

NGÔ HẢI ANH

NGUYÊN LÝ HỆ ĐIỀU HÀNH

BÀI TẬP THỰC HÀNH

NGÔ HẢI ANH

Hà Nội - 2023

NGUYÊN LÝ HỆ ĐIỀU HÀNH

Mục lục

1	GIỚI THIỆU	3
2	PROCESSES AND THREADS	4
3	MEMORY MANAGEMENT	5
3.1	Memory Abstraction	5
3.2	Virtual memory	5
3.3	Page Replacement Algorithms	6
4	FILE SYSTEMS	8
4.1	Files	8
4.2	Directories	8
4.3	File system implementation	8
4.4	File system management and optimization	9
5	INPUT/OUTPUT	10
5.1	Principles of I/O hardware	10
5.2	Disks	10
5.3	Clocks	11
5.4	User Interfaces	11
6	DEADLOCK	12
6.1	Bỏ qua deadlock	12
6.2	Phát hiện và điều chỉnh	12
6.3	Phòng tránh tự động	13
6.4	Ngăn ngừa deadlock	13
A	CÂU HỎI LÝ THUYẾT	14

Danh sách hình vẽ

3.1	Mối quan hệ giữa địa chỉ ảo và địa chỉ vật lý.	6
5.1	Cấu tạo của một ổ đĩa cứng (Hard Disk Drive-HDD).	11

Chương 1

GIỚI THIỆU

Một *hệ điều hành* là chương trình mà sau khi được nạp ban đầu vào máy tính bởi một chương trình nạp, sẽ quản lý toàn bộ các chương trình khác trong máy tính. Các chương trình khác được gọi là các *ứng dụng* hay các *chương trình ứng dụng*. Các chương trình ứng dụng sử dụng hệ điều hành bằng cách tạo ra các yêu cầu dịch vụ thông qua một *giao diện chương trình ứng dụng* (API) được định nghĩa từ trước. Ngoài ra, người dùng có thể tương tác trực tiếp với hệ điều hành thông qua một giao diện người dùng dạng *dòng lệnh* (command line) hoặc dạng *giao diện người dùng đồ họa* (GUI).

BÀI TẬP

- 1.1 Một hệ thống máy tính có đủ chỗ cho bốn chương trình trong bộ nhớ chính. Các chương trình này ở trạng thái chờ thao tác I/O hết một nửa thời gian. Hỏi đâu là phần trăm thời gian CPU bị lãng phí? Biết rằng hiệu suất CPU (CPU utilization) được tính bằng $1 - p^n$, nếu có n là số tiến trình trong bộ nhớ tại cùng thời điểm, và một tiến trình sẽ tiêu tốn p phần trăm thời gian của nó để hoàn tất I/O (trong thời gian đó CPU sẽ ở trạng thái idle), thì thời gian lãng phí CPU sẽ là p^n .
- 1.2 Một máy tính có 2 GB RAM, trong đó hệ điều hành luôn chiếm 256 MB. Giả sử mọi tiến trình đều đòi hỏi 128 MB, và có đặc điểm giống nhau. Nếu đặt mục tiêu hiệu suất CPU đạt 99% thì thời gian chờ thao tác I/O tối đa là bao nhiêu?

Chương 2

PROCESSES AND THREADS

BÀI TẬP

- 2.1 Có năm công việc đang chờ thực hiện. Thời gian chạy tương ứng là 9, 6, 3, 5 and X . Thứ tự chạy như thế nào sẽ có thời gian phản hồi (average response time) nhỏ nhất? (Câu trả lời phụ thuộc vào X).
- 2.2 Có năm công việc ký hiệu từ A đến E, cùng chạy trên một máy tính, tại cùng thời điểm. Thời gian chạy ước tính lần lượt là 10, 6, 2, 4, và 8 phút. Độ ưu tiên tương ứng là 3, 5, 2, 1, và 4, với 5 là độ ưu tiên cao nhất. Với mỗi thuật toán lập lịch dưới đây, xác định thời gian xoay vòng (turnaround time) trung bình của năm tiến trình. Các phụ phí như chuyển đổi giữa các tiến trình, thời gian chờ I/O,... có thể bỏ qua.
- (a) Round robin (xoay vòng)
 - (b) Priority (theo độ ưu tiên)
 - (c) First-come, first-served (theo thứ tự 10, 6, 2, 4, 8)
 - (d) Shortest job first

Với (a), giả sử hệ thống được hỗ trợ đa chương (multi-programmed), và mỗi công việc nhận được mức độ chia sẻ CPU công bằng so với các công việc khác. Với (b) đến (d) giả sử chỉ có một công việc chạy tại một thời điểm, cho đến khi công việc đó kết thúc.

Chương 3

MEMORY MANAGEMENT

3.1 Memory Abstraction

- Địa chỉ vật lý (physical address): là một vị trí thực trong bộ nhớ chính
- Địa chỉ luận lý (logical address): là một vị trí nhớ độc lập với cấu trúc, tổ chức vật lý của bộ nhớ
- Không gian địa chỉ (address space): là tập hợp các địa chỉ mà một tiến trình có thể dùng để định vị trong bộ nhớ
- Chuyển đổi địa chỉ (address translation): là quá trình ánh xạ một địa chỉ từ không gian địa chỉ này sang không gian địa chỉ khác
- Cơ chế đè (overlay): tại mỗi thời điểm chỉ giữ lại trong bộ nhớ những lệnh hoặc dữ liệu cần thiết, giải phóng các lệnh/dữ liệu chưa hoặc không cần dùng đến
- Cơ chế hoán chuyển (swapping): một tiến trình có thể tạm thời bị chuyển (swap) ra khỏi bộ nhớ chính và lưu trên một hệ thống lưu trữ phụ. Sau đó, tiến trình có thể được nạp lại vào bộ nhớ để tiếp tục quá trình thực thi
- Phân mảnh ngoại (external fragmentation): kích thước không gian nhớ còn trống đủ đáp ứng một yêu cầu cấp phát, tuy nhiên không gian nhớ này không liên tục
- Phân mảnh nội (internal fragmentation): kích thước vùng nhớ được cấp phát hơi lớn hơn vùng nhớ yêu cầu. Ví dụ cấp một khoảng trống 18.464 bytes cho một tiến trình yêu cầu 18.462 bytes
- Lỗ nhớ (Hole): một khối nhớ khả dụng (block of available memory). Vùng nhớ cấp phát cho một tiến trình cần phải được cấp từ một lỗ nhớ đủ lớn → How to satisfy a request of size n from a list of free holes?
 - First-fit: chọn khối nhớ trống phù hợp đầu tiên (first) kể từ đầu bộ nhớ
 - Next fit: chọn khối nhớ trống phù hợp đầu tiên kể từ vị trí cấp phát cuối cùng (left off last time)
 - Best-fit: chọn khối nhớ trống nhỏ nhất (smallest), thường phải tìm kiếm trên toàn bộ nhớ
 - Worst-fit: chọn khối nhớ trống lớn nhất (largest), thường phải tìm kiếm trên toàn bộ nhớ
 - First-fit and best-fit better than worst-fit in terms of speed and storage utilization

3.2 Virtual memory

- Bộ nhớ ảo: không gian nhớ tham chiếu bởi một địa chỉ logic → giúp cho không gian nhớ logic có thể lớn hơn nhiều so với không gian nhớ vật lý
- Hai phương pháp cài đặt bộ nhớ ảo: phân trang (paging) và phân đoạn (segmentation)
- Cơ chế phân trang (paging): thường sử dụng bảng trang (Page Table) để phiên dịch địa chỉ logic thành địa chỉ vật lý.
- Mỗi trường (entry) trong bảng trang chứa các thông tin (các bit thông báo):
 - Caching disabled: trạng thái trang có bị khóa (locked) hay không?
 - Referenced: trạng thái trang có được gọi gần đây không? (dùng trong các thuật toán thế trang)
 - Modified: trạng thái trang có thay đổi từ khi được nạp vào bộ nhớ không? (dùng trong các thuật toán thế trang)
 - Protection: cho phép mức độ truy cập trang (read, write,...)

- Present/Absent: liên quan đến thông tin xảy ra lỗi trang (page fault)
- Page frame number: số hiệu trang
- Bảng trang 2 mức (two-level page table): ví dụ không gian nhớ 32-bit có thể gồm 9-bit top-level page table và 11-bit second-level page table; khi đó số trang (ảo) sẽ là $2^{32-(9+11)} = 2^{20}$ (pages), độ lớn mỗi trang là $2^{32-20} = 4$ KB
- Hiện tượng lỗi trang (page fault): tham chiếu đến một trang không có trong bộ nhớ. Cách xử lý: tìm một khung nhớ nạn nhân (victim) trên bộ nhớ chính rồi tạm thời loại bỏ (thay thế) để “lấy chỗ” cho trang lỗi. Vấn đề: làm thế nào tìm ra được “nạn nhân” phù hợp? → các thuật toán thế trang

3.3 Page Replacement Algorithms

- Optimal (OPT hoặc giải thuật MIN)
 - Thay thế trang nhớ được tham chiếu trễ nhất trong tương lai
 - Không thực tế (cần lưu lại mọi quá trình chạy trước đó của tiến trình)
- NRU (Not Recently Used)
 - Mỗi mục (entry) trong Page Table có thêm 2 bit M (modified) và R (referenced: read, write)
 - Khi có lỗi trang, NRU thay thế trang nhớ đầu tiên ở loại nhỏ hơn trước (căn cứ vào bit R, M)
- FIFO (First-In, First-Out)
 - Duy trì danh sách liên kết (linked list) theo thứ tự đến của các trang
 - Trang đầu của danh sách sẽ được thế
- Second Change: kết hợp FIFO nhưng kèm điều kiện NRU
- LRU (Least Recently Used): thay thế trang không được tham chiếu (không được dùng) lâu nhất

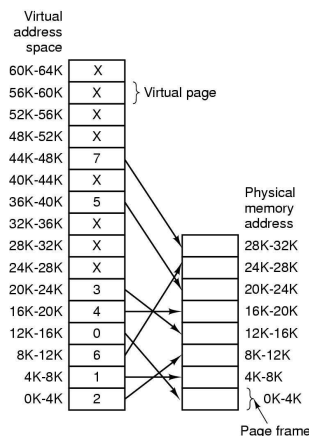
BÀI TẬP

3.1 Một hệ thống trao đổi (swapping system) trong đó bộ nhớ chứa các lỗ nhớ (hole) kích thước lần lượt là: 10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB, và 15 KB. Lỗ nhớ nào sẽ được lấy khi lần lượt có các yêu cầu như sau:

- (a) 12 KB
- (b) 10 KB
- (c) 9 KB

để đáp ứng yêu cầu *first-fit*? Câu hỏi tương tự với *best-fit*, *worst-fit*, và *next-fit*.

3.2 Sử dụng bảng trang như Hình 1 bên dưới, trong đó: mọi trang đều bắt đầu bằng một bội số của 4096 và kết thúc bằng cộng thêm 4095, do vậy ký hiệu 4K-8K nghĩa là 4096-8191, và từ 8K tới 12K tương đương với 8192-12287. Tìm *địa chỉ vật lý* (physical address) tương ứng với mỗi *địa chỉ ảo* (virtual addresses) sau:



Hình 3.1: Mối quan hệ giữa địa chỉ ảo và địa chỉ vật lý.

- (a) 20
- (b) 4100
- (c) 8300

3.3 Một máy tính với 32-bit địa chỉ sử dụng một *bảng trang hai mức* (two-level page table). Các địa chỉ ảo được chia vào 9-bit trường bảng trang cấp một, và 11-bit trường bảng trang cấp hai. Hỏi các trang có độ lớn bao nhiêu và có bao nhiêu địa chỉ?

3.4 Một máy tính có bốn khung trang. Thời gian nạp (loading), thời gian truy cập cuối (last access), và các bit R và M cho mỗi trang được cho như bên dưới:

Page	Loaded	Last Ref.	R	M
0	126	280	1	0
1	230	265	0	01
2	140	270	0	0
3	110	285	1	1

- (a) Trang nào sẽ bị thay thế bởi thuật toán NRU?
- (b) Trang nào sẽ bị thay thế bởi thuật toán FIFO?
- (c) Trang nào sẽ bị thay thế bởi thuật toán LRU?
- (d) Trang nào sẽ bị thay thế bởi thuật toán second chance?

Chương 4

FILE SYSTEMS

4.1 Files

- Khái niệm tệp (file)
 - Files are logical units of information created by processes
 - Files provide a way to store information on the disk and read it back later
 - Cách “tổ chức” thông tin của tệp: theo byte (byte sequence), theo bản ghi (record sequence) hoặc cấu trúc cây (tree)
- Một số tính chất (properties) của file
 - File name: có (không có) phân biệt chữ hoa, chữ thường (UNIX/MS-DOS), đa số hệ điều hành hỗ trợ tên tệp gồm 2 phần (two-part file names) gọi là tệp mở rộng (file extension), ví dụ: file.bak là dạng tệp dự phòng, file.txt là tệp văn bản thông thường,...
 - File structure: chính là cách “tổ chức” thông tin của tệp
 - File types: thường các hệ điều hành hỗ trợ hai kiểu chính là Regular files (các tệp thông thường, có phần mở rộng, lưu trữ thông tin cụ thể) và Directories (là tệp hệ thống – system files, thường lưu trữ cấu trúc tệp – file system hay đơn giản lưu trữ các tệp thông thường và các thư mục con)
 - File access: việc truy cập thông tin của tệp có thể là tuần tự (sequential access) hoặc ngẫu nhiên (random access)
 - File attributes: thuộc tính của tệp, đôi khi còn gọi là metadata, cho biết các thông tin như: ai là người sở hữu (tạo ra) tệp, được phép ghi (write) hay chỉ-đọc (read-only), kích cỡ tệp,...
 - File operations: các thao tác với tệp như tạo/xóa/sửa, đổi tên, thiết lập thuộc tính, mở/đóng,...

4.2 Directories

- Khái niệm thư mục (directory/folder): Bản chất cũng là tệp, nhưng là tệp đặc biệt → tệp hệ thống (system files)
- Trong thư mục có thể có một/nhiều thư mục con, do đó cấu trúc thư mục có thể là đơn mức (single-level directory systems) hoặc phân cấp (hierarchical directory systems)
- Cần hiểu rõ tên đường dẫn (path name) của thư mục: gốc (root), hiện hành (current/working), cha, con, người dùng (home directory), đăng nhập (login directory), dấu phân cách thư mục (/ hoặc \),...
- Các thao tác với thư mục (directory operators): tạo, xóa, sửa, đổi tên, link, unlink,...

4.3 File system implementation

- File system layout
 - Hệ thống các file được lưu trên đĩa (disk), đĩa lại được chia thành các phân vùng (partition), mỗi partition lại gồm nhiều khối (block)
 - Sector 0 của disk được gọi là MBR (Master Boot Record), dùng để boot
 - Boot block: chạy đầu tiên mỗi khi hệ thống khởi động
 - Superblock: chứa số magic number để biết (FAT/NTFS), số blocks, thông tin quản lý,...
- Cài đặt tệp (Implementing files): cách lưu trữ nội dung các tệp trên đĩa, có 3 cách:

- Liên tiếp (Contiguous allocation):
 - * Cho phép random access,
 - * Lãng phí không gian, có thể gây ra hiện tượng phân mảnh đĩa (disk fragmentation)
 - * Thường dùng cho CD/DVD
- Danh sách liên kết (Linked List allocation): quản lý được vùng trống, không random access
- Đánh chỉ số (Indexed allocation) hay còn gọi là i-node (index-node)
 - * Duy trì bảng chỉ số (index table) gồm các con trỏ trỏ đến địa chỉ block
 - * Bảng FAT (File Allocation Table)
 - * Có hỗ trợ random access
- Các tệp được chia sẻ (shared files): Khái niệm liên kết (link), gồm hai loại là *liên kết biểu tượng* (symbolic link) hoặc *liên kết cứng* (hard link)

4.4 File system management and optimization

Quản lý không gian đĩa (Disk space management) cần chú ý hai yếu tố:

- Tỷ lệ dữ liệu (data rate) tỷ lệ thuận với kích cỡ khối (block size) bởi vì thời gian truy cập khối phụ thuộc vào thời gian tìm (seek time) và độ trễ xoay (rotational delay)
- Small blocks are bad for performance but good for disk space utilization → tỷ lệ nghịch

Chương 5

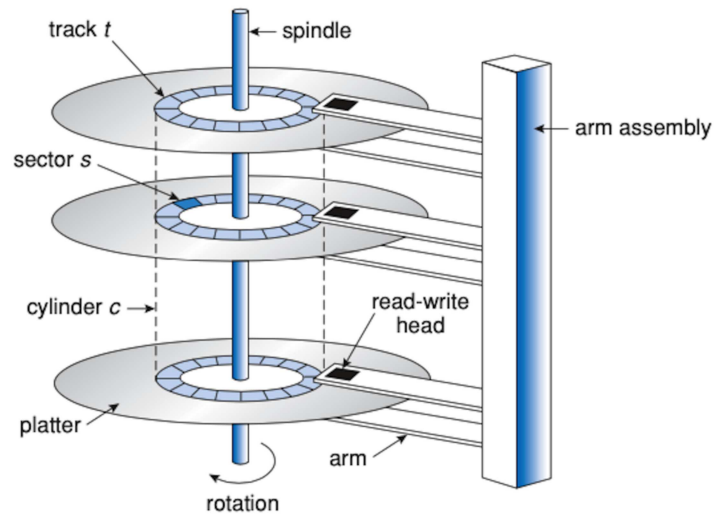
INPUT/OUTPUT

5.1 Principles of I/O hardware

- Các thiết bị I/O được chia thành 2 nhóm chính: dạng khối (block) và dạng ký tự (character), ngoài ra còn một số cách phân lớp thiết bị khác:
 - Truy cập tuần tự/truy cập ngẫu nhiên
 - Thiết bị dạng đồng bộ/không đồng bộ
 - Thiết bị chuyên dụng (dedicated) hoặc có thể chia sẻ (sharable)
 - Phân loại theo tốc độ hoạt động
 - Thiết bị chỉ đọc, chỉ ghi, đọc-ghi
- Một số khái niệm
 - I/O Device Controller: bộ điều khiển thiết bị
 - I/O Port: cổng chuyên dụng dành cho các mục đích I/O khác nhau: serial, parallel,...
 - I/O Bus: kết nối các Controller
 - I/O Address: được sử dụng bởi các lệnh I/O trực tiếp hoặc I/O được ánh xạ vào bộ nhớ
 - Memory-Mapped I/O: CPU đọc/ghi các controller giống như việc nó đọc/ghi một vùng bộ nhớ
- Các phương pháp chuyển dữ liệu I/O (giữa CPU và thiết bị I/O)
 - Programmed I/O (được lập trình sẵn)
 - * Mọi thao tác I/O phải được lập trình và được điều khiển
 - * Sử dụng vòng lặp busy-wait (còn gọi là polling) để kiểm tra tính sẵn sàng của thiết bị (ví dụ khi in tệp thì CPU cần kiểm tra máy in sẵn sàng nhận ký tự tiếp theo chưa)
 - * Ưu điểm: có thể lập trình, không cần hỗ trợ ngắt hay DMA
 - * Nhược điểm: tốn thời gian CPU (việc kiểm tra của CPU là liên tục)
 - Interrupt-driven I/O: gồm ngắt chính xác (precise) và ngắt không chính xác (imprecise)
 - * Ưu điểm: giúp CPU không bị busy-wait (tiết kiệm thời gian CPU)
 - * Nhược điểm: tốn chi phí khởi tạo, điều khiển ngắt
 - Direct Memory Access (truy cập bộ nhớ trực tiếp)
 - * Khi có thao tác I/O, ví dụ đọc một khối dữ liệu từ đĩa, nếu dùng CPU đọc bit trạng thái và đưa dữ liệu vào thanh ghi điều khiển 1 byte một lần → chi phí truyền dữ liệu rất lớn → dùng một bộ xử lý chuyên dụng gọi là DMA
 - * CPU viết một khối lệnh DMA (source, destination, byte) vào bộ điều khiển DMA rồi lại tiếp tục công việc khác
 - * Bộ điều khiển DMA tiếp tục các thao tác với bus bộ nhớ mà không cần sự hỗ trợ của CPU (CPU thực tế chỉ giữ nhiệm vụ kích hoạt bộ điều khiển DMA)
 - * Ưu điểm: giải phóng CPU khỏi các thao tác I/O
 - * Nhược điểm: bộ xử lý DMA chậm hơn CPU

5.2 Disks

- Disk hardware: các khái niệm track, sector, cylinder



Hình 5.1: Cấu tạo của một ổ đĩa cứng (Hard Disk Drive–HDD).

- CD-ROM, CD-Recordable, DVD
- Disk formatting
- Disk Arm Scheduling Algorithms
 - Read/write time factors
 - * **Seek time** (the time to move the arm to the proper cylinder).
 - * Rotational delay (the time for the proper sector to rotate under the head).
 - * Actual data transfer time.
 - Algorithms
 - * First-Come, First Served
 - * Shortest Seek First (SSF)
 - * Elevator
 - * Modified elevator

5.3 Clocks

Đồng hồ (hay còn gọi là bộ tính thời gian – timer), có thể lập trình được, thường gồm các thành phần:

- Bộ tạo dao động (crystal oscillator)
- Bộ đếm
- Thanh ghi

5.4 User Interfaces

- Keyboard, Mouse, Monitor
- X Windows System

Chương 6

DEADLOCK

Khái niệm *khóa chết*, hay *tắc nghẽn* (deadlock) được hiểu như sau “một tập hợp các tiến trình được coi là bị deadlock nếu mỗi tiến trình trong tập hợp đó đang chờ một sự kiện mà chỉ có thể gây ra bởi một tiến trình khác cũng đang ở trong tập hợp.”

Các điều kiện (conditions) cần thiết để xảy ra deadlock:

1. *Loại trừ lẫn nhau* (mutual exclusion): mỗi tài nguyên không được sử dụng bởi nhiều hơn một tiến trình tại một thời điểm;
2. *Giữ và chờ* (hold and wait): các tiến trình giữ tài nguyên và có thể yêu cầu thêm tài nguyên mới;
3. *Không có trưng dụng tài nguyên* (no preemption): các tài nguyên đang bị giữ không thể bị đòi lại, chúng chỉ có thể được giải phóng bởi chính tiến trình chiếm giữ chúng;
4. *Chờ vòng tròn* (circular wait): các tiến trình giữ tài nguyên và chờ các tài nguyên bị giữ bởi tiến trình khác, tạo thành một chu trình.

Các chiến lược xử lý deadlock được tiếp cận theo các hướng: xem như không có deadlock và không tìm cách xử lý, phát hiện có deadlock và tìm cách khôi phục lại trạng thái như khi chưa có deadlock, phòng tránh tắc nghẽn một cách tự động, và ngăn ngừa hoàn toàn deadlock.

6.1 Bỏ qua deadlock

Phương pháp này còn được gọi là *thuật toán Đà điểu* (Ostrich algorithm) vì tiếp cận theo hướng xem như không có vấn đề gì: deadlock rất hiếm khi xảy ra và nếu xảy ra thì chi phí ngăn ngừa deadlock cũng rất cao. Các hệ điều hành UNIX, Windows tiếp cận theo cách này. Đây là sự đánh đổi giữa công sức sửa chữa (correctness) và sự tiện lợi (convenience).

6.2 Phát hiện và điều chỉnh

Phương pháp này gồm hai bước: đầu tiên là phát hiện xem có deadlock không, khi đã phát hiện thì điều chỉnh theo một số cách:

- Thu hồi (preemption): tùy theo bản chất tài nguyên mà có thể tiến hành thu hồi.
- Quay ngược (rollback): kiểm tra định kỳ các tiến trình và lưu lại trạng thái để khi cần có thể khởi động lại tiến trình.
- Hủy chạy (killing): cách này tuy cứng nhắc nhưng là cách đơn giản nhất để thoát khỏi tình trạng deadlock, khi bắt một tiến trình ngừng chạy thì tiến trình khác có thể lấy tài nguyên của nó, cách này có hiệu quả nếu chọn ra được tiến trình nào mà có thể chạy lại mà không gây lỗi.

Thuật toán phát hiện deadlock: gọi E, A, C, R lần lượt là các ma trận biểu diễn các tình trạng bốn loại tài nguyên *hiện có*, *sẵn dùng*, *cấp phát*, *yêu cầu* với ví dụ như sau:

$$E = \begin{pmatrix} 4 & 2 & 3 & 1 \end{pmatrix} A = \begin{pmatrix} 2 & 1 & 0 & 0 \end{pmatrix} C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{bmatrix} R = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix}$$

Sau vòng lặp thứ nhất sẽ tính ra $A = (2\ 2\ 2\ 0)$, vòng lặp thứ hai $A = (4\ 2\ 2\ 1)$,... và cứ như vậy sẽ không có deadlock.

6.3 Phòng tránh tự động

Cách tiếp cận này sẽ tìm cách loại bỏ các khả năng có thể dẫn đến deadlock. Dựa trên một số khái niệm: *trạng thái an toàn* (safe state) là khi hệ thống tìm cách thỏa mãn các yêu cầu tài nguyên tối đa của các tiến trình mà vẫn không dẫn đến deadlock; theo đó một *chuỗi cấp phát an toàn* có thể được thực hiện: một thứ tự của các tiến trình $\langle P_1, P_2, \dots, P_n \rangle$ là an toàn đối với tình trạng cấp phát hiện hành nếu với mỗi tiến trình P_i nhu cầu tài nguyên của P_i có thể được thỏa mãn với các tài nguyên còn tự do của hệ thống, cộng với các tài nguyên đang bị chiếm giữ bởi các tiến trình P_j khác, với $j < i$. Một *trạng thái an toàn* không thể có deadlock, nhưng một *trạng thái không an toàn* (unsafe) có thể dẫn đến deadlock.

Do vậy hệ thống sẽ dùng các thuật toán xác định trạng thái an toàn để đáp ứng hay không yêu cầu cấp phát tài nguyên của các tiến trình. Thuật toán **Banker** giúp xác định trạng thái là *safe* hay *unsafe*. Ví dụ có năm tiến trình A, B, C, D, E đang tiêu thụ bốn loại tài nguyên, các ma trận $E = (6\ 3\ 4\ 2)$, $P = (5\ 3\ 2\ 2)$, $A = (1\ 0\ 2\ 0)$ biểu diễn tương ứng các tài nguyên *hiện có*, *đang cấp phát*, *sẵn dùng*. Nếu thứ tự chạy các tiến trình lần lượt là D, E, A, B, C thì vector A lần lượt sẽ là $(2\ 1\ 2\ 1)$, $(2\ 1\ 2\ 1)$, $(5\ 1\ 3\ 2)$, $(5\ 2\ 3\ 2)$, $(6\ 3\ 4\ 2)$, và hệ thống luôn ở trạng thái an toàn (safe). Có thể xem thêm hai ma trận bên dưới biểu diễn lượng tài nguyên đã cấp (assigned) và lượng tài nguyên cần thêm (needed).

$$Assigned = \begin{bmatrix} 3 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad Needed = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 2 \\ 3 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 \end{bmatrix}$$

6.4 Ngăn ngừa deadlock

Cách tiếp cận này sẽ tìm cách phủ định một trong bốn điều kiện xảy ra deadlock:

1. *Loại trừ lẫn nhau* (mutual exclusion): spooling giúp tài nguyên có thể được chia sẻ;
2. *Giữ và chờ* (hold and wait): yêu cầu các tài nguyên phải đăng ký trước khi sử dụng;
3. *Không có trưng dụng tài nguyên* (no preemption): cho phép hệ thống được thu hồi tài nguyên từ các tiến trình deadlocked và cấp phát trở lại cho tiến trình khi nó thoát khỏi tình trạng deadlock.
4. *Chờ vòng tròn* (circular wait): đánh số thứ tự các tài nguyên.

Phụ lục A

CÂU HỎI LÝ THUYẾT

CÂU HỎI

Câu 1: Lựa chọn nào sau đây là một cấu phần của hệ điều hành?

- a) Process Management
- b) Space Management
- c) Speed Management
- d) Time Management

Câu 2: Lựa chọn nào sau đây là đặc điểm phù hợp với bộ nhớ truy cập ngẫu nhiên (RAM)?

- a) Can only be read sequentially
- b) Is typically faster than cache memory
- c) Is volatile
- d) Stores all the files on the computer

Câu 3: Lựa chọn nào sau đây là mô tả thích hợp về hệ điều hành?

- a) A program that acts as an intermediary between computer processor and computer memory
- b) A program that acts as an intermediary between a user of a computer and a user of another computer
- c) A program that acts as an intermediary between computer software and computer hardware
- d) A program that acts as an intermediary between a user of a computer and the computer hardware

Câu 4: Lựa chọn nào sau đây là mô tả thích hợp về hệ điều hành nhúng?

- a) Multiple CPU
- b) Time-sharing
- c) Many I/O devices
- d) Restriction of memory size, speed of CPU, screen size, powers

Câu 5: Giá trị nào sau đây là của “mode bit” trong “user mode”?

- a) 0
- b) 1
- c) 00
- d) 11

Câu 6: Lựa chọn nào sau đây **không phải** là một giải thích đúng về hệ điều hành UNIX?

- a) It is a single-user and multi-task OS
- b) Provides an interactive human interface that uses character-based commands.
- c) Provides network functions that easily implement distributed processing.
- d) Since its specifications have been released to the public and it has a high portability, it has been adopted in a wide range of devices.

Câu 7: Chỉ lệnh (instruction) nào sau đây chỉ nên được cho phép ở “user mode”?

- a) Disable all interrupts
- b) Read the time-of-day clock
- c) Set the time-of-day clock
- d) Change the memory map

Câu 8: Lựa chọn nào dưới đây không phải là chức năng chính của một hệ điều hành?

- a) Provide the users with an extended (virtual) machine
- b) Manage the I/O devices
- c) Provide user interfaces
- d) Support virtual memory

Câu 9: Lựa chọn nào dưới đây là sai về các hệ thống timesharing và multiprogramming?

- a) In a timesharing system, multiple users can access the system simultaneously
- b) In a multiprogramming system, one user can run several processes simultaneously
- c) All timesharing systems are multiprogramming systems
- d) All multiprogramming systems are timesharing systems

Câu 10: Lựa chọn nào dưới đây là sự khác nhau giữa máy tính cá nhân và máy tính lớn (mainframe)?

- a) Personal computers are always interactive
- b) Mainframe computers are mostly batch systems with many users
- c) Protection is much more important on mainframe computers
- d) All of the above

Câu 11: Lựa chọn nào dưới đây là một chuyển trạng thái hợp lý?

- a) waiting → running
- b) running → ready
- c) waiting → terminated
- d) ready → terminated

Câu 12: Lựa chọn nào dưới đây không thể được chia sẻ giữa các luồng (thread) của một tiến trình (process)?

- a) File handles
- b) Process data
- c) Process code
- d) Stack

Câu 13: Lựa chọn nào dưới đây phù hợp với thuật toán định thời Shortest Job First?

- a) Avoid Starvation
- b) Maximize average response time
- c) Maximize average waiting time
- d) Minimize average waiting time

Câu 14: Nếu trong một đồ thị cấp phát tài nguyên (Resource Allocation Graph – RAG), mỗi loại tài nguyên chỉ có đúng một instance, lựa chọn nào dưới đây gây ra một tình huống deadlock?

- a) The graph has at least one cycle.
- b) The graph has no cycle.
- c) The graph is connected.
- d) The graph is not connected.

Câu 15: Lựa chọn nào dưới đây là việc thoát tiến trình một cách tự nguyện (voluntary-condition)?

- a) Job error

- b) Killed by another process
- c) Error exit
- d) Killed by user

Câu 16: What is Software proposal in the solution of Mutual exclusion with Busy waiting?

- a) Lock Variables
- b) Strict Alternation
- c) Peterson's Solution
- d) All of the above

Câu 17: Which is not a goal of a scheduling algorithm for batch systems?

- a) Fairness
- b) Throughput
- c) Turnaround time
- d) Response time

Câu 18: Which of the following is a preemptive scheduling algorithm?

- a) FCFS
- b) Shortest Job First
- c) Round Robin
- d) None of the above

Câu 19: Which is a wrong statement about the “*quantum*” used in Round Robin algorithm?

- a) If the quantum is very large, RR is essentially FCFS
- b) If the quantum is very small, the CPU efficiency is reduced
- c) A reasonable value of quantum is around 20-50 ms
- d) None of the above

Câu 20: Preemption is essential in which of the following systems?

- a) Batch system
- b) Interactive system
- c) Real time system
- d) None of the above

Câu 21: The ways to keep track of memory usages is?

- a) Memory Management with Bit Maps
- b) Memory Management with Linked Lists
- c) a and b
- d) None of the above

Câu 22: When a virtual memory system manages memory in fixed length units, which of the following terms correctly represents its unit?

- a) Frame
- b) Page

- c) Sector
- d) Segment

Câu 23: What is not the technique of implementation for Virtual Memory?

- a) Demand segmentation
- b) Demand partition
- c) Demand paging
- d) All of the above

Câu 24: In terms of storage utilization the best method of Dynamic Storage-Allocation is:?

- a) Next fit
- b) First-fit
- c) Best-fit
- d) Worst-fit

Câu 25: Which of the following information bits in the entry of page table is used to indicate Page Fault?

- a) Present/absent bit
- b) Status bit
- c) Referenced bit
- d) Modified bit

Câu 26: Which of the following is not the advantage of segmentation with paging?

- a) User can have a clear logical view of memory.
- b) Different access protections can be associated with different segment of memory.
- c) No external fragmentation.
- d) More efficient in time than pure segmentation and pure paging.

Câu 27: A page fault means that we referenced a page?

- a) outside the memory boundaries
- b) with an incorrect I/O request
- c) that was not in secondary storage
- d) that was not in main memory

Câu 28: Which of the following is appropriate to determine program size and create page table?

- a) Process creation
- b) Process execution
- c) Page fault time
- d) Process termination time

Câu 29: Which of the following information bits used by the various page replacement policies indicates if the page has been called lately?

- a) Locality bit
- b) Status bit
- c) Referenced bit

- d) Modified bit

Câu 30: Which of the following is appropriate to release page table and pages?

- a) Process creation
- b) Process execution
- c) Page fault time
- d) Process termination time

Câu 31: How does Windows 98 store long filenames using directory entries?

- a) By having 32-byte length entries
- b) By having 64-byte length entries
- c) By using possibly more than one entries per name
- d) By using a heap to store variable length filenames

Câu 32: Which of the following is specified to indicate the directory where the file is located?

- a) Extension
- b) Path
- c) Root directory
- d) Sub-directory

Câu 33: Which of the following is true about the block size is disk space management?

- a) the larger the block size is the lower the data rate is
- b) the larger the block size is the worse the disk space utilization is
- c) none of the above
- d) the larger the block size is lesser the disk space is

Câu 34: The i-nodes are used in which of the following allocation methods?

- a) Contiguous allocation
- b) Linked allocation
- c) Indexed allocation
- d) Linked allocation using FAT

Câu 35: What is correct about contiguous allocation of files?

- a) It is simple to implement
- b) It leads to excellent read performance
- c) It does not cause disk fragmentation
- d) It is widely used on CD-ROMs

Câu 36: Which of the following is not correct about hard links and symbolic links?

- a) Symbolic links need space to store the name and the file pointed to
- b) Hard links do not require extra disk space
- c) Symbolic links can point to files in the network
- d) Hard links can point to files on other machines

- Câu 37:** If i-node contains 10 direct addresses of 4 byte and all disk blocks are 1024 KB, what is largest possible file?
- a) 10 MB
 - b) 100 MB
 - c) 1GB
 - d) 10 GB
- Câu 38:** File Structure can be:?
- a) byte sequence
 - b) record sequence
 - c) tree
 - d) All of the above
- Câu 39:** The special files are:?
- a) character special file
 - b) block special file
 - c) None of a. and b.
 - d) Both a. and b
- Câu 40:** What are the allocation methods of disk blocks for files:?
- a) Contiguous allocation
 - b) Linked allocation
 - c) Indexed allocation
 - d) All of the above
- Câu 41:** Which of the following actions generates an external interrupt?
- a) An input/output operation is completed.
 - b) A page that does not exist in the main memory is accessed by the virtual storage management.
 - c) A system call instruction is executed.
 - d) Division by zero occurs.
- Câu 42:** Which is the right order between the 4 I/O software layers?
- a) User-level I/O software, Device drivers, Interrupt handlers, Device-independent OS software
 - b) User-level I/O software, Interrupt handlers, Device drivers, Device-independent OS software
 - c) Device-independent OS software, user-level I/O software, Device drivers, Interrupt handlers
 - d) User-level I/O software, Device-independent OS software, Device drivers, Interrupt handlers
- Câu 43:** In terms of speed the best technique of I/O Data transfer is:?
- a) Programmed I/O
 - b) Interrupt-Driven I/O
 - c) Direct Memory Access
 - d) None of the above
- Câu 44:** Device Driver is normally written by:?

- a) Device's Manufacturer
- b) OS's Manufacturer
- c) Computer's Manufacturer
- d) All of the above

Câu 45: For device-independent I/O software with double buffering in the kernel, how many buffers are really used:?

- a) 3
- b) 2
- c) 1
- d) 0

Câu 46: The functions of Device Driver are:?

- a) To accept abstract Read or Write request from device- independent software
- b) To initialize device
- c) To manage power requirement and log-events
- d) All of the above

Câu 47: Which of the following statements is not correct about DMA?

- a) DMA controller has access to the system bus independent of the CPU
- b) DMA helps reduce the number of interrupts (in comparison with interrupt-driven I/O)
- c) DMA controller is usually faster than CPU
- d) The operating system can only use DMA if the hardware has a DMA controller

Câu 48: Which of the following statements is not correct about “device independence”?

- a) Files and devices are accessed in the same way, independent of their physical nature
- b) A system has to maintain only one set of system calls for both writing on a file and writing on the console
- c) Device independence requires all programmers to deal with different devices directly
- d) Device independent interfaces should be given to programmers

Câu 49: Which of the following I/O software device layers is done by user-level software?

- a) Computing the track, sector, and head for a disk read
- b) Writing commands to the device registers
- c) Checking to see if the user is permitted to use the device
- d) Converting binary integers to ASCII for printing

Câu 50: Which of the following is correct about optical disks?

- a) They have higher recording density than that of magnetic disks
- b) They are faster than magnetic disks
- c) They have only one standard
- d) They are much more expensive than magnetic disks

Câu 51: The simplest way to break a deadlock is to?

- a) preempt a resource
- b) rollback
- c) kills one of the processes
- d) locks one of the processes

Câu 52: What is the correct approach with the Hold and Wait condition to prevent Deadlock?

- a) Order resources numerically
- b) Request all resources initially
- c) Spool everything
- d) Take resources away

Câu 53: What is the correct approach with the Mutual Exclusion condition to prevent Deadlock?

- a) Order resources numerically
- b) Request all resources initially
- c) Spool everything
- d) Take resources away

Câu 54: If in a resource-allocation graph, each resource type has exactly one instance, which of the following indicate a deadlock situation?

- a) The graph has at least one cycle.
- b) The graph has no cycle.
- c) The graph is connected.
- d) The graph is not connected.

Câu 55: The simplest way to break a deadlock is to?

- a) preempt a resource
- b) rollback
- c) kill one of the processes
- d) lock one of the processes

Câu 56: What is the correct approach with the circular wait condition to prevent Deadlock?

- a) Order resources numerically
- b) Request all resources initially
- c) Spool everything
- d) Take resources away

Câu 57: If system is in Deadlock, the processes can...?

- a) run
- b) release resources
- c) be awakened
- d) do nothing of above actions

Câu 58: A possibility of deadlock can occur:?

- a) If a system is in safe state

- b) If a system is in unsafe state
- c) If a system is in instable state
- d) None of the above

Câu 59: What is a correct sequence of events required to use a resource?

- a) release the resource, request the resource, use the resource
- b) use the resource, release the resource, request the resource
- c) release the resource, use the resource, request the resource
- d) request the resource, use the resource, release the resource

Câu 60: A job may be in starvation if algorithm to allocate a resource is:?

- a) shortest job first
- b) First-come, first-serve
- c) None of the above
- d) Both of the above

ĐÁP ÁN

- Câu 1:** a) Process Management.
- Câu 2:** c) Is volatile.
- Câu 3:** a) A program that acts as an intermediary between computer processor and computer memory.
- Câu 4:** d) Restriction of memory size, speed of CPU, screen size, powers.
- Câu 5:** b) 1.
- Câu 6:** a) It is a single-user and multi-task OS.
- Câu 7:** b) Read the time-of-day clock.
- Câu 8:** c) Provide user interfaces.
- Câu 9:** d) All multiprogramming systems are timesharing systems.
- Câu 10:** d) All of the above.
- Câu 11:** b) running → ready.
- Câu 12:** d) Stack.
- Câu 13:** d) Minimize average waiting time.
- Câu 14:** a) The graph has at least one cycle.
- Câu 15:** c) Error exit.
- Câu 16:** d) All of the above.
- Câu 17:** d) Response time.
- Câu 18:** c) Round Robin.
- Câu 19:** d) None of the above.
- Câu 20:** b) Interactive system.
- Câu 21:** c) a and b.
- Câu 22:** b) Page.
- Câu 23:** b) Demand partition.
- Câu 24:** c) Best-fit.
- Câu 25:** a) Present/absent bit.
- Câu 26:** b) Different access protections can be associated with different segment of memory.
- Câu 27:** d) that was not in main memory.
- Câu 28:** a) Process creation.
- Câu 29:** c) Referenced bit.
- Câu 30:** d) Process termination time.
- Câu 31:** c) By using possibly more than one entries per name.
- Câu 32:** b) Path.
- Câu 33:** b) the larger the block size is the worse the disk space utilization is.
- Câu 34:** c) Indexed allocation.
- Câu 35:** c) It does not cause disk fragmentation.
- Câu 36:** d) Hard links can point to files on other machines.
- Câu 37:** a) 10 MB.
- Câu 38:** a) All of the above.
- Câu 39:** d) Both a. and b.
- Câu 40:** d) All of the above.
- Câu 41:** a) An input/output operation is completed.
- Câu 42:** d) User-level I/O software, Device-independent OS software, Device drivers, Interrupt handlers.
- Câu 43:** c) Direct Memory Access.
- Câu 44:** a) Device's Manufacturer.

- Câu 45:** a) 3.
- Câu 46:** d) All of the above.
- Câu 47:** c) DMA controller is usually faster than CPU.
- Câu 48:** c) Device independence requires all programmers to deal with different devices directly.
- Câu 49:** d) Converting binary integers to ASCII for printing.
- Câu 50:** a) They have higher recording density than that of magnetic disks.
- Câu 51:** c) kills one of the processes.
- Câu 52:** b) Request all resources initially.
- Câu 53:** c) Spool everything.
- Câu 54:** a) The graph has at least one cycle.
- Câu 55:** c) kill one of the processes.
- Câu 56:** a) Order resources numerically.
- Câu 57:** d) do nothing of above actions.
- Câu 58:** b) If a system is in unsafe state.
- Câu 59:** d) request the resource, use the resource, release the resource.
- Câu 60:** a) shortest job first.