

## ÔN TẬP CHƯƠNG 8

### 1. Tại sao cần phải có bộ nhớ ảo?

- Bộ nhớ ảo (virtual memory): là một kỹ thuật cho phép xử lý một tiến trình không được nạp

toàn bộ vào bộ nhớ vật lý.

- Cần phải có bộ nhớ ảo vì:

+ Số lượng process trong bộ nhớ nhiều hơn.

+ Một process có thể thực thi ngay cả khi kích thước của nó lớn hơn bộ nhớ thực.

+ Giảm nhẹ công việc của lập trình viên.

### 2. Có bao nhiêu kỹ thuật cài đặt bộ nhớ ảo? Mô tả sơ lược các kỹ thuật đó?

Có hai kỹ thuật:

- Phân trang theo yêu cầu (Demand Paging)

- Demand paging: các trang của tiến trình chỉ được nạp vào bộ nhớ chính khi được yêu cầu.

- Khi có một tham chiếu đến một trang mà không có trong bộ nhớ chính (valid bit) thì phần

cứng sẽ gây ra một ngắt (gọi là page-fault trap) kích khởi page-fault service routine (PFSR)

của hệ điều hành.

- Phân đoạn theo yêu cầu (Demand Segmentation)

### 3. Các bước thực hiện kỹ thuật phân trang theo yêu cầu?

□ PFRS:

- Bước 1: Chuyển process về trạng thái blocked

- Bước 2: Phát ra một yêu cầu đọc đĩa để nạp trang được tham chiếu vào một frame trống; trong khi đợi I/O, một process khác được cấp CPU để thực thi

- Bước 3: Sau khi I/O hoàn tất, đĩa gây ra một ngắt đến hệ điều hành; PFSR cập nhật page table và chuyển process về trạng thái ready.

### 4. Mô tả các giải thuật thay thế trang FIFO, OPT, LRU?

|       | FIFO   | OPT  | LRU   |
|-------|--|--|---|
| Mô tả | <p>-Thay thế trang nhớ có thời gian vào sớm nhất trong các trang nhớ trong 3 khung trang</p> <p>- Bất thường (anomaly) Belady: số page fault tăng mặc dầu quá trình đã được cấp nhiều frame hơn.</p> | <p>Thay thế trang nhớ sẽ được tham chiếu trễ nhất trong tương lai -&gt; cần phải biết trước các trang sẽ được tham chiếu trong tương lai</p> | <p>Mỗi trang được ghi nhận (trong bảng phân trang) thời điểm được tham chiếu <math>\Rightarrow</math> trang LRU là trang nhớ có thời điểm tham chiếu nhỏ nhất (OS tốn chi phí tìm kiếm trang nhớ LRU này mỗi khi có page fault) • Do vậy, LRU cần sự hỗ trợ của phần cứng và chi phí cho việc tìm kiếm. Ít CPU cung cấp đủ sự hỗ trợ phần cứng cho giải thuật LRU</p> |

##### 5. Giải pháp tập làm việc hoạt động như thế nào?

- Được thiết kế dựa trên nguyên lý locality.
- Xác định xem process thực sự sử dụng bao nhiêu frame.
- Định nghĩa:
  - $WS(t)$  - các tham chiếu trang nhớ của process gần đây nhất cần được quan sát.
  - $\Delta$  - khoảng thời gian tham chiếu
  - Định nghĩa: Working set của process  $P_i$ , ký hiệu  $WS_i$ , là tập gồm  $\Delta$  các trang được sử dụng gần đây nhất.
- Nhận xét:

- $\Delta$  quá nhỏ  $\Rightarrow$  không đủ bao phủ toàn bộ locality.
- $\Delta$  quá lớn  $\Rightarrow$  bao phủ nhiều locality khác nhau.
- $\Delta = \infty \Rightarrow$  bao gồm tất cả các trang được sử dụng.
- Dùng working set của một process để xấp xỉ locality của nó
- Định nghĩa:  $WSS_i$  là kích thước của working set của  $P_i$ :
- $WSS_i =$  số lượng các trang trong  $WS_i$
- Đặt  $D = \sum WSS_i =$  tổng các working-set size của mọi process trong hệ thống.
- Nhận xét: Nếu  $D > m$  (số frame của hệ thống)  $\Rightarrow$  sẽ xảy ra thrashing.
- Giải pháp working set:
- Khi khởi tạo một quá trình: cung cấp cho quá trình số lượng frame thỏa mãn working-set size của nó.
- Nếu  $D > m \Rightarrow$  tạm dừng một trong các process.
- Các trang của quá trình được chuyển ra đĩa cứng và các frame của nó được thu hồi
- WS loại trừ được tình trạng trì trệ mà vẫn đảm bảo mức độ đa chương