Examen Soper

Lógica = virtual

Tamaño página = X

Tamaño Max segmento = Z

Numero de segmentos = Y

Numero de paginas = G

Memoria virtual:

* Bits direccionar el desplazamiento dentro de cada página(offset) = a -> X = 2a
* Bits para la página = b -> Z/X = F = 2b , G = 2b (numero de pagina)
* Bits para el segmento = c -> Y <= 2c
* Total = 2a+b+c

Sin segmento:

* Bits direccionar el desplazamiento dentro de cada página(offset) = a -> X = 2a
* Bits para la página = b -> Z/X = F = 2b
* Total = 2a+b

Memoria Física = R:

* Bits para el marco = d -> R/X = 2d
* Total = 2d+a

Conjunto Residente:

* Suponemos que ocupa toda la memoria física -> Total = 2d+a

Expresado en palabras:

Un tamaño de página de 1024 palabras y una memoria física formada por 1024 marcos

Un máximo de 16 segmentos

Memoria virtual:

* Punto de vista del programador: 4 bits para los segmentos y 20 bits para el offset del segmento.
* Punto de vista del sistema: 4 bits para los segmentos + 10 bits para las páginas dentro de cada segmento + 10 bits para el offset dentro de la página.
* 224 palabras

Memoria física:

* La dirección física está formada por 20 bits, 10 bits para el número de marcos (1024 marcos) y 10 bits para el desplazamiento dentro del marco. tamaño de la memoria física: 220 palabras.

Nos dan tabla: 0x7ECB 0x7CED 0x7D31 0x15B2 0x7A86 0x7823 0x7B01 0x3FA2 0x5C56

¿Cuántos marcos de memoria física tiene asignado el proceso en este momento?

Todos los que tengan el bit de validez activado de los datos de la tabla.

Una captura de pantalla de un celular

Descripción generada automáticamenteTabla

Descripción generada automáticamente

Algoritmos de Remplazo:

* Tabla

  Descripción generada automáticamente con confianza mediaOptimo:
* LRU: Selecciona el marco en el que está la página que lleva más tiempo sin acceso
* Tabla

  Descripción generada automáticamenteFifo:
* NRU: Utiliza los bits de Uso (U) y Modificado (M) de las entradas de la tabla de páginas. Se definen cuatro clases de páginas. El algoritmo NRU elimina una página de manera aleatoria de la primera clase con el número más bajo
  + Clase 0: Páginas no referenciadas recientemente ni modificadas (U=0,M=0).
  + Clase 1: Páginas no referenciadas recientemente, modificadas (U=0,M=1).
  + Clase 2: Páginas referenciadas recientemente, no modificadas (U=1,M=0).
  + Clase 3: Páginas referenciadas recientemente, modificadas (U=1,M=1).

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

* Algoritmo del Reloj Mejorado:
  + Algoritmo de selección de página a reemplazar: Buscar entre las de clase 0 (U=0, M=0)
  + Si no se encuentra ninguna (el puntero ha dado la vuelta entera), buscar de clase 1 (U=0, M=1) y durante la búsqueda cambiar los U=1 que vaya encontrando por U=0.
  + Si no se encuentra ninguna, buscar de nuevo de clase 0.
  + Si no encuentra ninguna, buscar de nuevo de clase 1 (alguna encontrara, pues todos los bits de uso se han puesto a 0 en la 2a vuelta).

Ejemplo LRU: direcciones lógicas como las físicas están formadas por 16 bits de los cuales 10 bits corresponden al desplazamiento (offset)

* Lógica = 6 bits página y 10 bits offset
* Tabla

  Descripción generada automáticamenteFísica = 6 bits marco y 10 bits offset

Dada la memoria según la tabla 1 del anexo calcula los valores solicitados a continuación. Condiciones:

* El acceso a la memoria se hace en el formato big endian.
* La tabla de páginas del proceso se encuentra en el marco A.
* El sistema de direccionamiento es de un nivel con páginas de B bytes.
* La memoria física máxima direccionable es de C bytes.
* Cada tabla de páginas tiene un tamaño de una página.
* El bit más alto indica la presencia o ausencia de la página. (1-presente, 0-ausente)
* El bit contiguo al de presencia por debajo es el del permiso de lectura.(1-legible, 0-no legible)
* El bit contiguo al de lectura por debajo es el del permiso de escritura.(1-escribible, 0-no escribible)
* Los bits indicando los marcos de página siempre están en la parte menos significativa de los elementos de las tablas.

Número de bits para direccionar un byte en una página (offset) = x -> A = 2X

Numero de bits para direccionar todos los marcos de memoria = z -> C/B = 2z

Número de bits para direccionar un elemento de una tabla de páginas = s -> Z + bits de control = T bits =< Y bytes -> A/Y = 2s

Tamaño del puntero virtual en bits = x + s (nivel1), x + s + s (nivel2)

Tamaño del puntero físico en bits = x + z

Tamaño máximo de la memoria virtual en bytes (B, kB, MB ...) = 2x+s

Búsqueda ejemplo:



S = 4 = bits página = 0

X= 5 = bits offset = 10010 = 18

Busco en la tabla que me pidan en el marco A, la pagina 0(2bytes) ->

E0 1A = 1110 0000 0001 1010, tiene los 3 bits activados, marco = z = 6 bits -> 01 1010 = 24

Buscamos en el marco 24 desplazamiento 18(1byte) -> FB

Si nos pide lectura y bit de lectura desactivado -> Fallo de protección

Si nos pide escritura y bit de escritura desactivado -> Fallo de protección

Si el bit de presencia está a 0 -> Fallo de pagina

¿Qué es la TLB? ¿Cuál es su función? Solución: Buffer de traducción adelantada. Es una memoria caché asociativa que guarda los marcos de página donde están almacenadas las páginas más recientemente referenciadas. Su función es la de reducir el tiempo de búsqueda de la dirección real a partir de una dirección virtual, ahorrando el acceso a la tabla de páginas

TLB, Ejemplo:

Número de páginas = 13 bits

Tamaño de páginas = 15 bits

Numero de marcos = 9 bits

Los tres bits más significativos de cada entrada de la tabla de páginas se refieren a bits de control, de los cuales los dos más significativos se corresponden al bit de presencia y al bit de modificación, respectivamente.

Tabla

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteEl tiempo de acceso a disco es de 120 ms, a memoria de 50 ns y a la TLB de 10 ns

Archivos:

Bloques de 4 kB

Punteros directos y un puntero indirecto, un indirecto doble y uno triple. Cada puntero ocupa 4 bytes

Parte informativa del inodo ocupa 36 bytes

**Calcula el tamaño máximo del disco según esta estructura**:

Como los punteros a los bloques son de 4 bytes = 32 bits entonces puedo direccionar 232 bloques. Como cada bloque es de 4 kB (212 B) entonces el tamaño máximo direccionable de un disco con este sistema de archivos es de 232 × 212 B, es decir 16 TB = 244.

**Calcula el tamaño máximo de un fichero:**

4096(Tam bloque) – 36(parte informativa) = 4060 B.

Tamaños punteros = 4bytes-> 4060/4= 1015 punteros, de los cuales 1012 son directos. En cada bloque de punteros caben 4 kB / 4 bytes punteros, es decir, 1024 punteros:

* **Punteros directos:** 1012 punteros × 4 kb. Es decir, el tamaño máximo de un fichero con punteros directos es de 4145152 B.
* **Puntero indirecto simple:** es un puntero que apunta a un bloque de punteros que apuntan a bloques de datos -> 10242 x 4 kB = 4GB. Es decir, el fichero podría ocupar con el indirecto simple lo que ocupa con directos más 4 MB.
* **Puntero indirecto doble:** es un puntero que apunta a un bloque de punteros indirectos simples -> 10243 x 4 kB = 4TB Por tanto un fichero podría ocupar lo mismo que en el caso anterior más 4 GB. ´
* **Puntero indirecto triple**: es un puntero que apunta a un bloque de punteros indirectos dobles-> 10244 x 4 kB = 4TB x 4Kb. Por tanto, un fichero podría ocupar lo mismo que en el caso anterior más 4 TB.
* **Tamaño máximo del fichero** = 4 TB + 4 GB + 7,9 MB > 4 TB.

Ejemplo raro:

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Bytes i-nodo = 4Kb / 32 = 18

Atributos i-nodo = sumar todo menos los punteros

Punteros directos = 128 / 27 – 4punteros = 21,…

Texto

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media