PHẦN MỀM HỆ THỐNG
QUẢN LÝ BỘ NHỚ

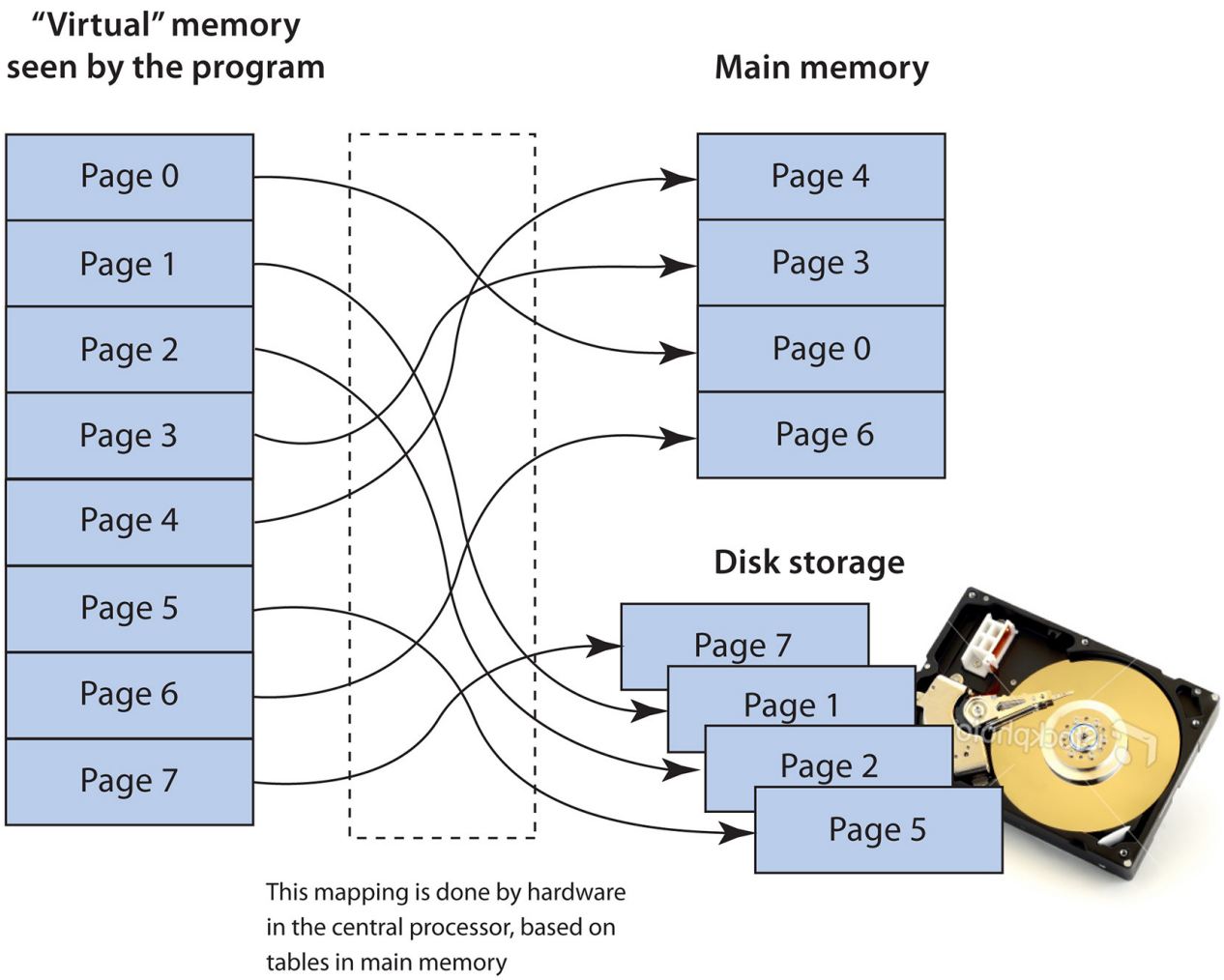
Nguyễn Hữu Đức

Viện Công nghệ thông tin và Truyền thông
Trường Đại học Bách khoa Hà Nội

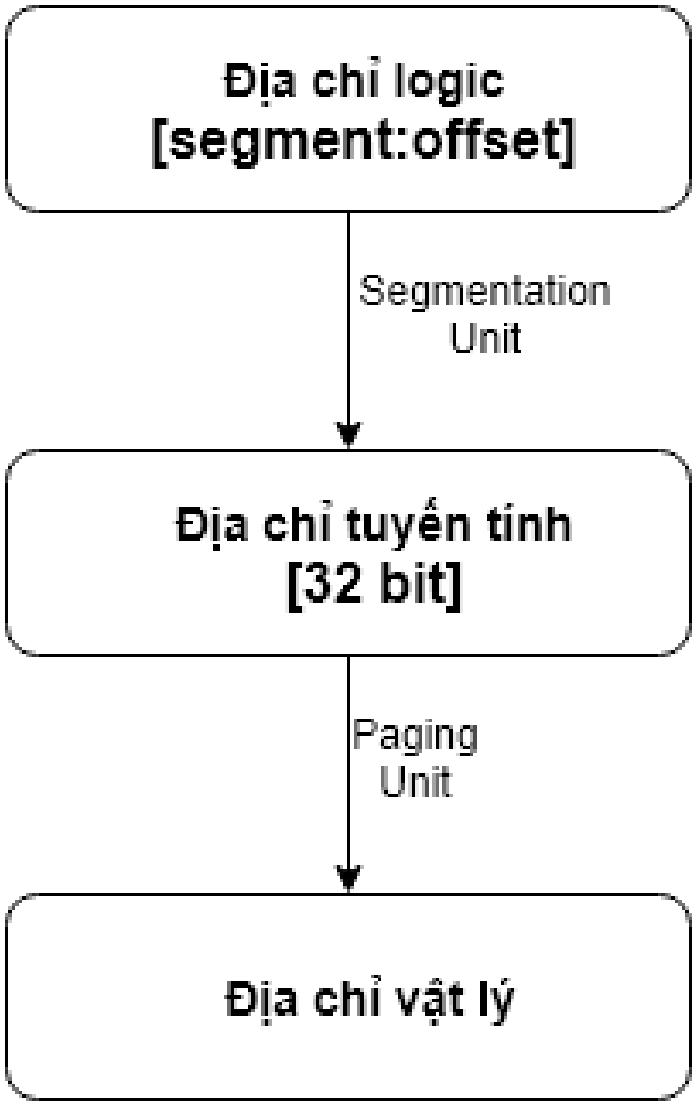
Tổng quan về quản lý bộ nhớ

- Quản lý bộ nhớ trên các hệ điều hành đơn nhiệm
  - Các chương trình và hệ điều hành chia sẻ cùng một không gian bộ nhớ
  - Chỉ một chương trình truy nhập bộ nhớ tại một thời điểm
- Quản lý bộ nhớ trên các hệ điều hành đa nhiệm
  - Nhiều tiến trình (chương trình) chia sẻ chung một không gian bộ nhớ vật lý
  - Mỗi tiến trình không "nhìn" thấy bộ nhớ của OS và các tiến trình khác
  - Tăng hiệu quả sử dụng bộ nhớ vật lý

Bộ nhớ ảo


"Virtual" memory
seen by the program
Main memory
Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 4
Page 3
Page 0
Page 6
Disk storage
Page 7
Page 1
Page 2
Page 5
This mapping is done by hardware
in the central processor, based on
tables in main memory

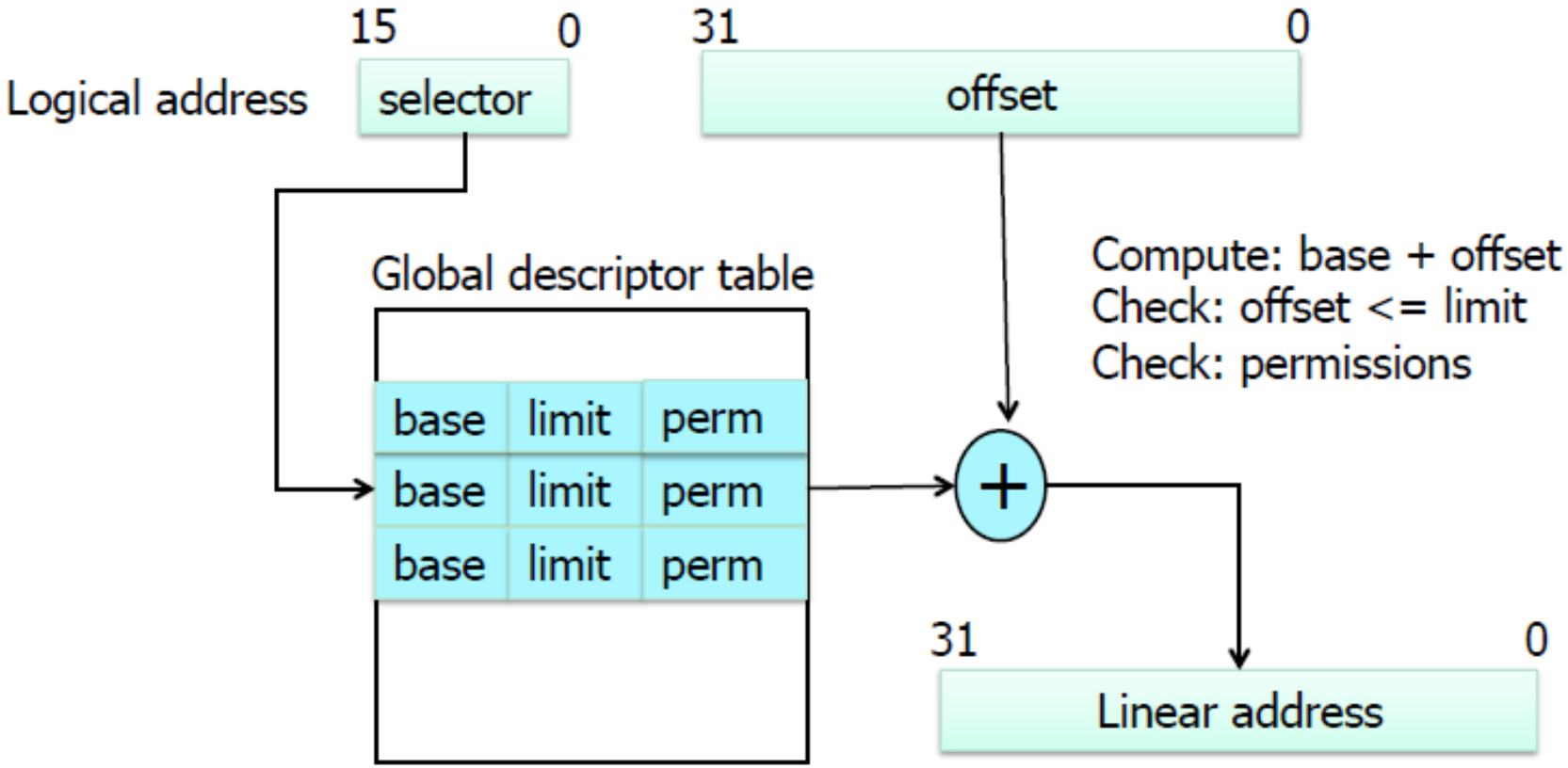
Chuyển đổi địa chỉ bộ nhớ trên x86



Phân đoạn - Segmentation

- Chia không gian bộ nhớ thành các đoạn (vd. text, data, ...)
  - Địa chỉ logic: `[segment:offset]`
- Bảng phân đoạn
  - **base**: Địa chỉ cơ sở của đoạn
  - **limit**: Kích thước đoạn
  - **perm**: Quyền truy nhập

Phân đoạn trên x86

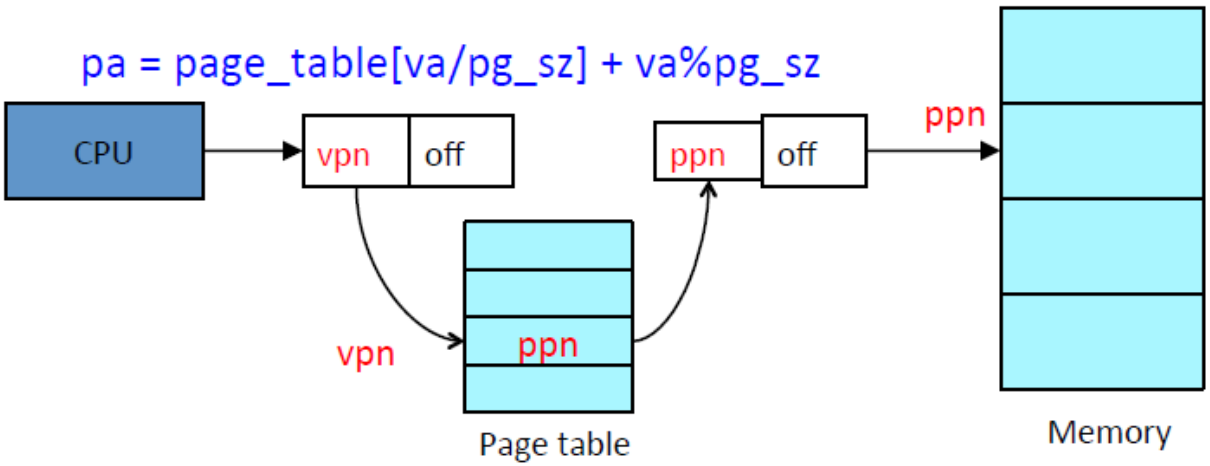


Phân trang

- Mục đích
  - Hạn chế sự phân mảnh bộ nhớ
  - Tránh cấp phát bộ nhớ khi chưa sử dụng
  - Chia sẻ bộ nhớ
- Bộ nhớ (ảo và vật lý) được chia thành các trang nhớ có kích thước cố định (4K, 4M)

Chuyển đổi bộ nhớ

- Địa chỉ tuyến tính: `vpn + offset`
- Địa chỉ vật lý: `ppn + offset`



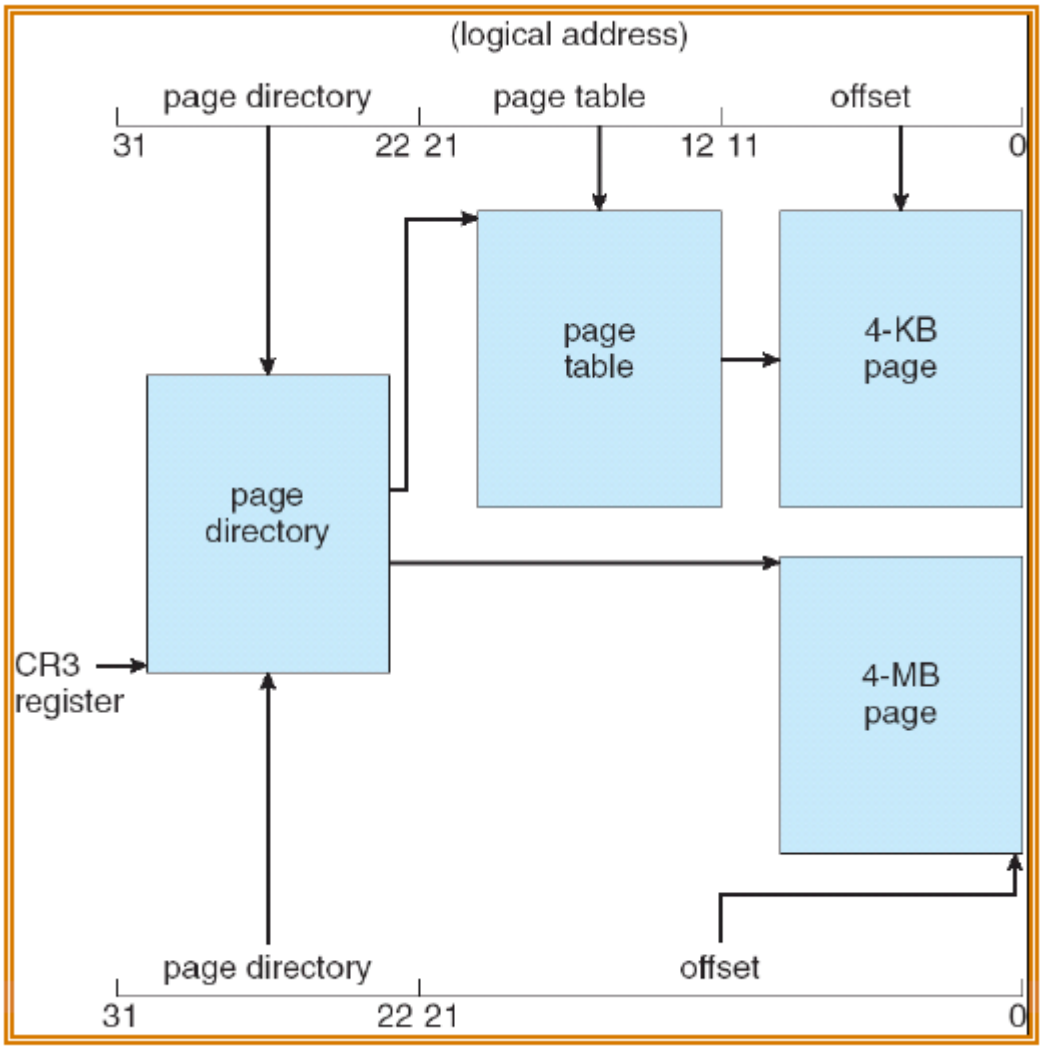
Các chế độ bảo vệ trang nhớ

- Gán các bit bảo vệ cho từng trang ảo
  - **present bit**: có ánh xạ tới trang vật lý không?
  - **read/write/execute bits**: có thể đọc/ghi/thực thi không?
  - **user bit**: được phép truy nhập ở user mode không?
- MMU kiểm tra quyền mỗi khi thực hiện truy nhập bộ nhớ

Vấn đề kích thước bảng phân trang

- Giả sử:
  - Không gian nhớ 4GB (32bit địa chỉ)
  - Kích thước trang nhớ: 4K
  - Mỗi entry trong bảng phân trang: 4 bytes
- Kích thước bảng phân trang sẽ là 4M cho mỗi tiến trình
- Thực tế: Không gian nhớ của các tiến trình thường thưa, do vậy sử dụng cấu trúc phân trang thông thường sẽ tốn kém bộ nhớ
- Giải pháp:
  - Hierachical paging
  - Hashed page tables
  - Inverted page tables

Phân trang trong x86



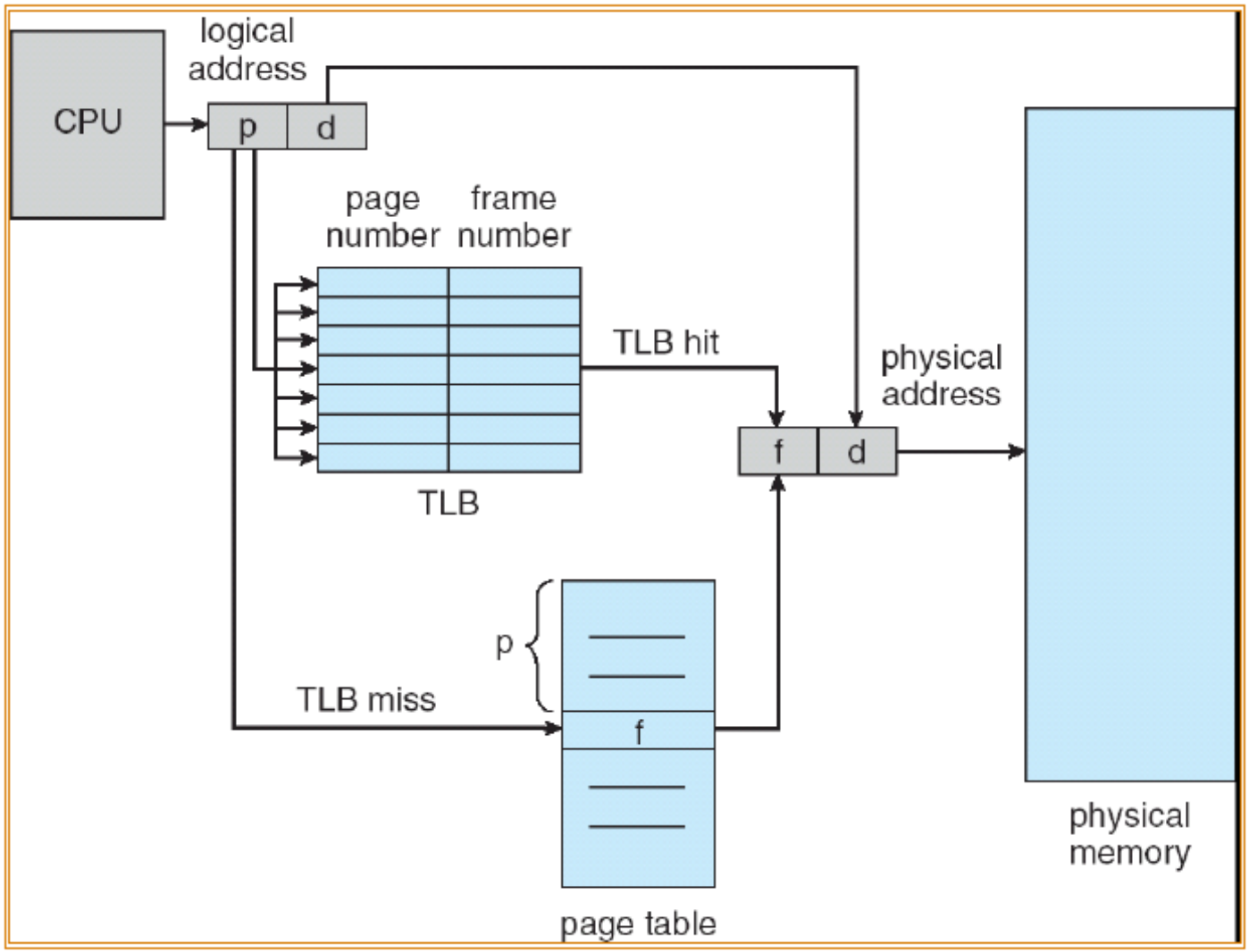
Phân trang trong x86-64

- Không sử dụng 64 bit địa chỉ (16EB), chỉ sử dụng 48bit
- Kích thước trang: 4K, 2M, 1G
- Phân cấp 4 mức
- PAE: địa chỉ ảo 48bit, địa chỉ vật lý 52bit

Translation Look-aside Buffer - TLB

- Quan sát: Các tiến trình thường chỉ sử dụng một số lượng nhỏ ánh xạ $vpn \rightarrow ppn$
- Để tăng tốc độ chuyển đổi địa chỉ, bộ nhớ liên kết (associative memory) được sử dụng cho các bảng ánh xạ tức thời (TLB) này.

Translation Look-aside Buffer - TLB



Translation Look-aside Buffer - TLB

- Xử lý TLB Miss
  - Phần cứng: x86
  - Phần mềm: MIPS, SPARC
- Giảm TLB Miss
  - Tăng kích thước TLB
  - Tăng kích thước trang
- Chuyển ngữ cảnh
  - Nạp lại toàn bộ TLB (x86)
  - Gắn PID vào từng mục trên TLB (MIPS, SPARC)

Chia sẻ trang nhớ

- Nhiều tiến trình có thể chia sẻ cùng một trang nhớ vật lý
- Tăng hiệu quả sử dụng bộ nhớ
  - Thư viên chia sẻ
  - Nhiều instance của cùng một ứng dụng
  - copy-on-write fork
- Dễ dàng trao đổi thông tin giữa các tiến trình

Chia sẻ trang nhớ