



Instituto Politécnico Nacional
Escuela Superior de Cómputo



TAREA 8

Sistemas Operativos

Integrantes:

Mora Ayala José Antonio
Ramírez Cotonieto Luis Fernando
Torres Carrillo Josehf Miguel Ángel
Tovar Jacuinde Rodrigo

Profesor:

Cortés Galicia Jorge
2CM17

Memoria Principal

Memoria

Está compuesta de una gran matriz de palabras o bytes, cada uno con su propia dirección.

Ciclo de Ejecución

Extraer una instrucción de la memoria
Decodificación
Extraer de memoria una serie de Operandos
Almacenamiento de Resultados

La unidad de memoria tan sólo ve un flujo de direcciones de memoria y no sabe cómo se generan esas direcciones (mediante el contador de programa, mediante indexación, indirección, direcciones literales, etc) ni tampoco para qué se utilizan (instrucciones o datos)

Hardware básico

La memoria principal y los registros integrados dentro del propio procesador son las únicas áreas de almacenamiento a las que la CPU puede acceder directamente.

Instrucción de Máquina

Toman como argumentos direcciones de memoria

¿Dirección de Disco?

No existe ninguna instrucción acepte direcciones de disco

Por lo que...

Todas las instrucciones en ejecución y los datos utilizados por esas instrucciones deberán encontrarse almacenados en uno de esos dispositivos de almacenamiento de acceso directo. Si los datos no se encuentran en memoria, deberán llevarse hasta antes de que la CPU pueda operar con ellos.

Acceso mediante Ciclo de Reloj

Puede accederse a los registros integrados en la CPU en un único ciclo del reloj del procesador. La mayoría de los procesadores pueden decodificar instrucciones y realizar operaciones simples con el contenido de los registros a la velocidad de una o más operaciones cada tic de reloj.

Memoria Principal

Se accede mediante una transacción del bus de memoria. El acceso a memoria puede requerir muchos ciclos de reloj del procesador para poderse completar, en cuyo caso el procesador necesitará normalmente detenerse, ya que no dispondrá de los datos requeridos para completar la instrucción que esté ejecutando.

Proporciona

Protección

No sólo debe preocuparnos la velocidad relativa del acceso a la memoria física, sino que también debemos garantizar una correcta operación que proteja al sistema operativo de los posibles accesos por parte de los procesos de los usuarios y que también proteja a unos procesos de usuario de otros.

Metodo 1

Asegurarnos de que cada proceso disponga de un espacio de memoria separado.

Debemos poder determinar el rango de direcciones legales a las que el proceso pueda acceder y garantizar también que el proceso sólo acceda a esas direcciones legales

Podemos proporcionar esta protección utilizando dos registros, usualmente una base y un límite

El registro base almacena la dirección de memoria física legal más pequeña, mientras que el registro límite especifica el tamaño del rango. Por ejemplo, si el registro base contiene el valor 300040 y el registro límite es 120900, entonces el programa podrá acceder legalmente a todas las direcciones comprendidas entre 300040 y 420940 (incluyendo los dos extremos).

Protección del Espacio de Memoria

Se consigue haciendo que el hardware la CPU compare todas las direcciones generadas en modo usuario con el contenido de esos registros.

Cualquier intento, por parte de un programa que se esté ejecutando en modo usuario, de acceder a la memoria del sistema operativo o a la memoria de otros usuarios hará que se produzca una interrupción. hacia el sistema operativo

Cualquier intento, por parte de un programa que se esté ejecutando en modo usuario, de acceder a la memoria del sistema operativo o a la memoria de otros usuarios hará que se produzca una interrupción. hacia el sistema operativo

Beneficios

Este esquema evita que un programa de usuario modifique (accidental o deliberadamente) el código y las estructuras de datos del sistema operativo o de otros usuarios.

Reasignación de Direcciones

Usualmente, los programas residen en un disco de memoria de archivos ejecutables binarios. Para poder ejecutarse, un programa deberá ser cargado en memoria y colocado dentro de un proceso.

Dependiendo del mecanismo de gestión de memoria que se utilice, el proceso puede desplazarse entre disco y memoria durante su ejecución

Procedimiento Normal

Seleccionar uno de los procesos de la cola de entrada y cargar dicho proceso en memoria.
A medida que se ejecuta el proceso, éste accede a las instrucciones y datos contenidos en la memoria.
Eventualmente, el proceso terminará su ejecución y su espacio de memoria será declarado como disponible

Reasignación de las instrucciones y los datos a direcciones de memoria

Pueden realizarse en cualquiera de los pasos

Cola de Entrada

Los procesos del disco que estén esperando a ser cargados en memoria para su ejecución forman lo que se denomina cola de entrada.

Tiempo de compilación.

Si sabemos en el momento de realizar la compilación dónde va residir el proceso en memoria, podremos generar código absoluto.

Por ejemplo, si sabemos que un proceso de usuario va a residir en una zona de memoria que comienza en la ubicación "R", el código generado por el compilador comenzará en dicha ubicación y se extenderá a partir de ahí. Si la ubicación inicial cambiase en algún instante posterior, entonces será necesario recompilar ese código. Los programas en formato .COM de MS-DOS se acoplan en tiempo de compilación.

Tiempo de carga.

Si no conocemos en tiempo de compilación dónde va a residir el proceso en memoria, el compilador deberá generar código reubicable.

Se retarda la reasignación final hasta el momento de la carga.

Si cambia la dirección inicial, tan sólo es necesario volver cargar el código de usuario para incorporar el valor modificado.

Tiempo de ejecución.

El proceso puede desplazarse durante su ejecución desde un segmento de memoria a otro

Es necesario retardar la reasignación hasta el instante de la ejecución.

Espacios de direcciones lógico y físico

Dirección Lógica

Dirección generada por la CPU

Dirección Física

Dirección vista por la unidad de memoria (es decir, la que se carga en el registro de direcciones de memoria de la memoria)

Espacio de Dirección Lógica

El conjunto de todas las direcciones lógicas generadas por un programa

Espacio de Dirección Física

El conjunto de todas las direcciones físicas correspondientes a estas direcciones lógicas

Metodo de reasignación en tiempo de Compilación y Carga

Generan direcciones lógicas y físicas idénticas.

Esquema de reasignación de Direcciones

En tiempo de ejecución hace que las direcciones lógica y física difieran

Los espacios de direcciones lógicas y físicas difieren.

Unidad de Gestión de Memoria

La correspondencia entre direcciones virtuales y físicas en tiempo de ejecución es establecida por un dispositivo hardware

Carga dinámica

Una rutina no se carga hasta que se la invoca; todas las rutinas se mantienen en disco en un formato de carga reubicable. Según este método, el programa principal se carga en la memoria y se ejecuta.

Cuando una rutina necesita llamar a otra rutina, la rutina que realiza la invocación comprueba primero si la otra ya ha sido cargada, si no es así, se invoca el cargador de montaje reubicable para que cargue en memoria la rutina deseada y para que actualice las tablas de direcciones del programa con el fin de reflejar este cambio.

Después, se pasa el control a la rutina recién cargada.

Ventaja

Una rutina no utilizada no se cargará nunca en memoria.

Util cuando

Se necesitan grandes cantidades de código para gestionar casos que sólo ocurren de manera infrecuente, como por ejemplo rutinas de error.

Requiere de

El mecanismo de carga dinámica no requiere de ningún soporte especial por parte del sistema operativo. Es responsabilidad de los usuarios diseñar sus programas para poder aprovechar dicho método. Sin embargo, los sistemas operativos pueden ayudar al programador proporcionándole rutinas de biblioteca que implementen el mecanismo de carga dinámica.

Montaje dinámico y bibliotecas compartidas

Con el montaje dinámico, se incluye un stub dentro de la imagen binaria para cada referencia a una rutina de biblioteca.

stub

Pequeño fragmento de código que indica cómo localizar la rutina adecuada de biblioteca residente en memoria o cómo cargar la biblioteca si esa rutina no está todavía presente.

Ejecución

Comprueba si la rutina necesaria ya se encuentra en memoria; si no es así, el programa carga en memoria la rutina.

En cualquiera de los casos, el stub se sustituye así mismo por la dirección de la rutina y ejecuta la rutina. Así, la siguiente vez que se ejecute ese segmento de código concreto, se ejecutará directamente la rutina de biblioteca, sin tener que realizar de nuevo el montaje dinámico.

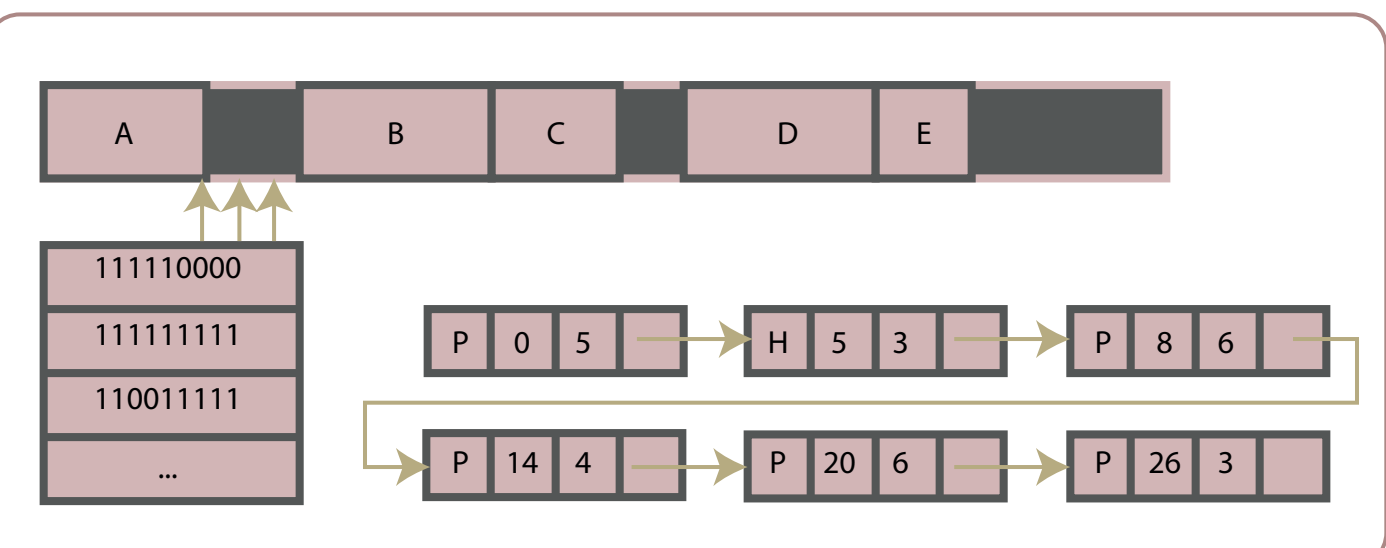
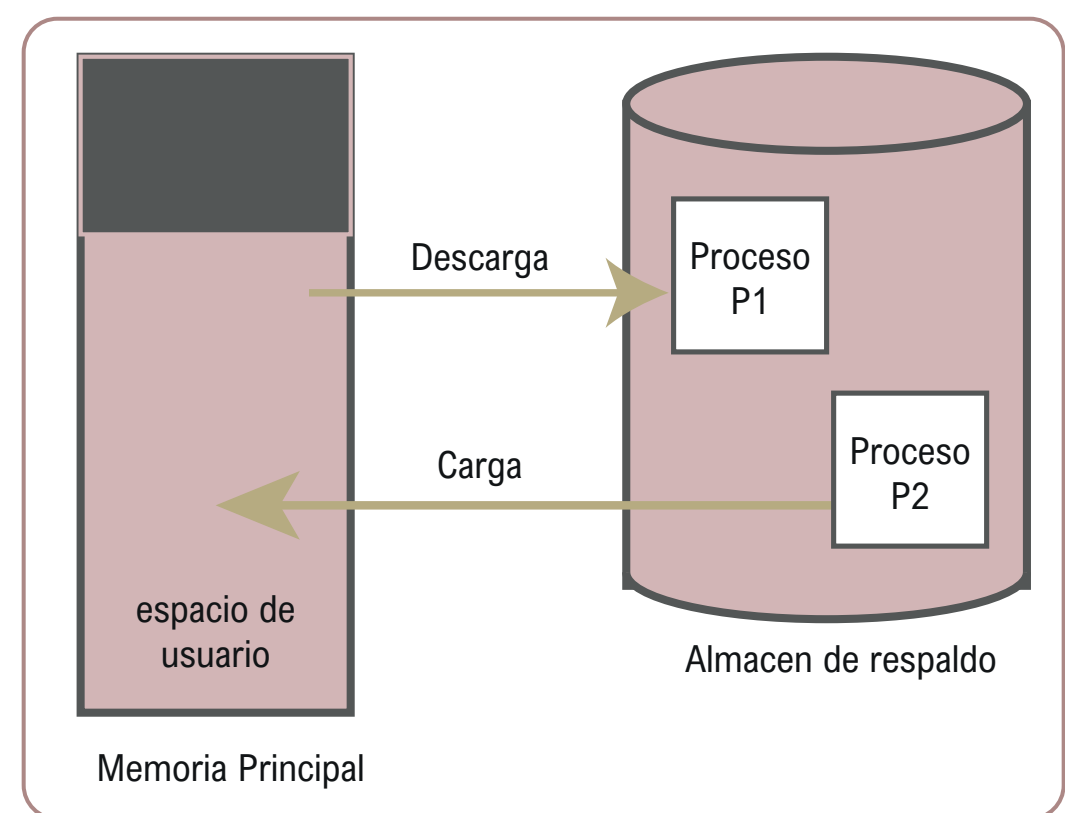
Sustitución de Biblioteca

Puede sustituirse una biblioteca por una nueva versión y todos los programas que hagan referencia a la biblioteca emplearán automáticamente la versión más reciente. Sin el mecanismo de montaje dinámico, sería necesario volver a montar todos esos programas para poder acceder a la nueva biblioteca.

Intercambio

Un proceso debe estar en memoria para ser ejecutado. Sin embargo, los procesos pueden ser intercambiados temporalmente, sacándolos de la memoria y almacenándolos en un almacén de respaldo y volviéndolos a llevar luego a memoria para continuar su ejecución. Por ejemplo, suponga que estamos utilizando un entorno de multiprogramación con un algoritmo de planificación de CPU basado en turnos.

Cada vez que un proceso termine su ejecución o sea asignado, se intercambiará por otro proceso. Idealmente, el gestor de memoria puede intercambiar los procesos con la suficiente rapidez como para que haya siempre algunos procesos en memoria, listos para ejecutarse



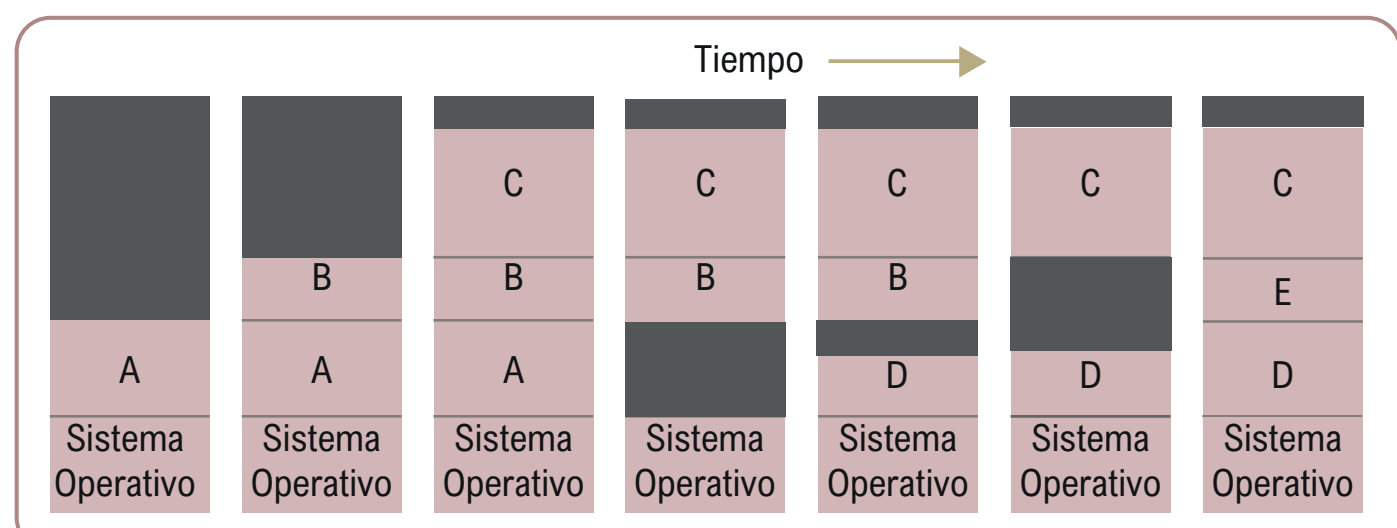
Suponiendo una latencia media de 8 milisegundos el tiempo de intercambio será de 256 milisegundos puesto que el intercambio requiere tanto una operación de carga como otra de descarga, el tiempo total necesario será de unos 516 milisegundo

Ejemplo

Para un uso eficiente de la CPU es necesario que el tiempo de ejecución de cada proceso sea largo en relación el tiempo de intercambio, así en un algoritmo de planificación de la CPU por turnos

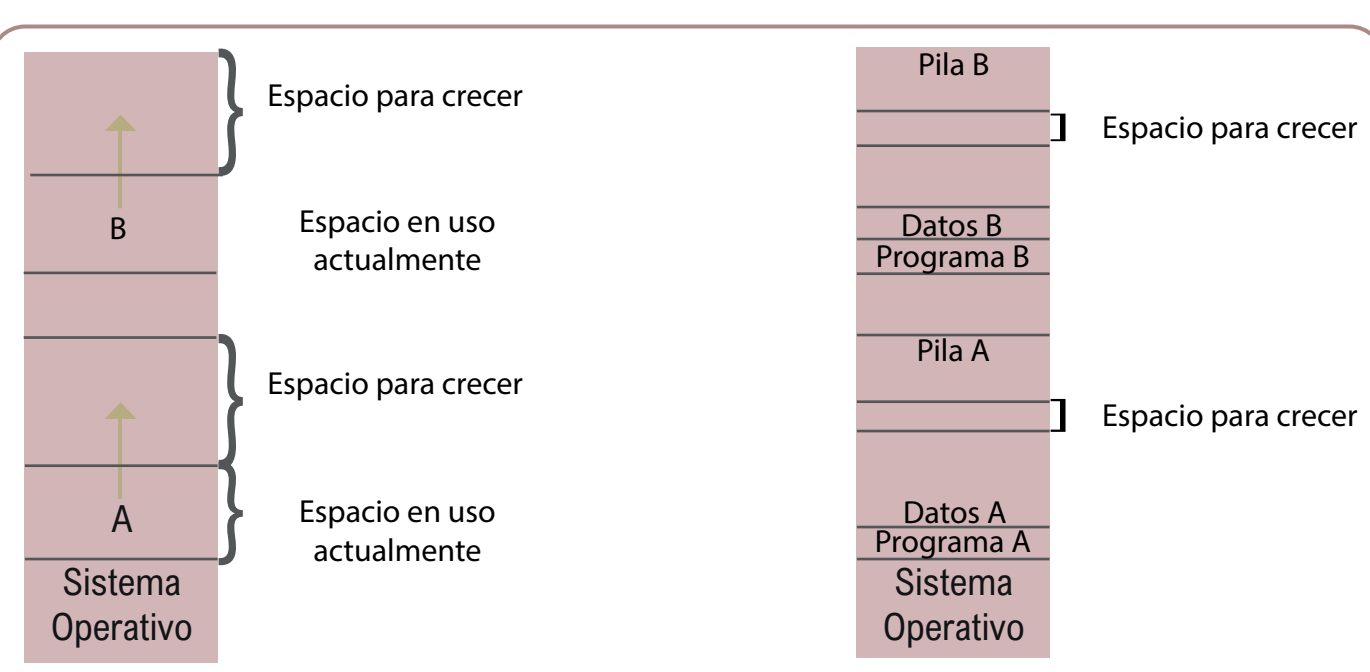
El proceso del usuario (mb)

$$\text{Tiempo (s)} = \frac{\text{Velocidad de transferencia del disco}}{\text{Velocidad de transferencia del disco}} \times \left(\frac{\text{Tiempo de intercambio}}{\text{Tiempo de ejecución}} \right)$$



Cualquier proceso que se hubiera descargado permanecia descargado hasta que el usuario lo volviera a seleccionar para ejecución

Las siguientes versiones aprovechaban las características avanzadas de MMU que ahora incorporan las PC



Antecedentes

Las primeras PC sin tantos métodos de gestión de memoria ejecutaban múltiples procesos de gran tamaño utilizando una versión modificada del mecanismo de intercambio

Una vez que la carga del sistema se reduce vuelve a desactivarse el mecanismo de intercambio

En UNIX se utiliza una variante del mecanismo de intercambio, normalmente se encuentra desactivado, pero se activa si se están ejecutando numerosos procesos y si la cantidad de memoria utilizada excede un cierto umbral

Lo que se puede encontrar en la mayoría de sistemas son versiones modificadas de este mecanismo de intercambio

Requiere un tiempo de intercambio muy alto y proporciona un tiempo de ejecución demasiado pequeño como para constituir una solución razonable de gestión de memoria

Estos mecanismos estándar de intercambio se utilizan en muy pocos sistemas

La memoria es un único bloque disponible para los procesos de usuario denominado

Cuando un proceso necesita memoria se busca un agujero lo suficientemente grande como para albergar este proceso asignándole solo la memoria necesaria, manteniendo el resto disponible

El OS toma en cuenta los requisitos de memoria en cada proceso y la cantidad de memoria disponible

El OS puede ordenar esta cola de entrada con algún algoritmo de planificación asignando memoria a los procesos hasta que finalmente los requisitos de memoria del siguiente proceso ya no queden bloques de memoria disponible

El OS puede esperar hasta que haya libre un bloque de memoria o puede examinar el resto de la cola de entrada para ver si pueden satisfacerse los requisitos de memoria de algún otro proceso

En cualquier momento determinado tendremos un conjunto de agujeros de diversos tamaños dispersos por la memoria

Si el nuevo agujero es adyacente a otro estos se combinan para formar otros de mayor tamaño reemplando de nuevo el ciclo

Funcionamiento

El intercambio es una técnica utilizada por los OS para extender la memoria donde los procesos pueden ser intercambiados, sacándolos de la memoria y dejándolos en un respaldo y regresándolos de nuevo a la memoria para continuar su ejecución

En un entorno de multiprogramación con un algoritmo basado en turnos, cuando termina un cuanto de tiempo, el gestor de memoria saca el proceso que acaba de terminar y carga otro proceso en ese espacio de memoria

El planificador de CPU asigna un cuanto de tiempo a algún otro proceso que ya se encuentre en memoria

Cada vez que un proceso termine su cuanto, asignado, se intercambiara por otro proceso

Si se utiliza reasignación en tiempo de ejecución si se puede mover el proceso a otro espacio de memoria

Si la reasignación se realiza en tiempo de ensamblado o en tiempo de carga entonces no es sencillo mover el proceso a una ubicación diferente

Un proceso descargado se volverá a cargar en el mismo espacio de memoria que ocupaba anteriormente

Los mecanismos de intercambio necesitan un almacén de respaldo que será un disco rápido y con suficiente espacio para albergar copias de todas las imágenes de memoria para todos los usuarios proporcionando un acceso directo a estas imágenes

Cada vez que el planificador de la CPU decide ejecutar un proceso llama al despachador que mira si el siguiente proceso de la cola se encuentra en memoria

El tiempo necesario de cambio de contexto en uno de estos sistemas de intercambio es relativamente alto, y se calcula con la siguiente fórmula

Si no hay memoria libre, el despachador intercambia el proceso deseado por otro proceso que este en memoria, después recarga los registros y transfiere el control del proceso seleccionado

Si un proceso de mayor prioridad desea ser servido, el gestor de memoria puede descargar el proceso de menor prioridad y cargar y ejecutar el de mayor prioridad

Los algoritmos de planificación con prioridad se utiliza una variante de esta política de intercambio

Otra estrategia, llamada memoria virtual, permite a los programas ejecutarse, aunque sólo estén parcialmente en la memoria principal

Suponemos que el administrador de memoria sabe cuanto memoria debe asignar

Si los procesos y agujeros se mantienen en una lista ordenada por dirección, se pueden usar varios algoritmos para asignar memoria a un proceso recién creado o trado a la memoria

Administración de memoria con listas enlazadas

Otra forma de contabilizar la memoria es mantener una lista enlazada de segmentos de memoria libres y asignados, donde un segmento es un proceso o bien un agujero entre dos procesos

El tamaño de la unidad de asignación es una cuestión de diseño importante, cuanto menor sea la unidad de asignación, mayor será el mapa de bits

Con una unidad de asignación de cuatro bits, 32 bits de memoria sólo requerirán un bit del mapa. Una memoria de 32n bits usará n bits de mapa, y el mapa sólo ocupará 1/32 de la memoria

El problema puede ser grave, podríamos tener un bloque de memoria libre entre cada dos procesos, si estos fragmentos de memoria estuviesen en un único bloque libre podríamos ser capaces de ejecutar varios procesos mas

La estrategia de primer ajuste y menor ajuste sufren de fragmentación externa, mientras se cargan los procesos en memoria y se eliminan, el espacio de memoria libre se descompone en una serie de fragmentos de pequeño tamaño

El problema es cuando hay un espacio de memoria, pero esos espacios disponibles no son contiguos, el espacio está fragmentado muchos agujeros pequeños

Dependiendo del mecanismo que usemos puede afectar el grado de fragmentación

Dependiendo de la cantidad total de espacio de memoria y del tamaño medio de los procesos esa fragmentación puede ser un problema grave o no

Este procedimiento constituye un caso concreto del problema general de asignación dinámica de espacio de almacenamiento, que se ocupa de como satisfacer una solicitud de tamaño n a partir de una lista de agujeros libres

Hay varias formas de seleccionar un agujero libre disponible mediante el primer ajuste, mejor ajuste y peor ajuste

Se asigna el primer agujero que sea lo suficientemente grande

La exploración puede comenzar desde el principio del conjunto de agujeros o en el punto en que hubiera terminado la exploración anterior

podemos detener la exploración en cuanto encontremos un agujero libre que sea lo suficientemente grande

Se asigna el agujero más pequeño que tenga el tamaño suficiente exacto

Debemos explorar la lista completa, hace que se genere un agujero más pequeño posible con la memoria que sobre en el agujero original

Debemos explorar la lista completa, genera un agujero mas grande posible con la memoria sobrante del agujero original lo que puede resultar más útil que el agujero más pequeño generado con la técnica de mejor ajuste

Las simulaciones muestran que las primeras 2 son mejores que la última en términos del tiempo necesario y de la utilización del espacio de almacenamiento, la de primer ajuste es la más rápida de implementar

Esta función resulta útil por que los OS contiene código y espacio en bufer para los controladores de dispositivo, si uno de estos no se utiliza, tanto su código como datos no son necesarios en la memoria denominada código transitorio

Es una forma efectiva de permitir que el tamaño del OS cambie dinámicamente

Todas las direcciones generadas por la CPU se comparan con estos registros, permitiendo proteger al OS como a los programas y datos de otros usuarios de las modificaciones que pudiera realizar este proceso en ejecución

Con estos registros cada dirección lógica debe ser inferior al valor contenido en registro limite

La MMU convertirá la dirección lógica dinámicamente sumándole el valor contenido en el registro de reubicación, la cual es la que se envía a la memoria

Utilizando un registro de reubicación, podemos contener el valor de la dirección física más pequeña mientras que el registro limite contiene el rango de las direcciones lógicas

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los procesos de usuario

El OS se sitúa tanto en la zona baja o alta de la memoria, debido a la ubicación del vector de interrupciones, ya que este se encuentra e menudo en la parte baja de la memoria

En este esquema cada proceso este contenido en una única sección contigua de memoria

La memoria debe albergar tanto el OS como los diversos procesos de usuario, se necesita asignar las distintas partes de la memoria principal de forma eficiente

Si tenemos un sistema informático con 512 MB de memoria principal y un OS que ocupa 25mb, el tamaño máximo de un proceso de usuario será de 487 MB

Pero muchos procesos de usuario pueden ser mucho más pequeños, como por ejemplo de 10MB

Un proceso de 10 mb podría intercambiarse en 256 milisegundos, comparado con los 6.4 segundos requeridos para intercambiar 256 MB, resultaría útil conocer cuanto memoria está usando un proceso de usuario

La memoria se divide en 2 particiones, una para el OS residente y otra para los