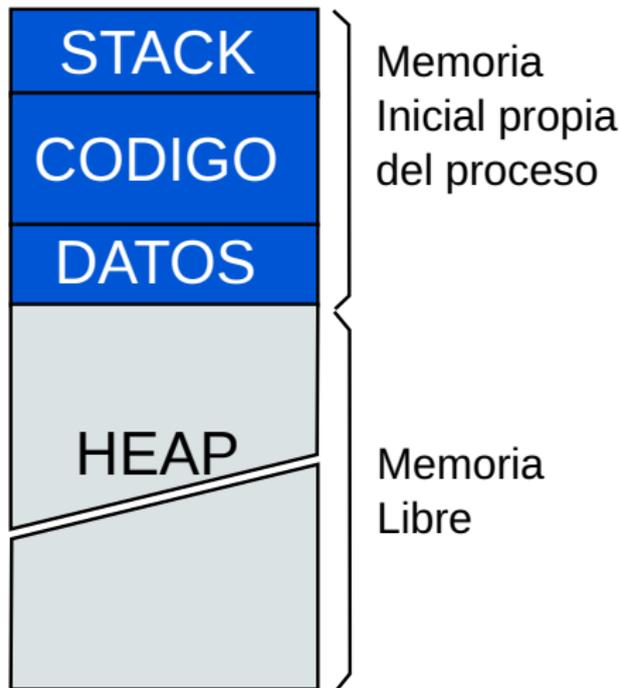


Visión de la memoria para un proceso



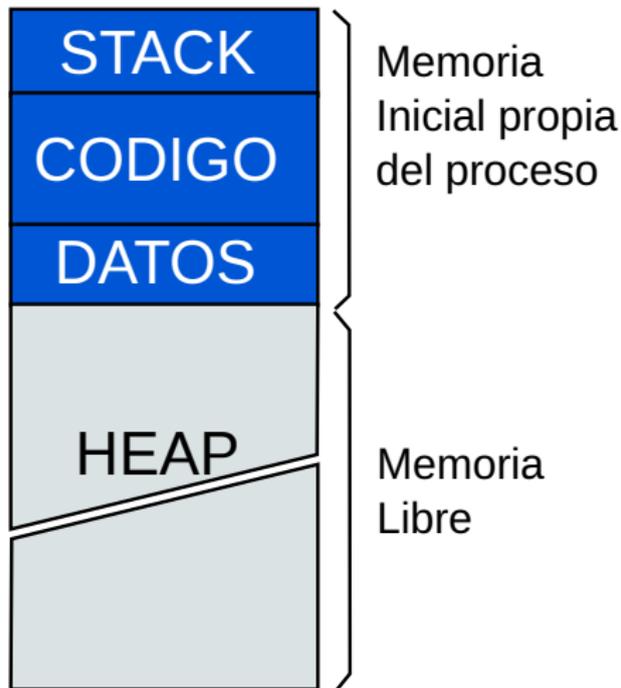
- Cada proceso tiene como mínimo una sección para código y otra para stack.

Visión de la memoria para un proceso



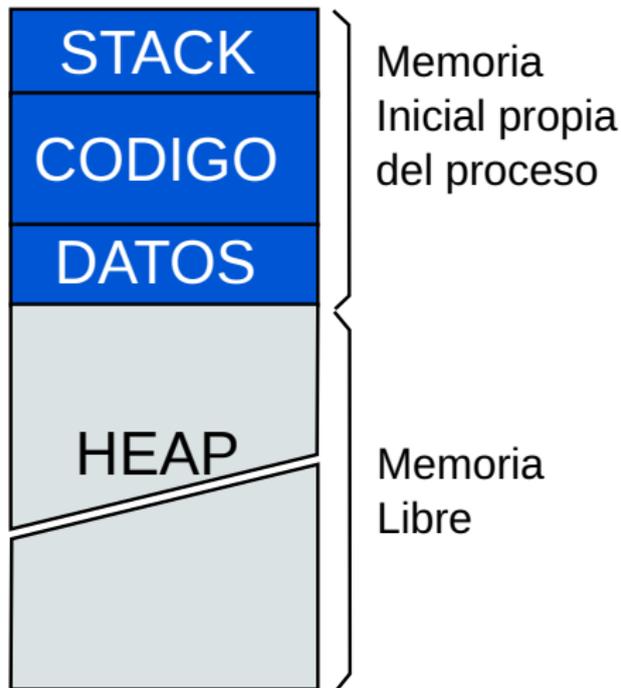
- Cada proceso tiene como mínimo una sección para código y otra para stack.
- La sección de datos, es opcional.

Visión de la memoria para un proceso



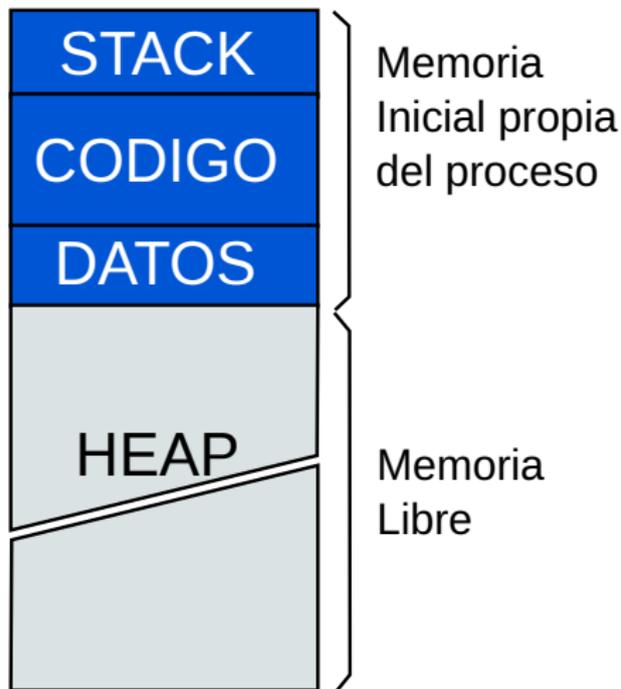
- Cada proceso tiene como mínimo una sección para código y otra para stack.
- La sección de datos, es opcional.
- El Heap es memoria disponible.

Visión de la memoria para un proceso



- Cada proceso tiene como mínimo una sección para código y otra para stack.
- La sección de datos, es opcional.
- El Heap es memoria disponible.
- El S.O. la administra a demanda de los procesos.

Visión de la memoria para un proceso



- Cada proceso tiene como mínimo una sección para código y otra para stack.
- La sección de datos, es opcional.
- El Heap es memoria disponible.
- El S.O. la administra a demanda de los procesos.
- Los procesos asumen un Heap de tamaño ilimitado.

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - **La Memory Management Unit**
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Administración de Memoria con MMU

Administración de Memoria con MMU

- En general no nos interesa demasiado la ubicación en memoria física de los diferentes bloques de nuestros programas.

Administración de Memoria con MMU

- En general no nos interesa demasiado la ubicación en memoria física de los diferentes bloques de nuestros programas.
- Intrínsecamente aceptamos que el Sistema Operativo los ubique donde le resulte mas apropiado, de acuerdo con sus criterios de administración de memoria.

Administración de Memoria con MMU

- En general no nos interesa demasiado la ubicación en memoria física de los diferentes bloques de nuestros programas.
- Intrínsecamente aceptamos que el Sistema Operativo los ubique donde le resulte mas apropiado, de acuerdo con sus criterios de administración de memoria.
- La división de la memoria en bloques, se puede realizar mediante dos criterios diferentes:

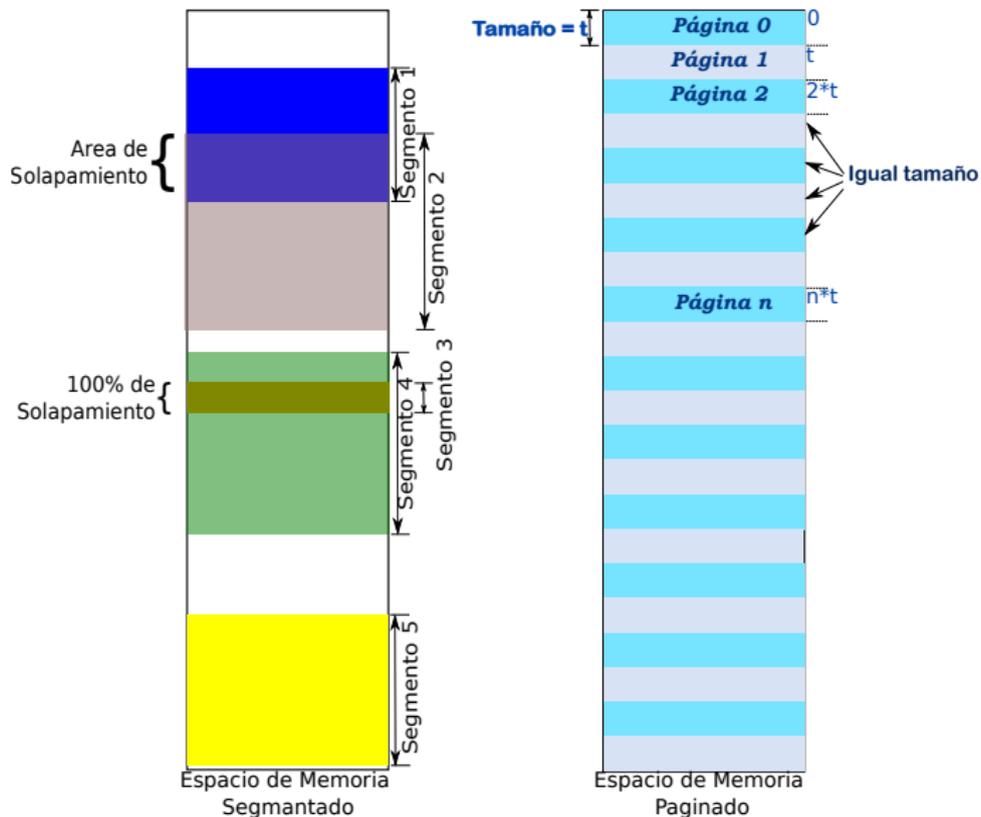
Administración de Memoria con MMU

- En general no nos interesa demasiado la ubicación en memoria física de los diferentes bloques de nuestros programas.
- Intrínsecamente aceptamos que el Sistema Operativo los ubique donde le resulte mas apropiado, de acuerdo con sus criterios de administración de memoria.
- La división de la memoria en bloques, se puede realizar mediante dos criterios diferentes:
 - Segmentación.

Administración de Memoria con MMU

- En general no nos interesa demasiado la ubicación en memoria física de los diferentes bloques de nuestros programas.
- Intrínsecamente aceptamos que el Sistema Operativo los ubique donde le resulte mas apropiado, de acuerdo con sus criterios de administración de memoria.
- La división de la memoria en bloques, se puede realizar mediante dos criterios diferentes:
 - Segmentación.
 - Paginación.

Segmentación vs. Paginación



Administración de Memoria por Segmentación

- Provee una visión de la memoria compatible con la del usuario.

Administración de Memoria por Segmentación

- Provee una visión de la memoria compatible con la del usuario.
- Organiza el espacio de direcciones en segmentos de diferente tamaño, identificados por un valor numérico.

Administración de Memoria por Segmentación

- Provee una visión de la memoria compatible con la del usuario.
- Organiza el espacio de direcciones en segmentos de diferente tamaño, identificados por un valor numérico.
- Para identificar un ítem dentro de un segmento se necesitan dos valores: El número de segmento y el desplazamiento relativo al origen del segmento del ítem que se intenta identificar.

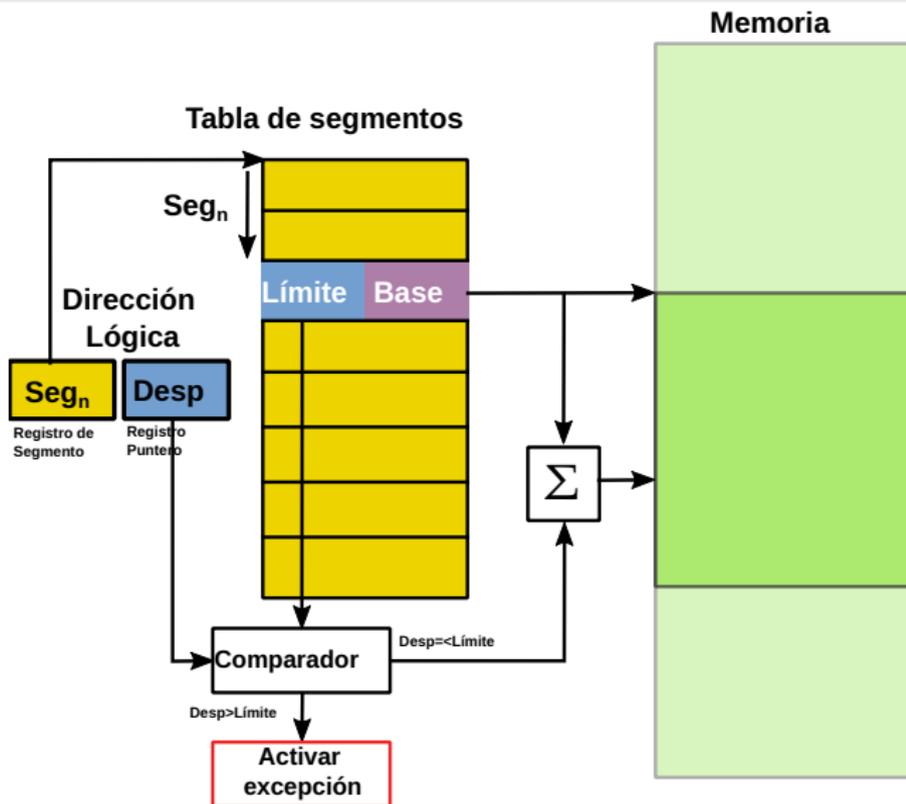
Administración de Memoria por Segmentación

- Provee una visión de la memoria compatible con la del usuario.
- Organiza el espacio de direcciones en segmentos de diferente tamaño, identificados por un valor numérico.
- Para identificar un ítem dentro de un segmento se necesitan dos valores: El número de segmento y el desplazamiento relativo al origen del segmento del ítem que se intenta identificar.
- En contraste, como veremos, cuando se administra por paginación el programador define la dirección de memoria mediante un valor llamado dirección virtual y el hardware lo convierte en una dirección física.

Administración de Memoria por Segmentación

- Provee una visión de la memoria compatible con la del usuario.
- Organiza el espacio de direcciones en segmentos de diferente tamaño, identificados por un valor numérico.
- Para identificar un ítem dentro de un segmento se necesitan dos valores: El número de segmento y el desplazamiento relativo al origen del segmento del ítem que se intenta identificar.
- En contraste, como veremos, cuando se administra por paginación el programador define la dirección de memoria mediante un valor llamado dirección virtual y el hardware lo convierte en una dirección física.
- El compilador y el linker construyen las aplicaciones organizadas en segmentos. En particular las herramientas que generan formato ELF (para Linux) las denominan secciones.

Administración de Memoria por Segmentación



Administración de Memoria por Paginación

Administración de Memoria por Paginación

- La administración por Paginación es mas ordenada y simple, pero mas rígida.

Administración de Memoria por Paginación

- La administración por Paginación es mas ordenada y simple, pero mas rígida.
- El Sistema Operativo utiliza como unidad mínima de memoria para asignar a un proceso la página.

Administración de Memoria por Paginación

- La administración por Paginación es mas ordenada y simple, pero mas rígida.
- El Sistema Operativo utiliza como unidad mínima de memoria para asignar a un proceso la página.
- En un típico caso de `malloc(SIZE_ELEMENTS)`, se asigna un número de páginas suficiente, pero a menos que `SIZE_ELEMENTS` sea múltiplo exacto del tamaño de página, la ultima tendrá bytes remanentes que no se utilizan.

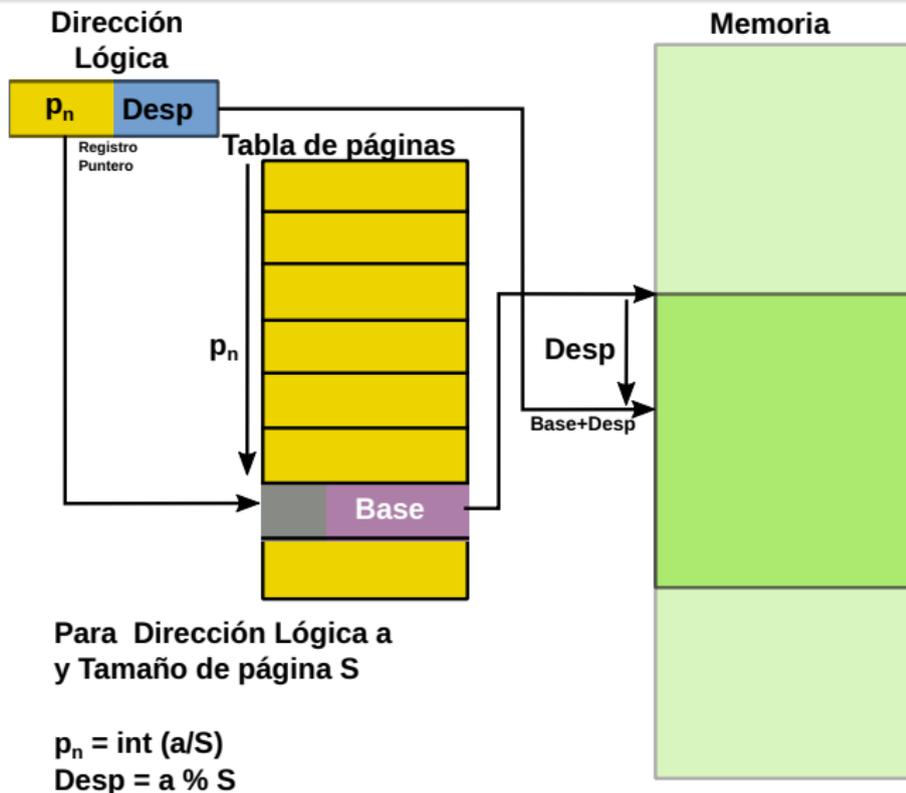
Administración de Memoria por Paginación

- La administración por Paginación es mas ordenada y simple, pero mas rígida.
- El Sistema Operativo utiliza como unidad mínima de memoria para asignar a un proceso la página.
- En un típico caso de `malloc(SIZE_ELEMENTS)`, se asigna un número de páginas suficiente, pero a menos que `SIZE_ELEMENTS` sea múltiplo exacto del tamaño de página, la ultima tendrá bytes remanentes que no se utilizan.
- La segmentación permite asignar en el caso `SIZE_ELEMENTS` bytes, pero es mas compleja su administración.

Administración de Memoria por Paginación

- La administración por Paginación es mas ordenada y simple, pero mas rígida.
- El Sistema Operativo utiliza como unidad mínima de memoria para asignar a un proceso la página.
- En un típico caso de `malloc(SIZE_ELEMENTS)`, se asigna un número de páginas suficiente, pero a menos que `SIZE_ELEMENTS` sea múltiplo exacto del tamaño de página, la ultima tendrá bytes remanentes que no se utilizan.
- La segmentación permite asignar en el caso `SIZE_ELEMENTS` bytes, pero es mas compleja su administración.
- La paginación genera cierto grado de desperdicio de memoria.

Administración de Memoria por Paginación



Administración de Memoria por Paginación

- En general la paginación divide el contenido del registro puntero en dos campos: El marco de la página (Page Frame), conformado por los bits mas significativos, y la palabra direccionada dentro de la página, compuesta por los bits menos significativos.

Administración de Memoria por Paginación

- En general la paginación divide el contenido del registro puntero en dos campos: El marco de la página (Page Frame), conformado por los bits mas significativos, y la palabra direccionada dentro de la página, compuesta por los bits menos significativos.
- Cada proceso requiere su propia tabla de páginas

Administración de Memoria por Paginación

- En general la paginación divide el contenido del registro puntero en dos campos: El marco de la página (Page Frame), conformado por los bits mas significativos, y la palabra direccionada dentro de la página, compuesta por los bits menos significativos.
- Cada proceso requiere su propia tabla de páginas
- Del mismo modo que con la segmentación, debe haber algún mecanismo de hardware que advierta cuando se produce un error (podría darse el caso de tener un valor de dirección que lleve a un elemento inválido en la Tabla de Páginas)

Administración de Memoria

Administración de Memoria

- Hay tres posibilidades entonces

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos
 - Administración de Memoria por páginas

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos
 - Administración de Memoria por páginas
 - Administración de Memoria por segmentos paginados

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos
 - Administración de Memoria por páginas
 - Administración de Memoria por segmentos paginados
- Los procesadores mas ancestrales utilizaron generalmente segmentación debido a que disponían de espacios de direccionamiento exiguos, y por lo tanto no se pueden dar el lujo de desperdiciar un solo byte.

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos
 - Administración de Memoria por páginas
 - Administración de Memoria por segmentos paginados
- Los procesadores mas ancestrales utilizaron generalmente segmentación debido a que disponían de espacios de direccionamiento exiguos, y por lo tanto no se pueden dar el lujo de desperdiciar un solo byte.
- En general los procesadores modernos (ARM, RISC V entre otros) emplean Paginación.

Administración de Memoria

- Hay tres posibilidades entonces
 - Administración de Memoria por segmentos
 - Administración de Memoria por páginas
 - Administración de Memoria por segmentos paginados
- Los procesadores mas ancestrales utilizaron generalmente segmentación debido a que disponían de espacios de direccionamiento exiguos, y por lo tanto no se pueden dar el lujo de desperdiciar un solo byte.
- En general los procesadores modernos (ARM, RISC V entre otros) emplean Paginación.
- La familia x86 de Intel inicia con segmentación en 1978 con el 8086. En 1985 el 80386 introduce Paginación. Debido al compromiso público de compatibilidad anunciado con el 8086, implementan Segmentos Paginados, porque sencillamente dicho compromiso le impide hasta hoy eliminar la segmentación.

Casos prácticos

Casos prácticos

- Burroughs 5500 soportaba 32 segmentos de 1024 palabras c/u. Resultó en un diseño insuficiente.

Casos prácticos

- Burroughs 5500 soportaba 32 segmentos de 1024 palabras c/u. Resultó en un diseño insuficiente.
- GE 645 (la máquina en la que se desarrolló MULTICS, el proyecto fallido previo a UNIX), armó una estructura de 256k Segmentos de hasta 64k word c/u.

Casos prácticos

- Burroughs 5500 soportaba 32 segmentos de 1024 palabras c/u. Resultó en un diseño insuficiente.
- GE 645 (la máquina en la que se desarrolló MULTICS, el proyecto fallido previo a UNIX), armó una estructura de 256k Segmentos de hasta 64k word c/u.
- PDP-11/20 es el equipo en el que se presenta la primer versión pública de UNIX en 1971. Capacidad Máxima de RAM 64 KiB. Método de administración Segmentación. 8 registros de segmento, que permitían alcanzar 8 KiB

Casos prácticos

- Burroughs 5500 soportaba 32 segmentos de 1024 palabras c/u. Resultó en un diseño insuficiente.
- GE 645 (la máquina en la que se desarrolló MULTICS, el proyecto fallido previo a UNIX), armó una estructura de 256k Segmentos de hasta 64k word c/u.
- PDP-11/20 es el equipo en el que se presenta la primer versión pública de UNIX en 1971. Capacidad Máxima de RAM 64 KiB. Método de administración Segmentación. 8 registros de segmento, que permitían alcanzar 8 KiB
- El factor común de todos ellos es la escasa capacidad de direccionamiento de memoria física.

Casos prácticos

- Burroughs 5500 soportaba 32 segmentos de 1024 palabras c/u. Resultó en un diseño insuficiente.
- GE 645 (la máquina en la que se desarrolló MULTICS, el proyecto fallido previo a UNIX), armó una estructura de 256k Segmentos de hasta 64k word c/u.
- PDP-11/20 es el equipo en el que se presenta la primer versión pública de UNIX en 1971. Capacidad Máxima de RAM 64 KiB. Método de administración Segmentación. 8 registros de segmento, que permitían alcanzar 8 KiB
- El factor común de todos ellos es la escasa capacidad de direccionamiento de memoria física.
- Cuando en mitad de los años 70 Intel desarrollaba su primer procesador de 16 bits (hay versiones que sostienen que el proyecto original era el 80286), no estaba tan claro si usar segmentación o paginación.

Casos prácticos: (Intel's Born..)

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.
- Intel se enfoca dos familias de microprocesadores la iAPx432 (para arquitecturas de 16/32 bits destinadas a “micromainframes”) y la iApx86.

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.
- Intel se enfoca dos familias de microprocesadores la iAPx432 (para arquitecturas de 16/32 bits destinadas a “micromainframes”) y la iApx86.
- Motorola se enfoca en el 68000 (a.k.a. 68k). Evolución del 6800

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.
- Intel se enfoca dos familias de microprocesadores la iAPx432 (para arquitecturas de 16/32 bits destinadas a “micromainframes”) y la iApx86.
- Motorola se enfoca en el 68000 (a.k.a. 68k). Evolución del 6800
- IBM habla con ambos para definir el procesador de su PC. Necesitaba el procesador en 1977 al menos en prototipo.

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.
- Intel se enfoca dos familias de microprocesadores la iAPx432 (para arquitecturas de 16/32 bits destinadas a “micromainframes”) y la iApx86.
- Motorola se enfoca en el 68000 (a.k.a. 68k). Evolución del 6800
- IBM habla con ambos para definir el procesador de su PC. Necesitaba el procesador en 1977 al menos en prototipo.
- El forecast del 68000 preveía su lanzamiento para 1979.

Casos prácticos: (Intel's Born..)

- En la década del 70 aparecen los primeros microprocesadores.
- El 8080 de Intel, el 6800 de Motorola y el Z80 de Zilog establecen un piso de evolución.
- Intel se enfoca dos familias de microprocesadores la iAPx432 (para arquitecturas de 16/32 bits destinadas a “micromainframes”) y la iApx86.
- Motorola se enfoca en el 68000 (a.k.a. 68k). Evolución del 6800
- IBM habla con ambos para definir el procesador de su PC. Necesitaba el procesador en 1977 al menos en prototipo.
- El forecast del 68000 preveía su lanzamiento para 1979.
- Leyenda urbana: El proyecto iAPx86 apuntaba a un Microprocesador de 16 bits con capacidades de multitasking: el 80286. Ante el requerimiento de IBM, lo recortan y proponen salir en pocos meses con un prototipo de 16 bits: el 8086

Casos prácticos: (Intel's Born..)

"Motorola, with its superior technology, lost the single most important design contest of the last 50 years". Wally Rhines, Mentor Graphics CEO.

The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor. IEEE Spectrum. Junio 2017.

Casos prácticos: (Intel's Born..)

“Motorola, with its superior technology, lost the single most important design contest of the last 50 years”. Wally Rhines, Mentor Graphics CEO.

The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor. IEEE Spectrum. Junio 2017.

- Más allá de estas anécdotas, la discusión sobre el manejo de la memoria no estaba por entonces saldada.

Casos prácticos: (Intel's Born..)

“Motorola, with its superior technology, lost the single most important design contest of the last 50 years”. Wally Rhines, Mentor Graphics CEO.

The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor. IEEE Spectrum. Junio 2017.

- Más allá de estas anécdotas, la discusión sobre el manejo de la memoria no estaba por entonces saldada.
- Intel y Zilog proponían administración con segmentos

Casos prácticos: (Intel's Born..)

“Motorola, with its superior technology, lost the single most important design contest of the last 50 years”. Wally Rhines, Mentor Graphics CEO.

The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor. IEEE Spectrum. Junio 2017.

- Más allá de estas anécdotas, la discusión sobre el manejo de la memoria no estaba por entonces saldada.
- Intel y Zilog proponían administración con segmentos
- Motorola, lanza el 68000 con administración por páginas de 2 Kbytes, y Big Endian.

Casos prácticos: (Intel's Born..)

“Motorola, with its superior technology, lost the single most important design contest of the last 50 years”. Wally Rhines, Mentor Graphics CEO.

The Inside Story of Texas Instruments' Biggest Blunder: The TMS9900 Microprocessor. IEEE Spectrum. Junio 2017.

- Más allá de estas anécdotas, la discusión sobre el manejo de la memoria no estaba por entonces saldada.
- Intel y Zilog proponían administración con segmentos
- Motorola, lanza el 68000 con administración por páginas de 2 Kbytes, y Big Endian.

¿Quién tenía Razón?

Casos prácticos: Nadie!!!

	CPU Speed	Motherboard Memory	Max. # of Half-Height Internal Drives	Display	Operating System	Warranty Period	Suggested List Price
Multitech 900	10 or 6 MHz Keyboard, Software or Hardware Switchable	512K RAM Up to 1MB	5	Hi-Res., Soft-white	MS-DOS 3.1	12-Months	\$2395
IBM PC/AT	8 MHz Only	512K RAM Up to 640K	3	Option	Option	12-Months	\$5295*
Sperry PC/IT	6, 7, 16 or 8 MHz Hardware Switchable	512K RAM Up to 1 MB	4	Option	MS-DOS 3.1	12-Months	\$3395
Compaq Deskpro 286	6 or 8 MHz Keyboard & Software Switchable	256K RAM Up to 2.2MB	4	Option	Option	3-Months	\$3395
Tandy 3000	8 MHz Only	512K RAM Up to 640K	3	Option	Option	3-Months	\$2599
Televideo Telecat-286	6 or 8 MHz Keyboard & Software Switchable	512K RAM Up to 1MB	3	Hi-Res., Green	Option	3-Months	\$2995**

Includes 30MB Hard Disk Drive
*Includes 20MB Hard Disk Drive

All prices reflect manufacturers' suggested list for base models as of June 4, 1986

Multitech 900 is a trademark of Multitech Electronics Inc.
All other product names are trademarks of their respective manufacturers

Inquiry 244

guaranteed *support* from Multitech, a company with a proven track-record in quality microcomputer manufacture; and, *Nationwide service** that is convenient and dependable. Call us today for more information about the Multitech 900 or our full line of personal computer products.



Committed to Excellence, Quality & Service

1012 Stewart Drive
Sunnyvale, CA 94086
Tel: (408) 773-8400
(800) 538-1542

*A Division of MSC International
Serviced Nationwide by TRW

SEPTEMBER 1986 • B Y T E 91

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Lo que es estrictamente cierto es que su iAPx86 fue el último diseño con Administración por segmentos.

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Lo que es estrictamente cierto es que su iAPx86 fue el último diseño con Administración por segmentos.

Fin de la historia.

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Lo que es estrictamente cierto es que su iAPx86 fue el último diseño con Administración por segmentos.

Fin de la historia.

Gol de Motorola.

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Lo que es estrictamente cierto es que su iAPx86 fue el último diseño con Administración por segmentos.

Fin de la historia.

Gol de Motorola.

Pero Intel gana el partido.

Casos prácticos: Conclusión

Sin dudas Motorola tuvo mejor visión de largo plazo, e Intel apremiado por la escasa capacidad relativa del 8086 optó por lo que hasta ese momento era un estándar no escrito.

Lo que es estrictamente cierto es que su iAPx86 fue el último diseño con Administración por segmentos.

Fin de la historia.

Gol de Motorola.

Pero Intel gana el partido.

Así son las cosas...

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - **Gestión del espacio libre**
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Se trata de la administración del Heap

Se trata de la administración del Heap

Asignación estática:

Asignación dinámica:

Se trata de la administración del Heap

Asignación estática:

Target: Micro computadores dedicados.

Asignación dinámica:

Target: Computadores de propósito general.

Se trata de la administración del Heap

Asignación estática:

Target: Micro computadores dedicados.

Tipo de CPU: Gama baja y media.

Asignación dinámica:

Target: Computadores de propósito general.

Tipo de CPU: Gama media y alta

Se trata de la administración del Heap

Asignación estática:

Target: Micro computadores dedicados.

Tipo de CPU: Gama baja y media.

Tareas: Se define su espacio de memoria al inicio y no se cambia.

Asignación dinámica:

Target: Computadores de propósito general.

Tipo de CPU: Gama media y alta

Tareas: solicitan memoria dinámicamente durante su ejecución.

Se trata de la administración del Heap

Asignación estática:

Target: Micro computadores dedicados.

Tipo de CPU: Gama baja y media.

Tareas: Se define su espacio de memoria al inicio y no se cambia.

Características: Administración simple. Protección delegada en el Hardware. Asignación rígida. Tiempo de acceso a memoria Determinístico.

Asignación dinámica:

Target: Computadores de propósito general.

Tipo de CPU: Gama media y alta

Tareas: solicitan memoria dinámicamente durante su ejecución.

Características: Administración compleja. Protección a cargo del S.O. y el hardware. Asignación flexible. Tiempo de acceso a memoria No Determinístico.

Asignación dinámica de la memoria

Asignación dinámica de la memoria

- ***Idea Fuerza:***

Asignación dinámica de la memoria

- **Idea Fuerza:**

- Cada vez que un proceso necesita un bloque de memoria, envía un requerimiento al SO (*allocation*).

Asignación dinámica de la memoria

- **Idea Fuerza:**

- Cada vez que un proceso necesita un bloque de memoria, envía un requerimiento al SO (*allocation*).
- Cuando no la necesita más la debe devolver para que sea asignada a otro proceso.

Asignación dinámica de la memoria

- **Idea Fuerza:**

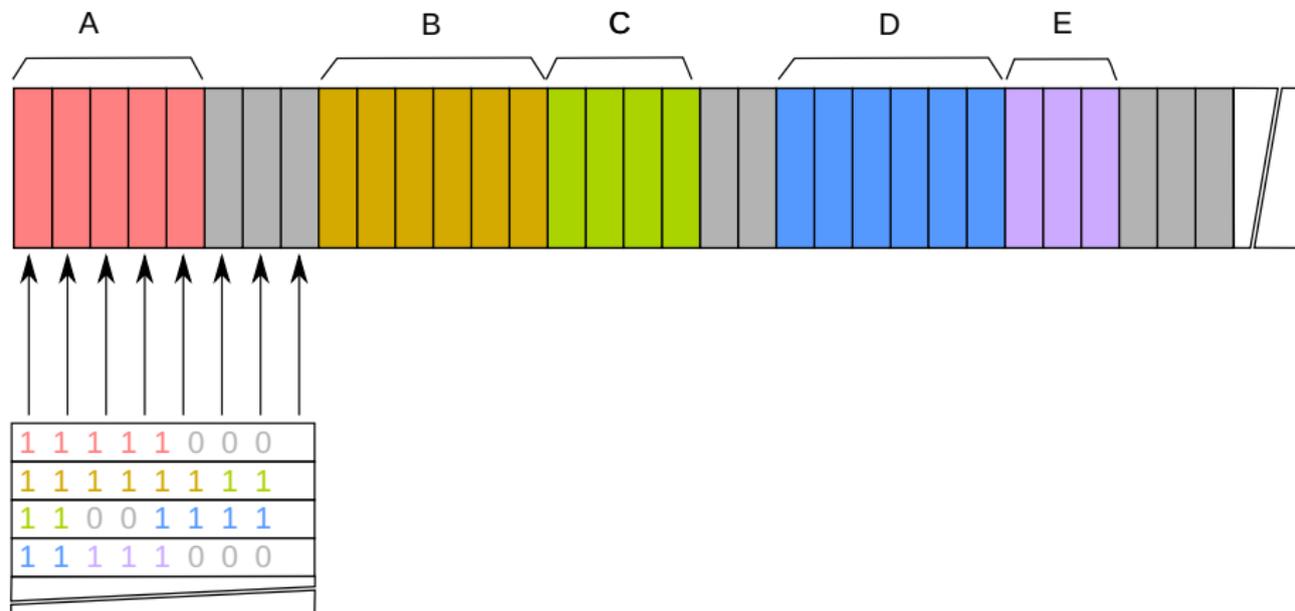
- Cada vez que un proceso necesita un bloque de memoria, envía un requerimiento al SO (*allocation*).
- Cuando no la necesita más la debe devolver para que sea asignada a otro proceso.
- Si en medio del uso el proceso necesita modificar el tamaño de un bloque que ya le ha sido asignado, entonces envía el S.O. un requerimiento de re asignación (*reallocation*)

Asignación dinámica de la memoria

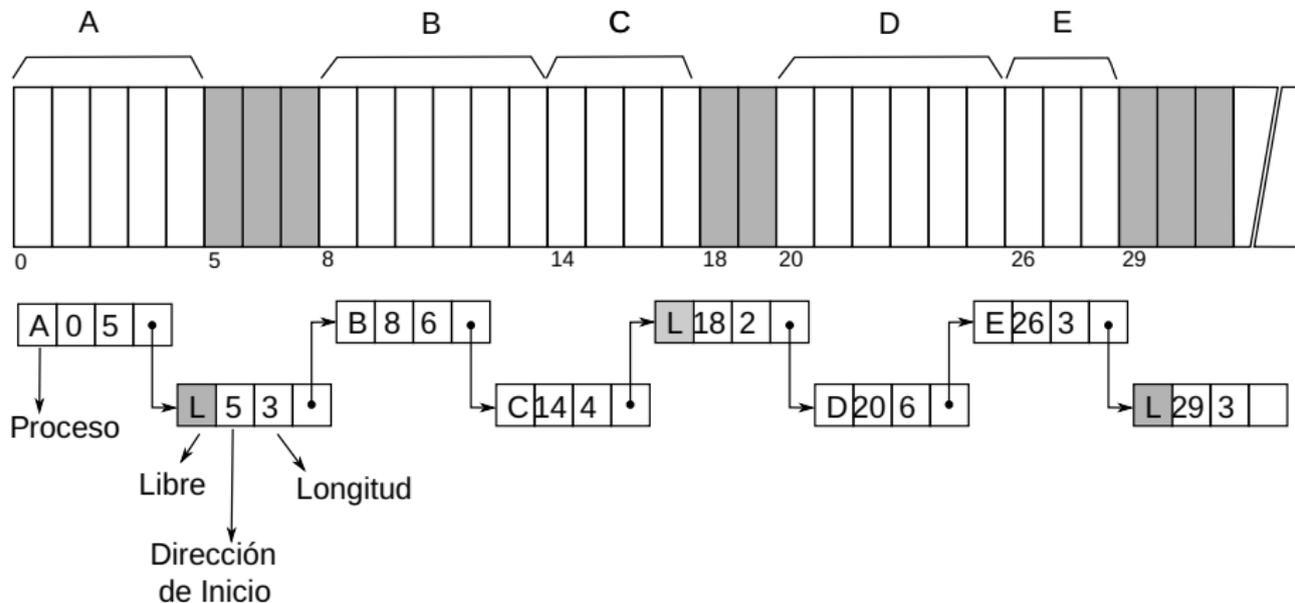
- **Idea Fuerza:**

- Cada vez que un proceso necesita un bloque de memoria, envía un requerimiento al SO (*allocation*).
- Cuando no la necesita más la debe devolver para que sea asignada a otro proceso.
- Si en medio del uso el proceso necesita modificar el tamaño de un bloque que ya le ha sido asignado, entonces envía el S.O. un requerimiento de re asignación (*reallocation*)
- El S.O. permanentemente asigna y libera en forma dinámica bloques de memoria.

Memoria dinámica manejada por Bitmap



Memoria dinámica manejada por Lista Enlazada



¿Y si el Heap se agota?

¿Y si el Heap se agota?

Memoria Virtual:

Servicio implementado en los Sistemas Operativos modernos con el fin de proveer a los programas de aplicación un espacio lógico o virtual de direccionamiento de memoria, independiente de la cantidad de memoria RAM físicamente disponible en el hardware del sistema.

Otra vez el Principio de Vecindad

Otra vez el Principio de Vecindad

- **Vecindad Temporal:** La probabilidad de un ítem de ser referenciado en lo inmediato es alta, si se lo está utilizando actualmente.

Otra vez el Principio de Vecindad

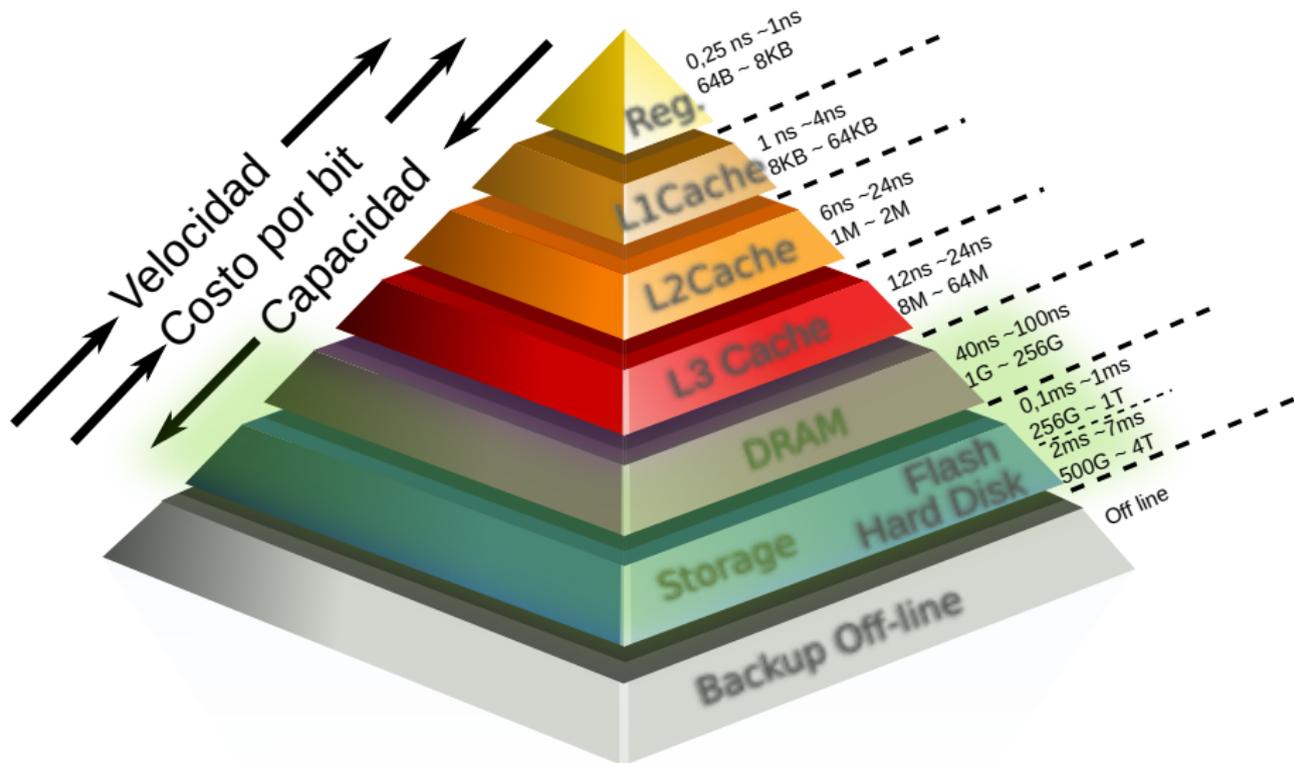
- **Vecindad Temporal:** La probabilidad de un ítem de ser referenciado en lo inmediato es alta, si se lo está utilizando actualmente.
- **Vecindad Espacial:** La probabilidad de un ítem de ser referenciado en lo inmediato es alta, si se están referenciando actualmente direcciones de memoria próximas.

Otra vez el Principio de Vecindad

- **Vecindad Temporal:** La probabilidad de un ítem de ser referenciado en lo inmediato es alta, si se lo está utilizando actualmente.
- **Vecindad Espacial:** La probabilidad de un ítem de ser referenciado en lo inmediato es alta, si se están referenciando actualmente direcciones de memoria próximas.

```
1 for ( i = 0 ; i < 256 ; i++)
2 {
3     suma = 0.0f;
4     for ( j = 0 ; (j <= i && j < 256) ; j++)
5         suma += v0[i-j] * v1[j];
6     fAux[i] = suma;
```

Delimitando el problema a tratar



Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - **Políticas de Administración de memoria**
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Políticas de Alojamiento

Políticas de Alojamiento

- Consiste en asignar bloques a un proceso, ya sea para código o para datos.

Políticas de Alojamiento

- Consiste en asignar bloques a un proceso, ya sea para código o para datos.
- En este punto es evidente la mayor simpleza y eficiencia de manejar bloques del mismo tamaño en lugar de bloques de tamaño diferente.

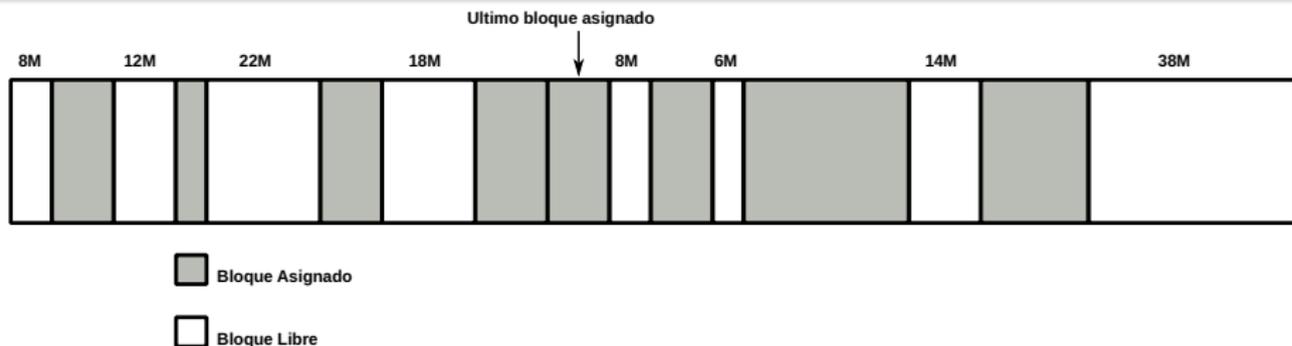
Políticas de Alojamiento

- Consiste en asignar bloques a un proceso, ya sea para código o para datos.
- En este punto es evidente la mayor simpleza y eficiencia de manejar bloques del mismo tamaño en lugar de bloques de tamaño diferente.
- Si los bloques son todos del mismo tamaño no hay que evaluar si lo que traemos cabe o no. Simplemente se usa el primero que se encuentra

Políticas de Alojamiento

- Consiste en asignar bloques a un proceso, ya sea para código o para datos.
- En este punto es evidente la mayor simpleza y eficiencia de manejar bloques del mismo tamaño en lugar de bloques de tamaño diferente.
- Si los bloques son todos del mismo tamaño no hay que evaluar si lo que traemos cabe o no. Simplemente se usa el primero que se encuentra
- Si los bloques son de diversos tamaños, entonces el algoritmo se complica muchísimo. Debe determinar antes que nada por cada bloque libre si su tamaño es suficiente para asignarle al proceso un bloque del tamaño requerido

Políticas de Alojamiento



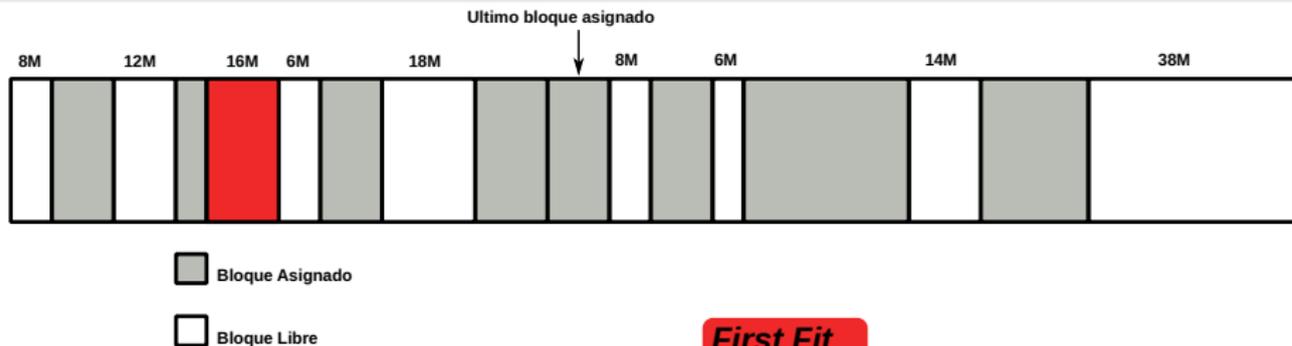
Consideremos un caso como el de la figura, en el que tenemos una distribución de bloque ocupados y libres en memoria, y en donde el último bloque que se ha alojado es el que se encuentra señalado.

Políticas de Alojamiento



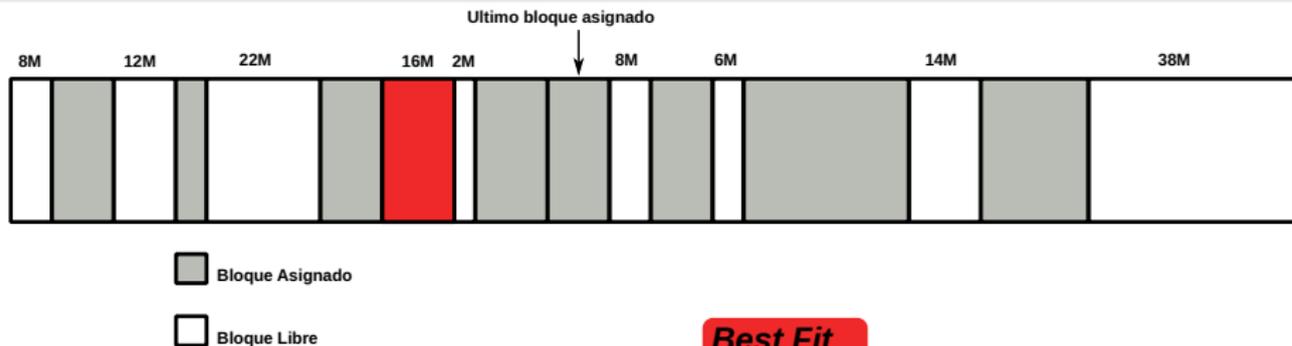
En estas condiciones un proceso requiere se le alojen 16 MiB de memoria. En este esquema en el que hay varios bloques de memoria libres, el SO deberá decir cual de todos asignar. Para ello existen tres políticas cuando la memoria se administra en bloques de diferentes tamaños.

Políticas de Alojamiento



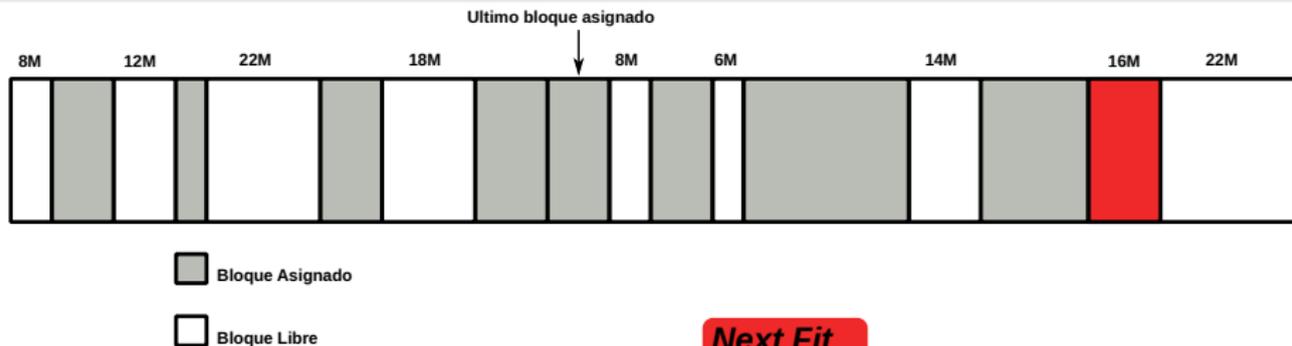
First Fit: Recorre la lista de bloques hasta encontrar el primero que tiene un tamaño mayor o igual al requerido. Ahí asigna y finaliza. Comparado con los próximos es el más sencillo y el más rápido. Pero a medida que trabaja, genera sobrantes en el área de direcciones menores, en general segmentos pequeños insuficientes para satisfacer nuevas demandas de alojamiento, pero a los que debe evaluar cada vez. Es decir, empeora con el tiempo. Requiere un compactado de memoria periódico.

Políticas de Alojamiento



Best Fit: Recorre la lista completa de bloques y asigna el que genera menos desperdicio. Curiosamente resulta el menos eficiente. A medida que trabaja, genera segmentos muy pequeños e insuficientes para satisfacer nuevas demandas de alojamiento, a los que debe evaluar cada vez. Es decir, empeora con el tiempo. Requiere un compactado de memoria periódico mas frecuente que el resto. Pero a diferencia del **First Fit**, tarda mucho mas ya que recorre toda la lista de bloques.

Políticas de Alojamiento



Next Fit

Next Fit: Recorre la lista de bloques a partir del último bloque asignado y asigna el primer bloque que encuentre cuyo tamaño sea mayor o igual al requerido. Es algo menos eficiente que **First Fit**, ya que tiende a compactar bloques pequeños en las direcciones más altas de memoria, ya que allí es donde se encuentran bloques grandes y este algoritmo los divide al principio con mucha eficiencia pero a medida que se van liberando, termina requiriendo compactado mucho más frecuentemente.

Temario

- 1 Introducción
 - Enfoque preliminar
 - Organización y Administración
- 2 Abstracción de memoria
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 Administración de memoria
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - **Antecedentes**
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 Relación con el hardware
 - MMU
 - Page faults
- 6 Paginación en un Sistema Operativo Real: Linux
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Antecedentes

Antecedentes

- 1 Durante los 60's y 70's se generó toda la investigación científica sobre Memoria Virtual.

Antecedentes

- 1 Durante los 60's y 70's se generó toda la investigación científica sobre Memoria Virtual.
- 2 Se probó que el software tiende a permanecer trabajando dentro un rango de direcciones de memoria contiguas durante lapsos significativos.

Antecedentes

- 1 Durante los 60's y 70's se generó toda la investigación científica sobre Memoria Virtual.
- 2 Se probó que el software tiende a permanecer trabajando dentro un rango de direcciones de memoria contiguas durante lapsos significativos.
- 3 Comenzar la ejecución de un proceso con un bloque de su código, otro sus datos y otro de su pila garantiza su funcionamiento durante un lapso considerable.

Antecedentes

- 1 Durante los 60's y 70's se generó toda la investigación científica sobre Memoria Virtual.
- 2 Se probó que el software tiende a permanecer trabajando dentro un rango de direcciones de memoria contiguas durante lapsos significativos.
- 3 Comenzar la ejecución de un proceso con un bloque de su código, otro sus datos y otro de su pila garantiza su funcionamiento durante un lapso considerable.
- 4 Cuando se pasa a trabajar dentro de otro rango de direcciones, los bloques dejan de requerirse, y pasará un lapso muy largo hasta que (eventualmente) se lo vuelva a requerir.

Trabajos relevantes

Trabajos relevantes

1er. antecedente de swapping por bloques de tamaño Fijo

John Fotheringham. "Dynamic storage allocation in the Atlas computer, including an automatic use of a backing store", in Communications of the ACM, Volume 4 , Issue 10 (October 1961). Pages: 435 - 436.

Trabajos relevantes

1er. antecedente de swapping por bloques de tamaño Fijo

John Fotheringham. "Dynamic storage allocation in the Atlas computer, including an automatic use of a backing store", in Communications of the ACM, Volume 4 , Issue 10 (October 1961). Pages: 435 - 436.

Demostración del Principio de Vecindad

Hatfield, D. "Experimets on page size, Program Access Patterns, and Virtual Memory Performance", IBM Journal Journal of Research and Development, January 1972.

Temario

- 1 Introducción
 - Enfoque preliminar
 - Organización y Administración
- 2 Abstracción de memoria
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 Administración de memoria
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - **Swapping**
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 Relación con el hardware
 - MMU
 - Page faults
- 6 Paginación en un Sistema Operativo Real: Linux
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Definiciones básicas

Definiciones básicas

- 1 Los procesos dividen su espacio de direccionamiento en bloques.

Definiciones básicas

- 1 Los procesos dividen su espacio de direccionamiento en bloques.
- 2 Los bloques de los procesos necesarios en un momento dado para su ejecución, se almacenan en la **memoria primaria** o **memoria física**. Es decir en la RAM.

Definiciones básicas

- 1 Los procesos dividen su espacio de direccionamiento en bloques.
- 2 Los bloques de los procesos necesarios en un momento dado para su ejecución, se almacenan en la **memoria primaria** o **memoria física**. Es decir en la RAM.
- 3 El resto de los bloques se almacena en la **memoria secundaria**.

Definiciones básicas

- 1 Los procesos dividen su espacio de direccionamiento en bloques.
- 2 Los bloques de los procesos necesarios en un momento dado para su ejecución, se almacenan en la **memoria primaria** o **memoria física**. Es decir en la RAM.
- 3 El resto de los bloques se almacena en la **memoria secundaria**.
- 4 **Memoria secundaria:** genera la “visión” de una mayor cantidad de memoria que la físicamente disponible.

Definiciones básicas

- 1 Los procesos dividen su espacio de direccionamiento en bloques.
- 2 Los bloques de los procesos necesarios en un momento dado para su ejecución, se almacenan en la **memoria primaria** o **memoria física**. Es decir en la RAM.
- 3 El resto de los bloques se almacena en la **memoria secundaria**.
- 4 **Memoria secundaria:** genera la “visión” de una mayor cantidad de memoria que la físicamente disponible.
- 5 Es finalmente la **Memoria Virtual**. Es decir, un área de disco rígido.

¿Cuándo se requiere swapping?

¿Cuando se requiere swapping?

Cuando un programa solicita acceso a un bloque de su código datos o pila, y este no se encuentra en la memoria RAM la CPU debe señalar esta situación al Sistema Operativo, que deberá disponer de un código muy bien programado capaz de recuperar desde el almacenamiento secundario este bloque faltante en la DRAM, y descargarlo.

¿Cuándo se requiere swapping?

Cuando un programa solicita acceso a un bloque de su código datos o pila, y este no se encuentra en la memoria RAM la CPU debe señalar esta situación al Sistema Operativo, que deberá disponer de un código muy bien programado capaz de recuperar desde el almacenamiento secundario este bloque faltante en la DRAM, y descargarlo.

El problema mas interesante aparece cuando no hay espacio suficiente en la RAM para almacenar el bloque requerido.

¿Cuándo se requiere swapping?

Cuando un programa solicita acceso a un bloque de su código datos o pila, y este no se encuentra en la memoria RAM la CPU debe señalar esta situación al Sistema Operativo, que deberá disponer de un código muy bien programado capaz de recuperar desde el almacenamiento secundario este bloque faltante en la DRAM, y descargarlo.

El problema mas interesante aparece cuando no hay espacio suficiente en la RAM para almacenar el bloque requerido.

En este caso es necesario intercambiar (swap) algún bloque residente en la RAM con el bloque requerido.

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - **Organización en bloques**
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Características de los Segmentos

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.
- 2 Definen espacios continuos y completos de direcciones.

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.
- 2 Definen espacios continuos y completos de direcciones.
- 3 Su tamaño *no es fijo*, y puede cambiar en forma dinámica durante la ejecución del proceso. Esto siempre que se implemente memoria virtual por segmentos.

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.
- 2 Definen espacios continuos y completos de direcciones.
- 3 Su tamaño *no es fijo*, y puede cambiar en forma dinámica durante la ejecución del proceso. Esto siempre que se implemente memoria virtual por segmentos.
- 4 Esta propiedad le permite a un programador definir estructuras de datos de tamaño incierto, sin preocuparse de la asignación de memoria. El SO alojará un espacio de memoria lineal y contiguo.

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.
- 2 Definen espacios continuos y completos de direcciones.
- 3 Su tamaño *no es fijo*, y puede cambiar en forma dinámica durante la ejecución del proceso. Esto siempre que se implemente memoria virtual por segmentos.
- 4 Esta propiedad le permite a un programador definir estructuras de datos de tamaño incierto, sin preocuparse de la asignación de memoria. El SO alojará un espacio de memoria lineal y contiguo.
- 5 Una vez mapeado en el espacio físico de memoria pueden quedar solapado con otro segmento en forma parcial o total.

Características de los Segmentos

- 1 Representan la memoria con la misma estructura lógica del programador: Un bloque que contiene el código, otro con los datos, otro con la pila, etc.
- 2 Definen espacios continuos y completos de direcciones.
- 3 Su tamaño **no es fijo**, y puede cambiar en forma dinámica durante la ejecución del proceso. Esto siempre que se implemente memoria virtual por segmentos.
- 4 Esta propiedad le permite a un programador definir estructuras de datos de tamaño incierto, sin preocuparse de la asignación de memoria. El SO alojará un espacio de memoria lineal y contiguo.
- 5 Una vez mapeado en el espacio físico de memoria pueden quedar solapado con otro segmento en forma parcial o total.
- 6 El algoritmo de swapping requiere detectar el bloque candidato a reemplazar, sumado a la comprobación de tamaño.

Características de las Páginas

Características de las Páginas

- 1 Tamaño *es fijo*. Establecido por hardware. No puede modificarse.

Características de las Páginas

- 1 Tamaño *es fijo*. Establecido por hardware. No puede modificarse.
- 2 Se mapean en el espacio físico de memoria una a continuación de otra sin dejar espacios libres.

Características de las Páginas

- 1 Tamaño **es fijo**. Establecido por hardware. No puede modificarse.
- 2 Se mapean en el espacio físico de memoria una a continuación de otra sin dejar espacios libres.
- 3 Cada página p_n , de tamaño t (donde n es el número de página, tal que $0 \leq n$, y $n \in \mathbb{N}$ comienza en la dirección $n * t$.

Características de las Páginas

- 1 Tamaño **es fijo**. Establecido por hardware. No puede modificarse.
- 2 Se mapean en el espacio físico de memoria una a continuación de otra sin dejar espacios libres.
- 3 Cada página p_n , de tamaño t (donde n es el número de página, tal que $0 \leq n$, y $n \in \mathbb{N}$ comienza en la dirección $n * t$.
- 4 Define un marco (frame) de direcciones que contendrá una parte de cada unidad de programa que compone el proceso en ejecución.

Características de las Páginas

- 1 Tamaño **es fijo**. Establecido por hardware. No puede modificarse.
- 2 Se mapean en el espacio físico de memoria una a continuación de otra sin dejar espacios libres.
- 3 Cada página p_n , de tamaño t (donde n es el número de página, tal que $0 \leq n$, y $n \in \mathbb{N}$ comienza en la dirección $n * t$.
- 4 Define un marco (frame) de direcciones que contendrá una parte de cada unidad de programa que compone el proceso en ejecución.
- 5 Las unidades de programación (Código, Datos, o Pila) pueden ubicarse en frames discontinuos en la memoria principal y a pesar de ello el procesador traduce las direcciones de modo que la secuencia de acceso no se altere.

Características de las Páginas

- 1 Tamaño **es fijo**. Establecido por hardware. No puede modificarse.
- 2 Se mapean en el espacio físico de memoria una a continuación de otra sin dejar espacios libres.
- 3 Cada página p_n , de tamaño t (donde n es el número de página, tal que $0 \leq n, y n \in \mathbb{N}$ comienza en la dirección $n * t$.
- 4 Define un marco (frame) de direcciones que contendrá una parte de cada unidad de programa que compone el proceso en ejecución.
- 5 Las unidades de programación (Código, Datos, o Pila) pueden ubicarse en frames discontinuos en la memoria principal y a pesar de ello el procesador traduce las direcciones de modo que la secuencia de acceso no se altere.
- 6 El algoritmo de swapping consiste solo en determinar el bloque a ser reemplazado.

Organización por bloques mas apta para swapping

Organización por bloques mas apta para swapping

La eficiencia del algoritmo de intercambio de bloques depende fuertemente del sistema adoptado para su organización, ya que como veremos hay un costo computacional implícito en determinar cual es el bloque candidato a ser reemplazado.

Organización por bloques mas apta para swapping

La eficiencia del algoritmo de intercambio de bloques depende fuertemente del sistema adoptado para su organización, ya que como veremos hay un costo computacional implícito en determinar cual es el bloque candidato a ser reemplazado.

Si el sistema elegido es segmentación, a esta complejidad se deberá agregar la necesidad de determinar además si el bloque seleccionado para desalojo liberará el espacio suficiente para escribir allí el bloque que se demanda alojar.

Organización por bloques mas apta para swapping

La eficiencia del algoritmo de intercambio de bloques depende fuertemente del sistema adoptado para su organización, ya que como veremos hay un costo computacional implícito en determinar cual es el bloque candidato a ser reemplazado.

Si el sistema elegido es segmentación, a esta complejidad se deberá agregar la necesidad de determinar además si el bloque seleccionado para desalojo liberará el espacio suficiente para escribir allí el bloque que se demanda alojar.

Si requiere desalojar mas de un bloque, la probabilidad que éstos sean contiguos, tiende a 0. Requiere reubicar al resto de los bloques. Cae estrepitosamente la Performance.

Conclusiones

Conclusiones

Los segmentos organizan muy bien las aplicaciones en memoria, pero son ineficientes para implementar Memoria Virtual. Las Páginas inversamente son muy aptas para esta última tarea.

Conclusiones

Los segmentos organizan muy bien las aplicaciones en memoria, pero son ineficientes para implementar Memoria Virtual. Las Páginas inversamente son muy aptas para esta última tarea.

Ambos generan Fragmentación de Memoria. La Paginación genera Fragmentación Interna. La Segmentación genera Fragmentación externa, que si bien puede resolverse con algoritmos de compactado, se requiere ciclos de cómputo adicionales.

Conclusiones

Los segmentos organizan muy bien las aplicaciones en memoria, pero son ineficientes para implementar Memoria Virtual. Las Páginas inversamente son muy aptas para esta última tarea.

Ambos generan Fragmentación de Memoria. La Paginación genera Fragmentación Interna. La Segmentación genera Fragmentación externa, que si bien puede resolverse con algoritmos de compactado, se requiere ciclos de cómputo adicionales.

Dennis Ritchie y Ken Thompson organizaron la gestión de memoria de UNIX en bloques de tamaño fijo. No obstante UNIX no incluyó memoria virtual en sus primeras versiones por falta de hardware de soporte.

Para Administración de memoria: Paginación

Para Administración de memoria: Paginación

El trabajo de Fotheringham, fue concluyente. A medida que la memoria fue dejando de ser dramáticamente escasa se impuso como criterio de diseño de Hardware.

Para Administración de memoria: Paginación

El trabajo de Fotheringham, fue concluyente. A medida que la memoria fue dejando de ser dramáticamente escasa se impuso como criterio de diseño de Hardware.

Una gran diferencia a favor de la Paginación está en el costo computacional del algoritmo de swapping ya que al tratar bloques de tamaño fijo, el swap es uno a uno directo.

Para Administración de memoria: Paginación

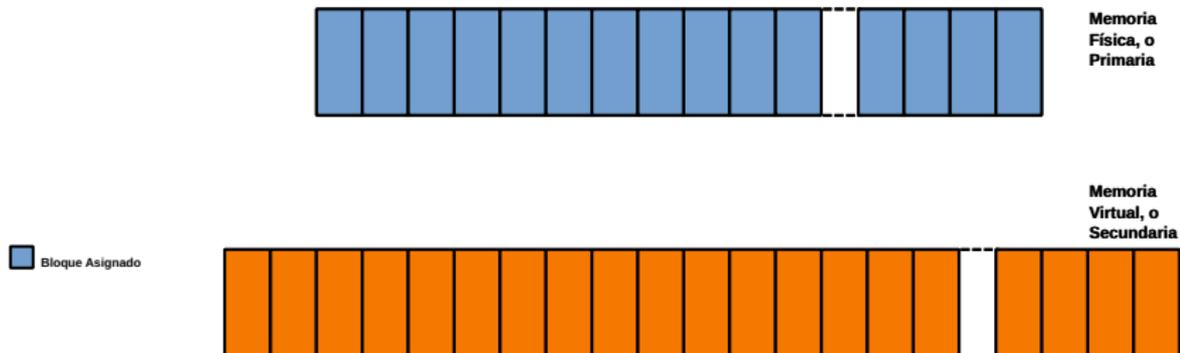
El trabajo de Fotheringham, fue concluyente. A medida que la memoria fue dejando de ser dramáticamente escasa se impuso como criterio de diseño de Hardware.

Una gran diferencia a favor de la Paginación está en el costo computacional del algoritmo de swapping ya que al tratar bloques de tamaño fijo, el swap es uno a uno directo.

La Paginación aloja en memoria una cantidad de procesos mucho mayor. Solo requiere una página e código y otra para datos y pila. El resto espera en Memoria Virtual hasta ser requerido. La segmentación obliga a bajar a memoria el segmento completo. Si éste es muy grande, no admite ser dividido.

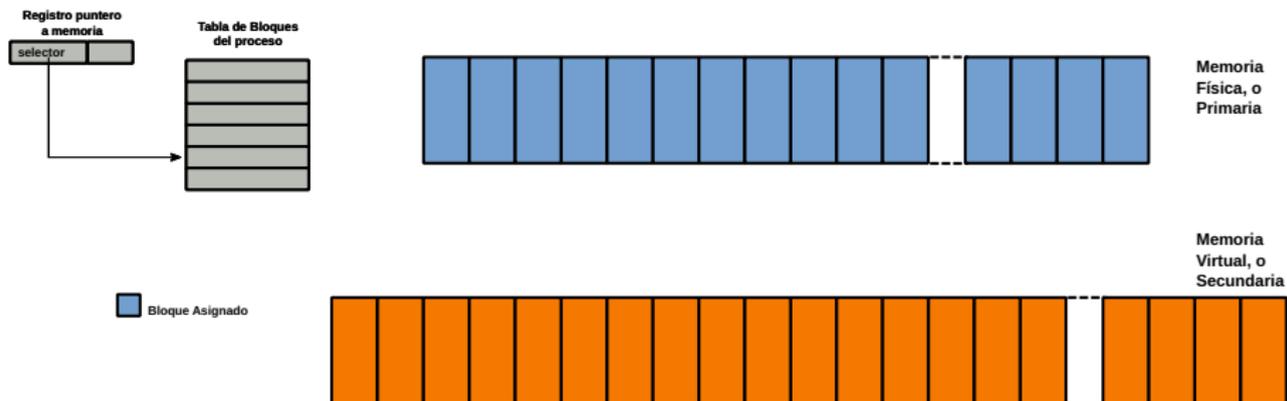
Swapping por Paginación

La memoria física se divide en páginas y una mayor cantidad de memoria secundaria o Virtual también se organiza en páginas. Ésta última tiene todos los bloques y la física solo los necesarios.



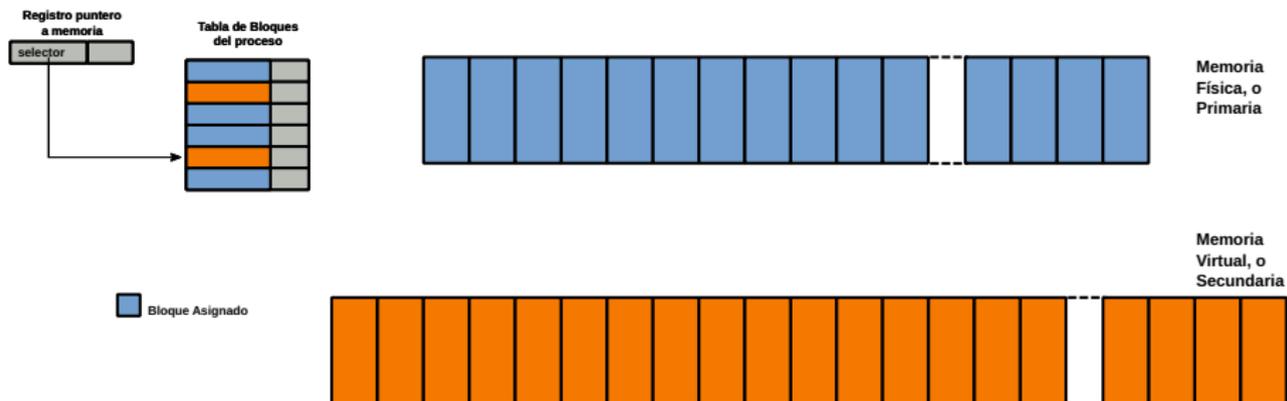
Swapping por Paginación

Cada proceso dispone de una tabla de páginas en las que referencian c/u de sus páginas. La MMU utiliza la cantidad necesaria de bits de cada registro como puntero de direcciones como entrada a esta tabla.



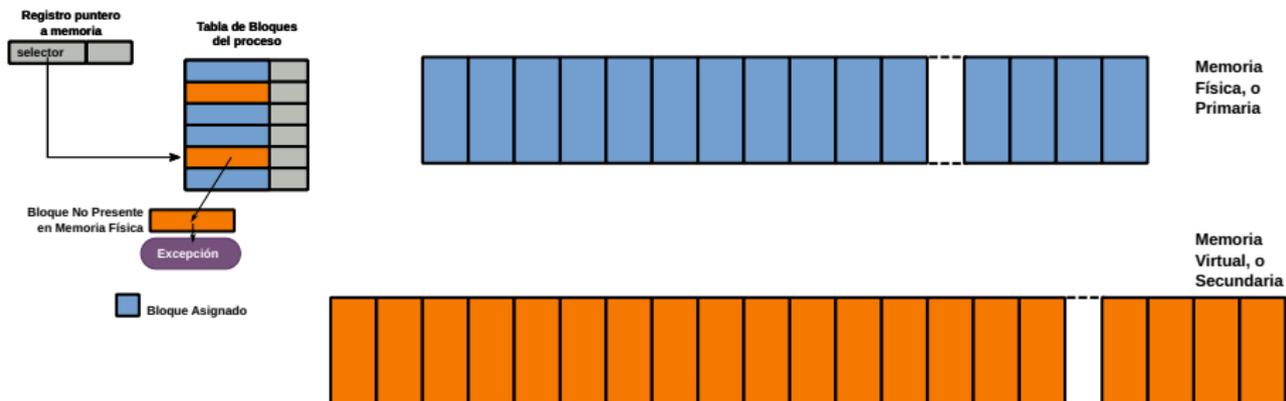
Swapping por Paginación

En la tabla de cada proceso están referenciadas todas las páginas. Algunas están en memoria física, otras solo en memoria Virtual. Cada referencia tiene un bit para señalar esta situación.



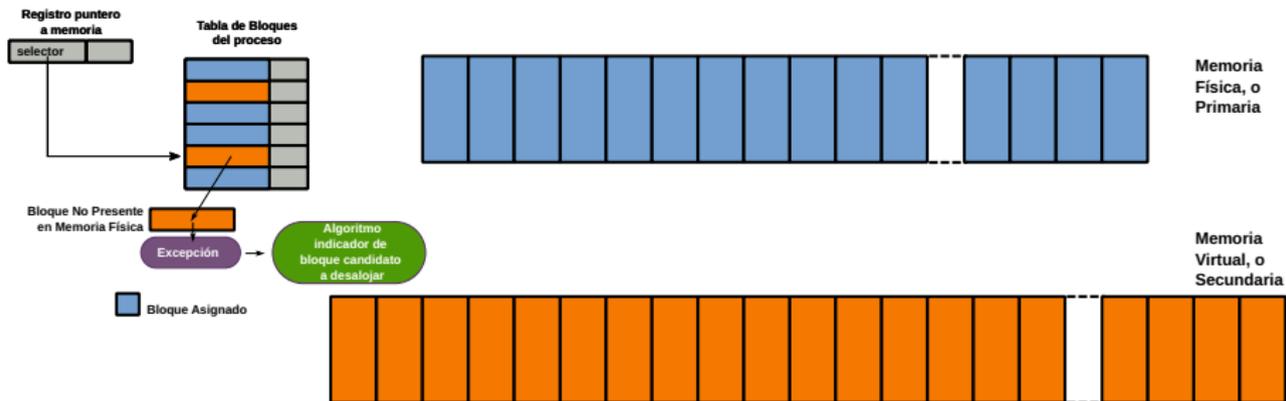
Swapping por Paginación

Los Bits mas significativos del registro puntero seleccionan (trabajando como índice) el elemento de la tabla de páginas que contiene la referencia a la página requerida. Si la página no está en memoria el procesador genera una excepción.



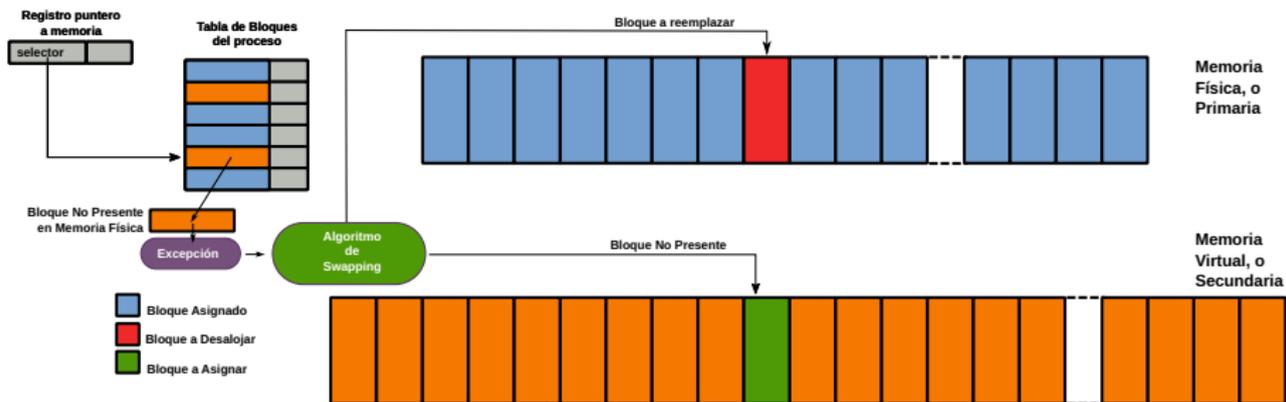
Swapping por Paginación

La excepción es manejada por el S.O. que pone en ese handler el algoritmo de Swapping. La memoria Física está al 100%.



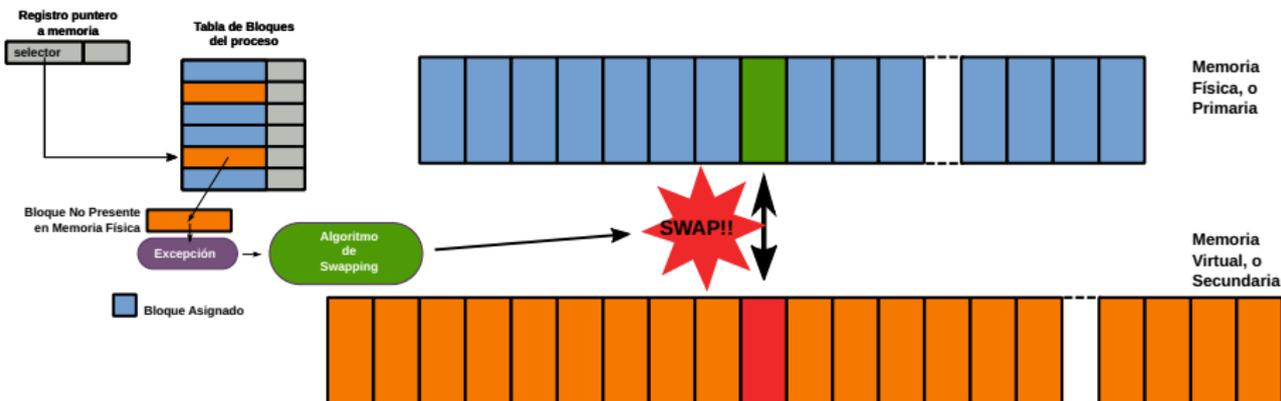
Swapping por Paginación

El algoritmo de Swapping, determina mediante criterios que explicaremos mas adelante en esta clase, cual de las páginas de la memoria Física debe ser desalojada para ubicar allí la página requerida. Además ubica a ésta última en la memoria Virtual.



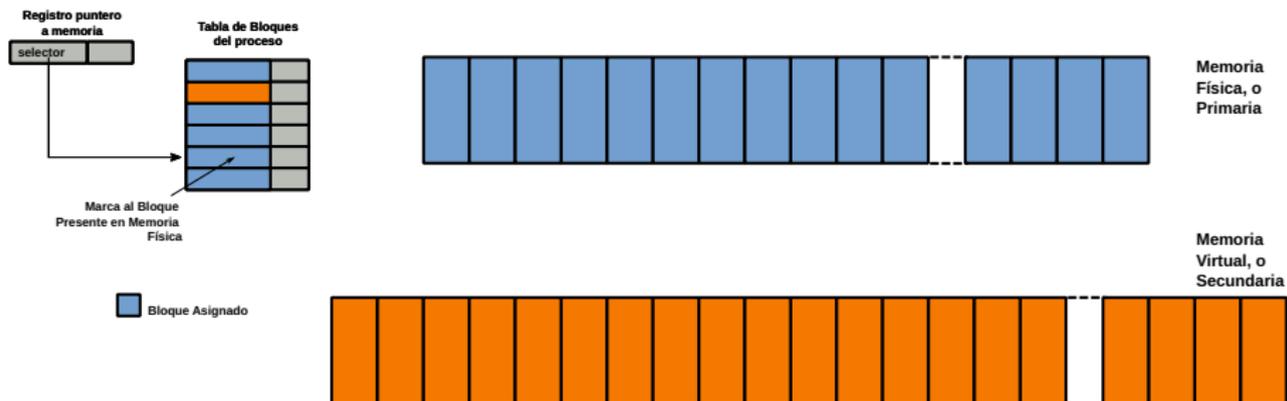
Swapping por Paginación

El algoritmo de Swapping, intercambia directamente ambas páginas sin comprobación adicional de tamaño. Si trabajásemos con segmentos en este punto hay que empezar a ver el tamaño de cada bloque..



Swapping por Paginación

Una vez realizado el intercambio (swap), se modifica el bit de presencia en la Tabla de Páginas del proceso, y los siguientes accesos a memoria se resuelven sin necesidad de swapping.



Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - **Políticas de Administración de memoria**
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Políticas de Fetch

Políticas de Fetch

Por Demanda Se carga una página a memoria principal cuando se genera un fallo de Página. Al principio habrá una cantidad grande de fallos, y luego el principio de vecindad se encargará de mantener la cantidad de fallos en un nivel muy bajo.

Políticas de Fetch

Por Demanda Se carga una página a memoria principal cuando se genera un fallo de Página. Al principio habrá una cantidad grande de fallos, y luego el principio de vecindad se encargará de mantener la cantidad de fallos en un nivel muy bajo.

Paginación Previa Consiste en aprovechar la capacidad de buffering de los dispositivos de disco actuales, y traer no solo la página referenciada sino sus vecinas, de modo que se aprovecha el tiempo de transferencia para traer más cantidad de páginas que de acuerdo con el principio de vecindad tienen alta probabilidad de ser accedidas en un futuro cercano. El problema es que se pueden traer páginas que finalmente no sean utilizadas.

Políticas y Algoritmos básicos de Reemplazo

Políticas y Algoritmos básicos de Reemplazo

- **Optima** Selecciona la página que mas deberá esperar para ser referenciada. Implica conocer de antemano el comportamiento futuro de cada proceso. Esto es prácticamente imposible. Sirve como referencia para comparar los algoritmos que si son implementables

Políticas y Algoritmos básicos de Reemplazo

- **Optima** Selecciona la página que mas deberá esperar para ser referenciada. Implica conocer de antemano el comportamiento futuro de cada proceso. Esto es prácticamente imposible. Sirve como referencia para comparar los algoritmos que si son implementables
- **LRU**: Least Recently Used. Reemplazará a aquella página que lleva mas tiempo sin ser referenciada. Obedece al principio de vecindad. Tiene mas costo computacional.

Políticas y Algoritmos básicos de Reemplazo

- **Optima** Selecciona la página que mas deberá esperar para ser referenciada. Implica conocer de antemano el comportamiento futuro de cada proceso. Esto es prácticamente imposible. Sirve como referencia para comparar los algoritmos que si son implementables
- **LRU**: Least Recently Used. Reemplazará a aquella página que lleva mas tiempo sin ser referenciada. Obedece al principio de vecindad. Tiene mas costo computacional.
- **FIFO**: First In First Out. Trata a los frames mediante un buffer circular desalojando las páginas de memoria mediante un simple algoritmo Round Robin. Es mas ágil que LRU pero no aplica el principio de vecindad, con lo cual su eficiencia es menor.

Políticas y Algoritmos básicos de Reemplazo

- **Optima** Selecciona la página que mas deberá esperar para ser referenciada. Implica conocer de antemano el comportamiento futuro de cada proceso. Esto es prácticamente imposible. Sirve como referencia para comparar los algoritmos que si son implementables
- **LRU**: Least Recently Used. Reemplazará a aquella página que lleva mas tiempo sin ser referenciada. Obedece al principio de vecindad. Tiene mas costo computacional.
- **FIFO**: First In First Out. Trata a los frames mediante un buffer circular desalojando las páginas de memoria mediante un simple algoritmo Round Robin. Es mas ágil que LRU pero no aplica el principio de vecindad, con lo cual su eficiencia es menor.
- **Clock**: Mejora la eficiencia de LRU, utilizando (si existe) un bit de la entrada de Tabla de páginas (o segmento), que indica que este bloque ha sido accedido.

Algoritmo de reloj

Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.

Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.
- Se arma un buffer circular con todos los descriptors de página presentes en la memoria RAM.

Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.
- Se arma un buffer circular con todos los descriptors de página presentes en la memoria RAM.
- Cada vez que se baja una página a RAM el bit de uso de su descriptor se pone en '1'.

Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.
- Se arma un buffer circular con todos los descriptors de página presentes en la memoria RAM.
- Cada vez que se baja una página a RAM el bit de uso de su descriptor se pone en '1'.
- El algoritmo recorre el buffer circular (como un reloj), buscando un descriptor con el bit de uso en 0.

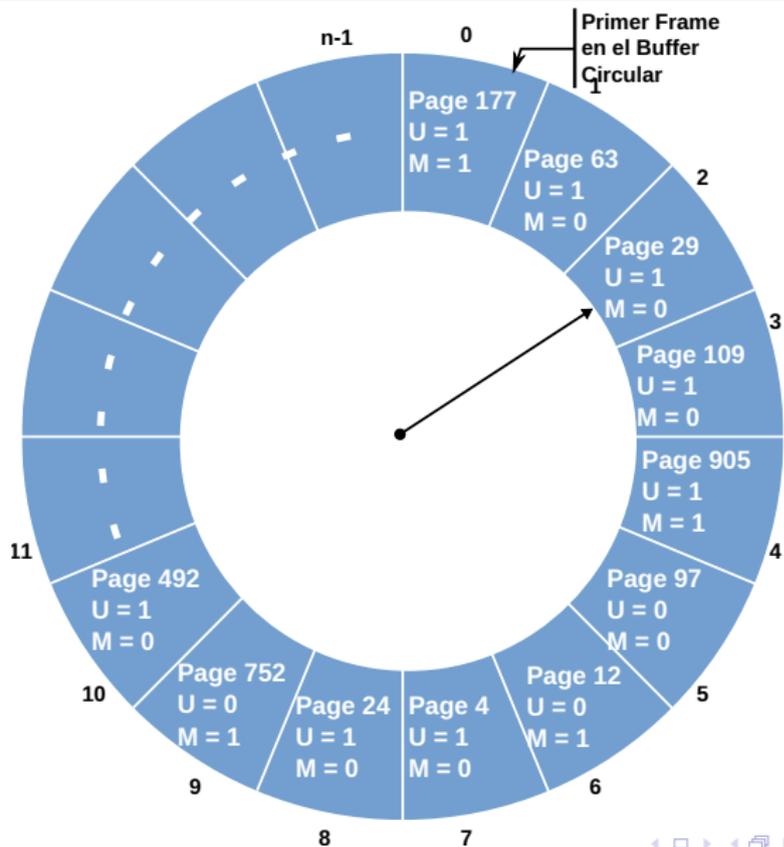
Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.
- Se arma un buffer circular con todos los descriptors de página presentes en la memoria RAM.
- Cada vez que se baja una página a RAM el bit de uso de su descriptor se pone en '1'.
- El algoritmo recorre el buffer circular (como un reloj), buscando un descriptor con el bit de uso en 0.
- Mientras encuentre con el bit de uso en '1', se lo cambia a '0' y pasa al siguiente.

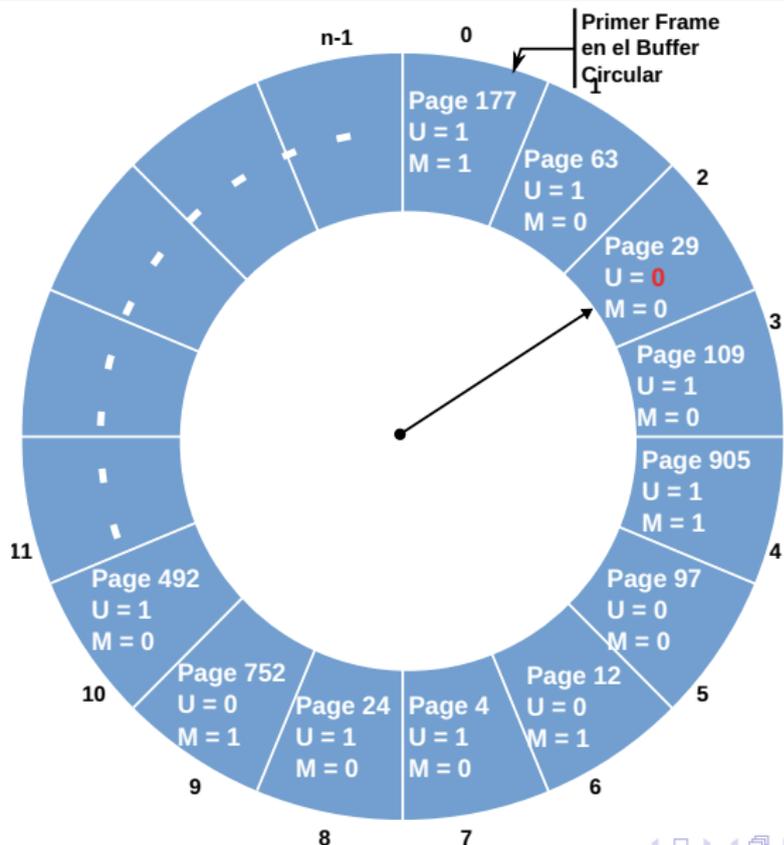
Algoritmo de reloj

- Se basa en que cada descriptor de página lleva un bit de Uso o Acceso.
- Se arma un buffer circular con todos los descriptores de página presentes en la memoria RAM.
- Cada vez que se baja una página a RAM el bit de uso de su descriptor se pone en '1'.
- El algoritmo recorre el buffer circular (como un reloj), buscando un descriptor con el bit de uso en 0.
- Mientras encuentre con el bit de uso en '1', se lo cambia a '0' y pasa al siguiente.
- El primero que encuentra en 0 es desalojado y en su lugar traída la página requerida

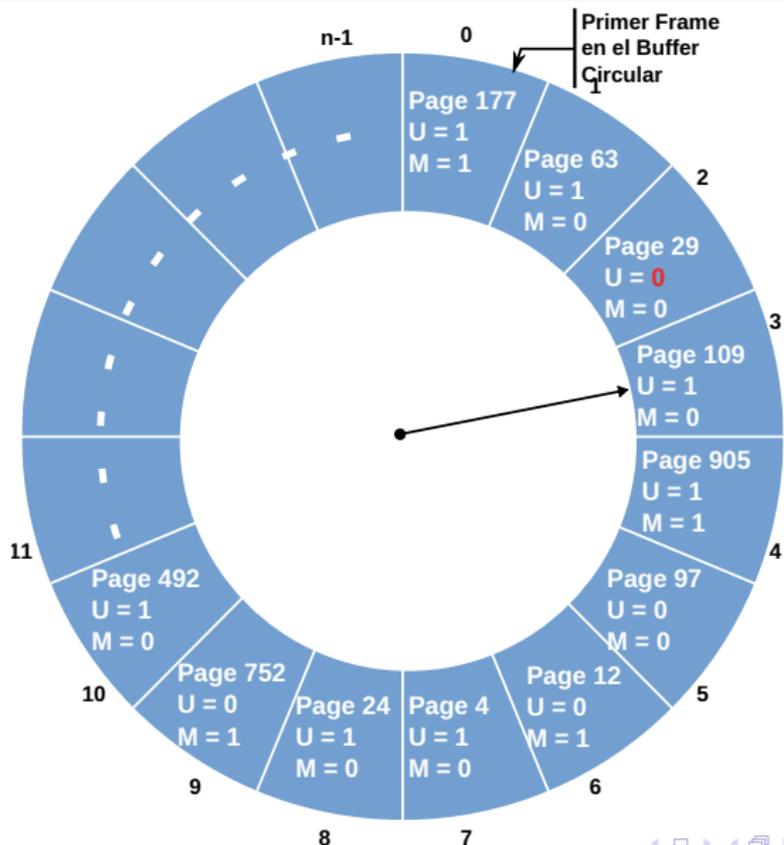
Algoritmo de reloj



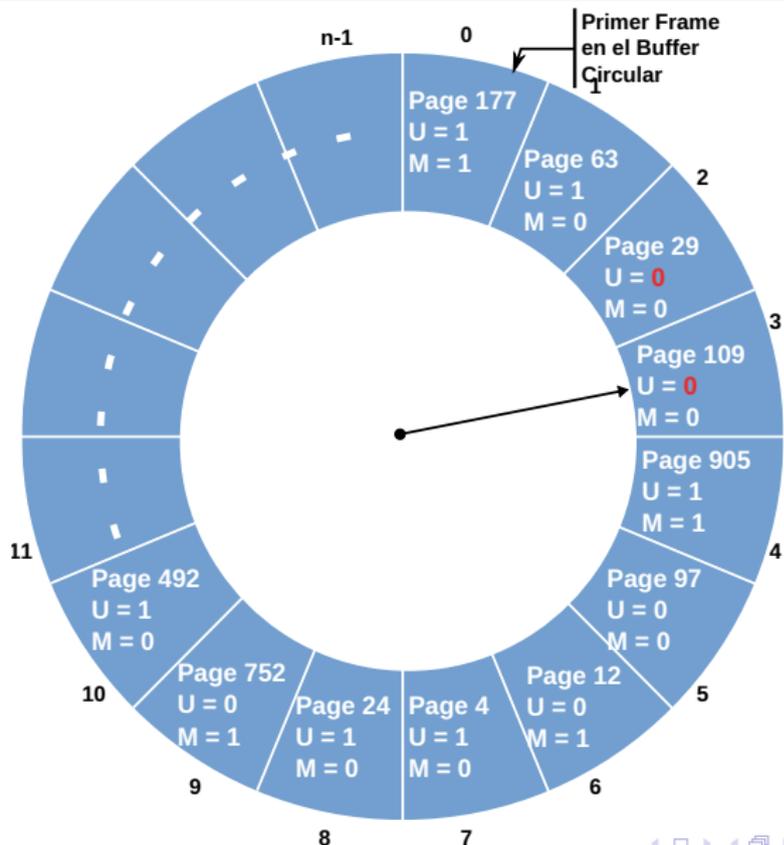
Algoritmo de reloj



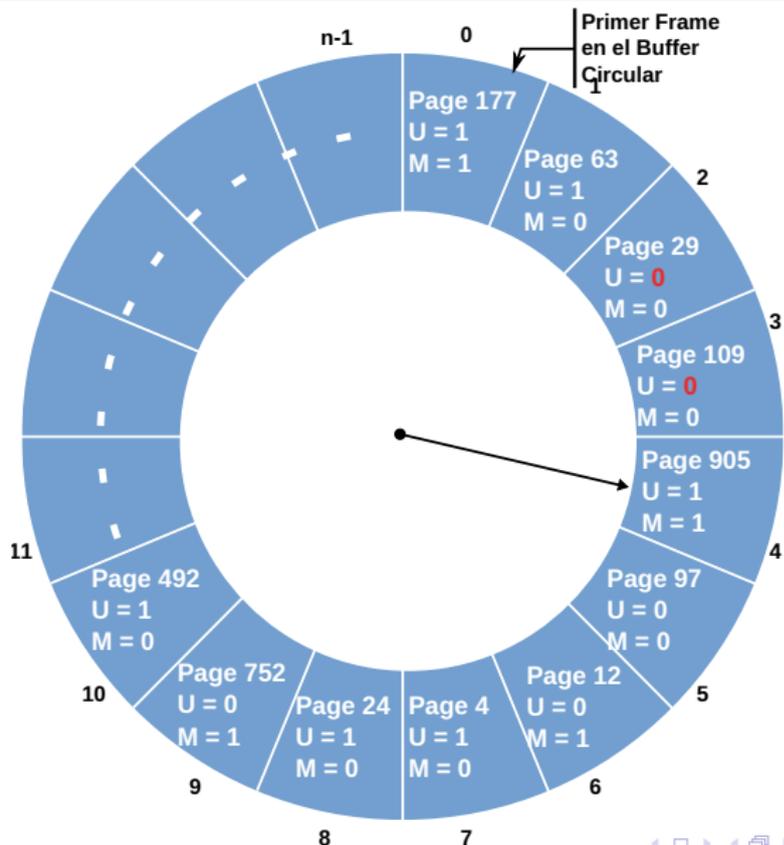
Algoritmo de reloj



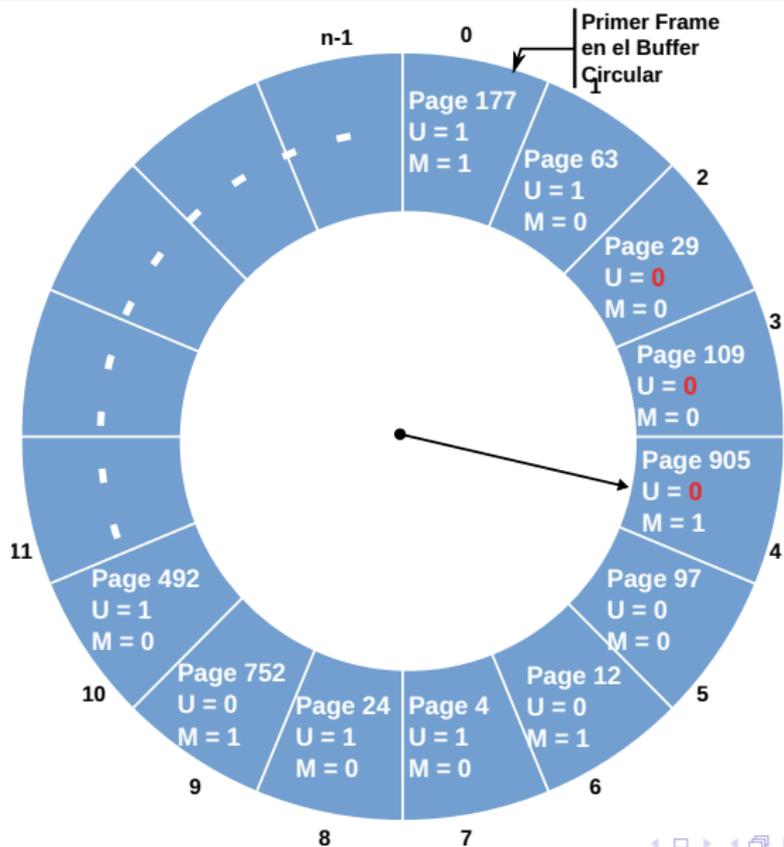
Algoritmo de reloj



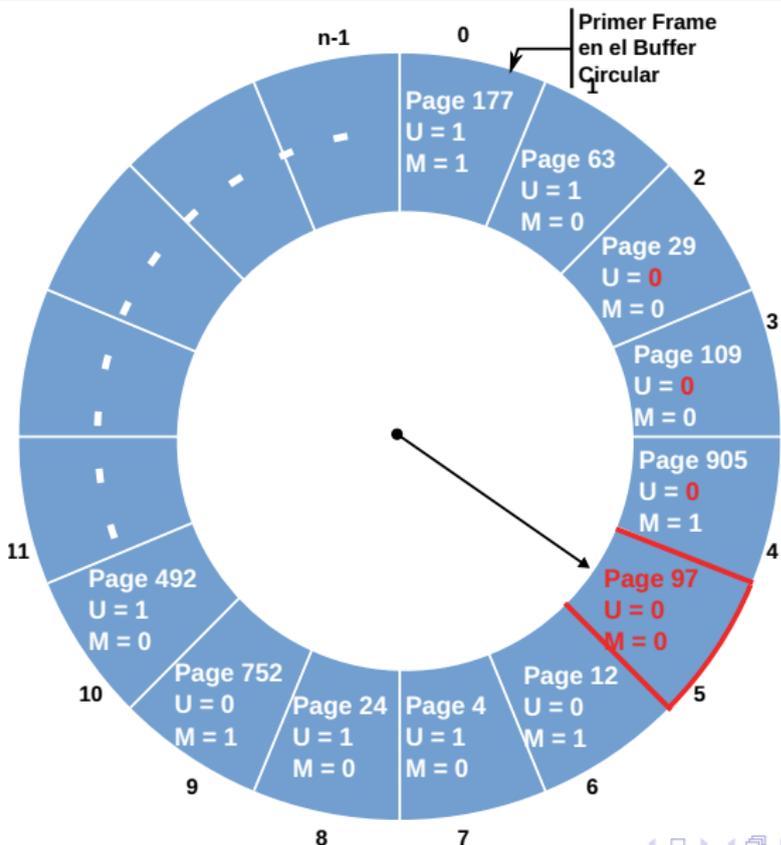
Algoritmo de reloj



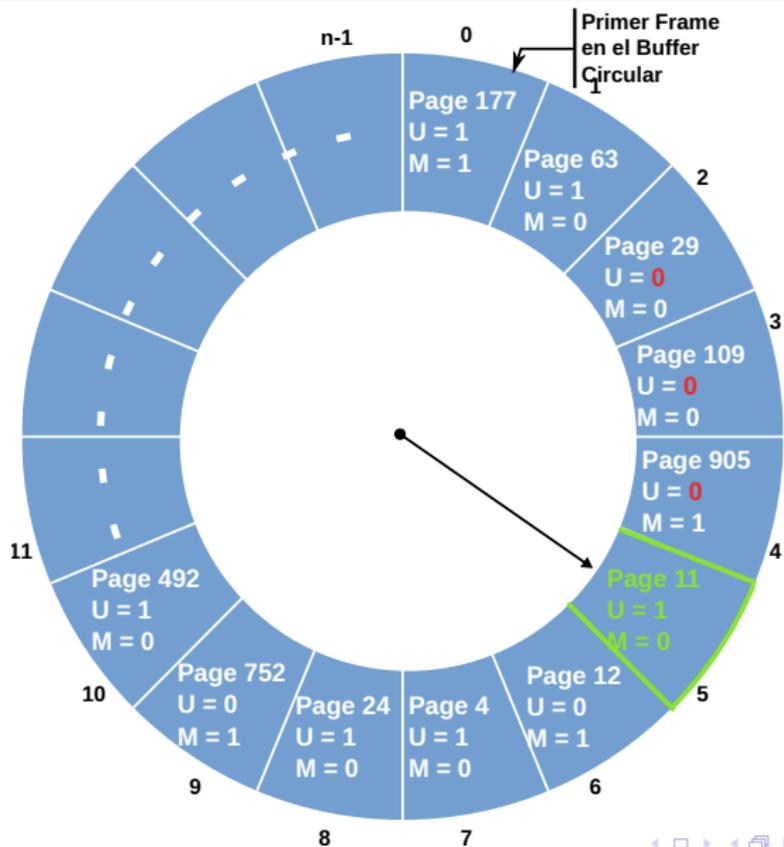
Algoritmo de reloj



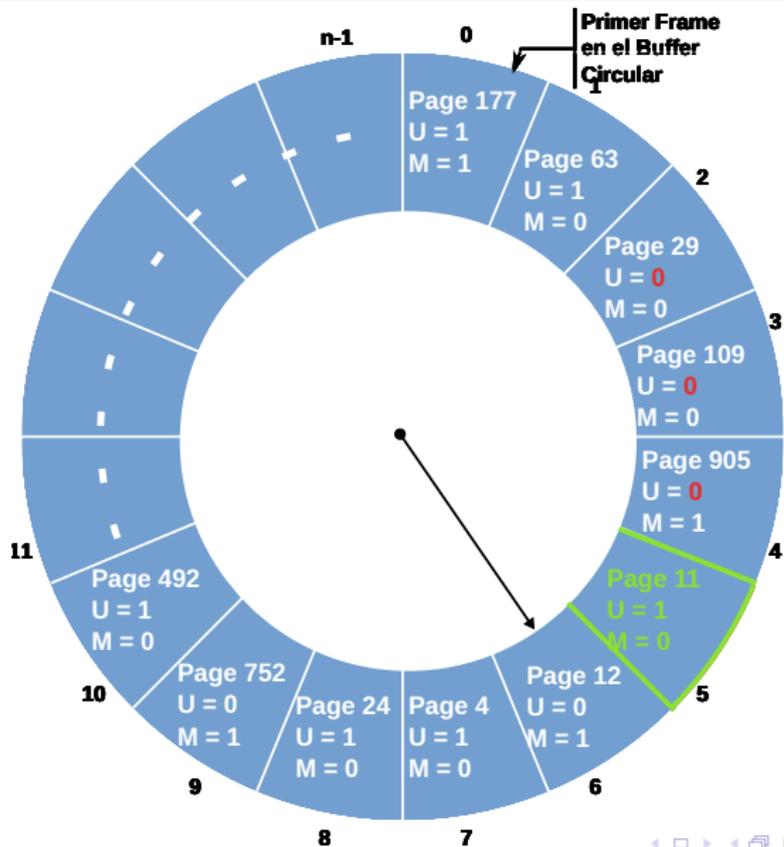
Algoritmo de reloj



Algoritmo de reloj



Algoritmo de reloj



Supongamos esta secuencia de acceso a frames:

2 3 2 1 5 2 4 5 3 2 5 2

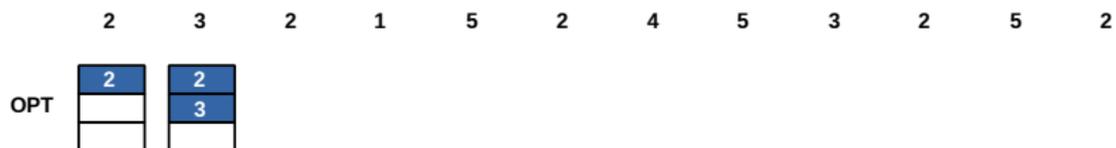
Algoritmo Optimo

2 3 2 1 5 2 4 5 3 2 5 2

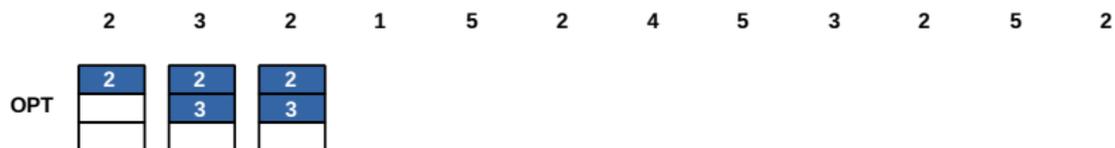
OPT

2

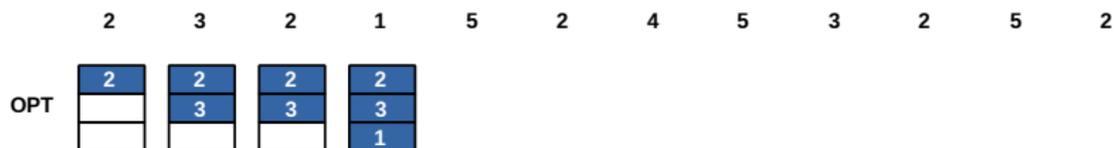
Algoritmo Optimo



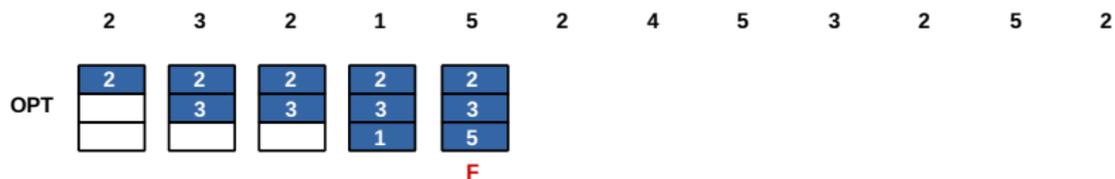
Algoritmo Optimo



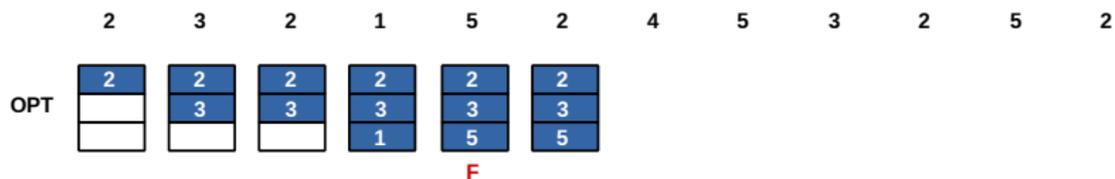
Algoritmo Optimo



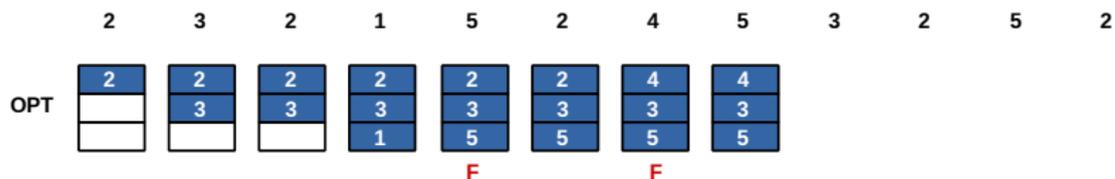
Algoritmo Optimo



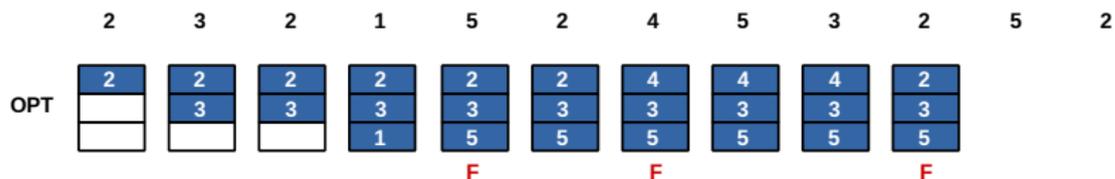
Algoritmo Optimo



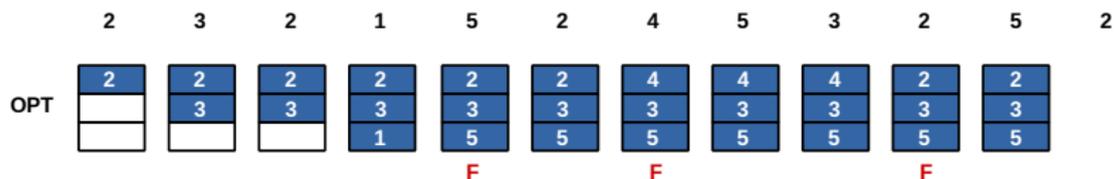
Algoritmo Optimo



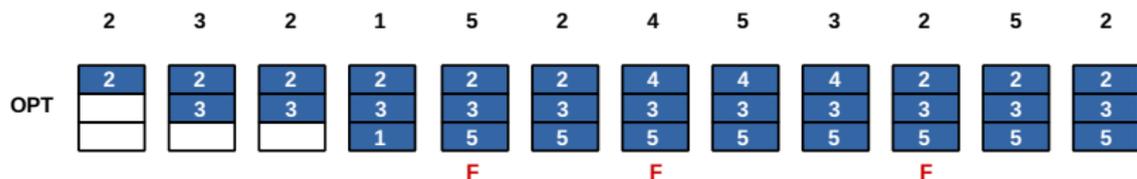
Algoritmo Optimo



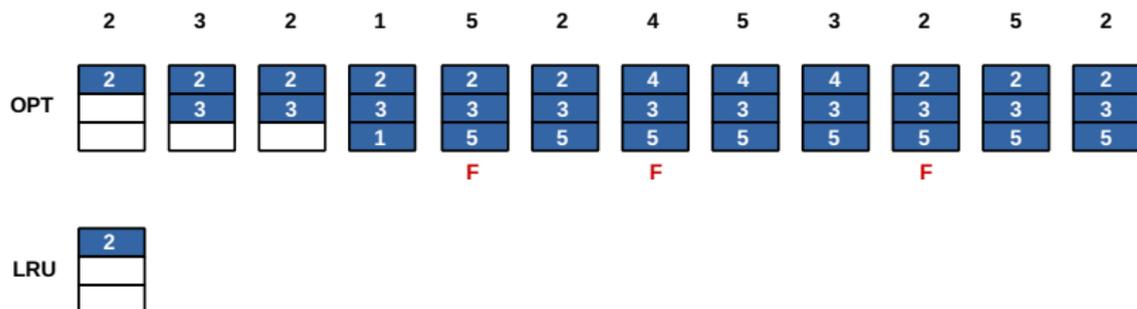
Algoritmo Optimo



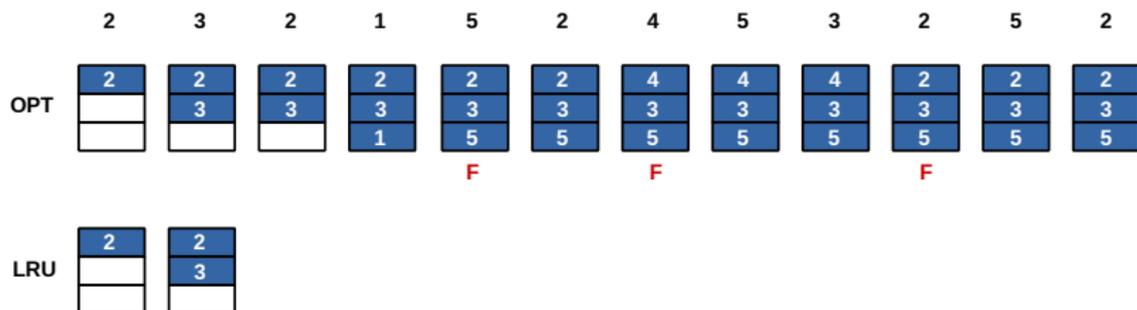
Algoritmo Optimo



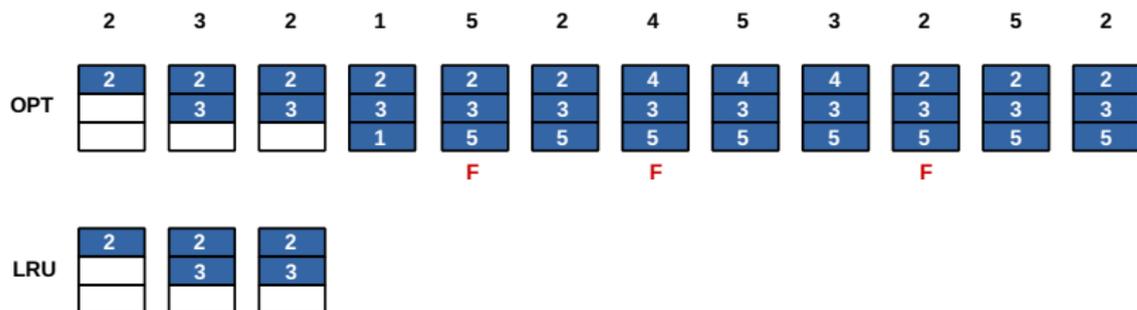
Algoritmo LRU



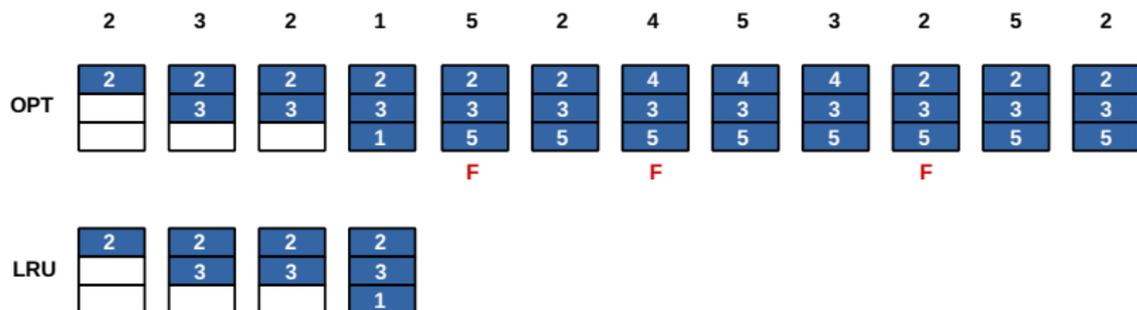
Algoritmo LRU



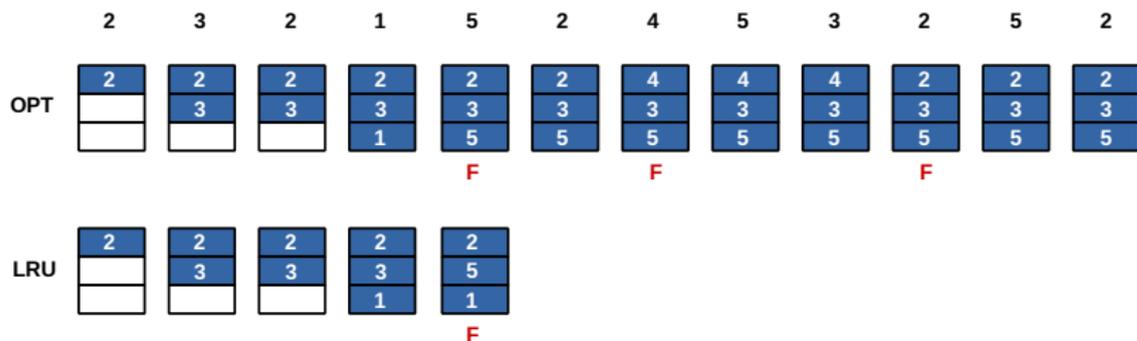
Algoritmo LRU



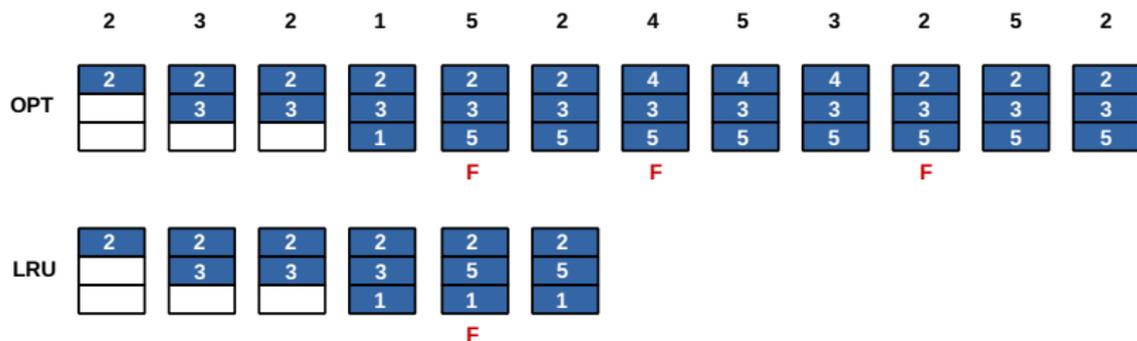
Algoritmo LRU



Algoritmo LRU



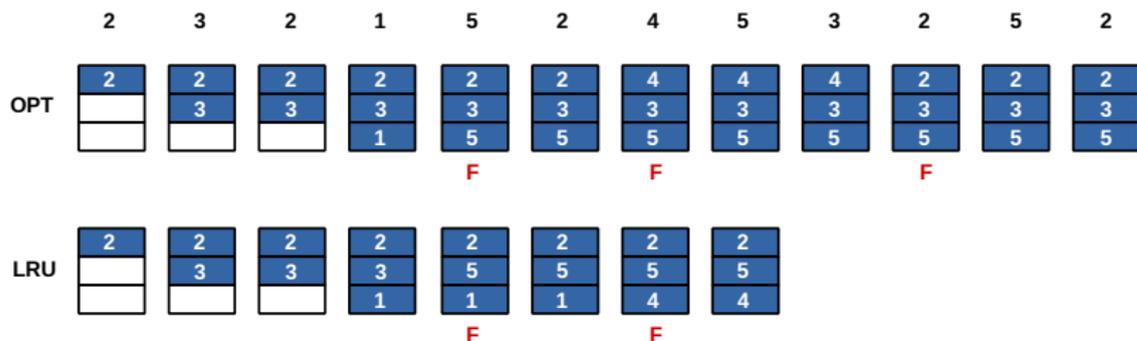
Algoritmo LRU



Algoritmo LRU

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4																				
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																

Algoritmo LRU



Algoritmo LRU

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4												
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																

Algoritmo LRU

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2								
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																

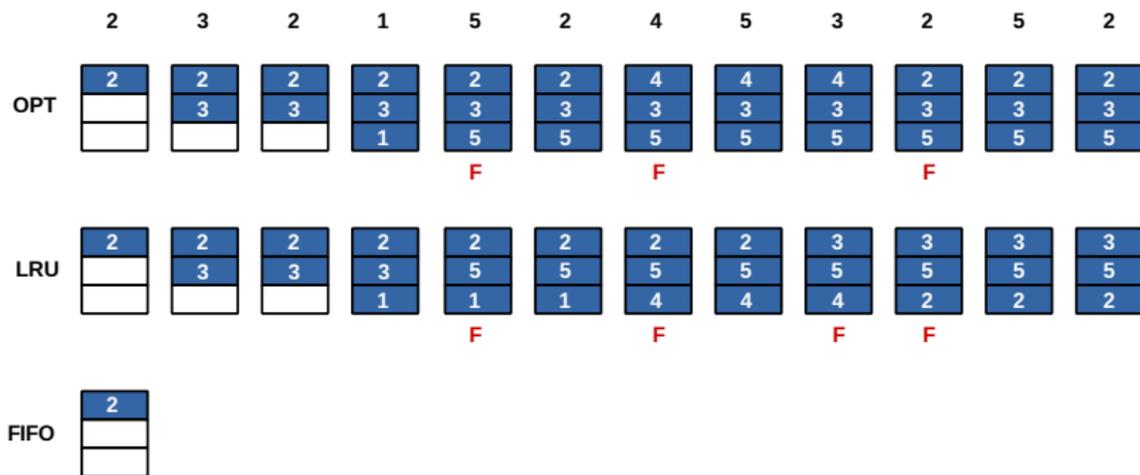
Algoritmo LRU

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2				
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																

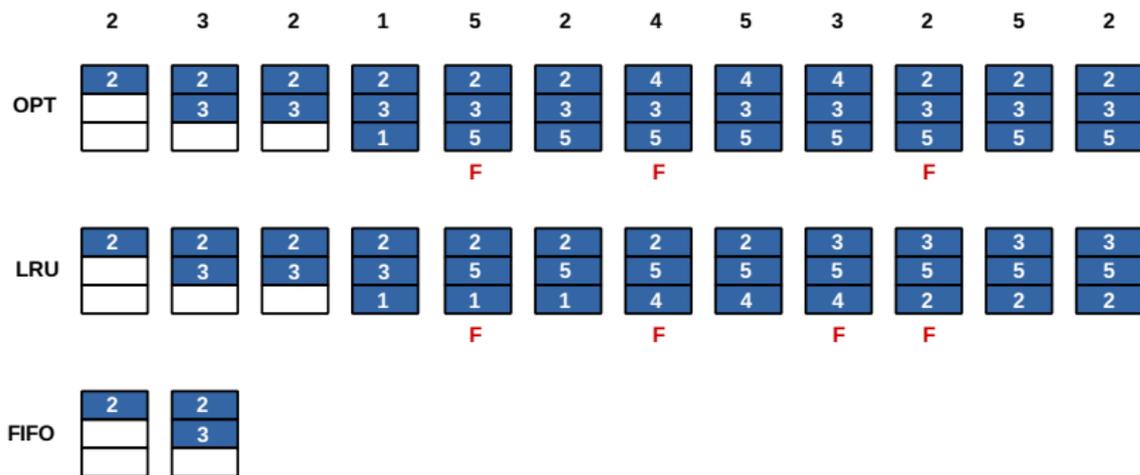
Algoritmo LRU

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																

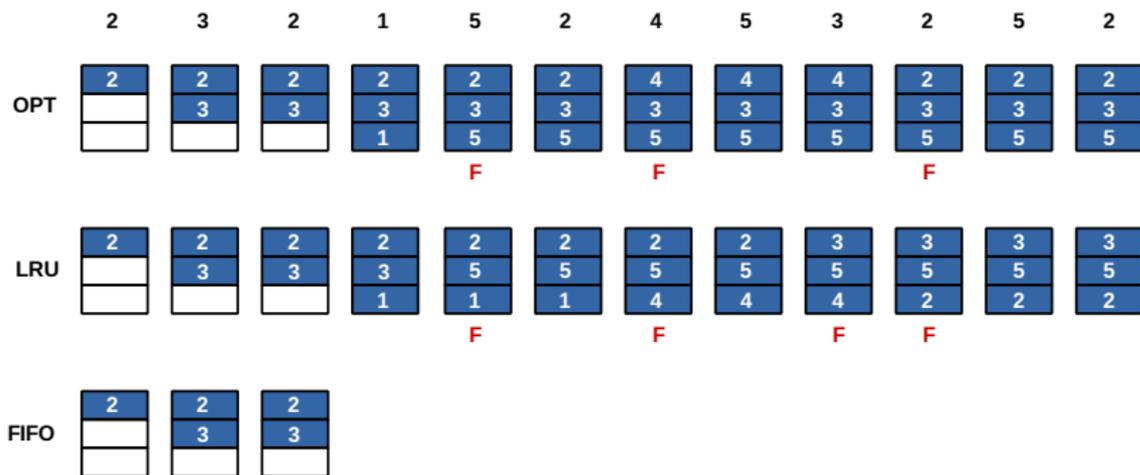
Algoritmo FIFO



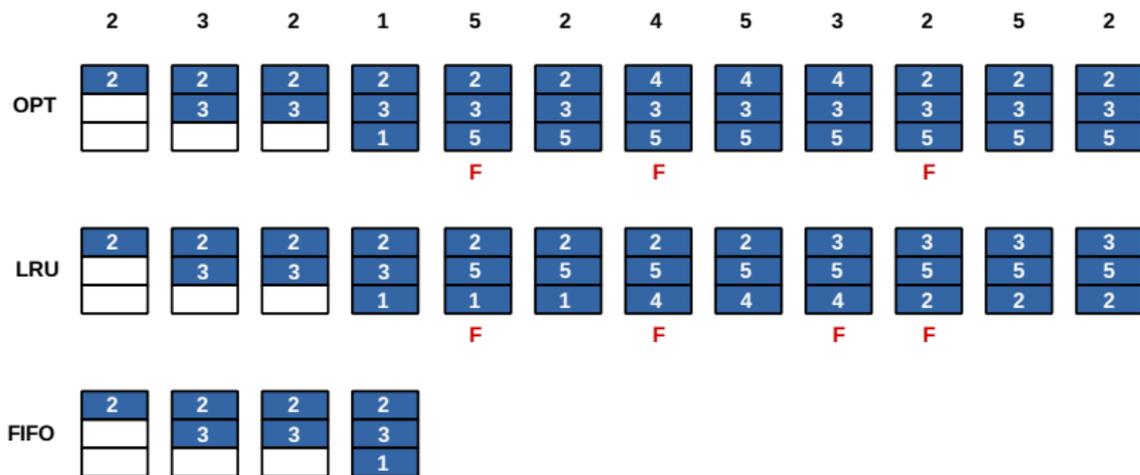
Algoritmo FIFO



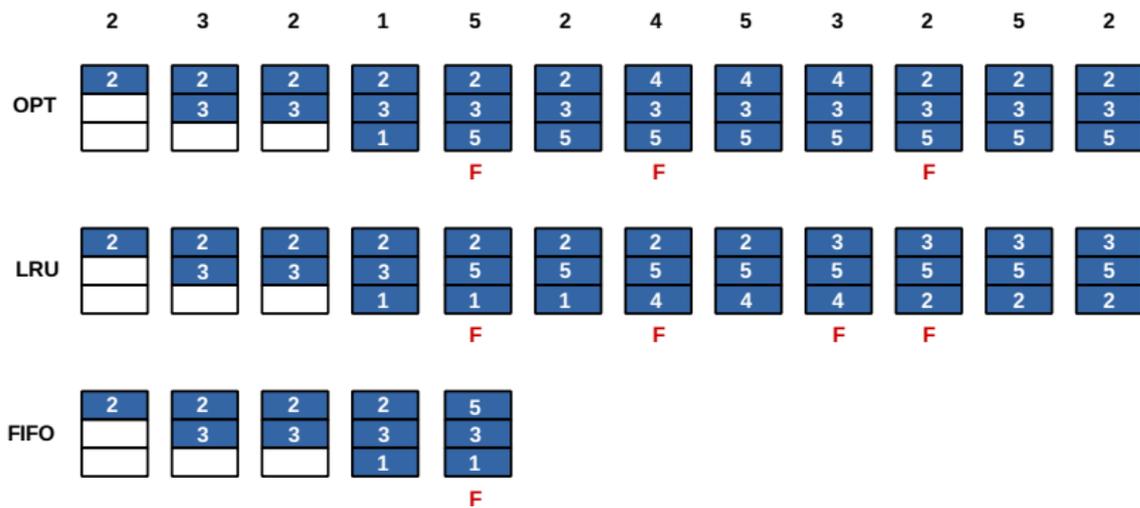
Algoritmo FIFO



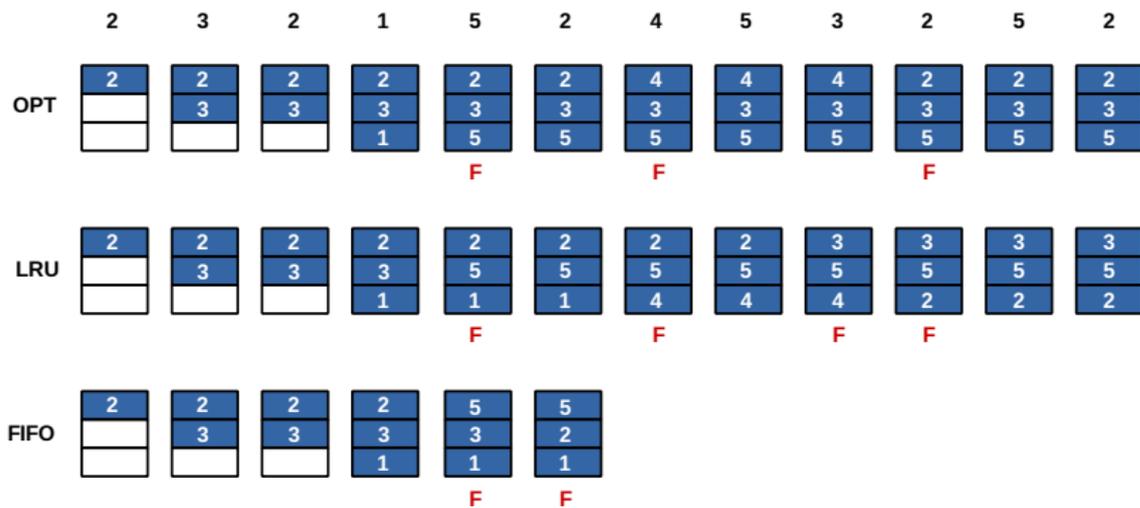
Algoritmo FIFO



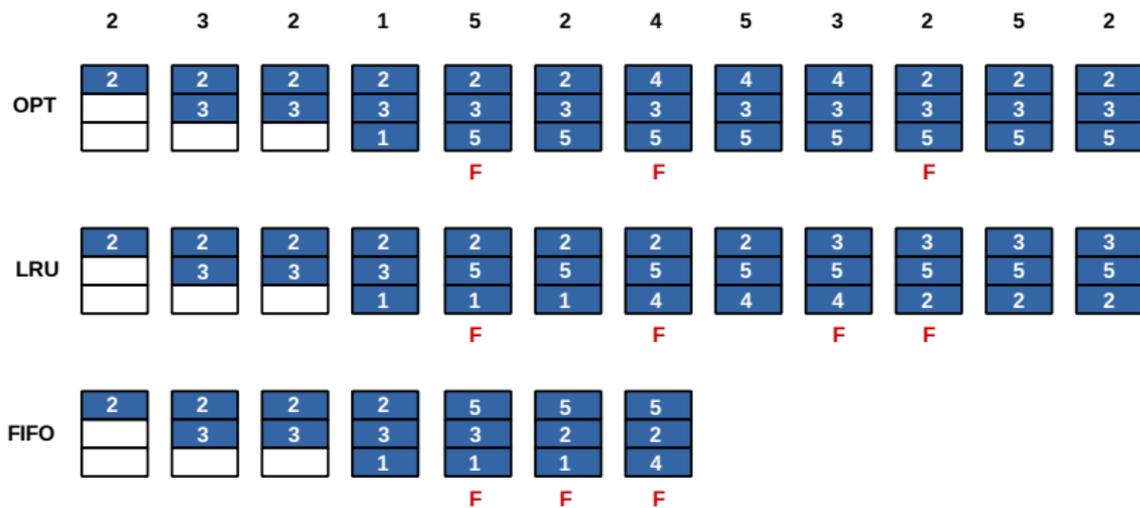
Algoritmo FIFO



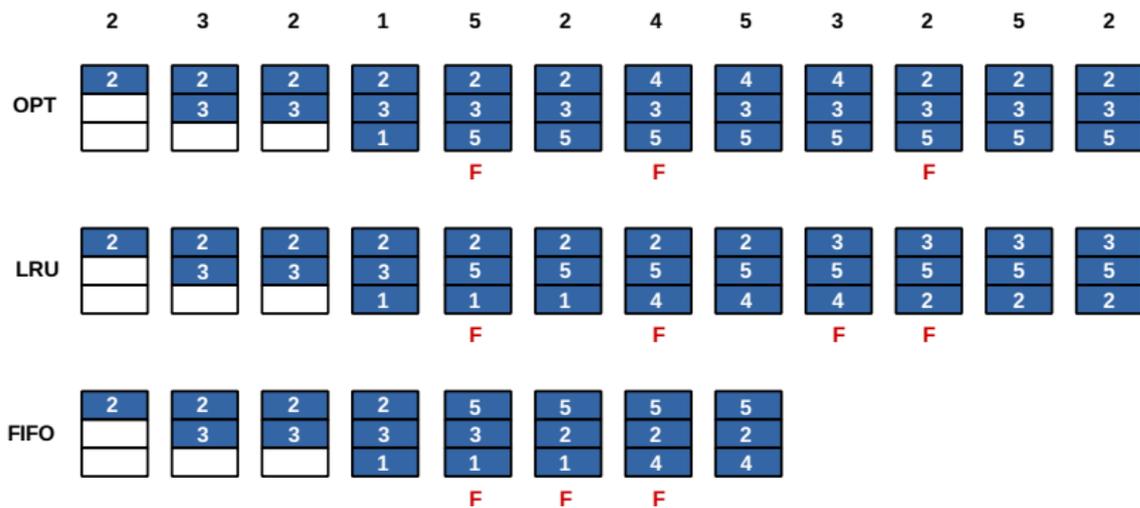
Algoritmo FIFO



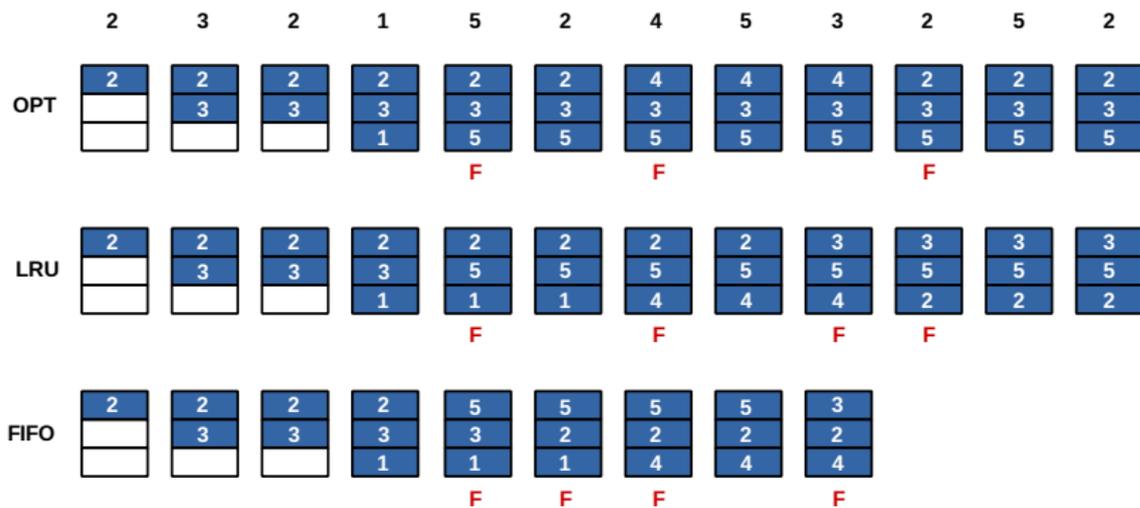
Algoritmo FIFO



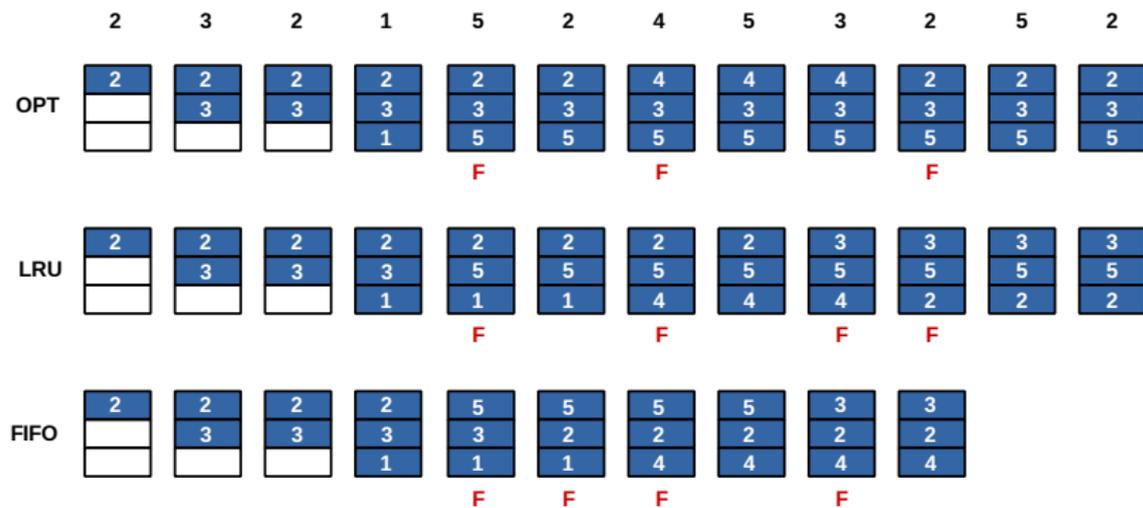
Algoritmo FIFO



Algoritmo FIFO



Algoritmo FIFO



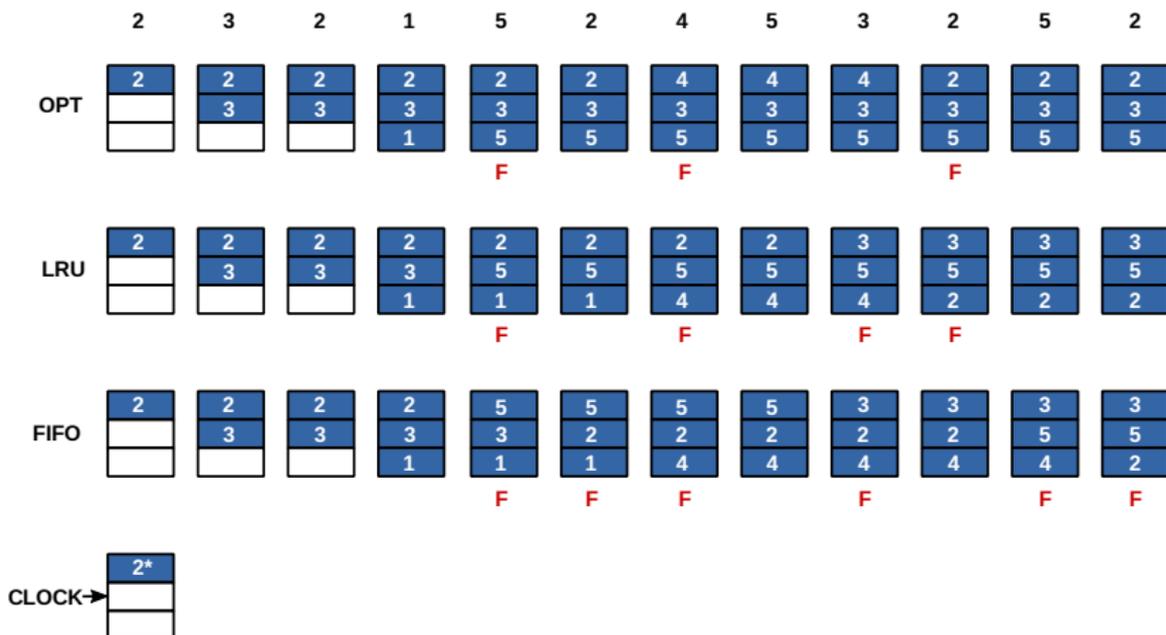
Algoritmo FIFO

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
FIFO	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>5</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table> F	5	3	1	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>1</td></tr></table> F	5	2	1	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	5	2	4	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	5	2	4	<table border="1"><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	3	2	4	<table border="1"><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	3	2	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4				
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
5																																																
3																																																
1																																																
5																																																
2																																																
1																																																
5																																																
2																																																
4																																																
5																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
5																																																
4																																																

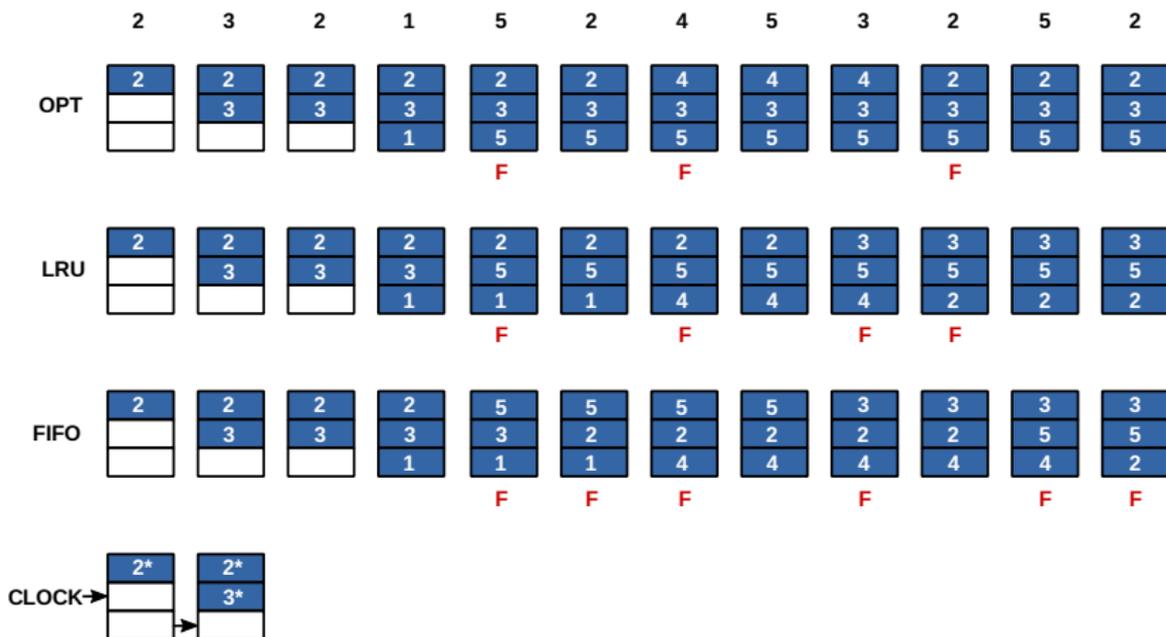
Algoritmo FIFO

	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table> F	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
LRU	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> F	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	2	5	4	<table border="1"><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
FIFO	<table border="1"><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table border="1"><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table border="1"><tr><td>5</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table> F	5	3	1	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>1</td></tr></table> F	5	2	1	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	5	2	4	<table border="1"><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	5	2	4	<table border="1"><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	3	2	4	<table border="1"><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	3	2	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table border="1"><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
5																																																
3																																																
1																																																
5																																																
2																																																
1																																																
5																																																
2																																																
4																																																
5																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																

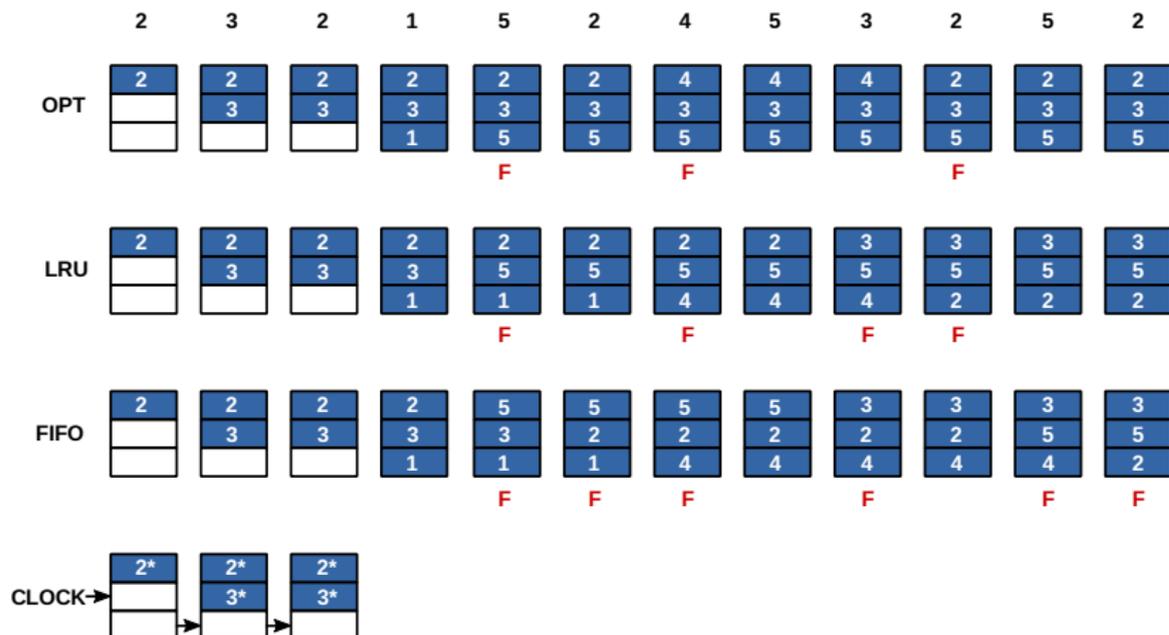
Algoritmo Reloj



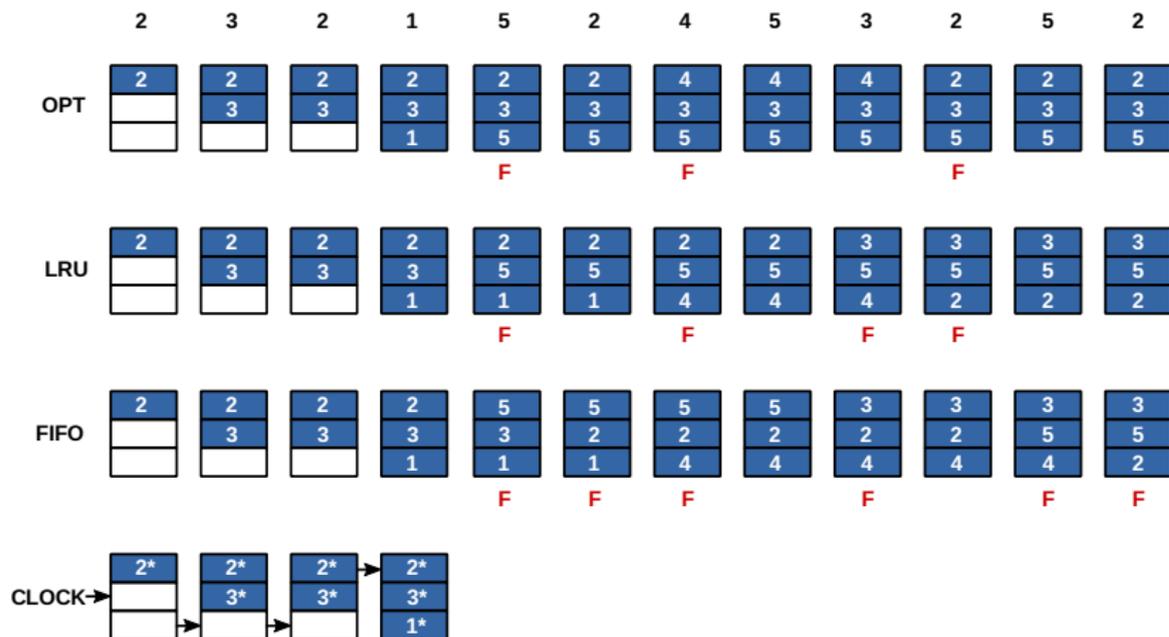
Algoritmo Reloj



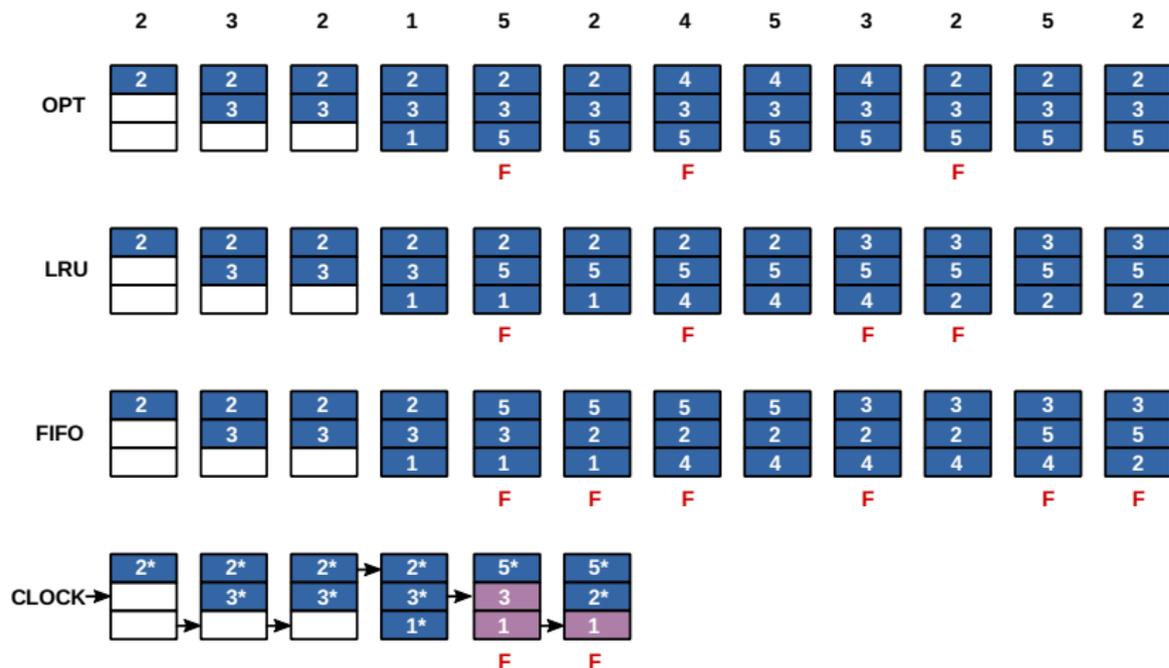
Algoritmo Reloj



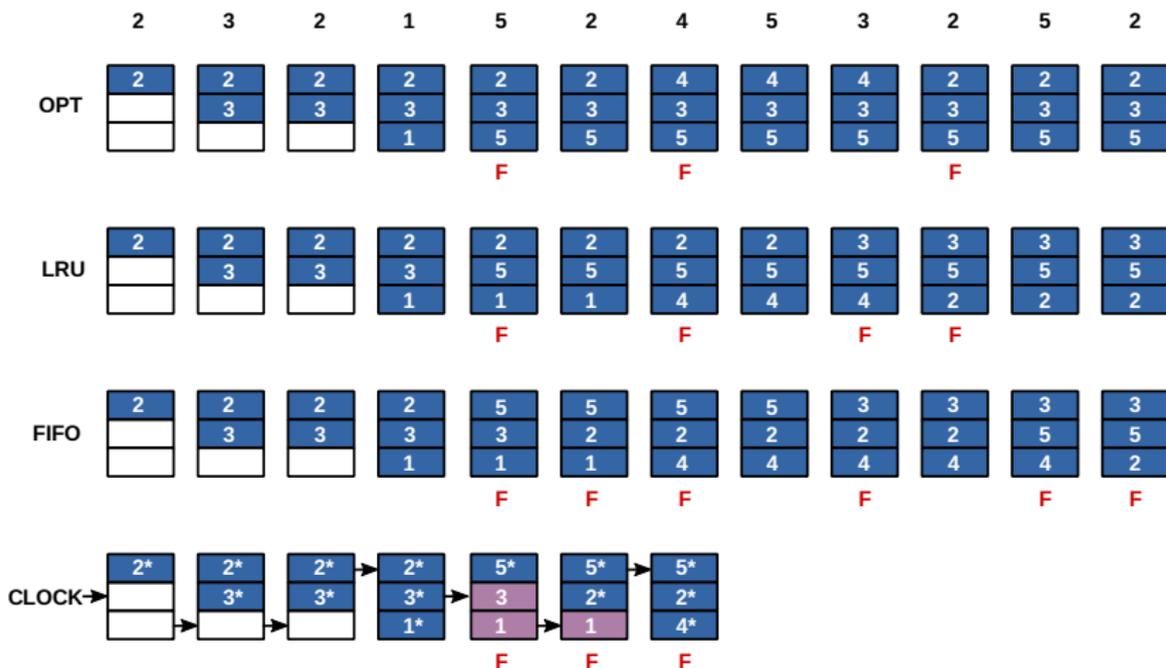
Algoritmo Reloj



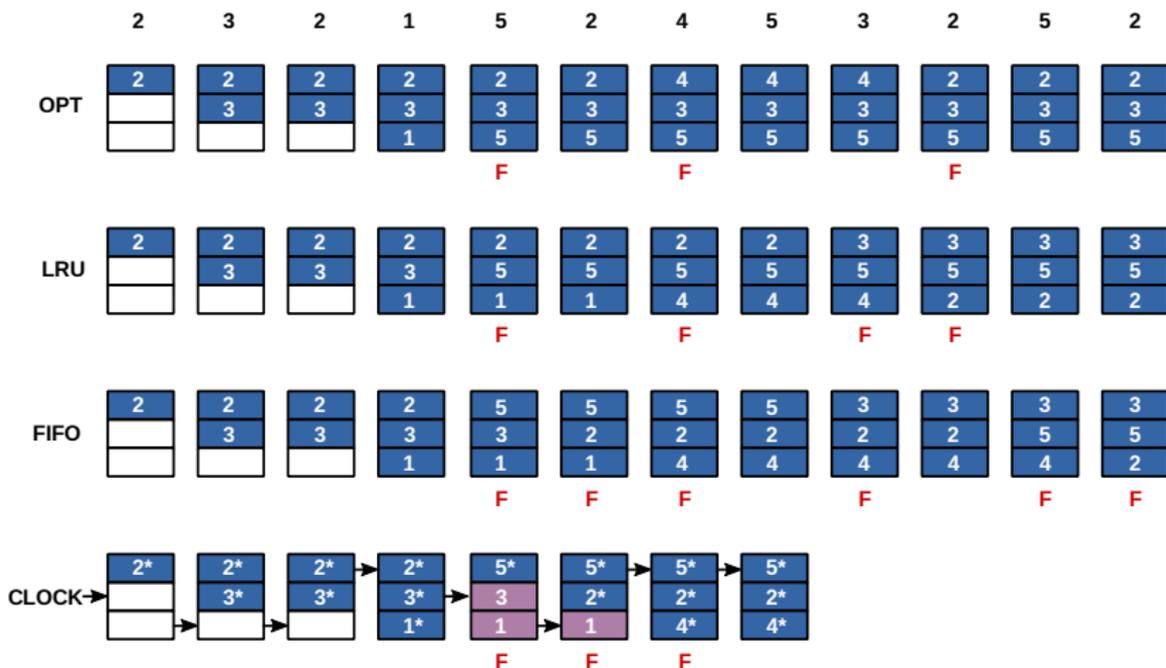
Algoritmo Reloj



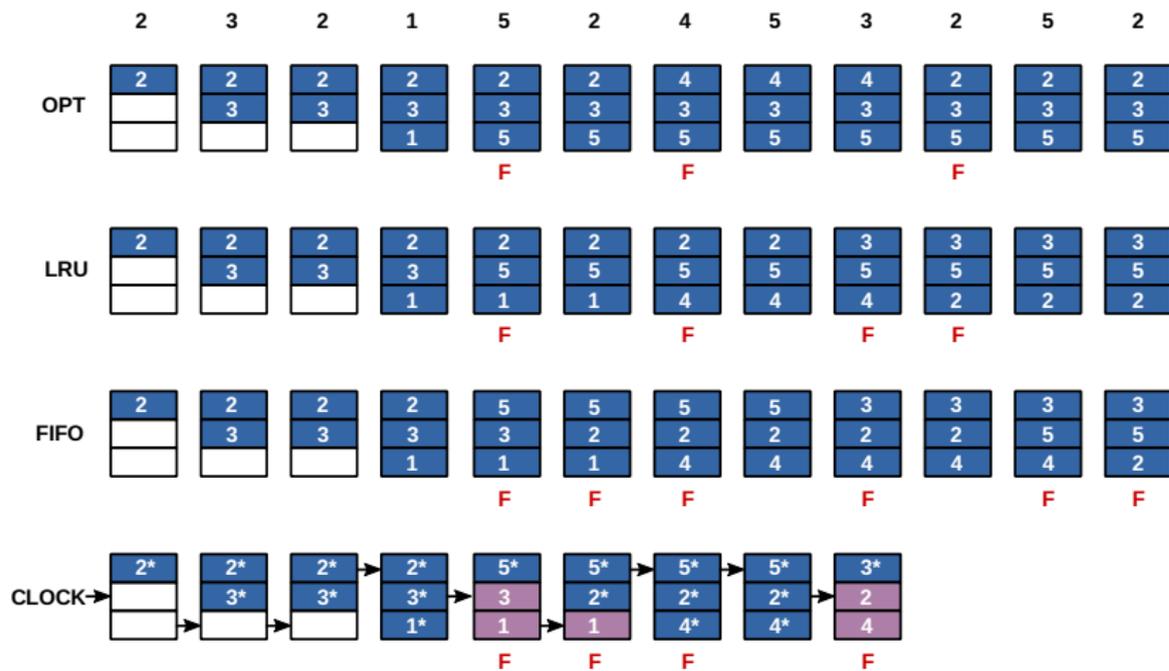
Algoritmo Reloj



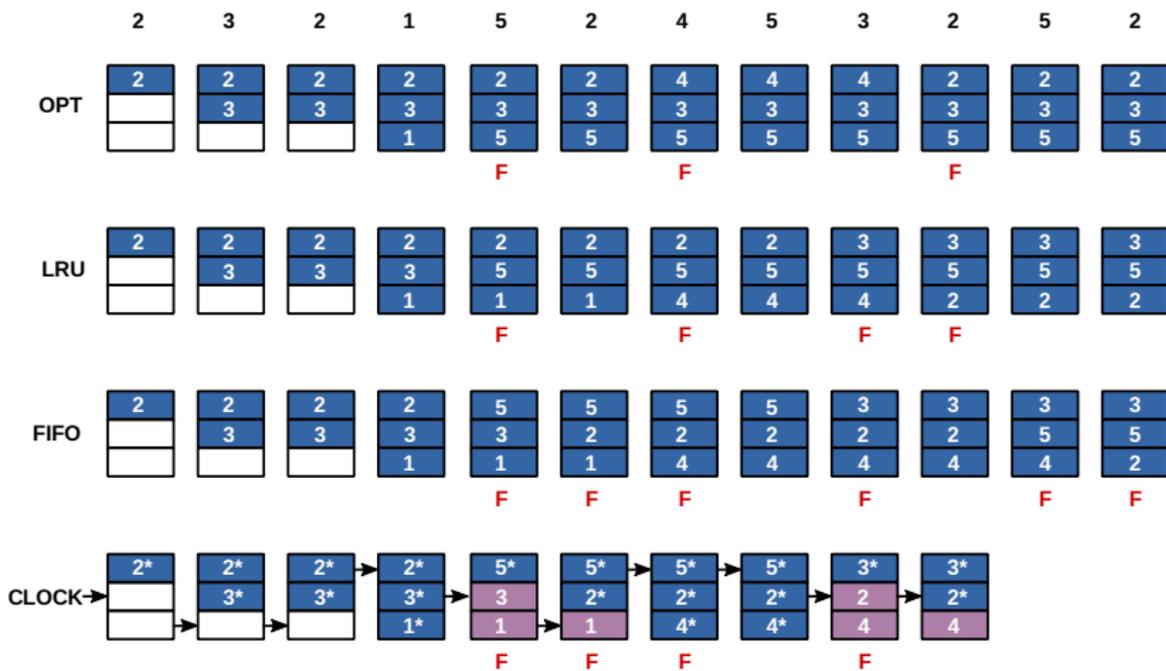
Algoritmo Reloj



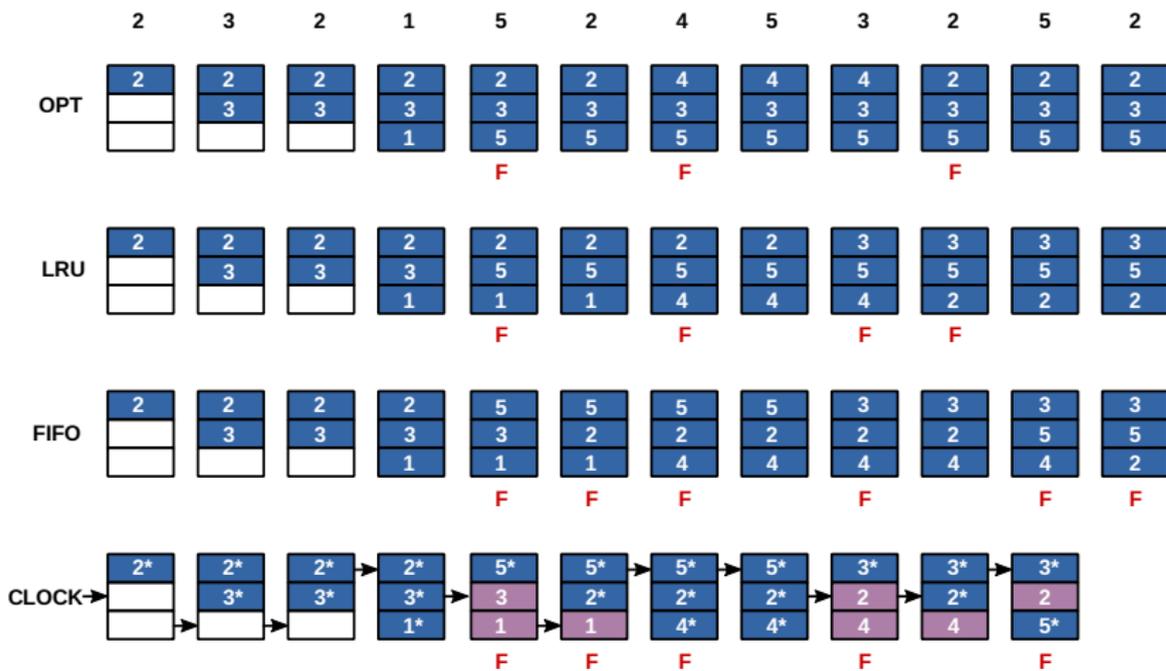
Algoritmo Reloj



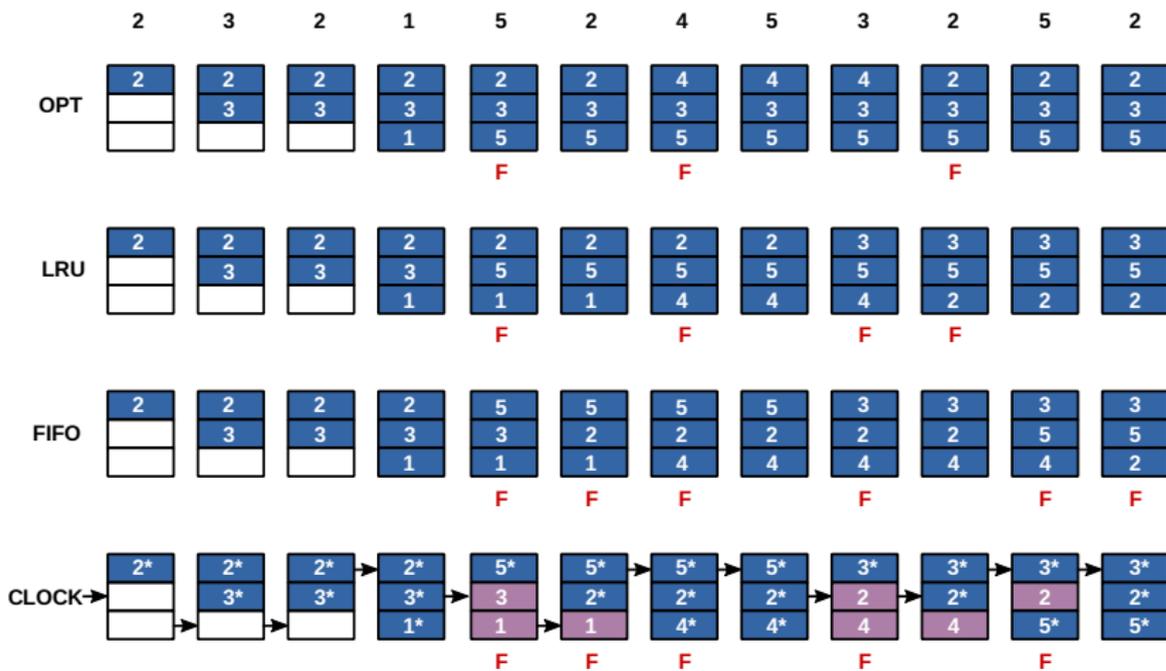
Algoritmo Reloj



Algoritmo Reloj



Algoritmo Reloj



Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - **MMU**
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Tablas de página

El sistema de Paginación consta de un grupo de tablas de página residentes en la memoria principal que permite interpretar para cada **Dirección Virtual** el número de página (page frame) y el offset.



Tablas de página en niveles jerárquicos

Si la memoria principal tiene un tamaño muy grande, utilizar una sola tabla de páginas puede ser muy costoso en términos de consumo de memoria.

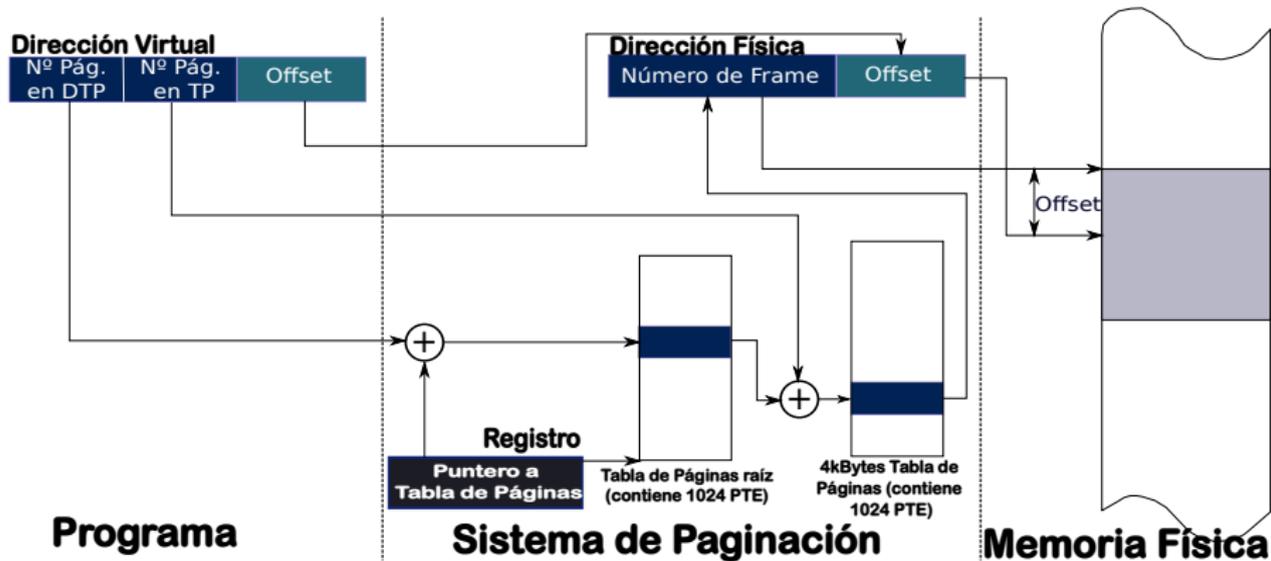
Un espacio de direccionamiento de 4 GiB requiere de una tabla con 1.048.576 descriptores de pagina. Si cada Descriptor de la tabla consume en el mejor de los casos 4 B, nuestra tabla de de descriptores de página ocuparía 4 MiB de memoria. Un disparate.

Una alternativa propuesta al estudiar el tema y que se impuso en todos los casos de procesadores conocidos es utilizar niveles jerárquicos de Tablas de Páginas.

Tablas de página en niveles jerárquicos



Dos niveles jerárquicos para 4 Gbytes de memoria



Buffer de traducciones

Buffer de traducciones

- Cualquiera sea el sistema de traducción de direcciones a implementar en la paginación, se requiere de tablas que contengan las direcciones físicas de comienzo de cada frame en uso, mas sus bits de control para permisos, y demás atributos.

Buffer de traducciones

- Cualquiera sea el sistema de traducción de direcciones a implementar en la paginación, se requiere de tablas que contengan las direcciones físicas de comienzo de cada frame en uso, mas sus bits de control para permisos, y demás atributos.
- Debido a su cantidad, éstas tablas no pueden residir en otro sitio que en la memoria principal del sistema.

Buffer de traducciones

- Cualquiera sea el sistema de traducción de direcciones a implementar en la paginación, se requiere de tablas que contengan las direcciones físicas de comienzo de cada frame en uso, mas sus bits de control para permisos, y demás atributos.
- Debido a su cantidad, éstas tablas no pueden residir en otro sitio que en la memoria principal del sistema.
- En un subsistema de paginación con n niveles jerárquicos, cada acceso a una dirección de memoria principal implicará previamente realizar n lecturas de descriptores desde memoria.

Buffer de traducciones

- Cualquiera sea el sistema de traducción de direcciones a implementar en la paginación, se requiere de tablas que contengan las direcciones físicas de comienzo de cada frame en uso, mas sus bits de control para permisos, y demás atributos.
- Debido a su cantidad, éstas tablas no pueden residir en otro sitio que en la memoria principal del sistema.
- En un subsistema de paginación con n niveles jerárquicos, cada acceso a una dirección de memoria principal implicará previamente realizar n lecturas de descriptores desde memoria.
- La penalización de tiempo de ejecución es evidente. Esto ocurrirá cada vez que se deba resolver la dirección a colocar en el bus de address.

Buffer de traducciones

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.
- Lo mismo con cada memory read o con cada memory write.

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.
- Lo mismo con cada memory read o con cada memory write.
- En estas condiciones el tiempo de ejecución podría medirse con reloj de arena. . .

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.
- Lo mismo con cada memory read o con cada memory write.
- En estas condiciones el tiempo de ejecución podría medirse con reloj de arena. . .
- La pregunta es: ¿Como es entonces que esto funciona en a práctica?

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.
- Lo mismo con cada memory read o con cada memory write.
- En estas condiciones el tiempo de ejecución podría medirse con reloj de arena. . .
- La pregunta es: ¿Como es entonces que esto funciona en a práctica?
- La razón es que se incluye un elemento que evita esta pérdida de rendimiento.

Buffer de traducciones

- Dicho muy claramente: Cada vez que el procesador deba fetchear una instrucción tiene que hacer previamente dos ciclos de bus de lectura de memoria para determinar la dirección, y luego un tercer ciclo para techear.
- Lo mismo con cada memory read o con cada memory write.
- En estas condiciones el tiempo de ejecución podría medirse con reloj de arena. . .
- La pregunta es: ¿Como es entonces que esto funciona en a práctica?
- La razón es que se incluye un elemento que evita esta pérdida de rendimiento.
- Por lo general todos los procesadores que cuentan con soporte de paginación incluyen un buffer de traducción conocido como **TLB**, por sus siglas en inglés de **T**raslation **L**ookaside **B**uffer.

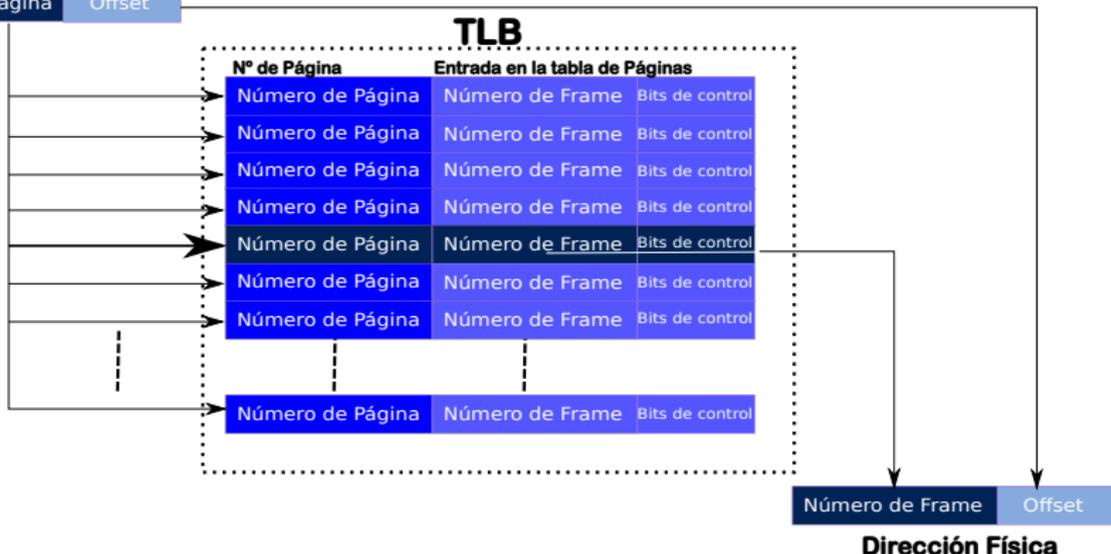
Traslation Lookaside Buffer

Dirección Virtual

Número de Página

Offset

TLB



Traducción por Tablas de página

Traducción por Tablas de página

La *Dirección Virtual* puede verse como compuesta por dos campos:

Traducción por Tablas de página

La *Dirección Virtual* puede verse como compuesta por dos campos:

- 1 Número de Página

Traducción por Tablas de página

La ***Dirección Virtual*** puede verse como compuesta por dos campos:

- 1 Número de Página
- 2 Desplazamiento (Offset) dentro de la página seleccionada.

Traducción por Tablas de página

La **Dirección Virtual** puede verse como compuesta por dos campos:

- 1 Número de Página
- 2 Desplazamiento (Offset) dentro de la página seleccionada.

Dirección Virtual

Número de Página

Offset

Entrada en la tabla de Páginas

Número de Frame

Bits de control
WAPM

Traducción por Tablas de página

La **Dirección Virtual** puede verse como compuesta por dos campos:

- 1 Número de Página
- 2 Desplazamiento (Offset) dentro de la página seleccionada.

Dirección Virtual

Número de Página

Offset

Entrada en la tabla de Páginas

Número de Frame

Bits de control WAPM

Con el Número de Página se ingresa a la tabla de páginas y se obtiene el frame de memoria física correspondiente.

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - **Page faults**
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Page faults

Page faults

- Parte del soporte de la CPU al Sistema Operativo para implementar Memoria Virtual, consiste en advertir cuando se requiere iniciar un swapping

Page faults

- Parte del soporte de la CPU al Sistema Operativo para implementar Memoria Virtual, consiste en advertir cuando se requiere iniciar un swapping
- Esto se puede lograr con algunos bits de estado adicionales cada entrada de tabla de páginas.

Page faults

- Parte del soporte de la CPU al Sistema Operativo para implementar Memoria Virtual, consiste en advertir cuando se requiere iniciar un swapping
- Esto se puede lograr con algunos bits de estado adicionales cada entrada de tabla de páginas.

Dirección Virtual

Número de Página

Offset

Entrada en la tabla de Páginas

Número de Frame

Bits de control WAPM

Bits de control de la entrada a la Tabla de Página

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.
- **P** indica si la página seleccionada está Presente en la memoria física. Si es '0' está en la memoria secundaria.

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.
- **P** indica si la página seleccionada está Presente en la memoria física. Si es '0' está en la memoria secundaria.
- Si se intenta acceder a una página cuyo bit **P** esté en '0', la CPU genera una excepción. El kernel toma el control y realiza el swapping.

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.
- **P** indica si la página seleccionada está Presente en la memoria física. Si es '0' está en la memoria secundaria.
- Si se intenta acceder a una página cuyo bit **P** esté en '0', la CPU genera una excepción. El kernel toma el control y realiza el swapping.
- **M** indica si una página fue ("1") o no ("0") modificada durante su permanencia en memoria física.

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.
- **P** indica si la página seleccionada está Presente en la memoria física. Si es '0' está en la memoria secundaria.
- Si se intenta acceder a una página cuyo bit **P** esté en '0', la CPU genera una excepción. El kernel toma el control y realiza el swapping.
- **M** indica si una página fue ("1") o no ("0") modificada durante su permanencia en memoria física.
- Si e "1" se salva la página en memoria secundaria, o se escribe el frame directamente en la página que se traerá de memoria secundaria.

Bits de control de la entrada a la Tabla de Página

- **W** indica si la página acepta escritura. Sino es Read Only. De gran utilidad para Copy-On-Write.
- **P** indica si la página seleccionada está Presente en la memoria física. Si es '0' está en la memoria secundaria.
- Si se intenta acceder a una página cuyo bit **P** esté en '0', la CPU genera una excepción. El kernel toma el control y realiza el swapping.
- **M** indica si una página fue ("1") o no ("0") modificada durante su permanencia en memoria física.
- Si e "1" se salva la página en memoria secundaria, o se escribe el frame directamente en la página que se traerá de memoria secundaria.
- **A** indica si la página ha sido Accedida o no. De gran utilidad para el algoritmo de selección de la página a desalojar de memoria.

Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - Copy-on-write

Generalidades

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone compila para diferentes microprocesadores.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.
- Implementar un código suficientemente flexible para un procesador x86 en lo que a paginación se refiere por si solo se hace un interesante problema, que implica detenerse a pensar de que forma definir una única estructura que sirva para implementar Paginación de 32 bits, PAE, o IA-32e.

Generalidades

- Linux es un sistema operativo multiplataforma, lo cual supone compila para diferentes microprocesadores.
- Esto crea la obligación en los desarrolladores de código de emplear estructuras de programación flexibles y portables.
- No basta para ello simplemente escribir código en C, sino hacerlo para que este resulte suficiente para cumplir estos objetivos.
- Implementar un código suficientemente flexible para un procesador x86 en lo que a paginación se refiere por si solo se hace un interesante problema, que implica detenerse a pensar de que forma definir una única estructura que sirva para implementar Paginación de 32 bits, PAE, o IA-32e.
- ARM también como veremos tiene diferentes modos.

Generalidades

Generalidades

- En general todos los procesadores tienen una estructura de paginación que básicamente se encarga de traducir una **dirección virtual** , en una **dirección física** , apta para enviar al exterior por el bus de direcciones.

Generalidades

- En general todos los procesadores tienen una estructura de paginación que básicamente se encarga de traducir una **dirección virtual** , en una **dirección física** , apta para enviar al exterior por el bus de direcciones.
- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.

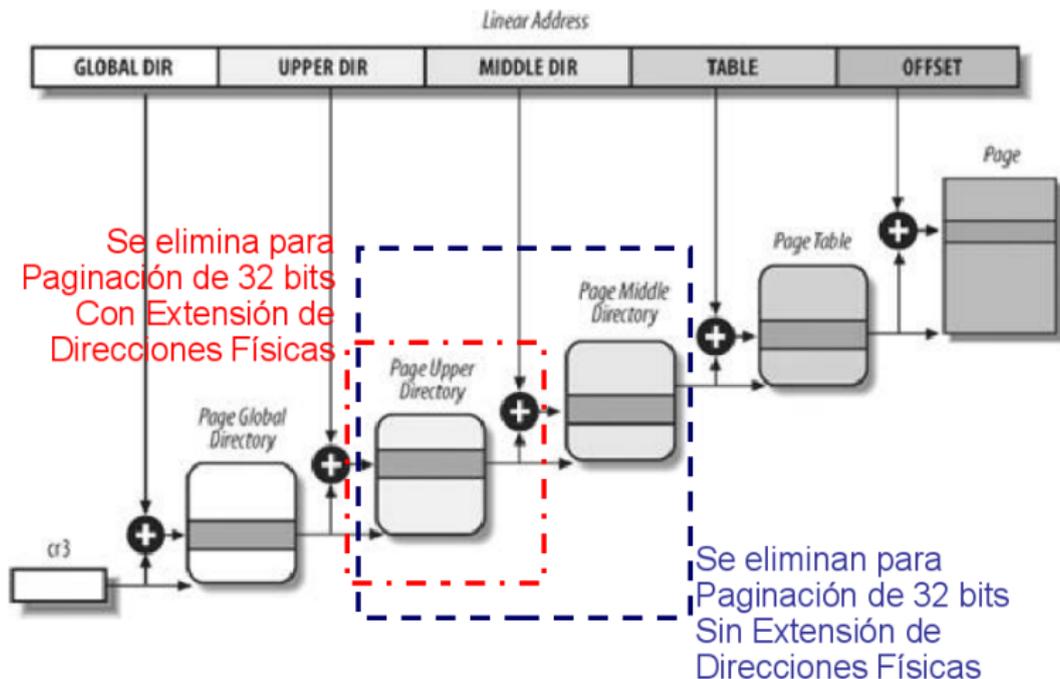
Generalidades

- En general todos los procesadores tienen una estructura de paginación que básicamente se encarga de traducir una ***dirección virtual*** , en una ***dirección física*** , apta para enviar al exterior por el bus de direcciones.
- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Habrá diferencias en la cantidad de bits de la ***dirección virtual*** , el tamaño de página, y otros detalles de implementación, en el mecanismo de traducción de cada arquitectura.

Generalidades

- En general todos los procesadores tienen una estructura de paginación que básicamente se encarga de traducir una ***dirección virtual*** , en una ***dirección física*** , apta para enviar al exterior por el bus de direcciones.
- A diferencia de la administración de memoria por segmentación, que es prácticamente una marca registrada de los Procesadores x86, la administración de memoria por paginación es común a todos los procesadores que implementan multitarea y lidian con algún mínimo esquema de protección de memoria entre tareas.
- Habrá diferencias en la cantidad de bits de la ***dirección virtual*** , el tamaño de página, y otros detalles de implementación, en el mecanismo de traducción de cada arquitectura.
- Para implementar una estructura suficientemente flexible de programación, Linux se basa en un modelo genérico.

Estructura de Paginación General de Linux



Paginación para diferentes procesadores en Linux

Arquitectura	PageSize	BitAddr	Niveles	Bits de cada campo
alpha	8 Kbytes	43	3	10 + 10 + 10 + 13
IA-64 (Itanium)	4 Kbytes	39	3	9 + 9 + 9 + 12
ppc64	4 Kbytes	41	3	10 + 10 + 9 + 12
sh64	4 Kbytes	41	3	10 + 10 + 9 + 12
Intel IA32	4 Kbytes	32	2	10 + 10 + 12
Intel IA32 PAE	4 Kbytes	32	3	2 + 9 + 9 + 12
Intel® 64	4 Kbytes	48	4	9 + 9 + 9 + 9 + 12
ARM CORTEX A	4 Kbytes	32	2	10 + 10 + 12

Así es que para cada arquitectura un conjunto de macros permiten definir la cantidad de bits de cada campo de la **dirección virtual** particionada en 4 niveles en la Figura anterior, pudiendo colapsar uno o hasta dos campos intermedios simplemente poniendo 0 como su valor a la macro que define la cantidad de bits de ancho de ese campo.

Lineamientos para Administración de la memoria

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

- 1 El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

- 1 El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas
- 2 El resto se denomina *memoria dinámica*, y consiste en un recurso fundamental, tanto para los procesos como para el mismo kernel.

Lineamientos para Administración de la memoria

Se trata de optimizar el uso de la memoria asignando a cada proceso y al mismo kernel solo la memoria necesaria, y liberándola ni bien haya finalizado su uso.

- 1 El kernel se asigna de modo permanente un área de memoria para su código y sus estructuras de datos estáticas
- 2 El resto se denomina *memoria dinámica*, y consiste en un recurso fundamental, tanto para los procesos como para el mismo kernel.
- 3 Como consecuencia de los dos ítems anteriores, la correcta administración de la *memoria dinámica* impacta directamente en la performance del sistema.

Lineamientos para Administración de la memoria

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan extensiones de paginación como PAE o Modo largo). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan extensiones de paginación como PAE o Modo largo). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

- El algoritmo de swaping es mas simple

Lineamientos para Administración de la memoria

Las distribuciones de Linux de 32 bits trabajan con páginas de 4 Kbytes solamente (aún las que implementan extensiones de paginación como PAE o Modo largo). Pesaron en la decisión los siguientes factores que se señalan como cruciales.

- El algoritmo de swaping es mas simple
- Si bien las transferencias de disco se hacen en múltiplos enteros de los tamaños de páginas de 4 Kbytes, lo cual permitiría emplear tamaños de páginas grandes, las transferencias de disco a memoria de menor tamaño son mas eficientes.

Detalle del Page Fault handler



Temario

- 1 **Introducción**
 - Enfoque preliminar
 - Organización y Administración
- 2 **Abstracción de memoria**
 - Espacio de abstracción
 - Gestión de la Memoria
- 3 **Administración de memoria**
 - La Memory Management Unit
 - Gestión del espacio libre
 - Políticas de Administración de memoria
- 4 **Memoria Virtual**
 - Antecedentes
 - Swapping
 - Organización en bloques
 - Políticas de Administración de memoria
- 5 **Relación con el hardware**
 - MMU
 - Page faults
- 6 **Paginación en un Sistema Operativo Real: Linux**
 - Lineamientos generales de administración de memoria
 - **Copy-on-write**

Bit W: Copy-On-Write -Caso práctico: LINUX

Bit W: Copy-On-Write -Caso práctico: LINUX

- Uno de los estándares mas empleados en el desarrollo de Sistemas Operativos modernos es POSIX

Bit W: Copy-On-Write -Caso práctico: LINUX

- Uno de los estándares mas empleados en el desarrollo de Sistemas Operativos modernos es POSIX
- Para crear un proceso se emplea la orden `fork ()`

Bit W: Copy-On-Write -Caso práctico: LINUX

- Uno de los estándares mas empleados en el desarrollo de Sistemas Operativos modernos es POSIX
- Para crear un proceso se emplea la orden `fork ()`
- Inicialmente el SO copiaba todos los recursos del proceso padre al proceso hijo

Bit W: Copy-On-Write -Caso práctico: LINUX

- Uno de los estándares mas empleados en el desarrollo de Sistemas Operativos modernos es POSIX
- Para crear un proceso se emplea la orden `fork ()`
- Inicialmente el SO copiaba todos los recursos del proceso padre al proceso hijo
- A partir del soporte de hardware de los Microprocesadores modernos se implementa PAGE COW

Bit W: Copy-On-Write -Caso práctico: LINUX

Bit W: Copy-On-Write -Caso práctico: LINUX

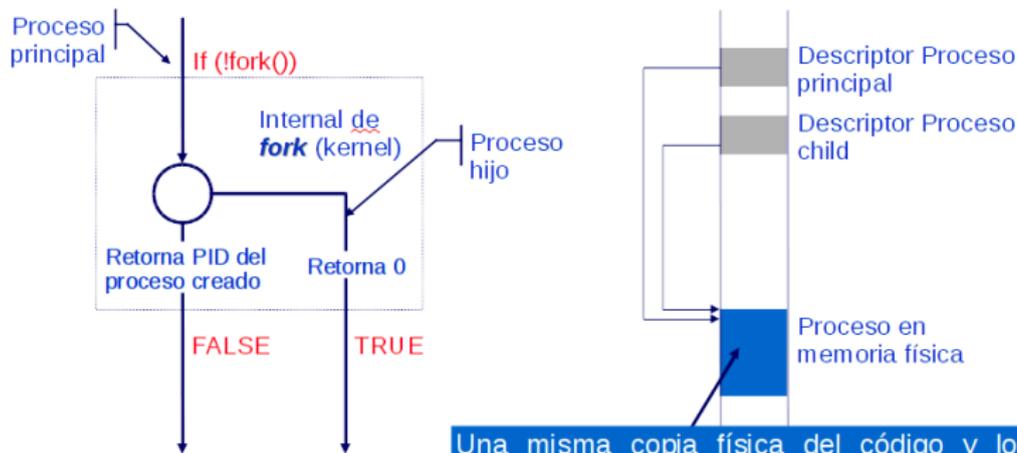
- Al crear el proceso hijo no se copia ninguna página.

Bit W: Copy-On-Write -Caso práctico: LINUX

- Al crear el proceso hijo no se copia ninguna página.
- En lugar de ello se duplica el espacio de direccionamiento del BCP del proceso padre en el BSP del proceso hijo.

Bit W: Copy-On-Write -Caso práctico: LINUX

- Al crear el proceso hijo no se copia ninguna página.
- En lugar de ello se duplica el espacio de direccionamiento del BCP del proceso padre en el BSP del proceso hijo.



Una misma copia física del código y los datos en memoria apuntada por dos descriptores de proceso diferentes.

Lightweight Process: Una copia = Dos procesos.