Memòria virtual i traducció d'adreces

Lluís Garrido – lluis.garrido@ub.edu

Grau d'Enginyeria Informàtica

Introducció

Es pot crear un "entorn virtual" per a aplicacions d'ordinador? Ho hem vist per al sistema de comunicació interprocés:

- Una aplicació no ha de saber, o inclús no sap, si l'entrada o sortida són fitxers, dispositius o altres processos.

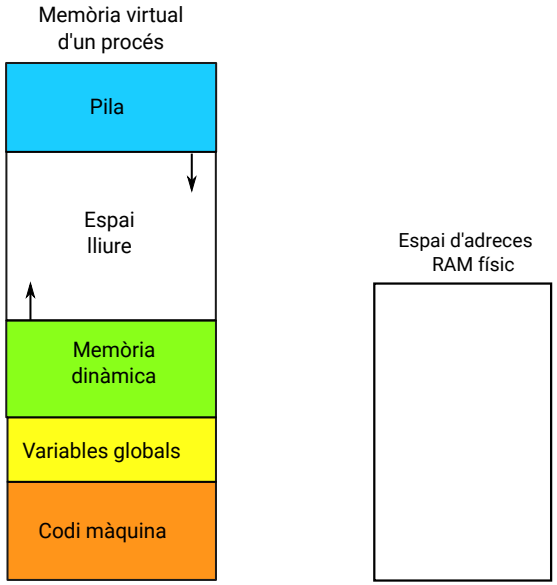
  ```
  $ ls | ./scanf > out.txt
  ```

  la comanda anterior executa la comanda `ls`, la pasa per una canonada a `scanf` i s'envia la sortida de `scanf` al fitxer `out.txt`.
- A nivell de programació (en C, Java, ...) només cal obrir el dispositiu de comunicació i enviar-hi les dades. El sistema operatiu s'encarrega fer les gestions necessàries perquè les dades arribin a destí.

Introducció

Es pot crear un "entorn virtual" per a aplicacions d'ordinador? Una cosa similar ocorre amb les direccions memòria:

- Cada procés "creu" que té assignat tot l'espai de memòria possible (64 bits) i que a més és "lineal". Però no és així...



Executeu el codi `exemple_malloc.c`. Observar que al malloc podem demanar més memòria que RAM física hi ha disponible.

Introducció

Estem acostumats a pensar que una direcció de memòria és només això: una direcció de memòria física.

- Per exemple, creiem que un punter apunta a la direcció de memòria RAM en què està emmagatzemada la variable o que el registre de comptador de programa apunta a la direcció de memòria RAM que s'està executant.
- Tot això és només una ficció! Executeu el codi `exemple.c` i `exemple_fork.c`: comproveu si els valors que surten per pantalla tenen sentit. No en tenen!

Introducció

Aquest "entorn de memòria virtual" s'aconsegueix gràcies a la col·laboració entreta entre maquinari i el sistema operatiu:

- El maquinari i el sistema operatiu s'encarreguen de gestionar un sistema que "tradueix" qualsevol direcció que genera un procés a una direcció física de memòria RAM.
- El sistema de traducció d'adreces és un concepte ben senzill però molt potent. Anem a veure'l! Com a programadors, però, és millor no pensar en tot aquest esquema a l'hora de programar...

Introducció

Gràcies al sistema de traducció d'adreces es poden implementar un munt de funcionalitats molt importants. Algunes d'elles són:

- Aïllament de processos o protecció dels processos entre sí. A més, permet a aplicacions construir "sandboxes" per executar aplicacions externes.
- Auto-aïllament d'un procés. Les diferents parts del codi estan protegides entre sí. Per exemple, la zona de codi executable no es pot modificar pel mateix procés.
- Execució d'un procés. Un procés pot executar-se sense tenir tot el codi executable a memòria RAM.
- Comunicació interprocés. Permet tenir una zona de memòria compartida entre processos perquè puguin compartir dades.

Introducció

Més funcionalitats importants:

- Codi executable compartit. Diferents processos poden compartir codi; per exemple llibreries comunes.
- Gestió de memòria de la pila o memòria dinàmica. El sistema operatiu ubica memòria per aquestes zones a mesura que creixen.
- Fitxers mapats a memòria. Es pot accedir a un fitxer accedint a posicions de memòria com si fos un vector (i.e. sense fer servir les funcions read i write).

Introducció

Com s'implementa?

- La majoria de sistemes (els que nosaltres utilitzem) tenen un maquinari especialitzat que realitza aquesta traducció; el maquinari està gestionat pel sistema operatiu.
- En alguns sistemes la traducció està proveïda, per exemple, per un intèrpret "byte-code".
- En altres sistemes la traducció es realitza en part a nivell de maquinari i en part a nivell de programari.

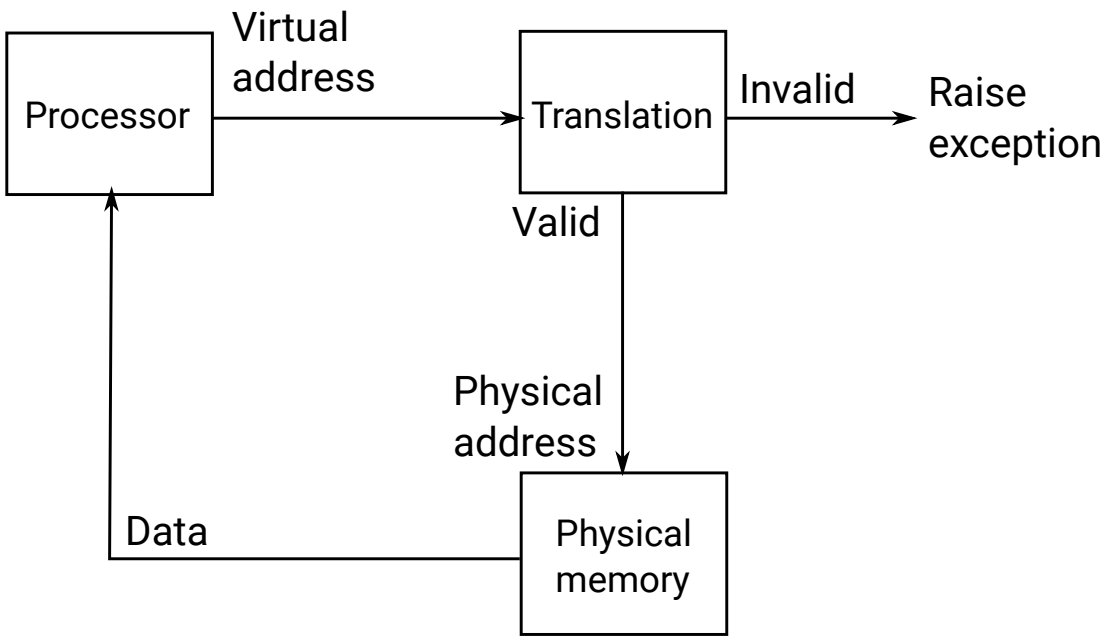
Contingut

El contingut d'aquestes transparències és

1. Concepte de traducció d'adreça. Els conceptes bàsics associats.
2. Traducció d'adreça flexible. Com es pot dissenyar el maquinari per proveir de la màxima flexibilitat al sistema operatiu.
3. Traducció eficient d'adreça. Quins mecanismes es poden utilitzar per fer una traducció eficient de l'adreça?
4. Traducció d'adreces a nivell de programari. Cada vegada més s'implementa la funcionalitat de traducció a nivell de programari. Què tal tenir en compte?

Concepte de traducció d'adreça

Totes (totes!) les direccions de memòria generades pels processos són "traduïdes" a una adreça de memòria física.



Concepte de traducció d'adreça

Respecte la traducció

- Un procés genera direccions de memòria virtual; cal realitzar la traducció a una adreça física.
- Abans de traduir una adreça, es comprova que l'adreça virtual és vàlida. Si no ho és, es produeix una excepció.
- La traducció acostuma a fer-se a nivell de maquinari, la Memory Mapping Unit (MMU). El sistema operatiu s'encarrega de gestionar i configurar aquest maquinari perquè realitzi aquesta tasca de traducció de forma correcta.

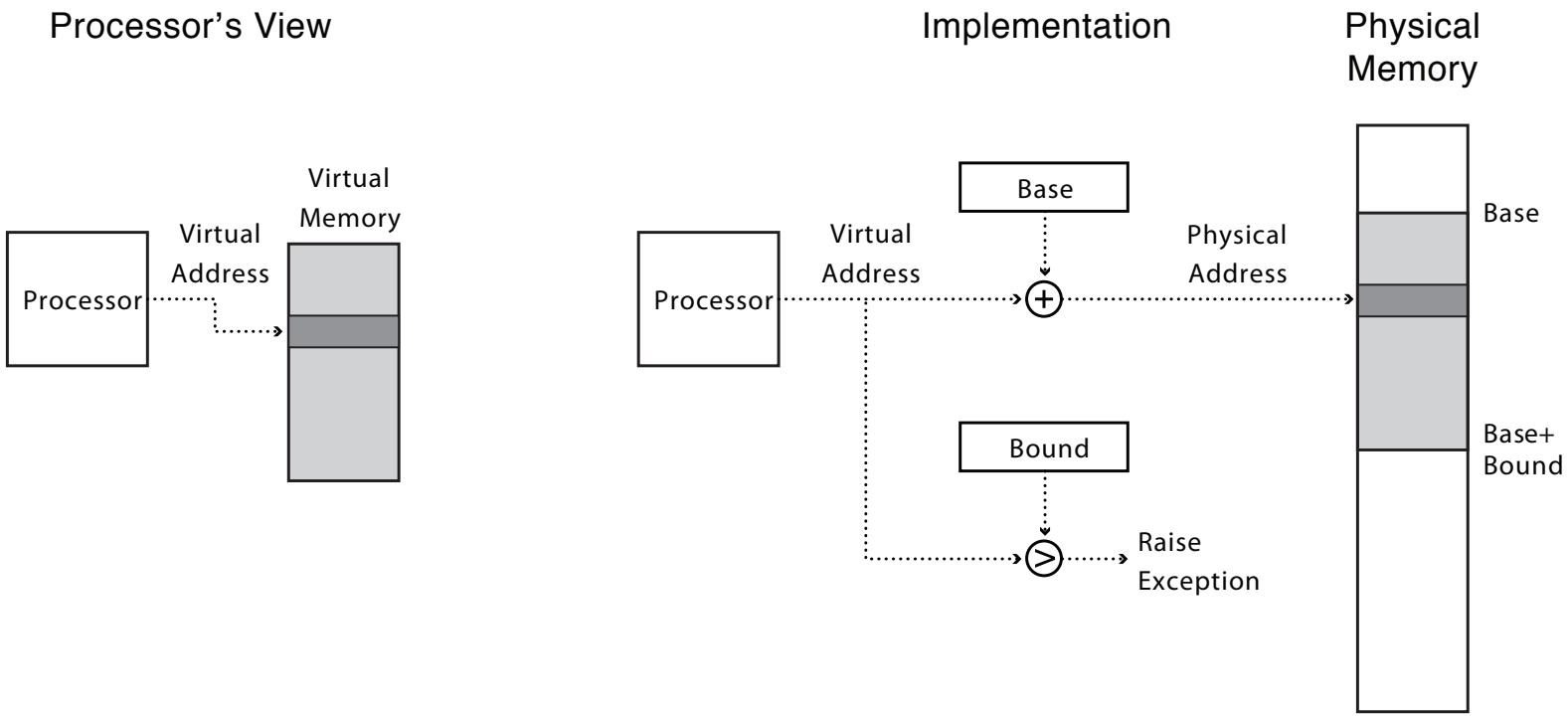
Concepte de traducció d'adreça

- Com es pot implementar aquesta traducció? Un vector? Un arbre? Una *hash table*? Totes les respostes són certes, totes s'han implementat en sistemes reals.
- A les transparències que vénen a continuació ens centrarem primer en com es tradueix; després ens centrarem en la traducció eficient.

Traducció d'adreça: *base* i *limit*

Un dels esquemes bàsics de traducció d'adreces fa servir dos registres: *base* i *limit*.

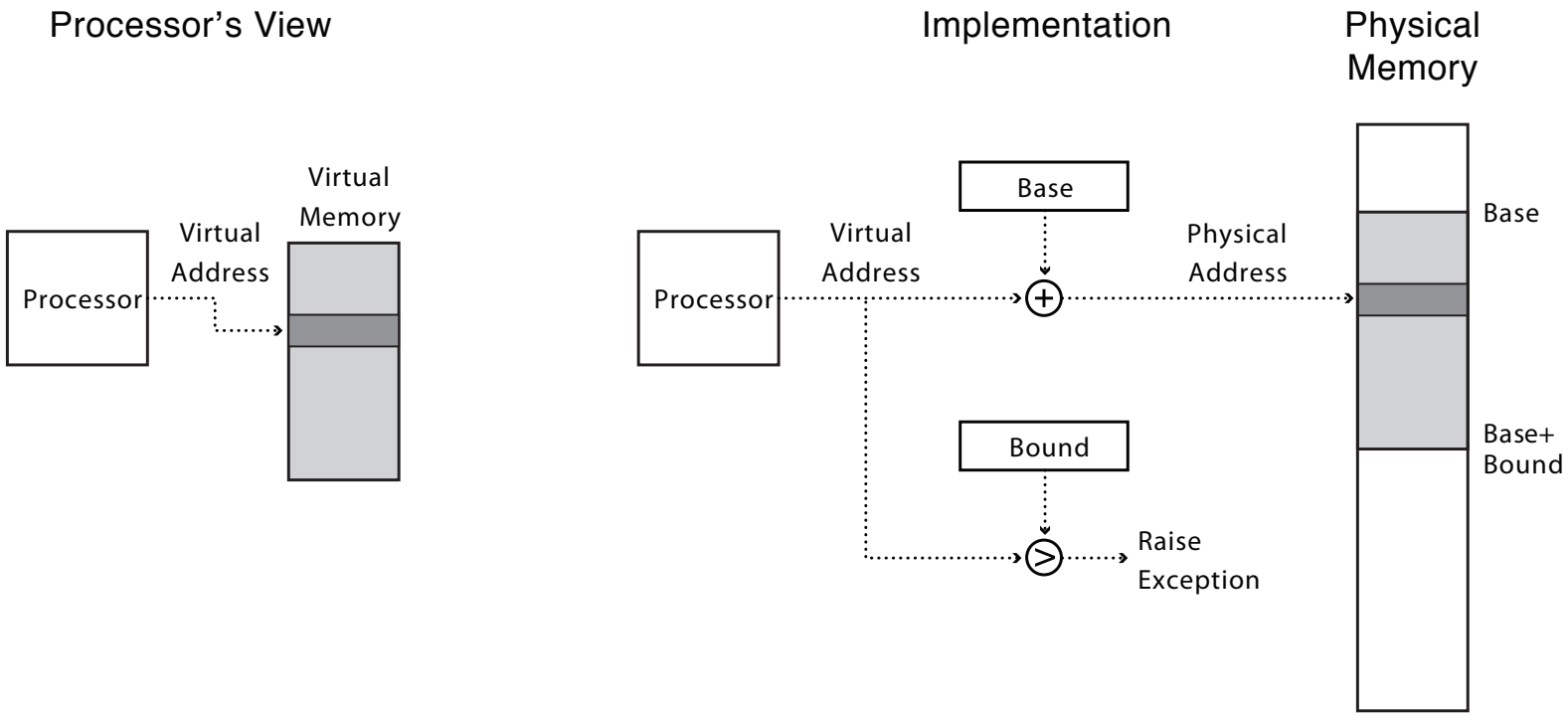
- Aquests registres només poden ser modificats per instruccions privilegiades.
- Cada procés té els seus propis registres base i limit.



Traducció d'adreça: *base* i *limit*

Traducció realitzada amb els registres *base* i *limit*

- Cada cop que s'accedeix a memòria, se suma la base a l'adreça. Si se supera el límit, es produeix una excepció.



Traducció d'adreça: *base* i *limit*

Esquema amb dos registres: *base* i *limit*. Té importants defectes

- Pila i memòria dinàmica expansible? Amb només dos registres cal preveure la memòria màxima que ocuparà el procés en el moment de carregar-lo de disc a memòria. Aquesta previsió és difícil de fer.
- Fragmentació de memòria? Amb aquest esquema tot el procés ocupa un espai continu, "lineal", que no es pot fragmentar en trossos.
- Auto-aïllament? Un procés pot sobreescriure la seva zona de codi executable, per exemple.
- Compartició de memòria? Amb aquest esquema no es pot aconseguir que diversos processos "comparteixin" una zona de memòria (per exemple, la de les instruccions màquina).
- Comunicació interprocés? Diversos processos no es poden comunicar entre sí compartint una zona de memòria.

Traducció d'adreça: memòria segmentada

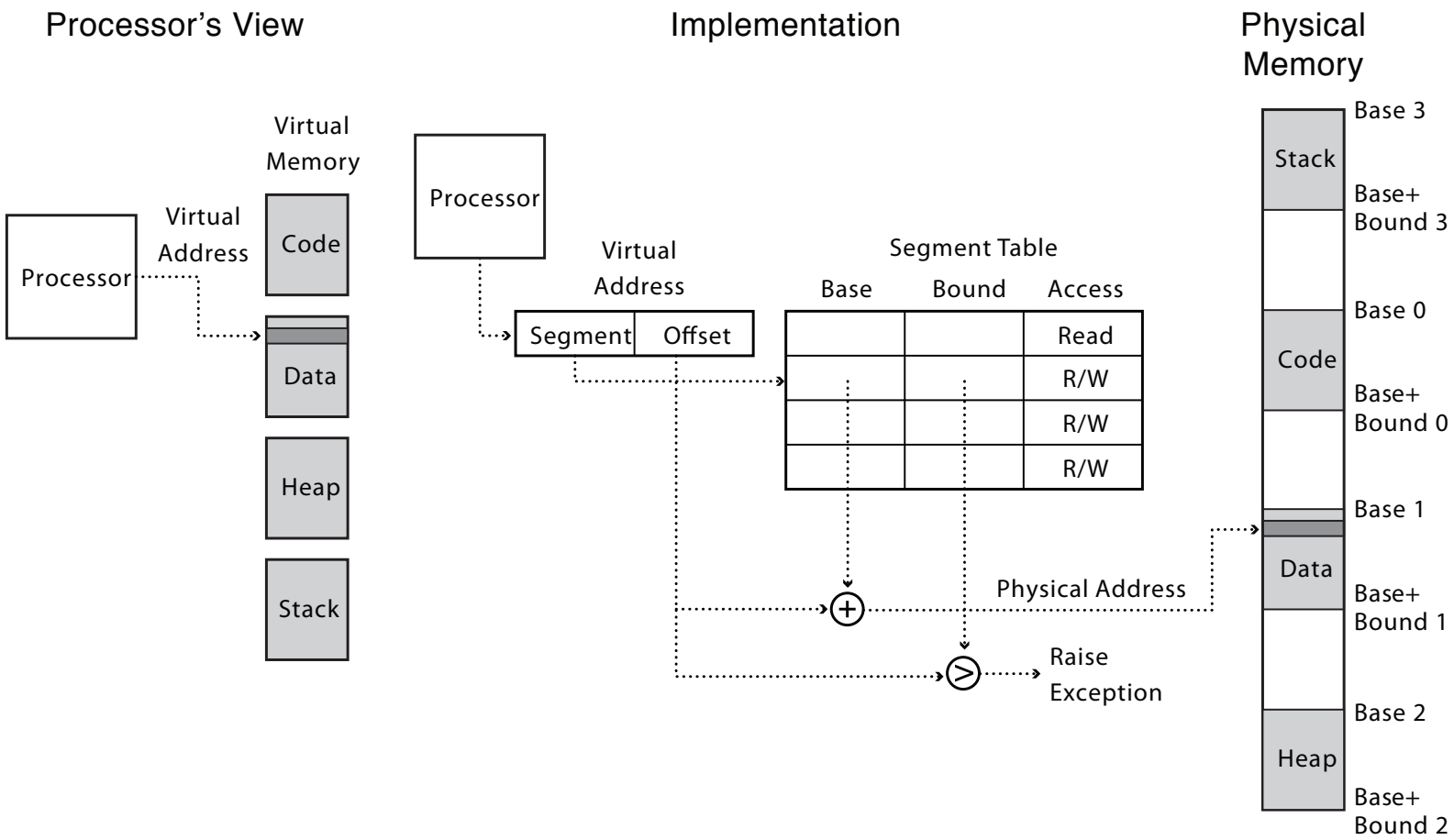
Molts dels problemes anteriors es poden solucionar de forma senzilla amb un canvi petit: en comptes de tenir un únic parell de registres *base* i *limit* per procés, el maquinari dóna suport per a múltiples registres *base* i *limit* per a cada procés.

És el que s'anomena segmentació.

- Cada parell de registres base i limit té associada una porció de l'espai d'adreces anomenat segment.
- Cada segment s'emmagatzema de forma contigua a memòria física i pot tenir una mida variable.

Traducció d'adreça: memòria segmentada

Dada adreça virtual té dos components: el número de segment (és a dir, el *base* i *limit* associat) i l'*offset* dins del segment.



Traducció d'adreça: memòria segmentada

Al sistema de memòria segmentada

- A una adreça virtual es bits alts estan associats al número de segment; els bits baixos a l'*offset*. El número de segments total possible depèn del nombre de bits alts associats.
- A nivell de maquinari es poden assignar diferents permisos d'accés (lectura, escriptura, execució) a cada segment. El sistema operatiu s'encarrega de gestionar-ho.
- Si un procés produeix una adreça de memòria virtual invàlida o accedeix hi accedeix de forma no vàlida, es produeix una excepció i el sistema operatiu genera un senyal SIGSEGV que s'envia al procés.

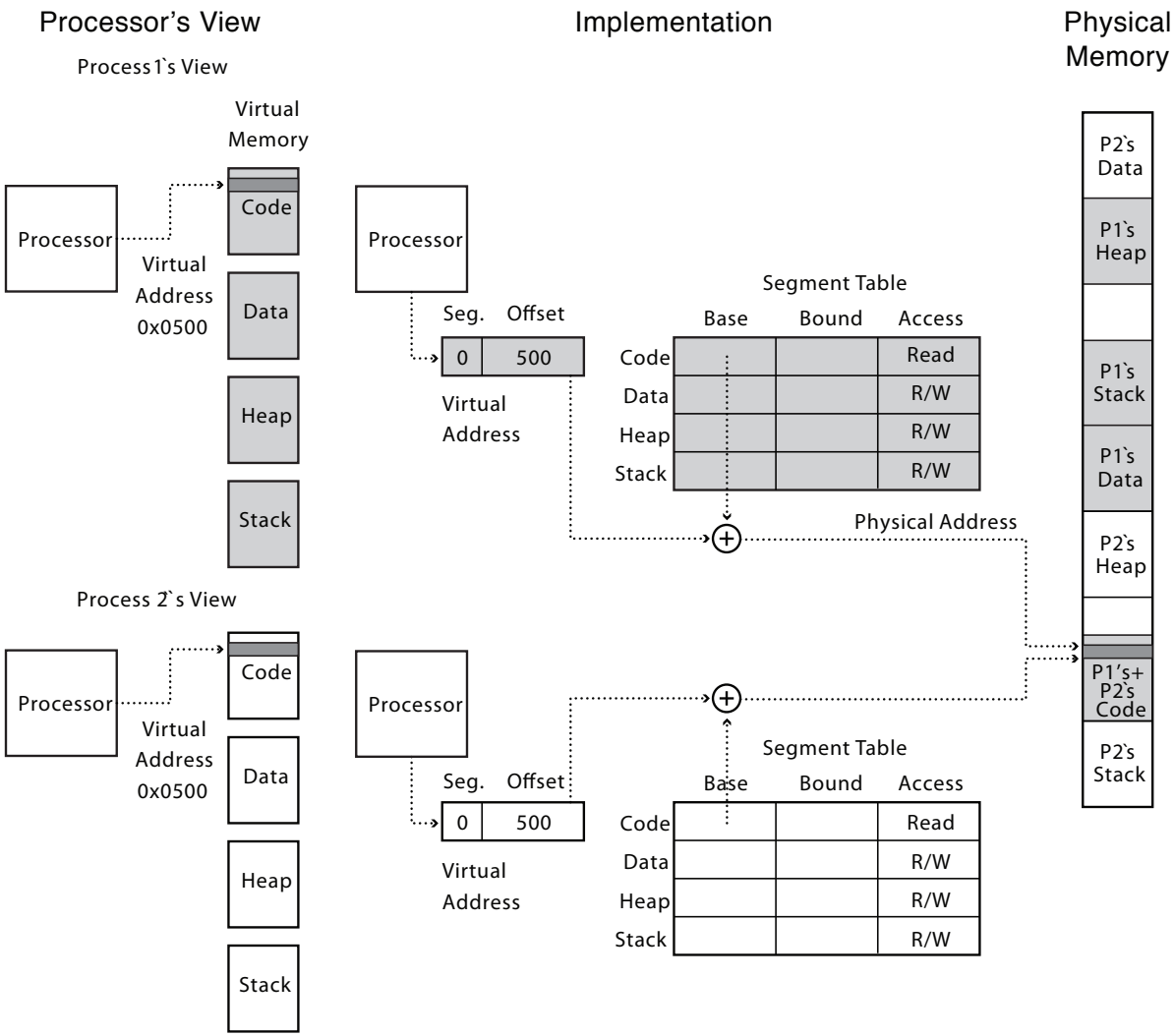
Traducció d'adreça: memòria segmentada

La memòria segmentada té importants avantatges respecte un esquema amb un sol parell de registres *base* i *limit*.

- Auto-aïllament. Permet que un procés es protegeixi a sí mateix diverses parts del codi.
- Fragmentació de memòria. L'espai de memòria d'un procés pot estar fragmentat en diversos trossos. A l'actualitat els segments estan associats a regions "gruixudes" com, per exemple, el codi, la pila, o la zona de memòria dinàmica.
- Compartició de segments. Diversos processos poden compartir codi (llibreries, etc.) compartint els registres *base* i *limit* d'un segment.
- Comunicació interprocés. Diversos processos poden comunicar-se entre sí compartint una zona de memòria.

Traducció d'adreça: memòria segmentada

Esquema de compartició de segments entre diversos processos:



Traducció d'adreça: memòria segmentada

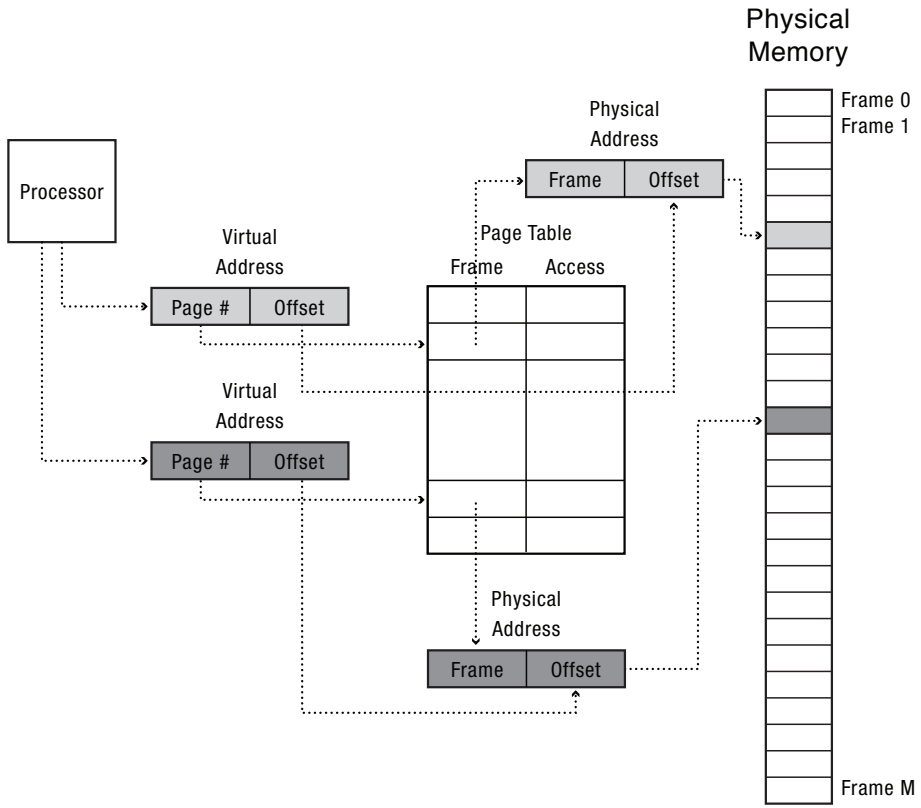
Aquest sistema té desavantatges. Per exemple, pot passar això:

1. A mesura que "passa el temps" la memòria física es divideix en trossos ocupats i lliures.
2. En crear un nou procés hem de crear nous segments. Es possible que no hi hagi cap zona lliure prou gran per a un segment. Però si "sumem" les regions lliures sí que hi pot haver una zona lliure prou gran per al segment.
3. En el cas anterior, el sistema operatiu pot "compactar" les regions per a unificar les regions lliures. Cal canviar els registres *base* i *limit* per a cada segment i moure les dades a memòria física. Les adreces virtuals no canvien.
4. L'operació de compactació és costosa! Un ordinador podria trigar un segon (!) a realitzar aquest procés.

Traducció d'adreça: memòria pàginada

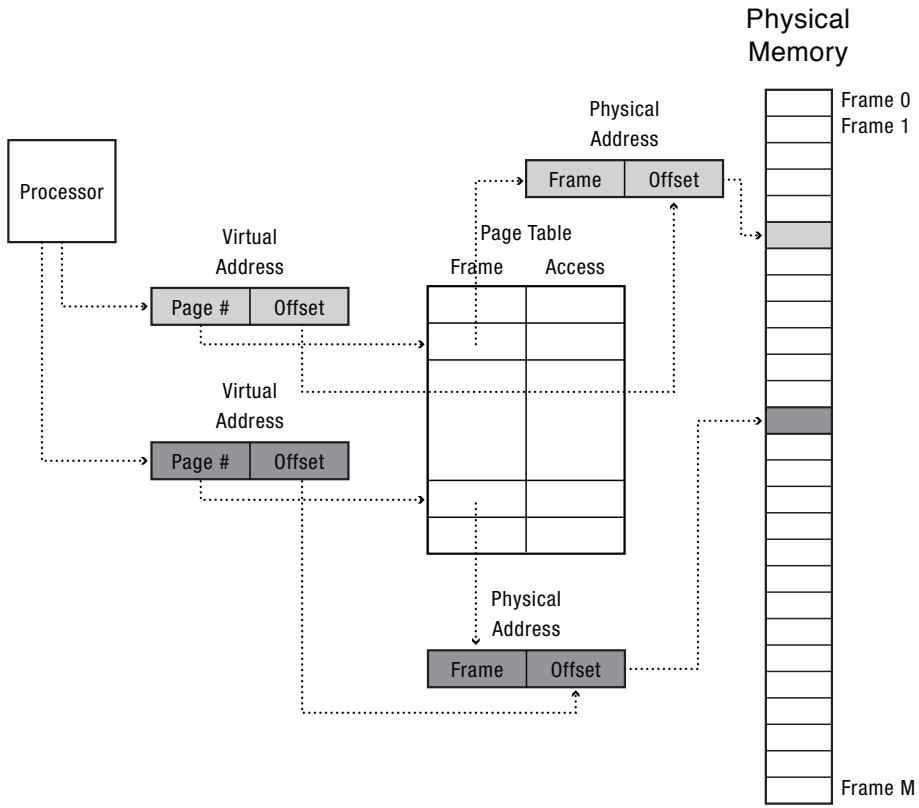
Una alternativa a memòria segmentada és la memòria paginada.

- Es fan servir blocs de mida fixa anomenats marcs de pàgina.
- La traducció es realitza mitjançant una taula. Cada procés té la seva pròpia taula i el sistema operatiu s'encarrega de gestionar-les.



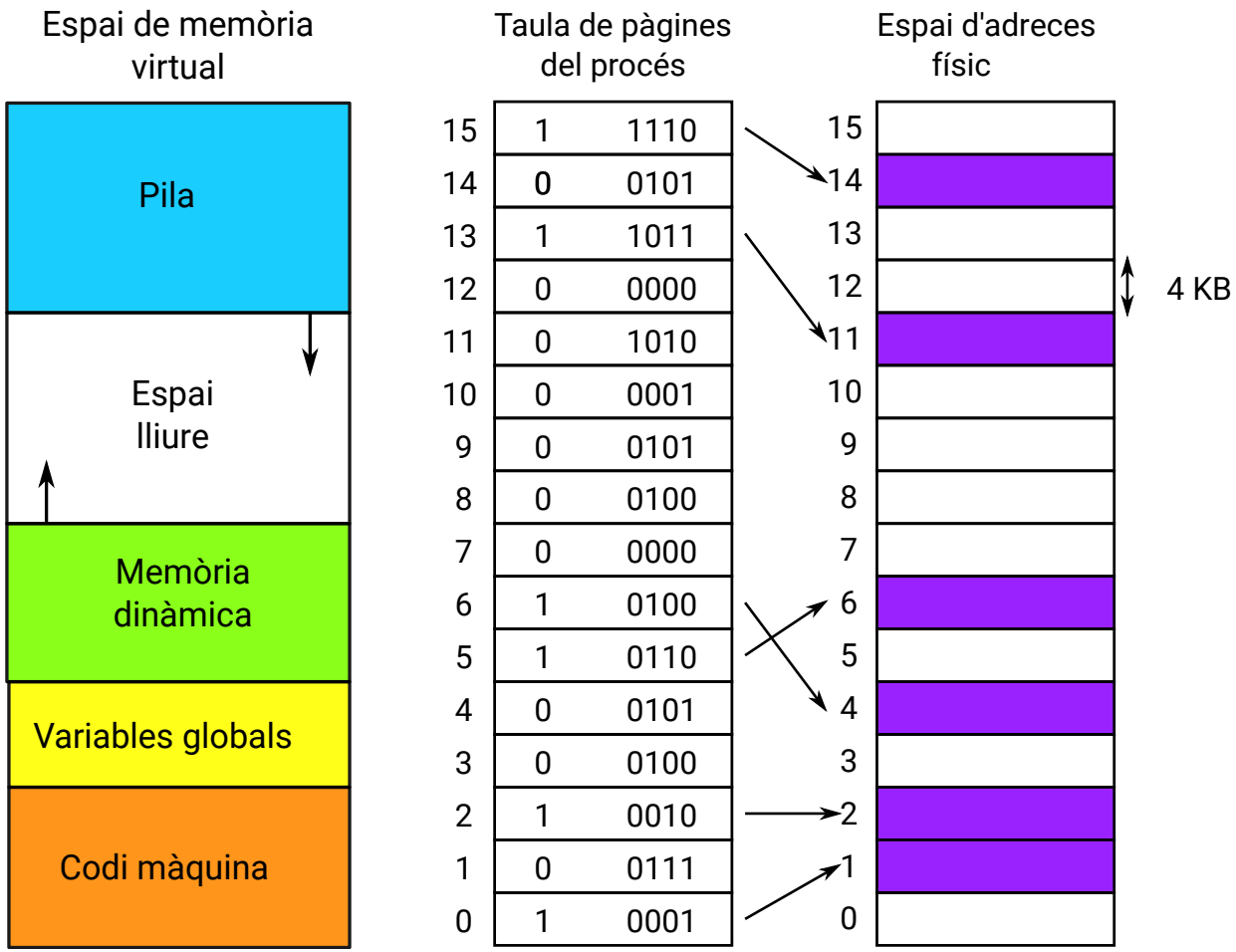
Traducció d'adreça: memòria paginada

- Els marcs de pàgina són de mida fixa i potència de 2: es fan servir els "bits alts" de l'adreça com a índex a la taula.
- Per fer la traducció només cal fer una "substitució" dels bits alts de l'adreça virtual pels bits indicats a la taula. No cal fer una suma com amb els segments.



Traducció d'adreça: memòria paginada

Exemple de mapat d'una adreça virtual a una adreça física amb arquitectura de 16 bits (veure següent transparència).



Traducció d'adreça: memòria paginada

Respecte l'esquema de la transparència anterior:

- Es un esquema de 16 bits amb marcs de pàgina de 4Kbytes (4096 bytes). Això dóna un total de $2^{16}/4096 = 16 = 2^4$ marcs de pàgina. Es faran servir doncs els 4 bits superiors de l'adreça per fer referència a un marc.
- Al dibuix anterior, el primer bit fa referència a si el marc virtual es troba a memòria física. Si hi és, els 4 bits següents fan referència al marc físic en què es troba. Si no hi és, els 4 bits no signifiquen res (és merda!)
- A l'exemple anterior els 4 primers bits de l'adreça virtual permeten "traduir" de l'adreça virtual a física:
  - L'adreça virtual 0000 1111 0000 1111 es mapa a 0001 1111 0000 1111.
  - L'adreça virtual 0001 1111 0000 1111 produeix una excepció.

Traducció d'adreça: memòria paginada

Unes conclusions del dibuix:

- El mapeig d'una adreça virtual a adreça física es realitza (típicament) a nivell de maquinari. Posicions contigües a memòria virtual no tenen perquè ser contigües a l'espai físic.
- Cada entrada a la taula conté la "traducció" al marc físic així com informació sobre les propietats associades (lectura, escriptura, execució, si està disponible o no, ...). Si un procés fa una operació invàlida, es produeix una excepció.
- Cada procés creu que té disponible "tot" l'espai de memòria possible: en sistemes de 32 bits són 4GB. A l'actualitat, els sistemes de 64 bits només es "veuen" realment 48 bits per motius electrònics.
- El sistema operatiu s'encarrega de gestionar les taules dels processos.

Traducció d'adreça: memòria paginada

Observar que el sistema de memòria pàgina és "molt txulo":

- Un procés "creu" que té assignat (tot) l'espai de memòria de forma lineal. Però...
- La memòria virtual d'un procés acostuma a estar "dispersa" entre la memòria física com si fos un mosaic.
  - Un element d'un vector es pot trobar en un marc de pàgina físic... i el següent element es pot trobar en un altre marc, una direcció física completament diferent.
  - O a l'inrevés: podem tenir marcs de pàgina contigus que corresponguin a dades completament diferents!

Traducció d'adreça: memòria paginada

Exemple amb dos processos: la memòria virtual permet gestionar la protecció de memòria entre aquests. El sistema operatiu és el responsable de gestionar les taules.