Memòria Virtual i Traducció d’Adreçes

1ra Part

1. Funcionalitats gràcies al sistema de traducció d’adreces
   1. Aïllament de processos (protecció)
   2. Auto-aïllament: les diferents part del codi estan protegides entre sí
   3. Execució d’un procés: un procés es pot executar sense tenir tot el codi a memòria RAM
   4. Comunicació interprocés: una zona de memòria compartida entre processos
   5. Codi executable compartit (llibreries)
   6. Gestió de memòria de la pila o mem. Dinàmica
   7. Fitxer mapats a memòria: accedir a un fitxer com si fos un vector amb posicions de memòria
2. Com s’implementa el sistema de traducció d’adreces
   1. La majoria: maquinari especialitzat controlat per l’SO
   2. Interprèt btye-code
   3. A nivell de maquinari i a nivell de programari
3. Concepte de traducció d’adreça
   1. Totes les direccions de memòria generades pels processos són traduïdes a una adreça de memòria física
   2. Respecte la traducció:
      1. Un procés genera direccions de memòria virtual: cal traduir-la a adreça física
      2. Abans, es comprova si és vàlida
      3. Maquinari: MMU (Memory Mapping Unit), controlat per l’SO
4. Traducció d’Adreça
   1. Base i Límit
      1. Dos registres, només poden ser modificats per instruccions privilegiades
      2. Cada procés té els seus propis
      3. Cada vegada que s’accedeix a memòria, es suma la base a l’adreça. Si supera el límit, excepció.
      4. Defectes:
         1. Previsió de memòria màxima que ocuparà el procés
         2. Espai continuu, lineal de memòria, no es pot fragmentar
         3. Perill de sobre-escriure la seva zona de codi executable
         4. Impossible compartir memòria
      5. Segmentació: tenir múltiples registres base-limit
         1. Cada un té una porció de l’espai d’adreces (segment) de mida variable
         2. Cada adreça virtual té:
            1. Número de segment (osea base i límit associat)
            2. Offset del segment
            3. Amb el segment s’entra ala taula de segments, s’agafa el límit, es mira si es més alt que l’offset i sinó s’entra a mem.
         3. Avantatges:
            1. Auto-aïllament
            2. Fragmentació de memòria
            3. Compartició de segments (compartició de codi)
            4. Comunicació interprocés
         4. Desavantatges:
            1. Memòria dividida en espais lliures i ocupats
            2. Cap zona de memòria lliure prou gran per crear un segment, però si es sumen si que n’hi ha
            3. Compactar significa canviar els registres
      6. Memòria paginada
         1. Marcs de pàgina: blocs de mida fixa de memòria i de potència de dos (Bits alts)
         2. Traducció mitjançant una taula (cada procés en té una)
         3. Es realitza a nivell de maquinàri
         4. La taula no només té la traducció al marc físic, sinó també propietats de lectura, escriptura, etc
         5. SO s’encarrega de les taules dels processos
         6. Per exemple:
            1. Esquema de 16 bits (Addr. Física)
            2. Marcs de 4Kbytes = 4096 bytes

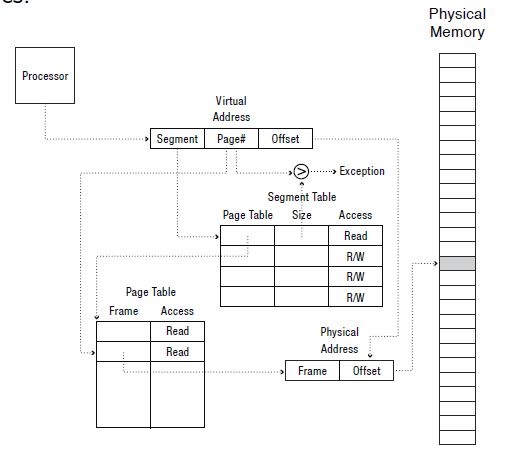
216 / 4096 = 16 marcs de pàgina

16 = 24; 4 bits fan referència a un marc

El 5è bit representa si està o no el marc a la mem. Física

2na Part

1. Memòria Paginada
   1. Traducció d’adreces i Fork
      1. El SO duplica la taula del procés, però no la info associada
      2. Copy – On – Write: les pàgines es marcquen com a lectura i es copien quan hi accedim per escriptura.
   2. Traducció d’adreces i Malloc
      1. Només es reserva l’espai virtual, no el físic.
      2. El físic s’assigna quan accedim a una pos. Del vector
   3. Zones de memòria lliures amb Memòria paginada:
      1. Problema solucionat. Només s’ha de trobar una pàgina lliure
      2. Si està plena: SO descarta
      3. Exec: SO inicialitza el mapa de memòria física, no carrega tot a memòria
   4. Esquemes d’Arbre (traducció multinivell)
      1. Com arribar a les fulles del arbre
      2. Paginació Segmentada:
         1. L’arbre té dos nivells, el primer és el segment que apunta a una taula, el segon, el que apunta als marcs de pàgina físics



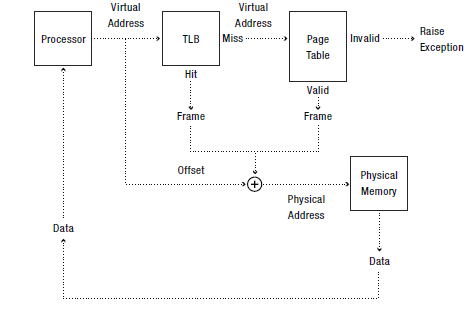
* + - 1. Paginació Multinivell
         1. L’arbre té múltiples nivells. La teva adreça virtual té diferents index que et diuen la pos. Dels diferents nivells.
      2. Paginació multinivell segmentada (juntar els dos)
         1. Cada segment té una taula multinivell
         2. Millor l’eficiència de la traducció sense perdre versatilitat:

Memòria Cache (TLB): una taula separada on es guarden les pàgines accedides més habitualment.

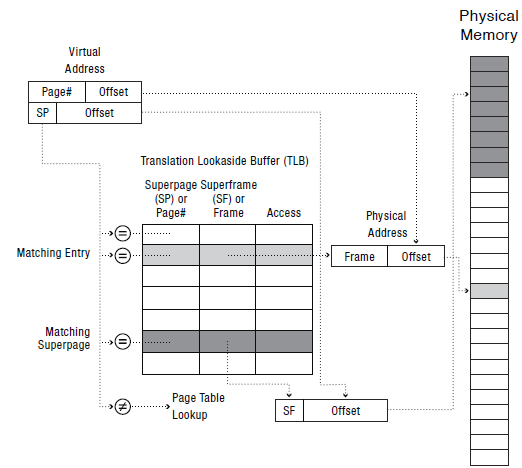
Coordinada per l’SO

Al fer un canvi de context s’invalida

Es comprova si hi ha hit, sinó -> multinivell



Encara es pot millorar mes: superpàgines, és a dir, pàgines contigües a mem. Física.



Encara més: utilitzar una cache de dades abans de la TLB (caché de primer nivell)

Indexada per adreces virtuals i emmagatzema el contingut de la seva posició de memòria física associada

Petita i ràpida

Encara més: una caché després de la TLB (caché de segon/tercer nivell)

Indexada per adreces físiques i emmagatzema una copia del contingut de la memòria física associada

1. Protecció de memòria a nivell de programari
   1. Intèrpret de codi ensamblador: lent
   2. Avantatges:
      1. Simplificació de maquinari
      2. Protecció a nivell d’aplicació
      3. Protecció a nivell de nucli
      4. Portabilitat (dispositius)

Part 3

1. Casos d’ús de la memòria virtual
   1. Fitxers mapats a memòria
      1. SO pot manipular fitxers “afegintlos” a memòria
      2. Exemples d’aplicació:
         1. Manipulació amb adreces virtuals
         2. Començar a executar un programa a disc sense que estigui completament a memòria.
         3. Paginació sota demanda: una app pot accedir a més memòria de la que hi és físicament
            1. Si una app accedeix a una pàgina no disponible de la RAM, el SO la porta de manera transparent de disc a memòria
            2. Com funciona:

Pàgina no trobada

SO captura l’excepció. Comprova que és una adreça virtual vàlida i la converteix a una posició de bloc al disc

Es llegeix el bloc del disc.

SO actualitza la taula multinivell amb la nova pàgina. Es possible que s’hagi expulsat alguna de carregada

SO recupera execució en el mateix punt

TLB miss igual però es troba a multinivell. Es posa a TLB

* 1. Per processos
  2. Memòria virtual: tots els segments estan mapats a un fitxer de disc
     1. Més memòria de la que hi ha
     2. 100 fallades max per segon
     3. Pila o memòria dinàmica es mapan a la partició swap perquè s’hi puguin emmagatzemar aquelles pàgines que no caben a memòria. (Es perden al sortir)