Main memory

Address space: Ram là một thiết bị có thể đánh địa chỉ (thường chia theo mỗi byte, mỗi byte có thể biết dữ liệu đang nằm ở đâu trên ram).

+ Virtual/ logical: Địa chỉ được tạo và quản lý bởi cpu. Địa chỉ mà các không gian tiến trình và cpu làm việc trên đó.

+Virtual address space: là tập hợp các địa chỉ ảo (program address space).

+Physical address: là địa chỉ thật trên bộ nhớ chính.

+Physical address space: là toàn bộ các địa chỉ vật lý.

Address binding: Thực hiện mapping 1 địa chỉ vào address space, có thể thực hiện tại Compile time, Load Time (load vào bộ nhớ) và hầu hết ở giai đoạn Execution time (chỉ khi nào đang được thi mới xác định địa chỉ vật lý).

Memory Management Unit (MMU): phần mạch tích hợp trong cpu, chuyển đổi địa chỉ cpu đang thao tác đến địa chỉ vật lý.

Contigunous Memory Allocation: Cấp phát vùng nhớ liên tục. Áp dụng cho hệ thống Main Frame. Toàn bộ tiến trình không thể được phân nhỏ, và phải được nạp vào một không gian liên tục của bộ nhớ chính. => Không phù hợp với hệ thống hiện tại
Nếu kích thước lớn.
Bộ nhớ chính có thể chia thành nhiều partition, chia sẵn hoặc hình thành khi nạp tiến trình. (Fixed: chia sẵn có thể không hoàn toàn phù hợp với tiến trình gây lãng phí; Variable: hình thành trong quá trình loading, cấp vừa đủ)

Memory Allocation Strategy:

First - Fit
Ưu điểm: Đơn giản hiệu quả.
Giảm sự phân mảnh Memory. Tốc độ của chiến lược phân bổ First - Fit nhanh do tìm vị trí trống đầu tiên có thể giữ process.
Nhược điểm: Hiệu suất kém trong bộ nhớ bị phân mảnh nhiều.
First - Fit có thể chọn phân bổ những vùng nhớ lớn hơn yêu cầu của tiến trình, dẫn đến lãng phí bộ nhớ và giảm khả năng đáp ứng các yêu cầu bộ nhớ khác.

Best - Fit
Ưu điểm: Tiết kiệm bộ nhớ.
Giảm tình trạng external fragmentation.
Nhược điểm: Tốc độ phân bổ chậm.
Tăng chi phí tính toán. Gia tăng khả năng gây internal fragmentation.

Worst - Fit
Ưu điểm:
Bằng cách chọn vùng nhớ trống lớn nhất, worst – fit tạo ra phân mảnh bên trong lớn, giúp tận dụng vùng nhớ để đặt các tiến trình nhỏ.
Nhược điểm: Hiệu suất kém trong bộ nhớ bị phân mảnh nhiều.
Phân bổ chậm do phải duyệt qua toàn bộ vùng nhớ để tìm vùng trống lớn nhất

Address Protection: được đánh từ 0 đến limit – 1, nếu hợp lý thì cộng với giá trị Relocation register

Swapping: Có thể swap giữa ram và phân vùng đặc biệt của ổ đĩa backing store hoặc fast disk. Quản lý bằng bit map hoặc linked list để swap vào fast disk.

Vấn đề phân mảnh:

External (ngoại vi): Khi các ô trống không liên tục, bị rời rạc (Tổng không gian trống đủ nhưng không liên tục để chứa tiến trình). Có thể dùng compaction để các khoảng trống lại về một phân vùng to hơn.

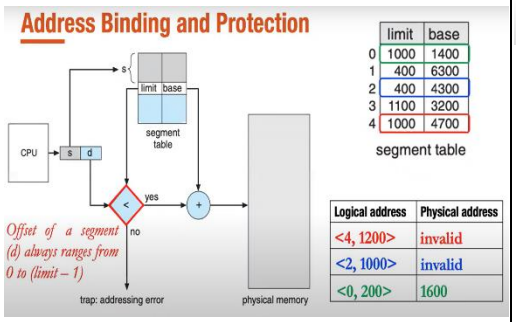
Internal (nội vi): khi vùng nhớ được chia sẵn thành các partition kích thước cố định. Và kích thước không phù hợp với tiến trình => tạo ra không gian trống không sử dụng. Giải pháp, thay đổi thuật toán cấp fphát.

Virtual memory

Noncontiguous memory allocation: chia tiến trình thành nhiều phần khác nhau, mỗi phần có thể load vào bộ nhớ chính và không cần liên tục.

Segmantation: Mỗi tiến trình được chia thành các segment (đoạn) khác nhau (Dựa trên các tiêu chí, luận lý riêng) rời rạc được nạp vào bộ nhớ chính không cần liên tục. Cần 2 thanh ghi Base và Limit.

<s: segment number, d: offset (0->Limit -1)>

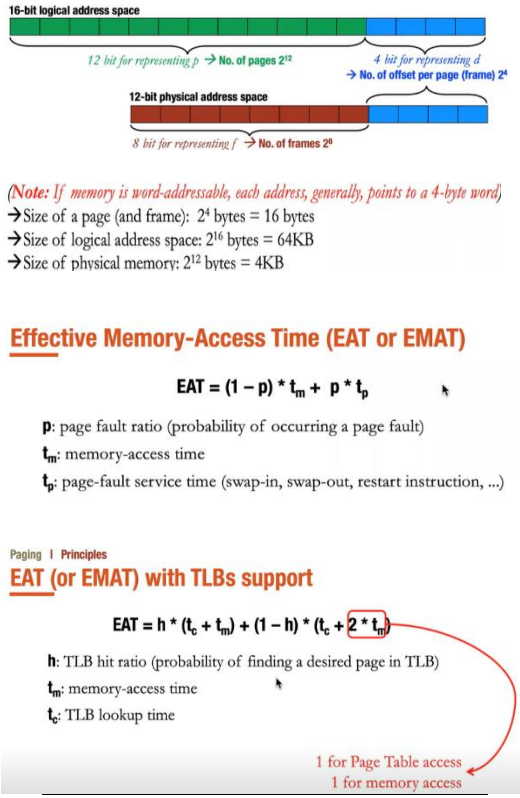


Paging: Tiến trình sẽ được chia thành các trang có kích thước bằng nhau do phần cứng quy định trước, còn bộ nhớ vật lý chia thành các khung trang, và các trang có thể được nạp không liên tục. Trang phải được nạp trọn vẹn vào một khung trang.

Địa chỉ logic <p: page: d: offset >

Địa chỉ vật lý <f: frame, d: offset>

Page table: dùng để mapping địa chỉ ảo sang địa chỉ vật lý.



FILE SYSTEM

File: Tập tin là một khái niệm trừu tượng, đơn vị lưu trữ ở mức logic, trừu tượng hoá byte dữ liệu trong ổ đĩa. (ordinary/Regular, Directory, shortcut, special). Gồm Name, size, location, creator, date, type, permission/ protection, etc.

Cách truy cập: Sequential tuần tự từng block theo đúng thứ tự, Direct random truy cập trực tiếp, indexed squential dựa trên inex để truy cập.

File -> chuỗi các byte hoặc lưu các thông tin dưới dạng record/ cây record.

Thư mục: tuyệt đối/ tương đối. tập tin thư mục ko có extention chứa tập hợp các file và directories con, gồm

Name, Path, Date, Permission/ protection.

Cấu trúc thư mục: Single level dễ cài đặt và tìm kiếm nhưng gặp vấn đề naming và grouping. Two-level: giải quyết vấn đề naming và tìm kiếm nhưng vẫn gặp vấn đề gouping (tầng1 cho người dùng). Tree-Structure: chia thành nhiều tầng hỗ trợ naming, grouping, searching nhưng truy cập phức tạp, vấn đề chia sẽ bộ nhớ. Graph: hổ trợ link thư mục này sang thư mục khác, cho phép chia sẽ dữ liệu nhưng tốn chi phí.

Protection: dùng Access Control List, gắn từng file để dịnh quyền Read, Write, Execute. 3 class user: Owner, group, universe (public). Dãy gồm 9 bit theo thứ tự class và theo RWX. (1 là bật)

Ổ cứng: có thể chia thành nhiều partitions. Mỗi vùng phải cài đặt file system có thể khác nhau.

FAT (flash disk, removable device): file allocation table, có tính tương thích cao với nhiều thiết bị do cũ chứa được tối đa 4GB/ file. exFat là phiên bản tối ưu hơn có thể lưu trữ 16EB/file nhưng tương thích kém hơn. Không bảo mặt, backup restore, lưu nhật ký.

NTFS (window): Là hệ thống tập tin mới, bảo mặt, ghi nhật ký, nén file, … hổ trợ tối đa 16EB/file.

Ext234 (linux): Hỗ trợ journaling (check sum), 16GB -> 16TB /file. Một thư mục tối đa có 6400 thư mục con

HFS, HFS+, APES (MacOS).

Cách mapping file thành block bên dưới ổ đĩa (Cấp phát block):

+ Contiguous Allocation: cấp liên tục, Tìm các block đĩa còn trống liên tục nhau để lưu file. Lưu vị Start và chiều dài block tính luôn block ban đầu, dễ thực hiện, hiệu suất cao, gây phân mảnh, file lớn thì không đủ không gian, và phải biết kích thước file để cấp phát.

+ Linked list Allocation: Dùng linked list, lưu block start và end, block n lưu địa chỉ của block n + 1. Giải quyết vấn đề phân mảnh, Mở rộng nhưng không hổ trợ random access, cần một không gian trong block để lưu pointer, mất 1 block sẽ mất liên kết.

+ Index Allocation: Index block, block chứa tất cả các block chứa dữ liệu trong file theo thứ tự. Cho phép random access, xử lý chuyện 1 block bị mất. Không phù hợp file nhỏ do phải dùng 1 block để lưu index gây lãng phí.

Window: Linked list + File Allcation table để lưu truy vết của tất cả các block.

UNIX: dùng: Indexed Allocation (I-nodes) multilevel indexed allocation. Khớp từng loại kích thước file.

DEAD LOCK

KN: Nhiều tiến trình, trong đó mỗi tiến trình chờ tài nguyên đang được cấp phát cho tiến trình khác tronng khi đang sở hữu tài nguyên nhất định => chờ => ngưng hoạt động => deadlock

Điều kiện: Độc quyền truy xuất, chiếm giữ và chờ đợi, không chiếm đoạt (nhả khi process đang giữ nhả), Tạo thành một cu trình ít nhất 2 process.

=> vi phạm thì ko có deadlock

Handle: Ignorance, preventiion, Avoidance, Detection, and recovery.

ĐỒNG BỘ HOÁ

Race condition: Nhiều thread chia sẽ một biến dùng chung, đồng thời cùng tranh đoạt điều khiển gây ra kết quả không mong đợi.

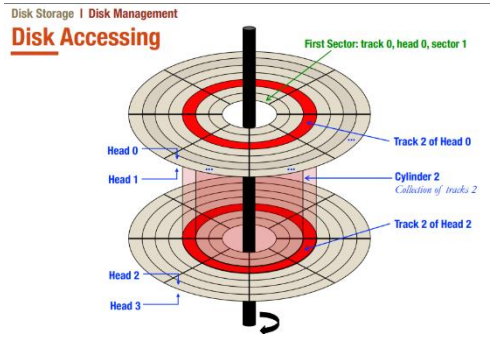
Critical Section: Là một đoạn code mà nhiều tiến trình hoặc tiểu trình có thể cùng cập nhật, thao tác dùng chung.

Tiêu chí Synchronization: Độc quyền truy xuất, process, thread ngoài không được block cái đang trong Critical Section, Thời gian đợi vào Critical Section được định sẵn.

Phương pháp:

+ Busy waiting: software (lock, strict, peterson's); hardware (Interupt disabling, TSL)

+ Sleep and wake up: semaphore (binary, counting); Monitor.



CHS to LBA address

$LBA = (C * N_{heads} + H) * N_{sectors} + (S - 1)$

LBA to CHS address

$C = LBA / (N_{heads} * N_{sectors})$

$H = (LBA / N_{sectors}) \% N_{heads}$

$S = (LBA \% N_{sectors}) + 1$

Mẹo đọc ổ đĩa: FCFS, SSTF Thì ok

SCAN: Đi đến đỉnh quay về gặp nào xử lý thg đó, C-SCAN Lên đỉnh về đáy rồi lên lại. LOOK: Lên thg trên xa nhất Xuống duyệt theo như scan. C-LOOK lên thg xa nhất về cũng thg xa nhất rồi lên lại.

Page fault (Lỗi trang)	Reference Bit	FIFO	f4	f3	f2	f1	
*	8	8				8	8
*	82	82			2	8	2
*	821	821		1	2	8	1
	821	821		1	2	8	2
*	8213	8213	3	1	2	8	3
*	0	2130	3	1	2	0	0
	02	2130	3	1	2	0	2
	021	2130	3	1	2	0	1
*	08	0218	8	1	2	0	8
							…