

Danh mục câu hỏi và Hướng dẫn cho điểm thi môn học
LÍ THUYẾT HỆ ĐIỀU HÀNH

Cấu trúc Đề thi:

Đề thi gồm 2 phần: Lí thuyết (50%) gồm 2 câu hỏi được chọn ngẫu nhiên từ 2 nhóm câu hỏi, mỗi nhóm 1 câu, và phần Bài tập (50%) được chọn ngẫu nhiên từ nhóm câu hỏi bài tập.

Phần 1: Lí thuyết

Nhóm 1:

Câu 1: Ý nghĩa của System Call. Hãy giải thích bình thường các phần mềm ứng dụng “khai thác” phần cứng của máy như thế nào?

- *Nêu đúng ý nghĩa (và định nghĩa) của System Call: System call cung cấp giao diện giữa một tiến trình và HĐH. Các system call thường sẵn dùng như các chỉ thị hợp ngữ, có thể được sử dụng trong các ngôn ngữ lập trình, thường được liệt kê trong các tài liệu hướng dẫn sử dụng. (3 điểm)*
- *API: Các system calls thuộc các HĐH windows hình thành nên Application Programmer Interface – API (2 điểm)*
- *Nêu các cách Truyền tham số khi gọi hàm hệ thống (5 điểm): Có ba phương pháp thông dụng để truyền tham số tới hệ điều hành. Phương pháp đơn giản nhất là truyền tham số trong các thanh ghi. Trong một vài trường hợp, các tham số thường lưu trữ trong một khối hay bảng trong bộ nhớ và địa chỉ của khối được truyền như một tham số trong thanh ghi. Các tham số cũng có thể được thay thế, hay được đẩy vào trong ngăn xếp bởi chương trình, và được lấy ra khỏi ngăn xếp bởi hệ điều hành. Một vài hệ điều hành dùng phương pháp khối hay ngăn xếp vì các phương pháp này không giới hạn số lượng hay chiều dài của tham số đang được truyền*

Câu 2: Một tiến trình yêu cầu Hệ điều hành chuyển trạng thái cho nó “ngủ” 5 giây. Hệ điều hành có đảm bảo được là sẽ kích hoạt tiến trình này đúng 5 giây sau đó không? Tại sao?

- *Giải thích OS sau khi chuyển Process từ trạng thái Running sang Block cho Proces ngủ 5 giây = cách thiết lập một đồng hồ báo thức (alarm clock) 5 giây để cho Proces ngủ. Giải thích khi clock báo động thông qua ngắt sau đúng 5 giây. Về nguyên tắc Ok với mô hình nào và không với mô hình nào (5 đ)*
- *Khi có tín hiệu ngắt đồng hồ, Process được chuyển sang trạng thái Ready và có thể được phục vụ. Giải thích trong trường hợp có có hệ thống mức độ ưu tiên khác nhau sẽ cho kết quả việc chờ của Process là đúng hay không (5đ)*

Câu 3: Nêu định nghĩa “Hệ điều hành”. Hệ điều hành thực hiện những chức năng chính gì?

- *Nêu được đúng Định nghĩa: HĐH là một chương trình quản lý phần cứng máy tính. Nó cung cấp nền tảng cơ sở cho các chương trình ứng dụng và hoạt động như một đối tượng trung gian giữa người sử dụng máy tính phần cứng máy tính (2,5đ)*

- *Nếu chức năng che phần cứng:* "Đồng nhất" hóa phần cứng đối với người sử dụng, (2,5 đ)
- *Nếu chức năng Abstract service:* Người sử dụng bao gồm cả người lập trình thay vì sử dụng các lệnh truy cập trực tiếp tới các địa chỉ phần cứng cụ thể, sẽ sử dụng khai thác chức năng dịch vụ của HĐH (2,5đ)
- *Nếu chức năng Quản lý cấp phát tài nguyên hoặc tối ưu hóa khai thác tài nguyên :* Bao gồm CPU, bộ nhớ các thiết bị ngoại vi(2,5 đ)
- *Nếu nếu được đúng các chức năng khác có thể cho 1 điểm khi chưa nêu rõ được các chức năng trên:* Cấp phát tài nguyên cho nhiều người dùng, bảo vệ...

Câu 4: Multi-tasking là gì? Giải thích sự khác nhau giữa các hình thức Multi-tasking (cooperative and pre-emptive multi-tasking).

- *Giải thích Multitasking:* Mô hình hệ thống đa nhiệm, hệ thống thực thi nhiều nhiệm vụ cùng lúc – trong khoảng thời gian nhất định (2,5đ)
- *Nhưng giải thích được bản chất của việc switching giữa các processes* (2,5đ)
- *M-Preemptive:* Giải thích được bản chất không tự nguyện chuyển trạng thái Running; OS theo định kỳ cưỡng bức running process sang trạng thái Ready to Run, hoặc trong mô hình có cho phép cấp phát chế độ ưu tiên (2,5đ)
- *M-cooperative:* Giải thích được bản chất và nêu sự khác biệt với Pre-emptive Multitasking; Process thông báo hoặc yêu cầu chuyển trạng thái. (2,5đ)

Câu 5: Sự giống nhau và khác nhau giữa Chương trình, Tiến trình và Luồng (Program, Process and Thread).

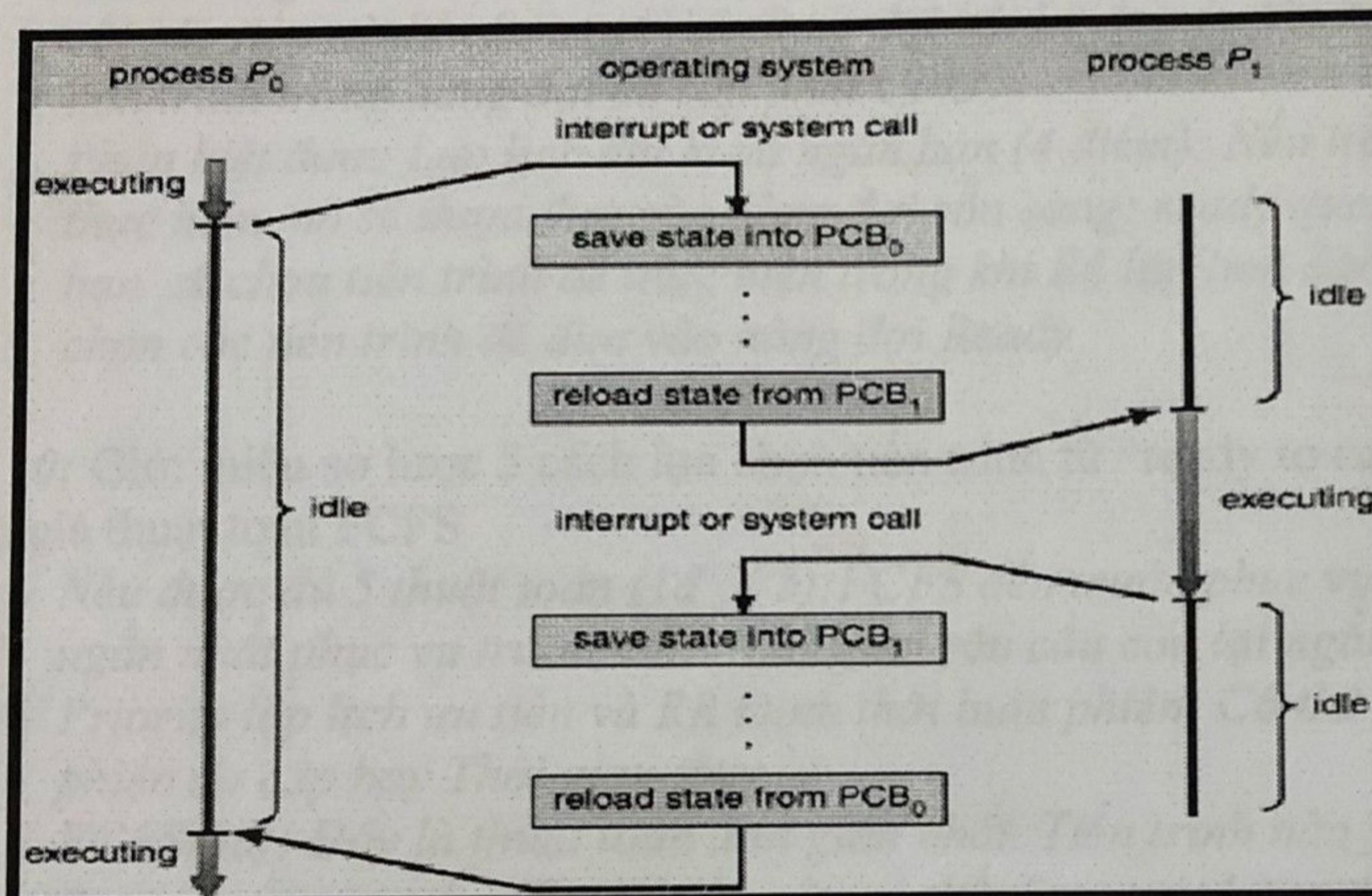
- *Phân biệt được:* Luồng (thread) là đơn vị con bên trong một process mà chúng có thể độc lập tương đối với nhau. Chúng có các thanh ghi và con trỏ lệnh riêng và có thể được lập lịch thực hiện độc lập NHƯNG các threads trong một process có chung cùng với process các thuộc tính như: không gian địa chỉ, chung các tài nguyên mở sẵn sang cho chúng. (5đ)
- *Phân biệt được Proceses với Program:* Đều là tập hợp các instruction, chương trình tổng quát hơn khi có thể ở trạng thái passive, lưu trữ ... trong khi tiến trình là chương trình hoặc phần chương trình active đang được nạp trong bộ nhớ trong của máy (5đ)
- *Nếu nếu được ví dụ đúng (2 đ)(Yêu cầu thêm nếu trả lời chưa rõ lắm các ý trên)*

Câu 6: So sánh ưu và nhược điểm giữa Hệ điều hành client-server và hệ điều hành đơn khối (monolithic) truyền thống.

- *OS đơn khối không phải switch, không phải gửi các message giữa các procedure - > có thể tăng được hiệu suất, nhanh hơn một chút (Có thể nêu tính chất của các procedure của OS truyền thống; ngang hàng, nhìn và gọi lẫn nhau) (5đ)*
- *Client-server OS lợi thế: do chia thành các phần với các chức năng phục vụ khác nhau như File service, Terminal service hay Memory service ... nên khi có một service nào đó có vấn đề thì các service khác vẫn hoạt động (2,5đ)*
- *Có thể sẵn sàng cho xây dựng hệ thống phân tán (distributed system) (2,5đ)*

Câu 7: Hãy giải thích cơ chế CPU chuyên từ việc thực hiện process này sang process khác. Làm sao có thể đảm bảo được việc thực hiện đúng các lệnh của process tương ứng?

- **Giải thích được context switching:** Chuyển CPU tới một quá trình khác yêu cầu lưu trạng thái của quá trình cũ và nạp trạng thái được lưu cho quá trình mới. **Ngữ cảnh** của quá trình được hiện diện trong PCB của quá trình; Nó chứa giá trị các thanh ghi, trạng thái quá trình và thông tin quản lý bộ nhớ. Khi chuyển ngữ cảnh ngữ cảnh xảy ra, nhân lưu ngữ cảnh của quá trình cũ trong PCB của nó và nạp ngữ cảnh được lưu của quá trình mới được định thời để chạy. Thời gian chuyển ngữ cảnh là chi phí thuần vì hệ thống không thực hiện công việc có ích trong khi chuyển. Tốc độ của nó khác từ máy này tới máy khác phụ thuộc vào tốc độ bộ nhớ, số lượng thanh ghi phải được chép và sự tồn tại của các chỉ thị đặc biệt: (2đ)
- **Save PCB và Load PCB tương ứng (1,5 đ x 2: Trình bày và Giải thích)**



- **Giải thích các thành phần và ý nghĩa chính của PCB, đặc biệt là PC- Thanh ghi địa chỉ chương trình:** Mỗi PCB của tiến trình chứa nhiều phần thông tin được gắn liền với tiến trình xác định, gồm: Trạng thái, bộ đếm chương trình, các thanh ghi, thông tin về throgram biểu, thông tin về bộ nhớ, thông tin tính toán, thông tin trạng thái xuất nhập. Đặc biệt thông tin về PC đảm bảo cho tiến trình được “liên tục” được thực hiện lệnh tương ứng sau khi phải nhường CPU cho tiến trình khác (3+2 đ)

Câu 8: Mục đích và nguyên tắc của việc lập lịch cho CPU (Process Scheduling)

- **Nêu được mục đích:** Chuyển CPU cấp phát cho các tiến trình khác nhau trong mô hình đa trinh đa nhiệm; tối ưu hóa CPU; Công bằng tương đối đối với các tiến trình/ các người sử dụng đối với hệ đa người sử dụng → lựa chọn tiến trình trong hàng đợi để cấp phát CPU (3 đ) **0,25**
- Chính là cấp phát, lựa chọn tiến trình để phân bổ CPU đối với các mô hình HĐH khác nhau, các thuật giải định thời khác nhau có các thuộc tính khác nhau và “thiên vị” với một loại tiến trình nào đó → Lựa chọn thuật toán phù hợp với các mục tiêu cụ thể trong yêu cầu cụ thể căn cứ vào (trọng số) các tiêu chí sau (2 đ) **0,5**
- **Nêu được 5 tiêu chí của việc lập lịch:** việc sử dụng CPU, Thông lượng tức là số lượng tiến trình thông qua trong 1 đơn vị thời gian, Thời gian hoàn thành là tổng thời gian chờ + thực thi và thực hiện xuất nhập, Thời gian chờ để được thực hiện, Thời gian đáp ứng tức là T được phục vụ lần đầu (1 đ x5) **1,25**

Câu 9: Ý nghĩa căn bản của Lập lịch, Sự khác nhau giữa Lập lịch dài kì và lập lịch ngắn hạn?

- Giải thích khái niệm lập lịch (3 điểm) Một tiến trình khi gia nhập hệ thống, sẽ được đưa vào các hàng đợi tương ứng và có thể có nhiều tiến trình nằm trong các hàng đợi khác nhau. Các tiến trình sẽ chuyển từ hàng đợi này sang hàng đợi khác và thực tế tại 1 thời điểm nhất định chỉ một tiến trình được thực hiện tức là được cấp phát CPU → vấn đề là chọn tiến trình nào để thực hiện/ cấp CPU hay chuyển hàng đợi
- Mục đích của Lập lịch (3 điểm): Chuyển CPU cấp phát cho các tiến trình khác nhau trong mô hình đa trinh đa nhiệm; tối ưu hóa CPU; Công bằng tương đối đối với các tiến trình/ các người sử dụng đối với hệ đa người sử dụng → lựa chọn tiến trình trong hàng đợi để cấp phát CPU
- Phân biệt được Lập lịch dài kì và ngắn hạn (4 điểm): Nếu tiến trình sẵn sàng để thực hiện, nó sẽ được đưa vào hàng đợi sẵn sàng: ready queue và Lập lịch ngắn hạn sẽ chọn tiến trình để thực hiện trong khi Bộ lập lịch dài hạn (long term) sẽ chọn các tiến trình để đưa vào hàng đợi Ready

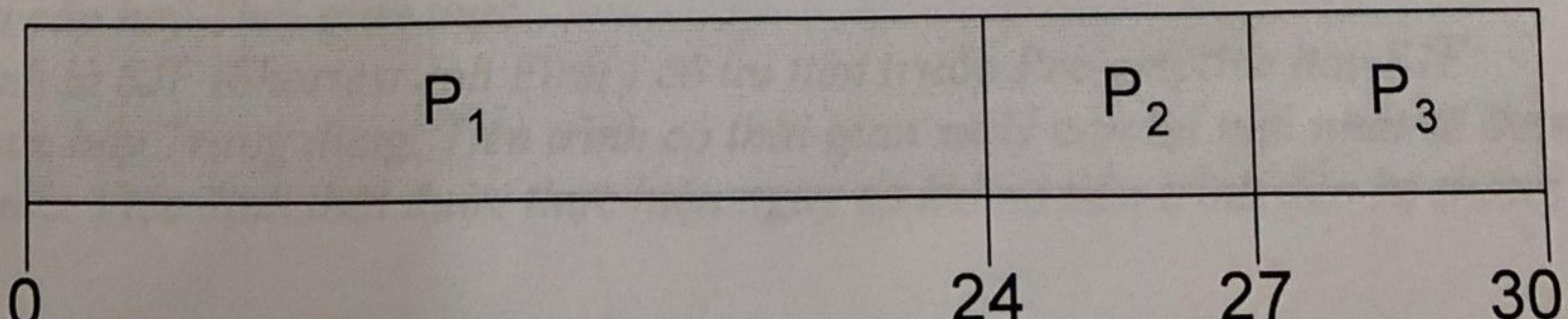
Câu 10: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện.

Đánh giá thuật toán FCFS

- Nếu được đủ 5 thuật toán (1đ X 5): FCFS đến trước phục vụ trước, SJF yêu cầu ngắn nhất phục vụ trước, SRF thời gian yêu cầu còn lại ngắn nhất phục vụ trước, Priority lập lịch ưu tiên và RR Định thời luân phiên; Có thể nêu định thời luân phiên đa cấp hay Thời gian thực...
- FCFS(5đ): Đây là thuật toán đơn giản nhất. Tiến trình nào yêu cầu CPU trước sẽ được phục vụ trước. Thuật toán này có thể dùng một hàng đợi kiểu FIFO. Khi một tiến trình ra nhập hàng đợi sẵn sàng, PCB của nó sẽ được liên kết tới đuôi của hàng đợi. Khi CPU rỗi, nó sẽ được cấp phát một tiến trình ở đầu hàng đợi, và tiến trình đang chạy sẽ được xoá khỏi hàng đợi. Điều kiện đánh giá thường là thời gian đợi trung bình.

<u>Process</u>	<u>Burst Time</u>
P1	24
P2	3
P3	3

- Giả sử các process đến theo thứ tự P1 , P2 , P3
- Giải đố Gantt cho việc định thời là:



- Thời gian đợi cho $P_1 = 0$, $P_2 = 24$, $P_3 = 27$

- Thời gian đợi trung bình: $(0 + 24 + 27)/3 = 17$

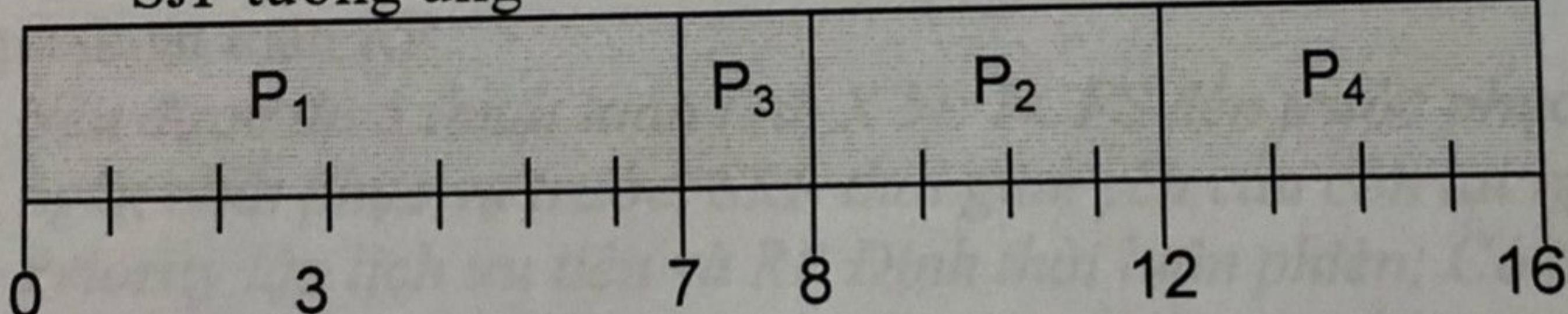
Câu 11: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện.

Đánh giá thuật toán SJF

- Nếu được đủ 5 thuật toán (1đ X 5): FCFS đến trước phục vụ trước, SJF yêu cầu ngắn nhất phục vụ trước, SRF thời gian yêu cầu còn lại ngắn nhất phục vụ trước, Priority lập lịch ưu tiên và RR Định thời luân phiên; Có thể nêu định thời luân phiên đa cấp hay Thời gian thực...
- Thuật toán SJF: định thời công việc ngắn nhất trước (shortest-job-first-SJF). Thuật toán này gán tới mỗi tiến trình chiều dài của chu kỳ CPU tiếp theo cho tiến trình sau đó. Khi CPU rỗi, nó được gán tới tiến trình có chu kỳ CPU kế tiếp ngắn nhất. Nếu hai tiến trình có cùng chiều dài chu kỳ CPU kế tiếp, định thời FCFS được dùng. Chú ý rằng thuật ngữ phù hợp hơn là chu kỳ CPU kế tiếp ngắn nhất được dùng. Chú ý rằng thuật ngữ phù hợp hơn là chu kỳ CPU kế tiếp ngắn nhất (shortest next CPU burst) vì định thời được thực hiện bằng cách xem xét chiều dài của chu kỳ CPU kế tiếp của tiến trình hơn là toàn bộ chiều dài của nó

Process	Thời điểm đến Burst Time	
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

SJF tương ứng



Thời gian đợi trung bình

$$(0 + 6 + 3 + 7)/4 = 4$$

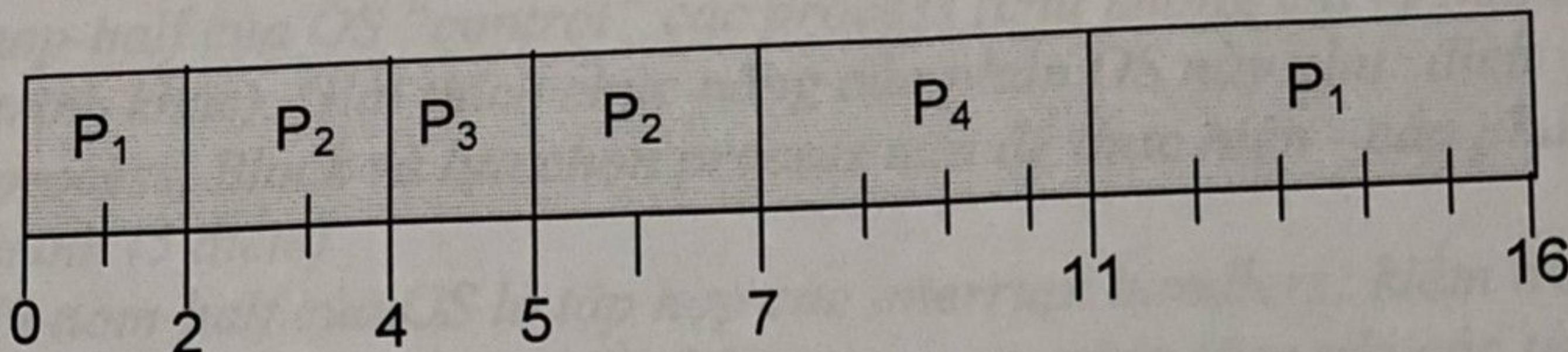
Câu 12: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện.

Đánh giá thuật toán SRF

- Nếu được đủ 5 thuật toán (1đ X 5): FCFS đến trước phục vụ trước, SJF yêu cầu ngắn nhất phục vụ trước, SRF thời gian yêu cầu còn lại ngắn nhất phục vụ trước, Priority lập lịch ưu tiên và RR Định thời luân phiên; Có thể nêu định thời luân phiên đa cấp hay Thời gian thực...
- SRT chính là SJF (Shortest Job First) có ưu tiên trước Pre-emptive hay SJF cưỡng bức hay Trung dung. Tiến trình có thời gian xử lý còn lại nhỏ nhất sẽ được xử lý trước. Việc định thời được thực hiện ngay cả khi có tiến trình đến hệ thống

<u>Process</u>	<u>Thời điểm đến</u>	<u>Burst Time</u>
P_1	0.0	7
P_2	2.0	4
P_3	4.0	1
P_4	5.0	4

- Shortest-Remaining-Time-First (SRTF)



- Thời gian đợi trung bình = $(9 + 1 + 0 + 2)/4 = 3$
Tốt hơn giải thuật nonpreemptive SJF

Câu 13: Giới thiệu sơ lược 5 cách lựa chọn tiến trình từ “ready to run” để thực hiện.

Đánh giá thuật toán RR

- Nếu được đủ 5 thuật toán (1đ X 5): FCFS đến trước phục vụ trước, SJF yêu cầu ngắn nhất phục vụ trước, SRF thời gian yêu cầu còn lại ngắn nhất phục vụ trước, Priority lập lịch ưu tiên và RR Định thời luân phiên; Có thể nếu định thời luân phiên đa cấp hay Thời gian thực...
- Round Robin là bộ lập lịch theo kiểu ưu tiên trước pre-emptive. Thuật toán này được dùng cho các hệ thống chia sẻ thời gian. Một khoảng thời gian nhỏ gọi là time quantum, được xác định. Thông thường nó từ 10 đến 100 mili giây, hàng đợi tiến trình sẵn sàng được dùng như một hàng đợi vòng tròn. Bộ lập lịch CPU sẽ đi vòng quanh hàng đợi, cấp phát CPU cho mỗi tiến trình với một khoảng thời gian là 1 quantum. Nếu CPU burst của tiến trình nhỏ hơn 1 quantum, thì tiến trình sẽ tự giác trả lại CPU ngay lập tức. Ngược lại, nếu lớn hơn tiến trình sẽ bị ngắt.

Tiến trình	CPU burst (ms)
P1	24
P2	3
P3	3

$$\text{Time quantum} = 4 \text{ ms}$$

Khi đó ta có:

P1	P2	P3	P1	P1	P1	P1	P1
1-4	5-7	8-10				11-30	

Thời gian chờ đợi trung bình:

$$T = (6+4+7)/3 = 5.66$$

Câu 14: Giải thích tại sao phần địa chỉ cao (top-half) của hệ điều hành không phải là tiến trình (process). Giải thích tại sao phần địa chỉ thấp (bottom-half) của hệ điều hành cũng không phải là tiến trình (process).

- *top-half của OS “control” các process (chứ không chỉ là tiến trình như các tiến trình khác). Giải thích chức năng của phần OS này như: dịch vụ I/O thay cho các process, Block và lựa chọn process mới để thực hiện - cấp phát CPU cho tiến trình (5 điểm)*
- *bottom-half của OS là tập hợp các interrupt handlers; kiểm tra trạng thái các device controller, trao đổi thông tin ... -> cộng tác với các tiến trình, đồng hành cung cấp các dịch vụ cho tiến trình (thay vì là một tiến trình) (5 điểm)*

Câu 15: Giải thích ngắn gọn trạng thái của các tiến trình, các mô hình tiến trình hệ thống (process models) chung.

- *Giải thích đúng mỗi trạng thái: Creation – Khởi tạo: tiến trình được active nạp vào bộ nhớ; Ready – sẵn sàng chờ được thực hiện tức là chờ được cấp CPU để thực hiện instruction tiếp theo; Blocked – chẵn hay chờ là trạng thái bị tạm ngừng thực hiện do yêu cầu hay chế độ đặt lịch của HDH đòi với CPU hay tiến trình chờ sự kiện nào đó liên quan tới tác nghiệp ngoại vi; Running – trạng thái tiến trình đang chạy được thực hiện bởi CPU; Termination là trạng thái kết thúc hoàn thành việc thực hiện và sẽ giải phóng các tài nguyên chiếm dụng bởi tiến trình (1đ X5)*
- *Giới thiệu đúng 1 mô hình: Mô hình simple monitor, Mô hình batch system, Mô hình batch system với blocked state và mô hình Interactive Multiprogramming (5đ)*

Câu 16: Giải thích vai trò của PCB (process control block).

- *Nêu được thành phần cần lưu chính của PCB (5 đ):*

- **Trạng thái tiến trình (process state):** trạng thái có thể là mới, sẵn sàng, đang chạy, chờ đợi, kết thúc, ...
- **Bộ đếm chương trình (program counter):** bộ đếm hiển thị địa chỉ của chỉ thị kế tiếp được thực thi cho quá trình này.
- **Các thanh ghi (registers) CPU:** các thanh ghi khác nhau về số lượng và loại, phụ thuộc vào kiến trúc máy tính. Chúng gồm các accumulators, các thanh ghi chỉ mục, các con trỏ ngăn xếp, và các thanh ghi đa năng (general-purpose registers), cùng với thông tin mã điều kiện (condition-code information).
- **Thông tin lập thời biểu CPU (CPU-scheduling information):** thông tin gồm độ ưu tiên của quá trình, các con trỏ chỉ tới các hàng đợi lập thời biểu, và bất kỳ tham số lập thời biểu khác.
- **Thông tin quản lý bộ nhớ (Memory-management information):** thông tin này có thể gồm những thông tin như giá trị của các thanh ghi nền và thanh ghi giới hạn, các

bảng trang hay các bảng phân đoạn, phụ thuộc hệ thống bộ nhớ được dùng bởi hệ điều hành.

- **Thông tin tính toán** (accounting information): thông tin này gồm lượng CPU và thời gian thực được dùng, công việc hay số quá trình,...
- **Thông tin trạng thái nhập/xuất** (I/O status information): thông tin này gồm danh sách của thiết bị nhập/xuất được cấp phát qua trình này, một danh sách các tập tin đang mở

- Nếu được ý nghĩa, nhằm đảm bảo tính “liên tục” của tiến trình (3 đ): Bộ đệm chương trình/thanh ghi chỉ mục phải được lưu khi một ngắt xảy ra, cho phép quá trình được tiếp tục một cách phù hợp sau đó, thanh ghi này là thành phần quan trọng của PCB.
- **Qua việc Load and Save PCB tương ứng** (2đ): Toàn bộ nội dung PCB của tiến trình đang chiếm dụng CPU được lưu trữ (Save) và các thông tin PCB của tiến trình mới được tải ra (Load) tức là được nạp vào các thanh ghi, thành phần tương ứng của CPU, bộ nhớ....

Câu 17: Giới thiệu và giải thích ngắn gọn các thành phần (components) của Hệ điều hành.

- **Nếu được 4 thành phần** (2đ x 4): Quản lý tiến trình/ cấp phát CPU cho các tiến trình, Quản lý Bộ nhớ - cấp phát Bộ nhớ cho từng tiến trình và quản lý không gian bộ nhớ chính, Quản lý Ngoại vi và Nhấn mạnh đến quản lý Bộ nhớ ngoài mà đặc biệt là ổ đĩa cứng:
 - Quản lý tiến trình bao gồm:** Tạo và xoá tiến trình người sử dụng cũng như tiến trình hệ thống, Dừng hoặc tiếp tục thực hiện tiến trình tức là lựa chọn sắp xếp các tiến trình vào các hàng đợi và lựa chọn tiến trình để cấp phát CPU, đồng bộ các tiến trình...
 - Quản lý bộ nhớ bao gồm:** Các công việc quản lý bộ nhớ liên quan, Theo dõi xem các phần nào của bộ nhớ đang bị sử dụng và ai đang sử dụng, Quyết định xem tiến trình nào sẽ được nạp vào bộ nhớ khi bộ nhớ rỗng, Cấp phát và thu hồi bộ nhớ khi cần thiết
 - Quản lý Ngoại vi:** Một trong những chức năng chính của HĐH là giấu những sự phức tạp và riêng biệt của thiết bị, tạo ra sự dễ sử dụng nhất đối với users. Mỗi phần điều khiển ngoại vi thường chứa một phần bộ nhớ gọi là buffer, nó thường có thành phần quản lý buffering, caching, spooling; chứa giao diện chung với các điều khiển thiết bị, phần điều khiển cho các thiết bị đặc biệt.
 - Quản lý Bộ nhớ ngoài:** Với hệ thống hiện đại ổ đĩa cứng là những thành phần tối quan trọng. Trong quá trình vận hành đĩa cứng không những dùng để lưu trữ các tệp mà còn được dùng như làm một phần của bộ nhớ chính; HĐH đảm đương việc quản lý vùng đĩa trống, Cấp phát các vùng đĩa và Lập lịch cho đĩa
- **Các thành phần khác**(2 đ): Hệ thống Bảo vệ, Thành phần biên dịch, Thành phần quản lý Tệp và Phần Quản lý kết nối mạng

Câu 18: Nếu vắn tắt các kĩ thuật cấp phát bộ nhớ (nạp chương trình vào bộ nhớ).

- Nếu được 3 phương cách, mỗi cách 3 điểm (3đ x 3 = 9đ):

Phân vùng cố định (Fixed partitioning): Bộ nhớ chia thành các phân vùng, các tiến trình được nạp vào từng vùng => xuất hiện phân mảnh nội, yêu cầu và quản lý overlay (3đ)

Phân vùng động (Dynamic partitioning): Số lượng và kích thước các vùng trên bộ nhớ có thể thay đổi. Cấp phát cho mỗi tiến trình số lượng bộ nhớ vừa đủ. Tạo ra các khoảng trống trong bộ nhớ, cấp phát cho các tiến trình tiếp theo vào các khoảng trống đó => xuất hiện phân mảnh ngoại, yêu cầu quản lý các vùng nhớ trống và lựa chọn khoảng trống cấp phát cho tiến trình mới (3đ).

Phân trang và phân đoạn đơn giản: Bộ nhớ chia thành các trang cố định có kích thước bằng nhau và các tiến trình được cấp phát một số lượng trang nhớ đủ lớn để nạp chương trình. Bộ nhớ chia thành các đoạn cố định có độ lớn không bằng nhau và các tiến trình được cấp phát một số đoạn tương ứng. Mỗi tiến trình được cấp phát nhiều vùng nhớ và không nhất thiết phải liền nhau => phân mảnh nội (3đ)
- Nếu được ý tổng hợp giữa 3 cách (1 điểm)

Câu 19. Trong kỹ thuật cấp phát bộ nhớ phân vùng động, khi nạp tiến trình mới vào bộ nhớ cần lựa chọn một vùng còn “rỗng” nào đó. Nếu các phương pháp/thuật toán lựa chọn.

Nguyên tắc và mục đích chung là *Cấp phát vùng nhớ trống nào cho tiến trình đang yêu cầu sao cho phân mảnh ngoại là nhỏ nhất.*

1. Best-fit: chọn khối nhớ trống nhỏ nhất. Yêu cầu trình bày ví dụ
2. First-fit: chọn khối nhớ trống phù hợp đầu tiên kể từ đầu bộ nhớ. Yêu cầu trình bày ví dụ
3. Worst fit: chọn khối nhớ trống lớn nhất. Yêu cầu trình bày ví dụ

Mỗi thuật toán trên 3đ, nếu ý nguyên tắc và/hoặc nếu thêm thuật toán 4 cho 1 điểm

Next-fit: chọn khối nhớ trống phù hợp đầu tiên kể từ vị trí cấp phát cuối cùng

Câu 20: Phân tích hai khái niệm Page và Paging (trang và phân trang).

- Giải thích được khái niệm Page (trang) với độ lớn (4 điểm): Trang có độ lớn cố định và tất cả các trang có độ lớn bằng nhau, các tiến trình cũng được “chia” thành các trang có độ lớn tương ứng. Bản thân bộ nhớ được quản lý theo trang thông qua việc quản lý “khung trang”. Trang có độ lớn từ 512byte đến 16kb
- Giải thích khái niệm và phương thức Paging (phân trang) (6 điểm) Cơ chế phân trang cho phép không gian địa chỉ thực (physical address space) của một tiến trình có thể không liên tục nhau:

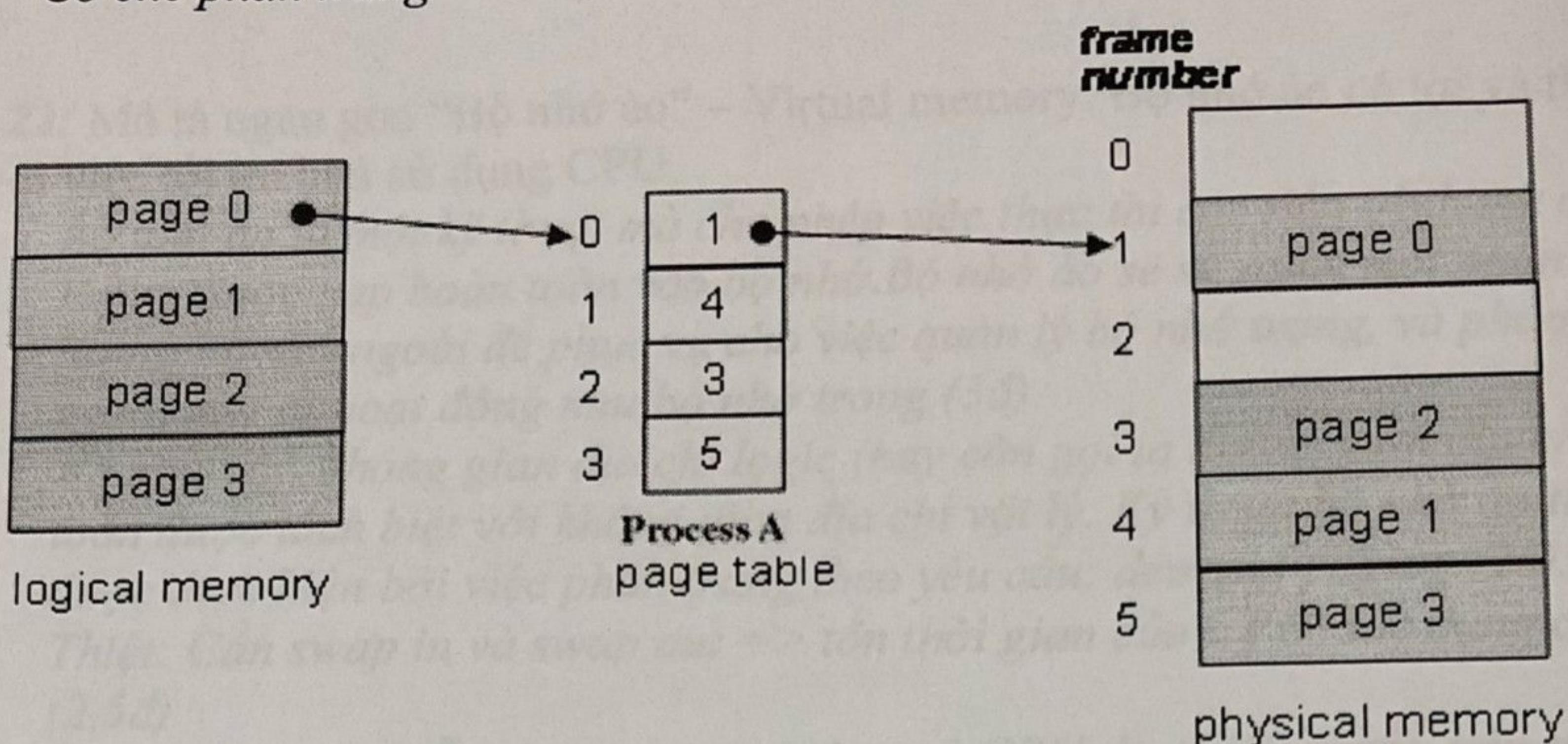
Hệ điều hành phải thiết lập một bảng phân trang (page table) để ánh xạ địa logic thành địa chỉ thực.

Mỗi tiến trình có một bảng phân trang (các trang bộ nhớ được cấp phát cho tiến trình này) được quản lý bằng một con trỏ và được lưu giữ trong PCB. Công việc nạp

bảng phân trang vào hệ thống (do bộ điều phối CPU dispatcher thực hiện) là một phần của công việc hoán đổi tiến trình.

Cơ chế phân trang khiến bộ nhớ bị phân mảnh nội, tuy nhiên lại khắc phục được phân mảnh ngoại.

VD: – Có chế phân trang



Câu 21: Giải thích cơ chế phân đoạn (segmentation) trong quản lý bộ nhớ của hệ điều hành.

Trong thực tế, dưới góc nhìn của người sử dụng, một chương trình cấu thành từ nhiều phân đoạn (segment). Mỗi phân đoạn là một đơn vị logic, ví dụ như: main program, procedure, function, local variables, global variables, stack, symbol table, ... (Giải thích khái niệm kèm hình vẽ 4đ)

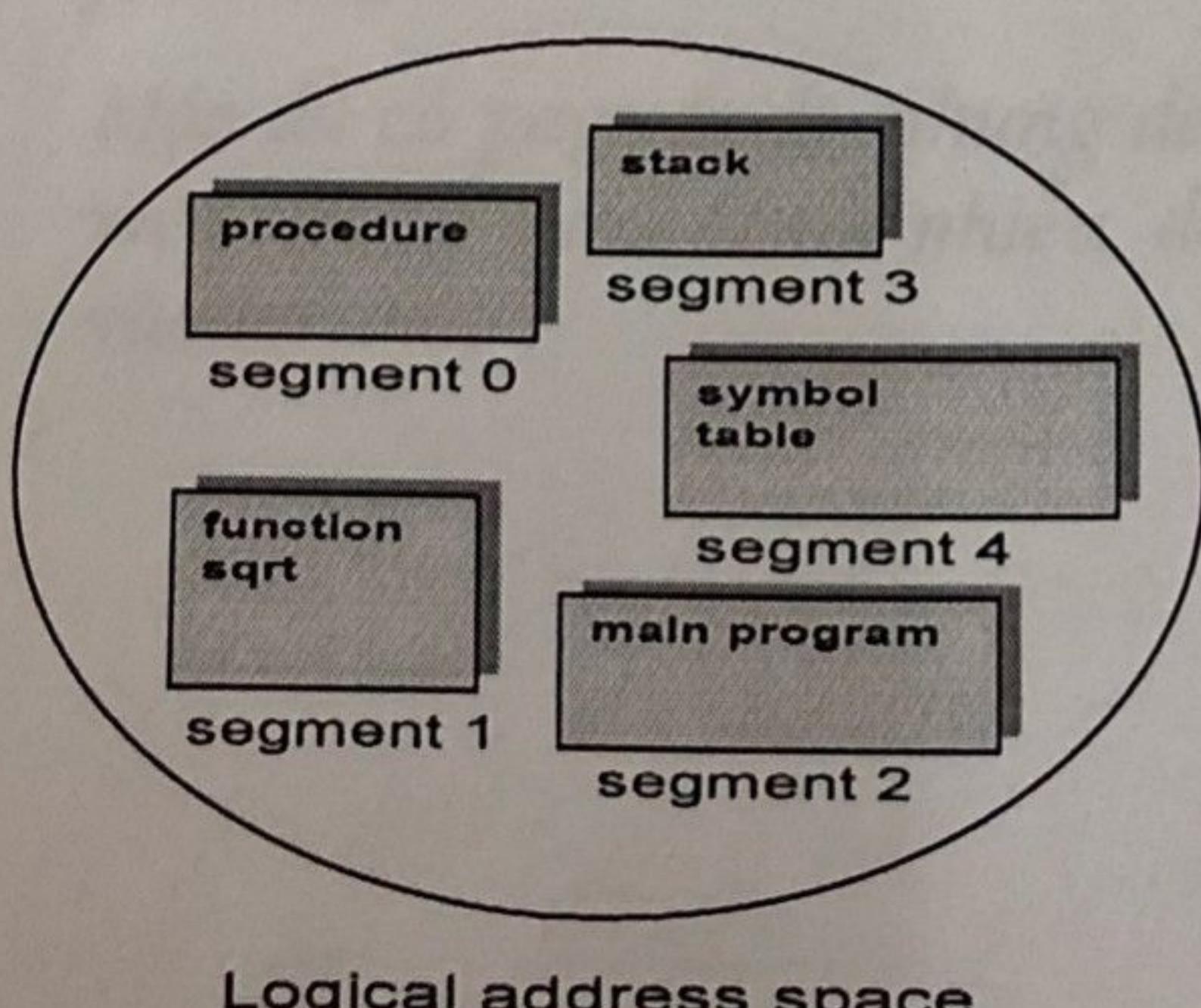
Kỹ thuật phân đoạn (6đ): Chia không gian địa chỉ ảo thành một tập các phân đoạn (segment), mỗi phân đoạn có tên và kích thước riêng. Một địa chỉ logic được định vị bằng tên phân đoạn và độ dời (offset) bên trong phân đoạn đó.

Địa chỉ logic là một cặp giá trị: <segment-number, offset>

Bảng phân đoạn (segment table)

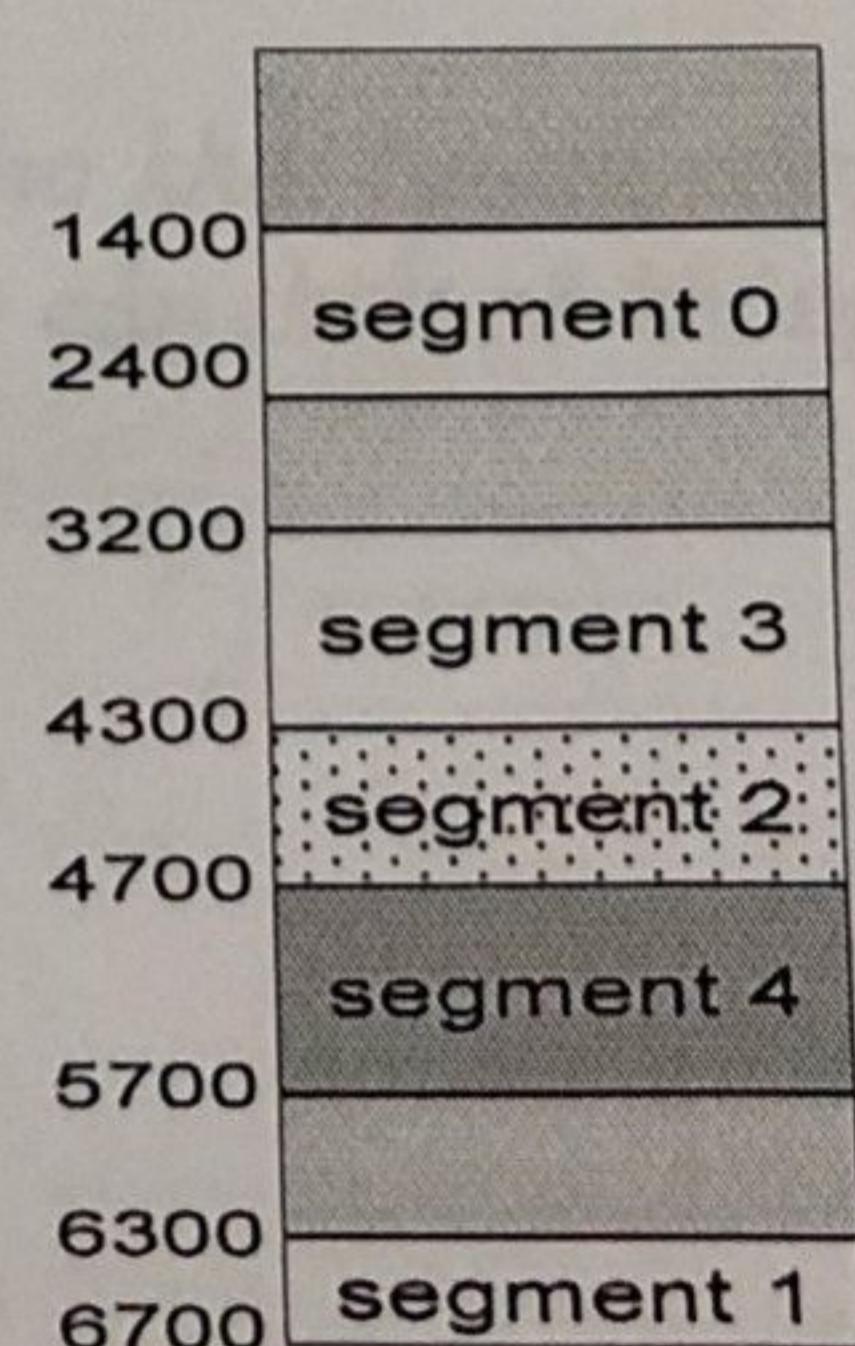
base – chứa địa chỉ khởi đầu của phân đoạn trong bộ nhớ

limit – xác định kích thước của phân đoạn



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table



physical memory space

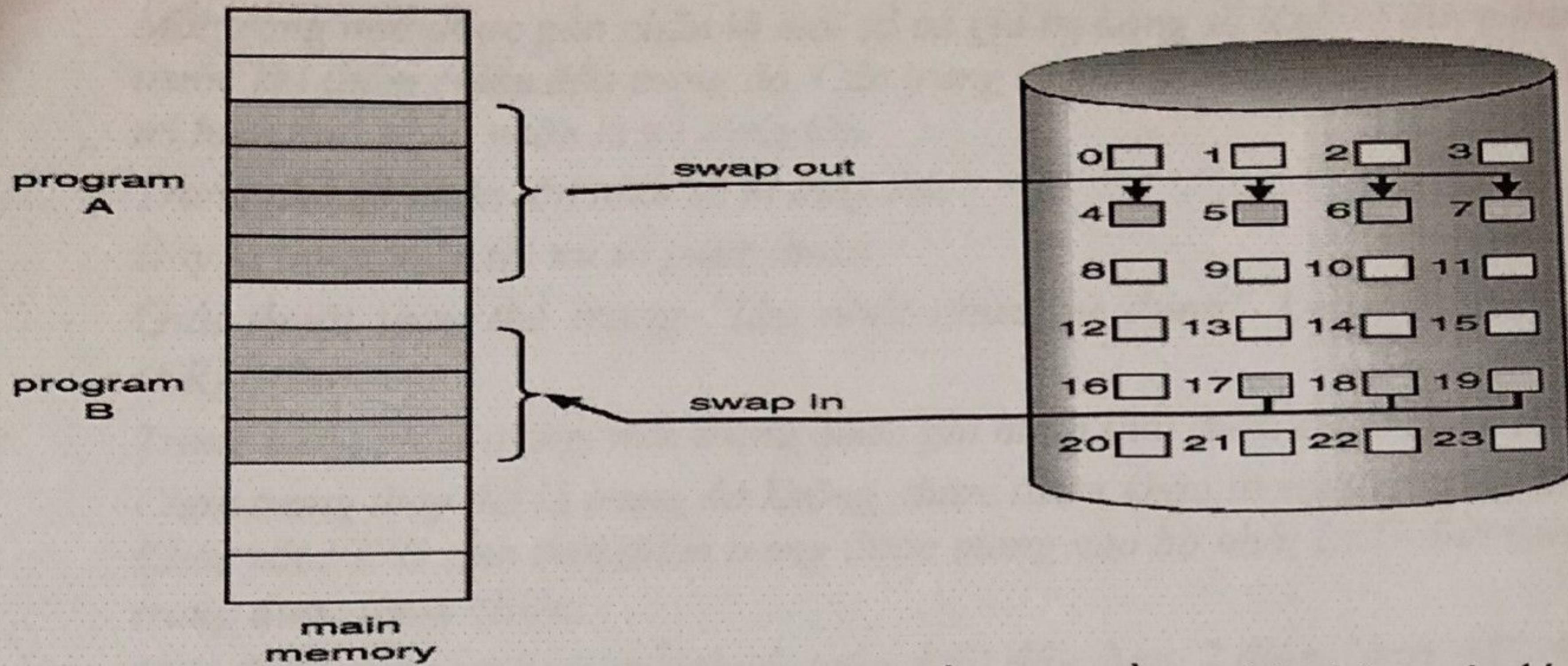
Câu 22: Mô tả ngắn gọn “Bộ nhớ ảo” – Virtual memory. Bộ nhớ ảo có lợi và thiệt hại gì đối với việc tối ưu hóa sử dụng CPU.

- *Bộ nhớ ảo là một kỹ thuật mà cho phép việc thực thi các tiến trình mà có thể không được nạp hoàn toàn vào bộ nhớ. Bộ nhớ ảo sẽ sử dụng một phần dung lượng bộ nhớ ngoài để phục vụ cho việc quản lý bộ nhớ trong, và phần bộ nhớ ngoài này sẽ hoạt động như bộ nhớ trong (3đ).*
- *Và như vậy, không gian địa chỉ logic (hay còn gọi là không gian địa chỉ ảo) hoàn toàn được tách biệt với không gian địa chỉ vật lý. Kỹ thuật bộ nhớ ảo thường được thực hiện bởi việc phân trang theo yêu cầu: demand paging. (2đ).*
- *Thiệt: Cần swap in và swap out => tốn thời gian của CPU; tác dụng của DMA (2,5đ)*
- *Ích lợi mang lại dẫn tới nâng cao hiệu quả CPU do nhiều users, tiến trình hơn (2,5 điểm)*

Câu 23: Trong kỹ thuật bộ nhớ ảo thường sử dụng Phân trang theo yêu cầu (demand paging). Hãy giới thiệu cơ chế này.

Nếu được mỗi ý sau được 2 điểm:

- *Hệ thống phân trang theo yêu cầu tương tự như hệ thống phân trang thông thường nhưng có hoán chuyển swapping*
- *Khi cần thiết các trang của tiến trình có thể được hoán chuyển ra kho chứa của bộ nhớ ảo*
- *Với kỹ thuật này hệ thống cần sự giúp đỡ của phần cứng để phân biệt những trang nào đã trong bộ nhớ, trang nào còn ở trên đĩa. Bit valid/invalid trong bảng phân trang được sử dụng cho mục đích này. Nếu valid là trang bộ nhớ có thể truy nhập hoặc là trang chưa được đưa vào bộ nhớ.*
- *Việc truy nhập vào trang mã được đánh dấu là invalid gây ra một lỗi gọi là page fault trap.*
- *Mặc dù có page fault, nhưng demand paging giúp cho khôi phục công việc như việc vào ra giảm đi rất nhiều, do những trang không cần thiết sẽ không phải nạp vào bộ nhớ.*



Câu 24: Trong kỹ thuật sử dụng Bộ nhớ ảo Hệ điều hành cần có bộ phận quản lý việc hoán chuyển các trang/đoạn giữa bộ nhớ thực và bộ nhớ ảo và có thể xảy ra “Lỗi trang” (page fault). Hệ điều hành giải quyết vấn đề đó như thế nào và mục tiêu cần đạt được?

Các công việc (5đ)

Chuyển trạng thái của tiến trình sang Chờ

Chọn một trang để thay thế (page replacement algorithm)

Khởi động việc nạp trang mới từ đĩa vào bộ nhớ

Chuyển thực thi cho tiến trình khác trong lúc đang thực hiện việc vào ra

Nhận interrupt báo I/O hoàn tất (i.e. đã nạp xong trang nhớ mới)

Chuyển trạng thái process về ready

Khi bộ nhớ chính có chỗ trống thì chúng ta tiến hành nạp chương trình vào trang nhớ đó → page fault.

Yêu cầu (5đ) : Tối thiểu số page fault và nguyên tắc chọn trang thay thế là

- Trang không còn dùng nữa

- Trang sẽ không dùng lại trong thời gian xa nhất

Các tiêu chuẩn (thực tế) để chọn trang thay thế

Các trang không bị thay đổi

Các trang không bị khóa

Các trang không thuộc quá trình nhiều page fault

Các trang không thuộc tập làm việc của quá trình

Câu 25: Đánh giá các thuật toán thay trang (Page Replacement) trong kỹ thuật sử dụng Bộ nhớ ảo.

- Thuật toán FIFO (2đ): Xem các frame được cấp phát cho process như là circular buffer

Trang nhớ cũ nhất sẽ được thay thế: first-in first-out, đây là thuật toán đơn giản nhất

- Thuật toán tối ưu (Optimal OPT) (2đ)

Thay thế trang nhớ được tham chiếu trễ nhất trong tương lai

Mỗi trang nhớ được gắn nhãn là một số có giá trị bằng số lệnh sẽ được thực thi trước khi tham chiếu đến trang đó. Các trang sẽ không được truy cập tiếp kề từ vị trí hiện thời sẽ có nhãn là vô cùng lớn.

Trang nhớ có nhãn lớn nhất sẽ bị thay thế.

Đây là thuật toán tối ưu số page faults

- Giải thuật thay thế trang “lâu nhất chưa sử dụng” Least Recently Used (LRU)(2đ)

Trong bảng phân trang, mỗi trang được ghi nhận thời điểm được tham chiếu.

Chọn trang thay thế là trang đã không được tham khảo trong thời gian lâu nhất.

Khác với FIFO tính thời điểm trang được mang vào bộ nhớ, LRU tính thời điểm trang được tham chiếu.

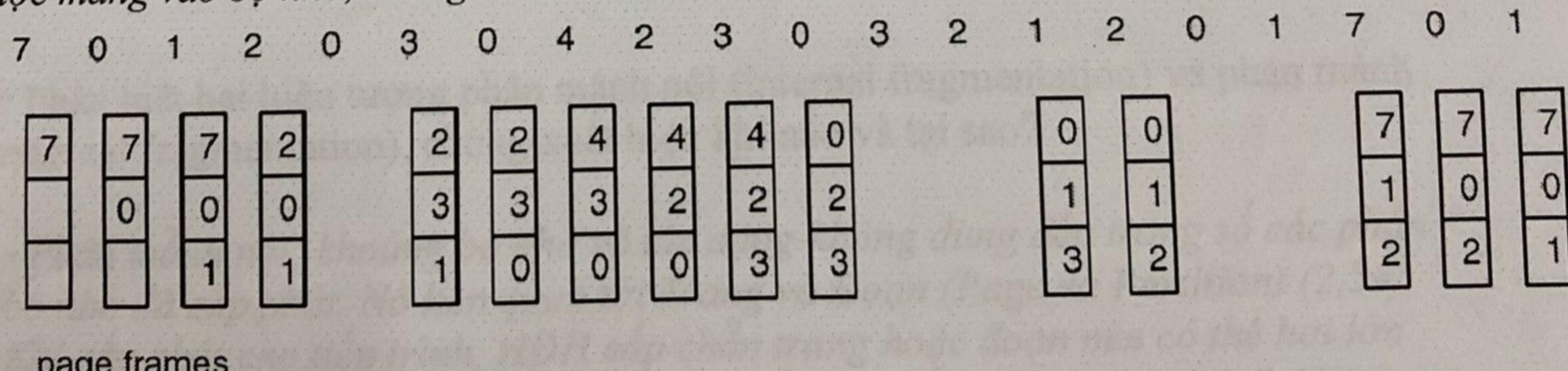
- Giới thiệu được sơ lược mỗi thuật toán dưới đây được 2 điểm (max 4đ): thay thế trang có tần suất sử dụng nhỏ nhất - LFU (Least Frequently Used, là giải thuật xấp xỉ LRU, Chọn trang thay thế là trang có tần suất được tham khảo là nhỏ nhất trong 1 khoảng thời gian nhất định; Thuật thay thế trang ít được sử dụng gần đây (NRU - Not Recently Used); Thuật toán Second chance tương tự FIFO nhưng cho cơ hội thứ hai còn có tên là đồng hồ vì dùng bộ đếm quay vòng giống đồng hồ; hoặc thuật toán MFU.

Câu 26: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán FIFO.

- Giới thiệu đủ các thuật toán FIFO, OPT, LRU và các thuật toán xấp xỉ LRU, second chance (4đ)

Giải thích Thuật toán FIFO (3đ), ví dụ (3đ):

Giải thuật này gắn với mỗi trang thời gian khi trang đó được mang vào trong bộ nhớ. Khi một trang phải được thay thế, trang cũ nhất sẽ được chọn. Chú ý rằng, nó không yêu cầu nghiêm ngặt để ghi thời gian khi trang được mang vào. Chúng ta có thể tạo một hàng đợi FIFO để quản lý tất cả trang trong bộ nhớ. Chúng ta thay thế trang tại đầu hàng đợi. Khi trang được mang vào bộ nhớ, chúng ta chèn nó vào đuôi của hàng đợi.

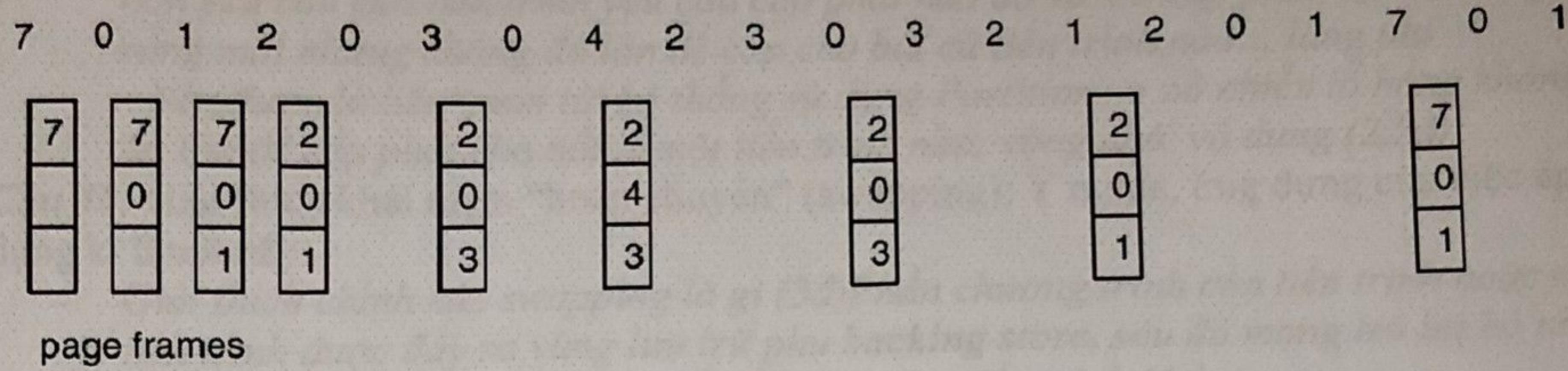


page frames

Câu 27: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán Tối ưu OPT.

- Giới thiệu đủ các thuật toán FIFO, OPT, LRU và các thuật toán xấp xỉ LRU, second chance (4đ)

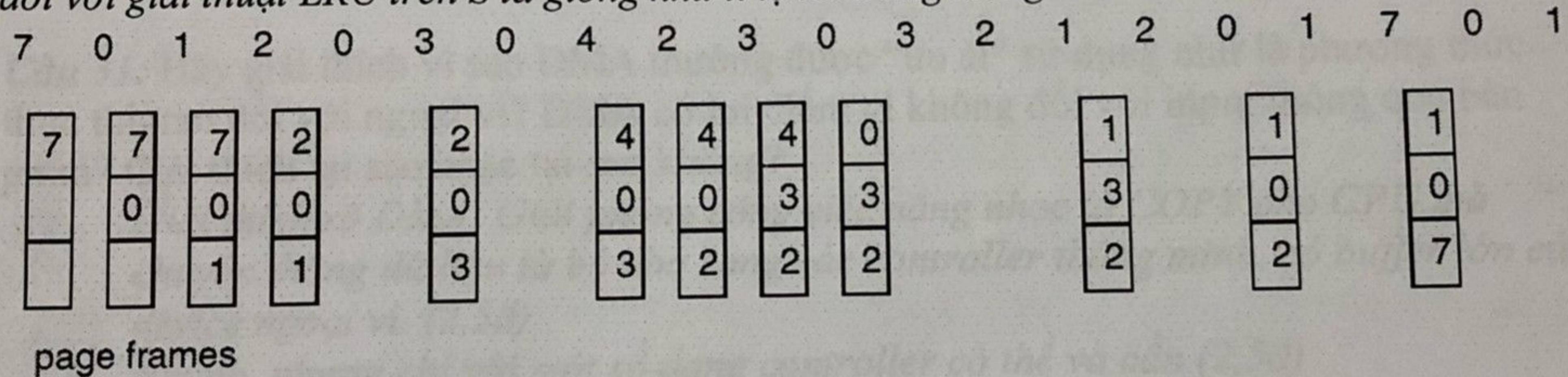
Giải thích Thuật toán OPT (3đ), ví dụ (3đ): Thuật toán thay thế trang tối ưu có tỉ lệ lỗi trang thấp nhất trong tất cả các giải thuật. OPT thay thế trang mà nó không được dùng cho một khoảng thời gian lâu nhất. Sử dụng giải thuật thay thế trang đảm bảo tỉ lệ lỗi trang nhỏ nhất có thể cho một số lượng khung cố định.



Câu 28: Nêu ngắn gọn các thuật toán Thay thế trang trong kỹ thuật bộ nhớ ảo, Trình bày và giải thích bằng ví dụ thuật toán LRU.

- Giới thiệu đủ các thuật toán FIFO, OPT, LRU và các thuật toán xấp xỉ LRU, second chance (4đ)

Giải thích Thuật toán OPT (3đ), ví dụ (3đ): Nếu chúng ta sẽ dụng quá khứ gần như một xấp xỉ của tương lai gần thì chúng ta sẽ thay thế trang mà nó không được dùng cho khoảng thời gian lâu nhất. Tiếp cận này là giải thuật ít được dùng gần đây nhất (least-recently-used (LRU)). Thay thế trang LRU gắn với mỗi trang thời gian sử dụng cuối cùng của trang. Khi một trang phải được thay thế, LRU chọn trang không được dùng trong một khoảng thời gian lâu nhất. Chiến lược này là giải thuật thay thế trang tối ưu tìm kiếm lùi theo thời gian hơn là hướng tới. (gọi S^R là trình tự ngược của chuỗi tham khảo S thì tỉ lệ lỗi trang cho giải thuật OPT trên S là tương tự như tỉ lệ lỗi trang cho giải thuật OPT trên S^R . Tương tự, tỉ lệ lỗi trang đối với giải thuật LRU trên S là giống như tỉ lệ lỗi trang cho giải thuật LRU trên S^R)



Câu 29: Phân biệt hai hiện tượng phân mảnh nội (internal fragmentation) và phân mảnh ngoài (external fragmentation), chúng xuất hiện khi nào và tại sao?

- Phân mảnh nội: khoảng bộ nhớ vô tác dụng-không dùng đến trong số các phần bộ nhớ đã cấp phát. Nó liên quan tới Trang và Đoạn (Page và Partition) (2,5đ)
Khi cấp phát cho tiến trình, HDH cấp chẵn trang hoặc đoạn nên có thể hơi lớn hơn nhu cầu, phần lớn hơn này không thể cấp cho tiến trình nào mặc dù không được sử dụng

- Nếu được là liên quan tới độ lớn trang hay segment và có đánh giá (2,5đ) Độ lớn các trang, đoạn là cố định. Trang/đoạn càng lớn phân mảnh càng tăng, nhưng khi chia trang có độ lớn nhỏ thì tốn công và tài nguyên để quản lý – liên quan đến Frame

- Phân biệt được Phân mảnh ngoại: là vùng nhớ không sử dụng được nằm giữa các vùng nhớ đã cấp phát (2,5đ): Khi cấp bộ nhớ cho tiến trình HDH cấp vừa đủ, khi tiến trình kết thúc, vùng nhớ này được giải phóng, vùng nhớ này có thể lớn

hơn yêu cầu của tiến trình yêu cầu cấp phát nào đó và khi cấp phát, tạo ra vùng rỗng mới nhưng không đủ lớn để cấp cho bất cứ tiến trình nào... lãng phí

- Nếu được là liên quan tới hệ thống sử dụng Partition -> có nhiều lỗ hổng không đủ lớn để cấp phát cho bất kỳ một tiến trình nào; vùng nhớ vô dụng (2,5đ)

Câu 30: Giải thích khái niệm “hoán chuyển” (swapping); Ý nghĩa, ứng dụng của việc áp dụng kỹ thuật này.

- Giải thích chính xác swapping là gì (3đ) Phần chương trình của tiến trình hoặc cả tiến trình được đẩy ra vùng lưu trữ phụ **Backing store**, sau đó mang trở lại bộ nhớ để việc thực thi được tiếp tục, để cấp phát cho tiến trình khác.
- Nếu swap tiến trình và swap trang (3+3 đ)
Một quá trình cần ở trong bộ nhớ để được thực thi. Tuy nhiên, một quá trình có thể được hoán vị (swapped) tạm thời khỏi bộ nhớ tới **Backing store**.
Thí dụ, giả sử một môi trường đa chương với giải thuật lập thời biểu CPU round-robin. Khi định mức thời gian hết, bộ quản lý bộ nhớ sẽ bắt đầu hoán vị ra (swap out) vùng lưu trữ phụ quá trình vừa mới kết thúc và hoán vị vào (swap in) một quá trình khác tới không gian bộ nhớ được trống. => swap tiến trình. Có thể chỉ cần swap một phần của tiến trình, cụ thể là một số trang ra => thay thế trang
- Nếu được yêu cầu và ý nghĩa của việc ứng dụng (1đ): cần sử dụng bộ nhớ ngoài, thông thường là **Đĩa cứng**, cần quan tâm tới thời gian

Nhóm 2:

Câu 31: Hãy giải thích vì sao DMA thường được “ưu ái” sử dụng như là phương thức thực thi trao đổi với ngoại vi? DMA có lợi điểm gì không đối với input thông qua bàn phím? Giải thích tại sao hoặc tại sao không?

- Giải thích rõ DMA: Giải phóng công việc nặng nhọc là COPY cho CPU mà chuyển thẳng dữ liệu từ bộ nhớ sang các controller thông minh, có buffer lớn của device ngoại vi. (2,5đ)
- Nhanh, nhưng chỉ với một số dạng controller có thể và cần (2,5đ)
- Bàn phím rất chậm, giải thích nếu sử dụng DMA cho bàn phím (2,5đ)
- → Rất ít hiệu quả (2,5đ) Trong trường hợp trả lời Có thì phải giải thích Bàn phím thực chất ở đây là hệ máy tính chuyên dụng dạng WordProcessing thực chất là trao đổi giữa 2 hệ thống

Câu 32: Các máy tính nguyên chiếc sử dụng phổ biến hiện nay thường cài đặt Hệ điều hành Windows XP hoặc Vista; đây là các hệ thống/ Hệ điều hành Đa trình đa nhiệm. Hãy giới thiệu một số dạng hệ thống/ hệ điều hành khác.

- Giới thiệu các phiên bản khác của HĐH do Microsoft phát triển (có thể gợi ý): Từ MS DOS, các phiên bản Windows 3.1, Windows 98, XP, nếu ngắn gọn sự phát triển, mô hình triển (4đ)
- Nếu được 3 dạng HĐH đa trình đa nhiệm phổ biến khác không loại bỏ (Unix/Linux, Mac OS/ OS2) (3đ)
- Giới thiệu các HĐH thời gian thực, Symbian, Mobile, Hệ thống chuyên dụng ..(3 điểm).

Câu 33: Khi học môn “Tin học đại cương” chúng ta đã được biết đến một cách chia các thiết bị ngoại vi thành 2 hoặc 3 loại (nhóm) là “VÀO”, “RA” và/hoặc vừa “VÀO vừa RA”; Dưới góc nhìn người thiết kế Hệ điều hành chúng ta có thể chia thiết bị ngoại vi thành 2 loại. Đó là gì và cách chia này dựa trên căn cứ nào? Hãy nêu ví dụ một số ngoại vi thuộc từng loại trên.

- *Nêu được cách nhìn dưới OS: Character và Block Device; Giới thiệu đặc tính chung của 2 nhóm này đó là bit by bit hay từng byte một và cả Block, khỏi dữ liệu, số lượng lớn; Cách chia này căn cứ chính dựa trên cơ sở nguyên lý truyền tin (5đ)*
- *Nêu ví dụ và chỉ dạng thông tin trao đổi (lưu ý thiết bị lưu trữ là Flash Disk qua cổng USB hay gọi là USB; cần nêu được 4 ví dụ tiêu biểu, 2 ví dụ cho mỗi loại: Bàn phím, con chuột, cổng mạng hay USB... và CD, FDD, ổ đĩa cứng, card màn hình...) (5đ)*

Câu 34: Giải thích sự khác biệt giữa “polled I/O” và “interrupt-driven I/O”. Nêu những ưu điểm của “interrupt-driven I/O” so với “polled I/O”.

- *Giải thích polled I/O: Phải quay vòng kiểm tra thanh ghi trạng thái của Device (2,5 đ) cơ chế IO ngang hàng*
- *Vì thế mất thời gian so với Interrupt → Multitasking không thể (2,5 đ)*
- *Giải thích interrupt Driven I/O: Cơ chế gửi tín hiệu ngắn và việc phục vụ ngắn (2,5 đ)*
- *Lợi thế: Nêu trên và khả năng cấp phát Cấp ưu tiên (2,5 đ) cho phép ngắn phục vụ ngắn tức là cho phép Tiến trình có mức ưu tiên cao hơn có thể ngắn tiến trình khác*

Câu 35: Một cách tương đối, ổ đĩa cứng chậm hơn rất nhiều so với CPU. Nêu các yếu tố đánh giá tốc độ truy xuất dữ liệu trên đĩa. Hãy nêu ngắn gọn các kĩ thuật mà Hệ điều hành sử dụng để nâng cao thông lượng (throughput) trung bình của đĩa.

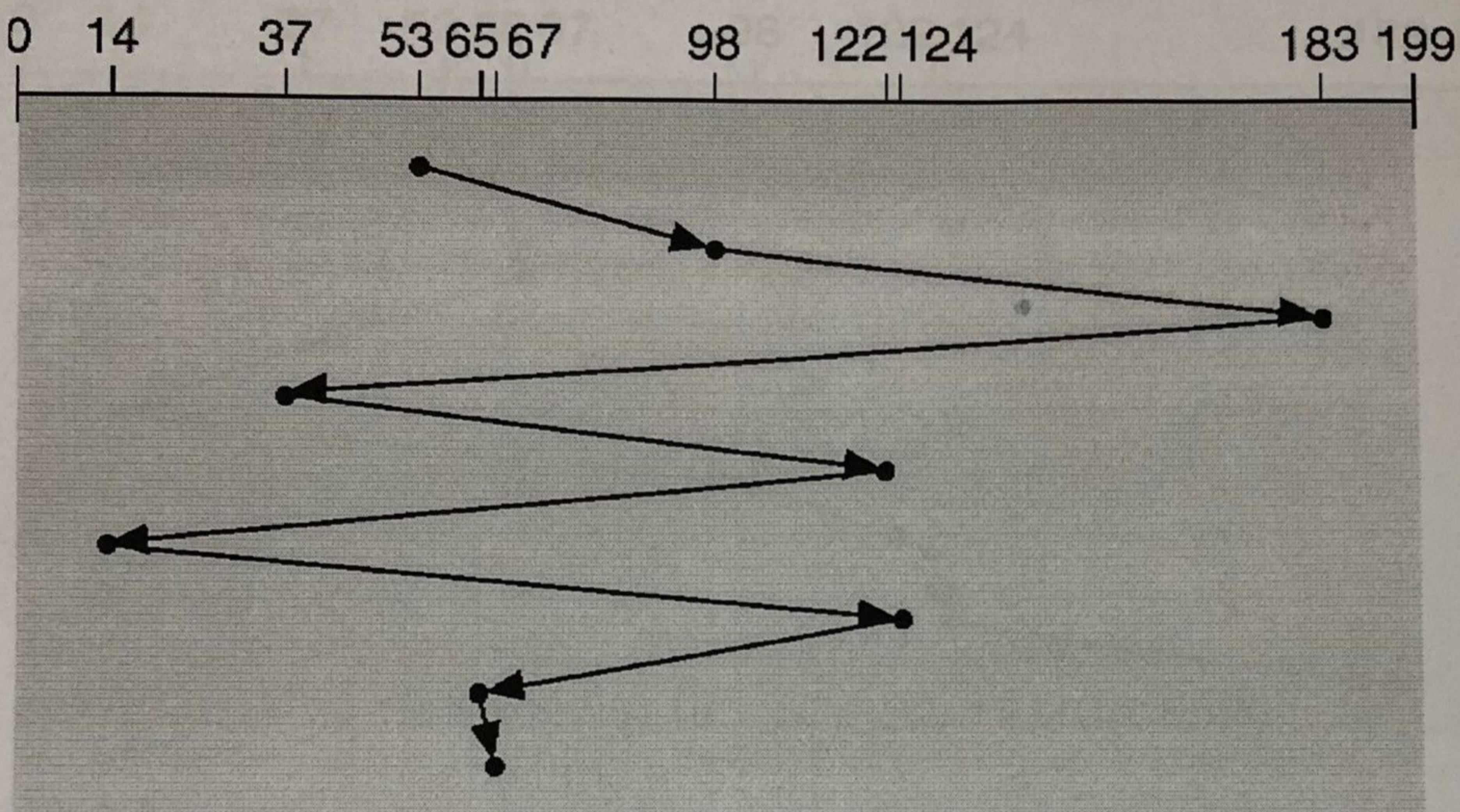
- *Tốc độ vòng quay, Thời gian truyền, Độ trễ và quan trọng nhất là Thời gian dịch chuyển đầu từ (3 điểm)*
- *Giải thích Nguyên tắc chính để nâng cao thông lượng chính là Arm Motion/ Arm scheduling (7 điểm, mỗi Thuật toán 1 điểm):
FCFS: Theo trình tự các sector lưu trữ trên đĩa (thực chất là trên cylinder)
SSTF: Đọc sector trên cylinder gần nhất-dịch chuyển đầu từ ngắn nhất từ vị trí hiện tại của đầu từ
SCAN: Đầu từ quét theo 1 hướng từ trong ra hoặc từ ngoài vào trên đường đi sẽ đọc dữ liệu tại các sector tương ứng
CSCAN: Chỉ quét theo 1 chiều, khi tới đầu ngược lại, đầu từ dịch ngay về cylinder đầu tiên và tiếp tục đọc
CLOOK: Tương tự CSCAN nhưng không quét đến cylinder cuối của đĩa mà chỉ quét đến cylinder chứa sector có dữ liệu cuối cùng và quay về cylinder chứa sector dữ liệu đầu*

Câu 36: Hãy giải thích thuật toán Lập lịch cho đầu từ FCFS là gì, cho ví dụ để mô tả để tính được seek time.

- *Theo trình tự các sector lưu trữ trên đĩa (thực chất là trên cylinder) (2 đ)*

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

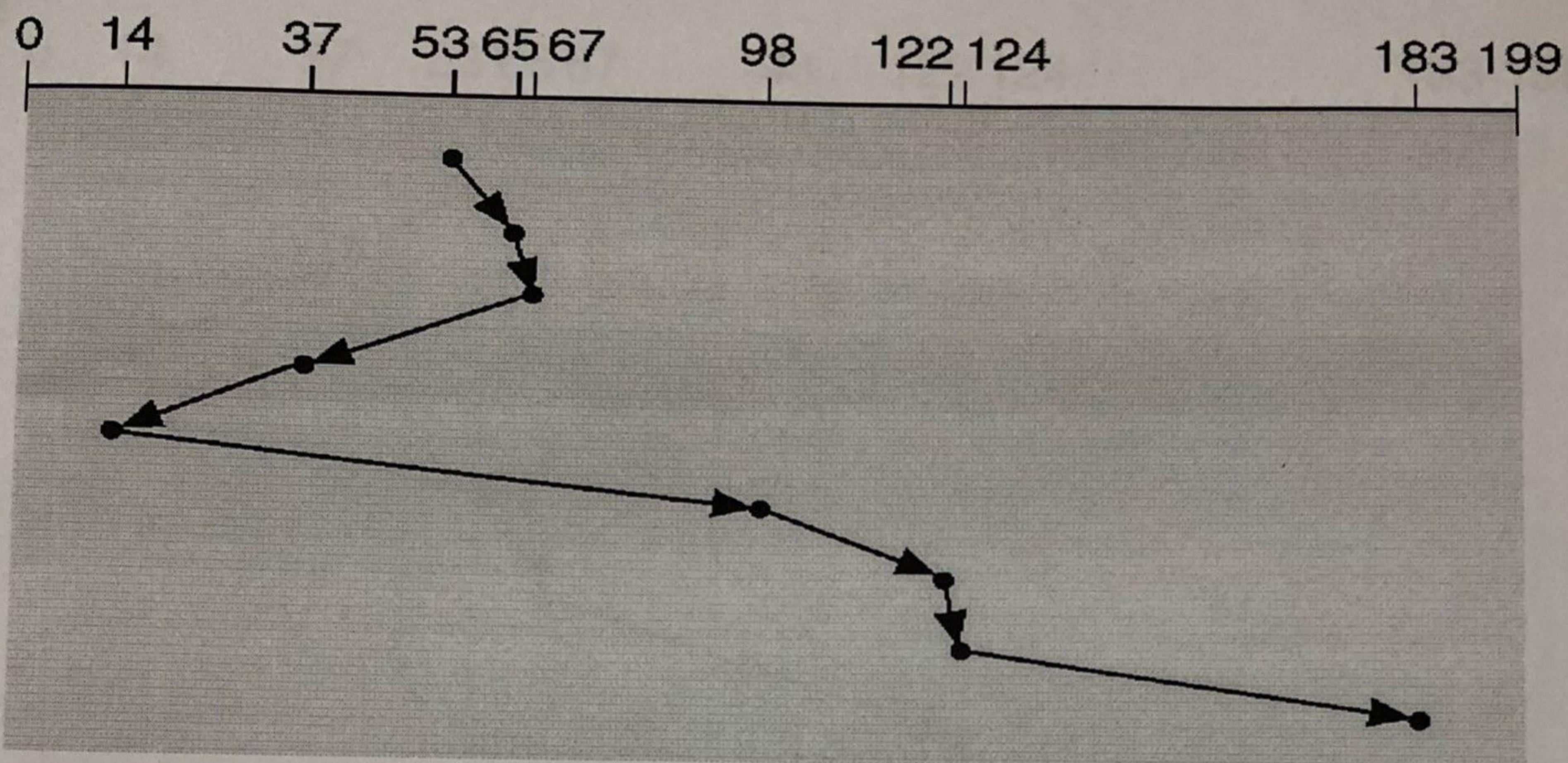


- Vẽ được hình ví dụ và giải thích đúng trình tự (6đ)
- Tính được tổng seek time sau khi cho thời gian dịch chuyển từ cylinder này sang cylinder kế tiếp (2đ)

Câu 37: Hãy giải thích thuật toán Lập lịch cho đầu từ SSTF là gì, cho ví dụ để mô tả để tính được seek time.

- SSTF: Đọc sector trên cylinder gần nhất-dịch chuyển đầu từ ngắn nhất từ vị trí hiện tại của đầu từ
- Vẽ được hình ví dụ và giải thích đúng trình tự (6đ)
- Tính được tổng seek time sau khi cho thời gian dịch chuyển từ cylinder này sang cylinder kế tiếp (2đ) 236 bước

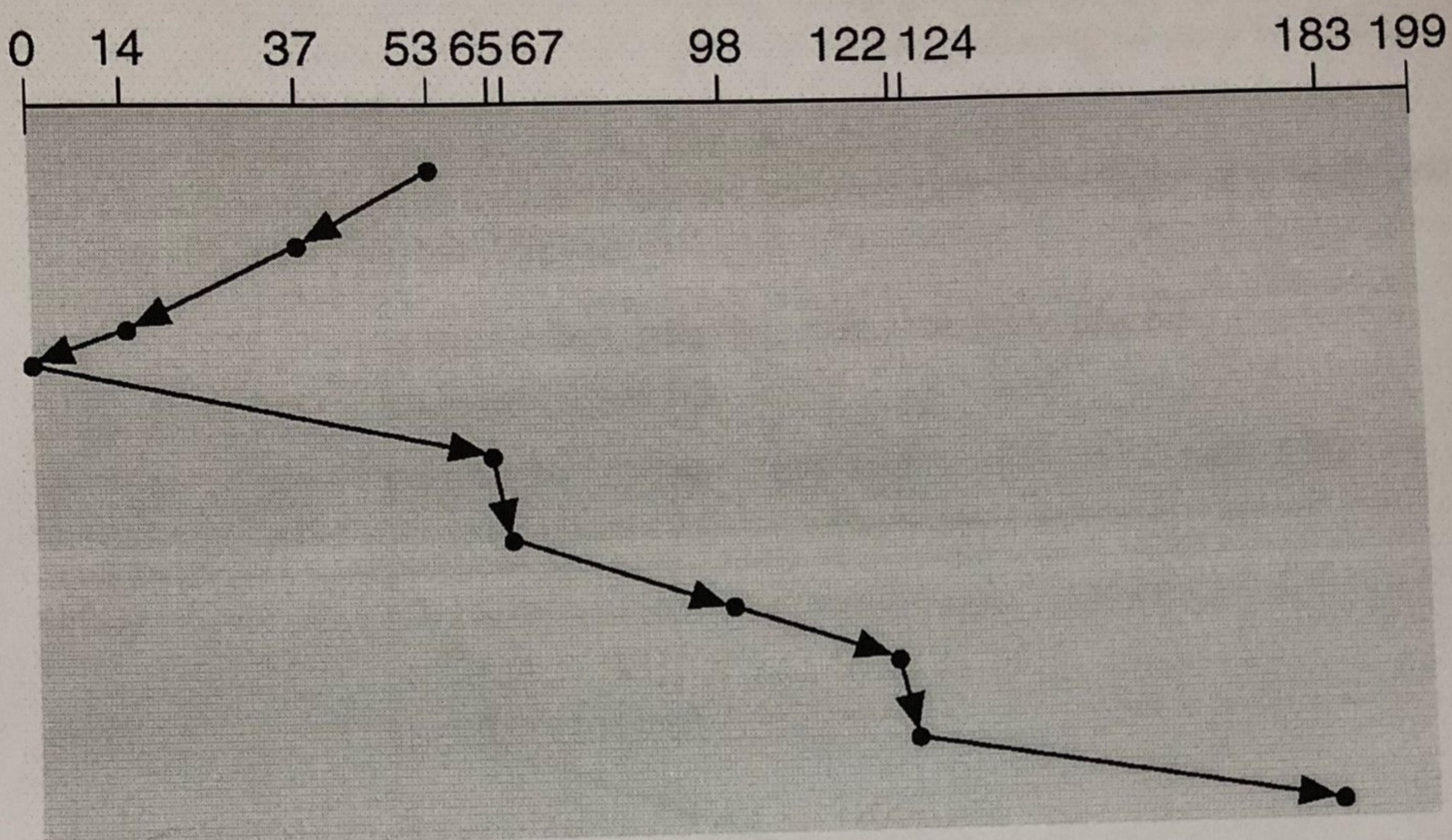
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Câu 38: Hãy giải thích thuật toán Lập lịch cho đầu từ SCAN là gì, cho ví dụ để mô tả để tính được seek time.

- SSTF: Đầu từ quét theo 1 hướng từ trong ra hoặc từ ngoài vào trên đường đi sẽ đọc dữ liệu tại các sector tương ứng
- Vẽ được hình ví dụ và giải thích đúng trình tự (6đ)
- Tính được tổng seek time sau khi cho thời gian dịch chuyển từ cylinder này sang cylinder kế tiếp (2đ) 208 bước

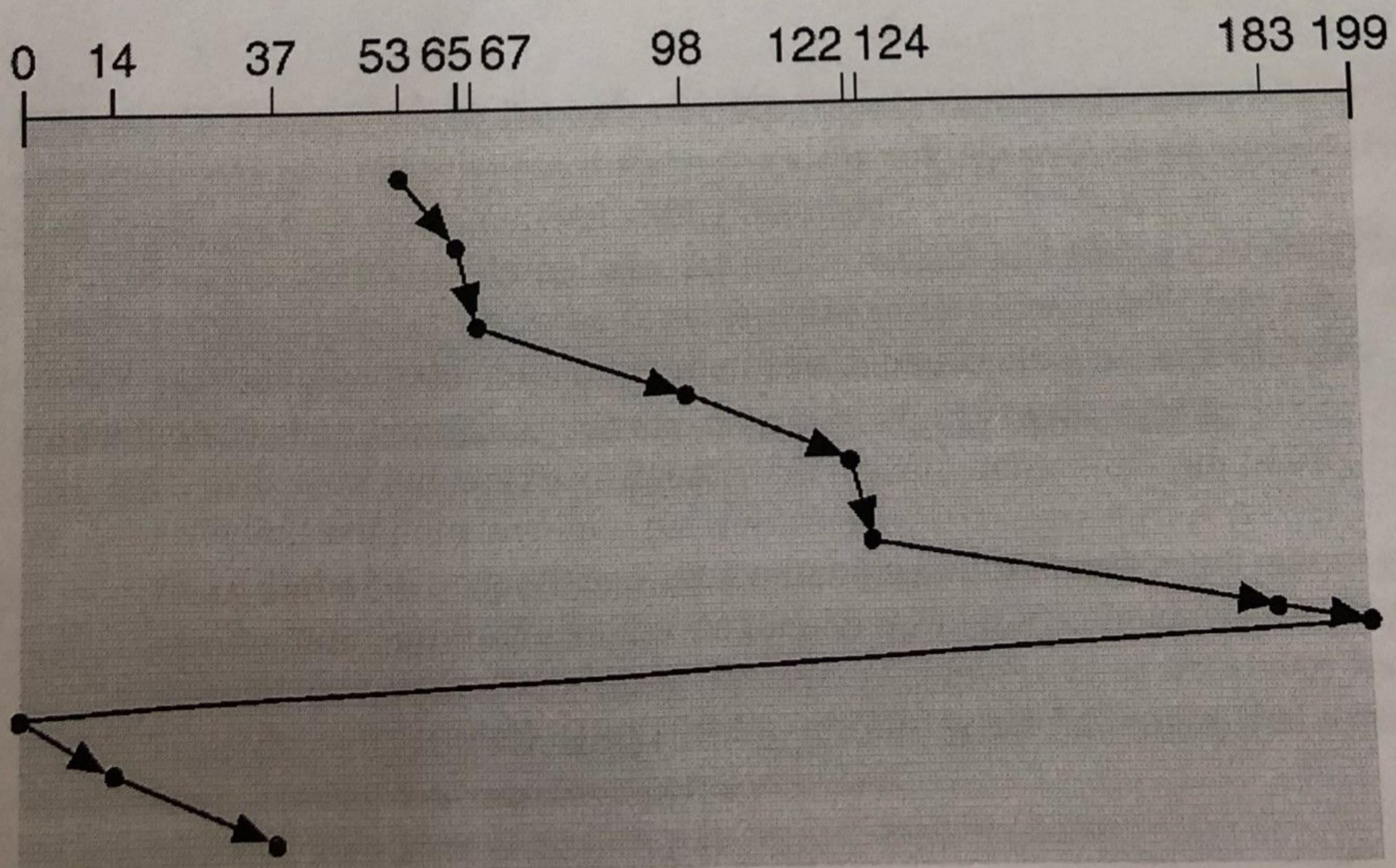
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Câu 39: Hãy giải thích thuật toán Lập lịch cho đầu từ C-SCAN là gì, cho ví dụ để mô tả để tính được seek time.

- CSCAN: Chỉ quét theo 1 chiều, khi tới đầu ngược lại, đầu từ dịch ngay về cylinder đầu tiên và tiếp tục đọc
- Vẽ được hình ví dụ và giải thích đúng trình tự (6đ)
- Tính được tổng seek time sau khi cho thời gian dịch chuyển từ cylinder này sang cylinder kế tiếp (2đ)

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53

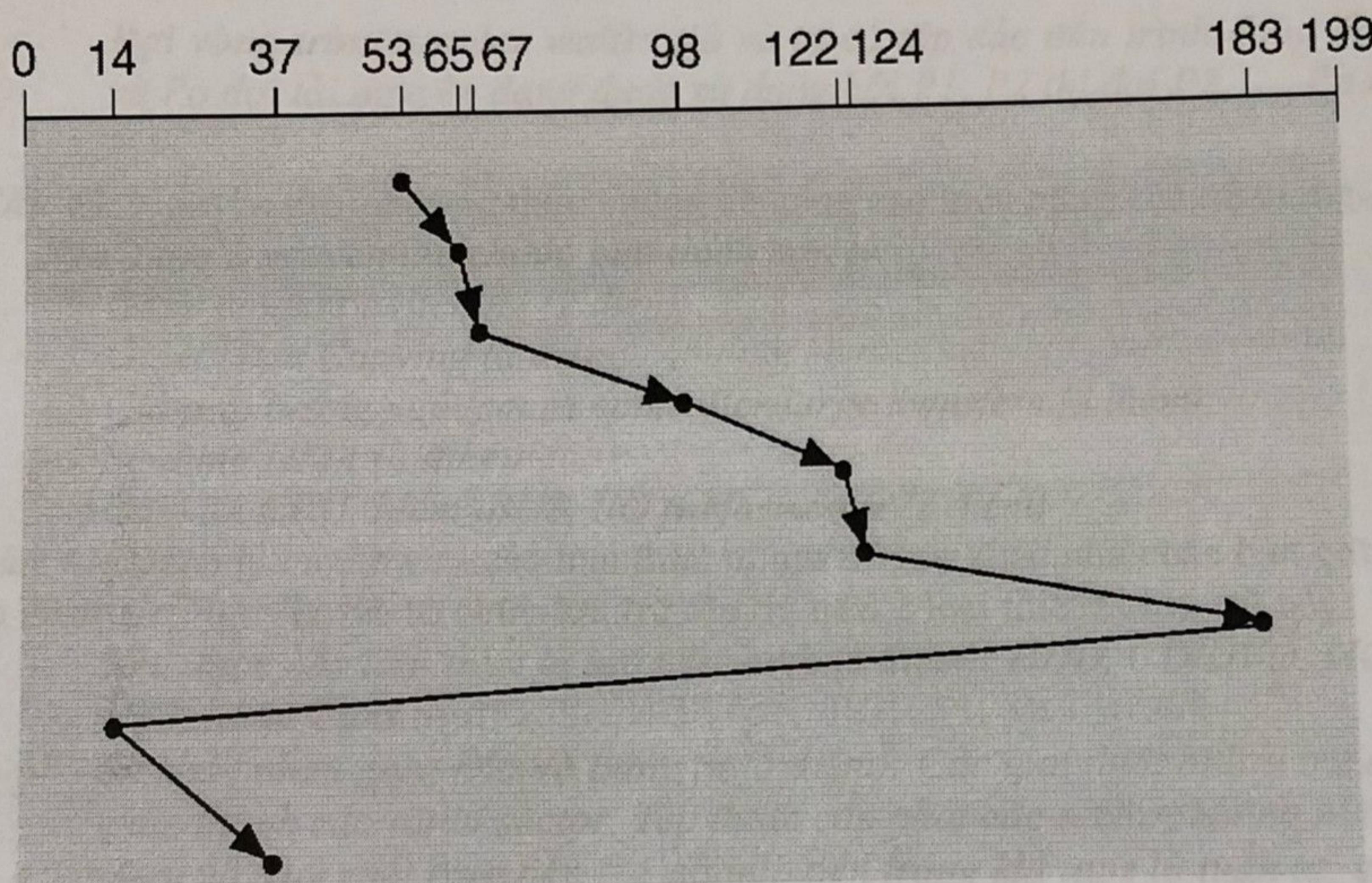


Câu 40: Hãy giải thích thuật toán Lập lịch cho đầu từ C-LOOK là gì, cho ví dụ để mô tả để tính được seek time.

- CLOOK: Tương tự CSCAN nhưng không quét đến cylinder cuối của đĩa mà chỉ quét đến cylinder chứa sector có dữ liệu cuối cùng và quay về cylinder chứa sector dữ liệu đầu
- Vẽ được hình ví dụ và giải thích đúng trình tự (6đ)
- Tính được tổng seek time sau khi cho thời gian dịch chuyển từ cylinder này sang cylinder kế tiếp (2đ)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



)

Câu 41: Có 2 dạng thiết bị đầu cuối nối tiếp (serial) và memory-mapped đều đầy kí tự ra màn hình. Hãy giải thích tại sao thiết bị đầu cuối nối tiếp phải sử dụng ngắt trong khi thiết bị đầu cuối memory-mapped không bắt buộc.

- Nếu cơ chế Write “ký tự” vào “bộ nhớ”; nhanh nên không cần thiết (5 điểm)
- Với dạng còn lại, chậm và vì thế sau khi truyền xong phải “báo cáo” thông qua ngắt bởi thực chất “nối tiếp” tức là phải truyền từng bit một (5 điểm)

Câu 42: Tắc nghẽn (deadlock) và các điều kiện để xảy ra tắc nghẽn.

Giải thích khái niệm tắc nghẽn (2 điểm):

*)

- Trong môi trường đa chương, nhiều quá trình có thể cạnh tranh một số giới hạn tài nguyên. Một quá trình yêu cầu tài nguyên, nếu tài nguyên không sẵn dùng tại thời điểm đó, quá trình đi vào trạng thái chờ. Quá trình chờ có thể không bao giờ chuyển trạng thái trở lại vì tài nguyên chúng yêu cầu bị giữ bởi những quá trình đang chờ khác. Trường hợp này được gọi là deadlock
- Nếu các điều kiện đồng thời gây nên tắc nghẽn (2 điểm X4):

2

Một deadlocks có thể nảy sinh nếu 4 điều kiện sau đây cùng xảy ra tại một thời điểm.

Loại trừ (mutual exclusion): ít nhất một tài nguyên không thể chia sẻ được, chỉ có thể một tiến trình có thể sử dụng mà thôi. Các tiến trình khác đều phải đợi lượt của mình

Giữ và đợi (hold and wait): Một tiến trình khác phải đang sử dụng ít nhất một tài nguyên và đợi sử dụng một tài nguyên mà đang được sử dụng bởi tiến trình khác.

Không thể dừng (no pre-emption): Các tài nguyên không thể bị dừng khi đang được sử dụng. Điều này có nghĩa là một tài nguyên chỉ có thể được trả lại khi tiến trình kết thúc việc sử dụng.

Đợi vòng tròn (circular wait): giả sử ta có tập các tiến trình đang đợi P_0, \dots, P_n và P_0 đợi tài nguyên đang được sử dụng bởi $P_1, P_2 \dots, P_n$ thì đợi P_2, \dots, P_n đợi P_0 .

Câu 43: Nguyên tắc/ phương thức chung để nâng cao hiệu năng của hệ thống.

Nêu được 5 nguyên tắc chính, giải thích hợp lý:

- Giảm context switching (2 điểm)
- Giảm Data Copying (2 điểm)
- Giảm số lượng ngắn/smart controller/large transfers (2 điểm)
- Sử dụng DMA (2 điểm)
- Cân đối CPU, Mem, BUS, I/O performance (2 điểm)

Câu 44: Đánh giá ngắn gọn các loại thiết bị lưu trữ; sự khác nhau căn bản giữa Đĩa cứng và Băng từ; Nguyên tắc tổ chức lưu trữ tệp tin trên 2 loại thiết bị lưu trữ này.

- Nêu được các loại thiết bị lưu trữ chính (3 điểm): HDD, CD/DVD, MO, Tape, USB, Mem Card
- Sự khác nhau giữa Đĩa và Băng từ (5 điểm): Các loại thiết bị lưu trữ dạng đĩa tổ chức thành các rãnh/sector, Tệp được cấp phát các sector, không nhất thiết liền nhau và truy xuất trực tiếp => nhanh, tiện trong khi tape là tuần tự, chậm, phải tua.
- Tổ chức tệp tin trên mỗi loại (2đ) Truy xuất tuần tự và truy xuất trực tiếp

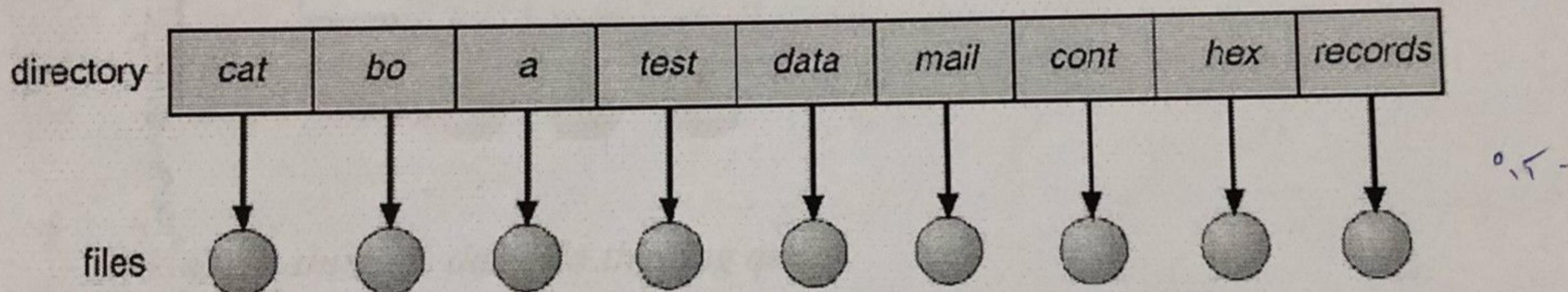
Câu 45: Các thuộc tính chính và các trạng thái của File các hệ điều hành khác nhau thường có?

- Nêu được 5 thuộc tính của File (1đ x 5): Một tập tin có một số thuộc tính khác mà chúng rất khác nhau từ một hệ điều hành này tới một hệ điều hành khác, nhưng điển hình chúng gồm:
 - **Tên (name):** tên tập tin chỉ là thông tin được lưu ở dạng mà người dùng có thể đọc
 - **Định danh (identifier):** là thẻ duy nhất, thường là số, xác định tập tin trong hệ thống tập tin
 - **Kiểu (type):** thông tin này được yêu cầu cho hệ thống hỗ trợ các kiểu khác nhau
 - **Vị trí (location):** thông tin này là một con trỏ chỉ tới một thiết bị và tới vị trí tập tin trên thiết bị đó.
 - **Kích thước (size):** kích thước hiện hành của tập tin.
 - **Giờ (time), ngày (date) và định danh người dùng (user identification).**
- Trình bày được các trạng thái của File và vai trò/tác nghiệp có thể thực hiện với File ứng với các trạng thái đó (5 điểm)
Hệ điều hành cung cấp lời gọi hệ thống để thực hiện các thao tác này

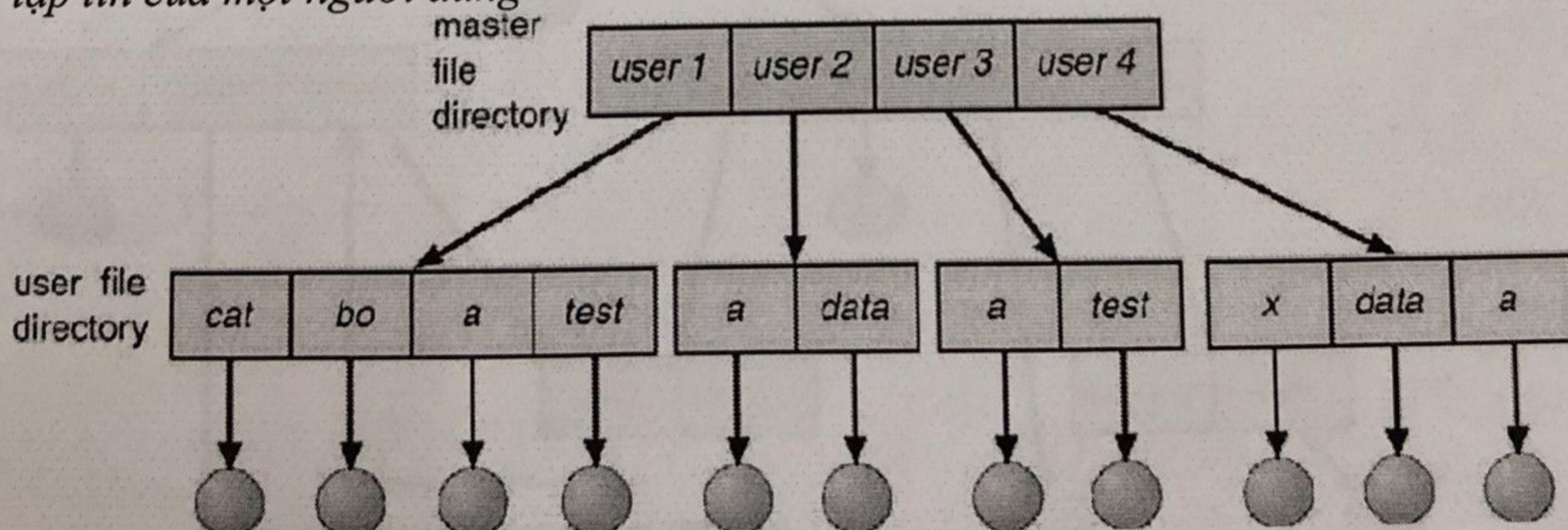
- Khởi tạo tập tin
- Mở là cho phép hệ thống thiết lập một số thuộc tính và địa chỉ đĩa trong bộ nhớ để tăng tốc độ truy xuất.
- Đóng: khi chấm dứt truy xuất, thuộc tính và địa chỉ trên đĩa không còn dùng nữa, tập tin được đóng lại để giải phóng vùng nhớ.
- Ghi: để ghi một tập tin
- Chèn cuối: giống thao tác ghi nhưng dữ liệu luôn được ghi vào cuối tập tin
- Đọc: để đọc từ một tập tin
- Xoá: xóa, giải phóng không gian tập tin để không gian này có thể dùng lại bởi tập tin khác và xoá mục từ thư mục.

Câu 46: Giới thiệu cấu trúc thư mục trong việc tổ chức lưu trữ tệp tin trên đĩa.

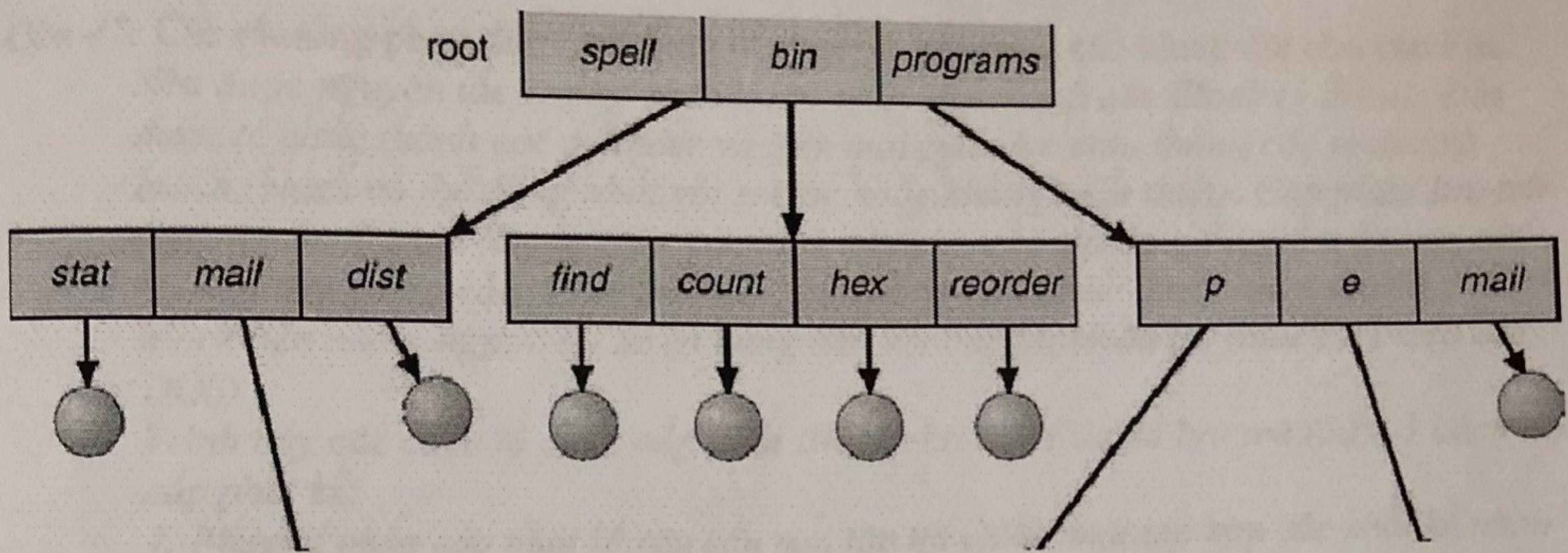
- Đĩa được chia thành một hay nhiều phân khu (partition) hay phân vùng (volumes). Diễn hình, mỗi đĩa trên hệ thống chứa ít nhất một phân khu. Phân khu này là cấu trúc cấp thấp mà các tập tin và thư mục định vị. Mỗi phân khu chứa thông tin về các tập tin trong nó. Thông tin này giữ trong những mục từ trong một thư mục thiết bị hay bảng mục lục phân vùng (volume table of contents). Thư mục thiết bị (được gọi đơn giản là thư mục) ghi thông tin như tên, vị trí, kích thước và kiểu-đối với tất cả tập tin trên phân khu (3đ)
- Cấu trúc thư mục đơn giản nhất là thư mục đơn cấp. Tất cả tập tin được chứa trong cùng thư mục (2đ)



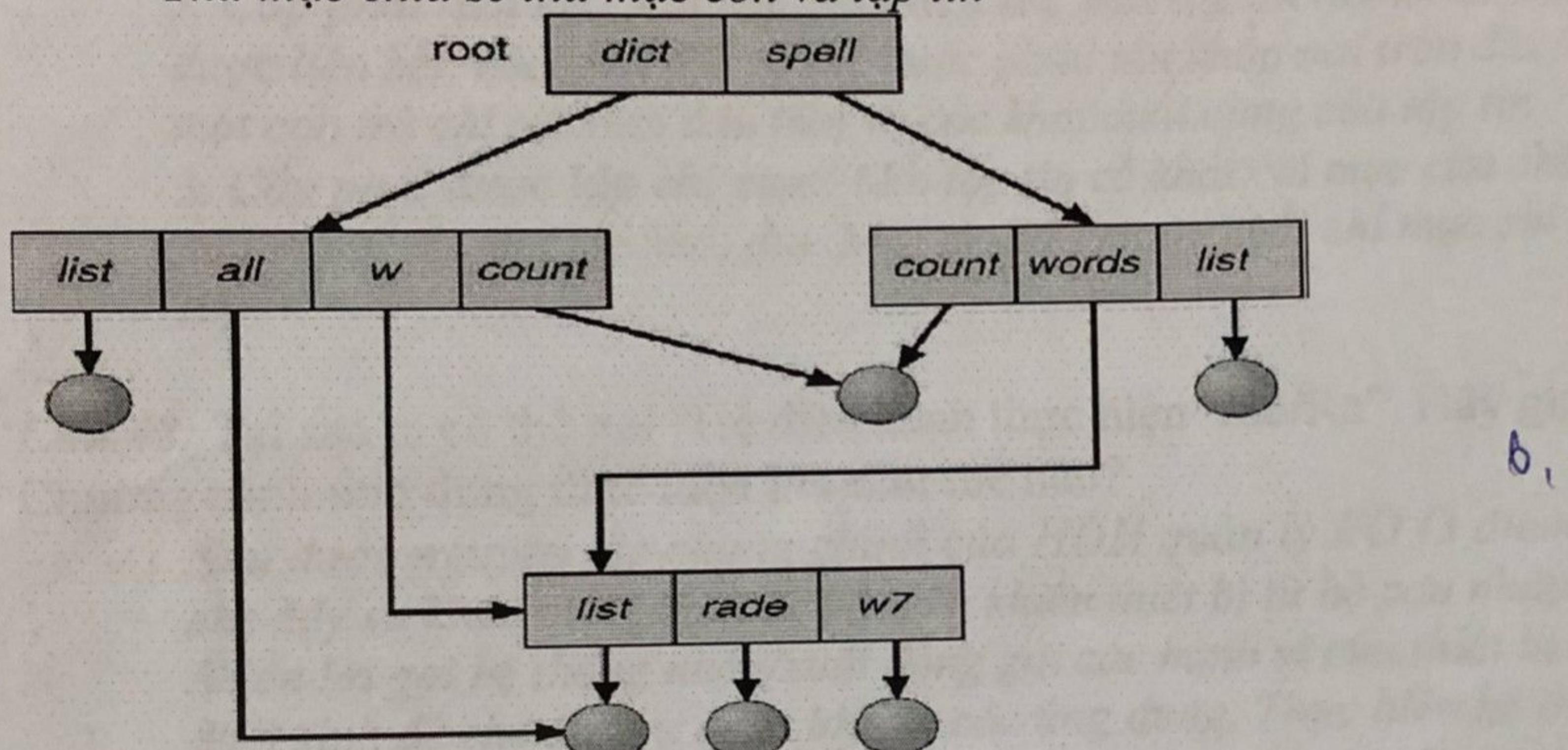
- Cấu trúc thư mục hai cấp, mỗi người dùng có thư mục tập tin riêng cho họ (user file directory-UFD). Mỗi UFD có một cấu trúc tương tự nhưng các danh sách chứa các tập tin của một người dùng



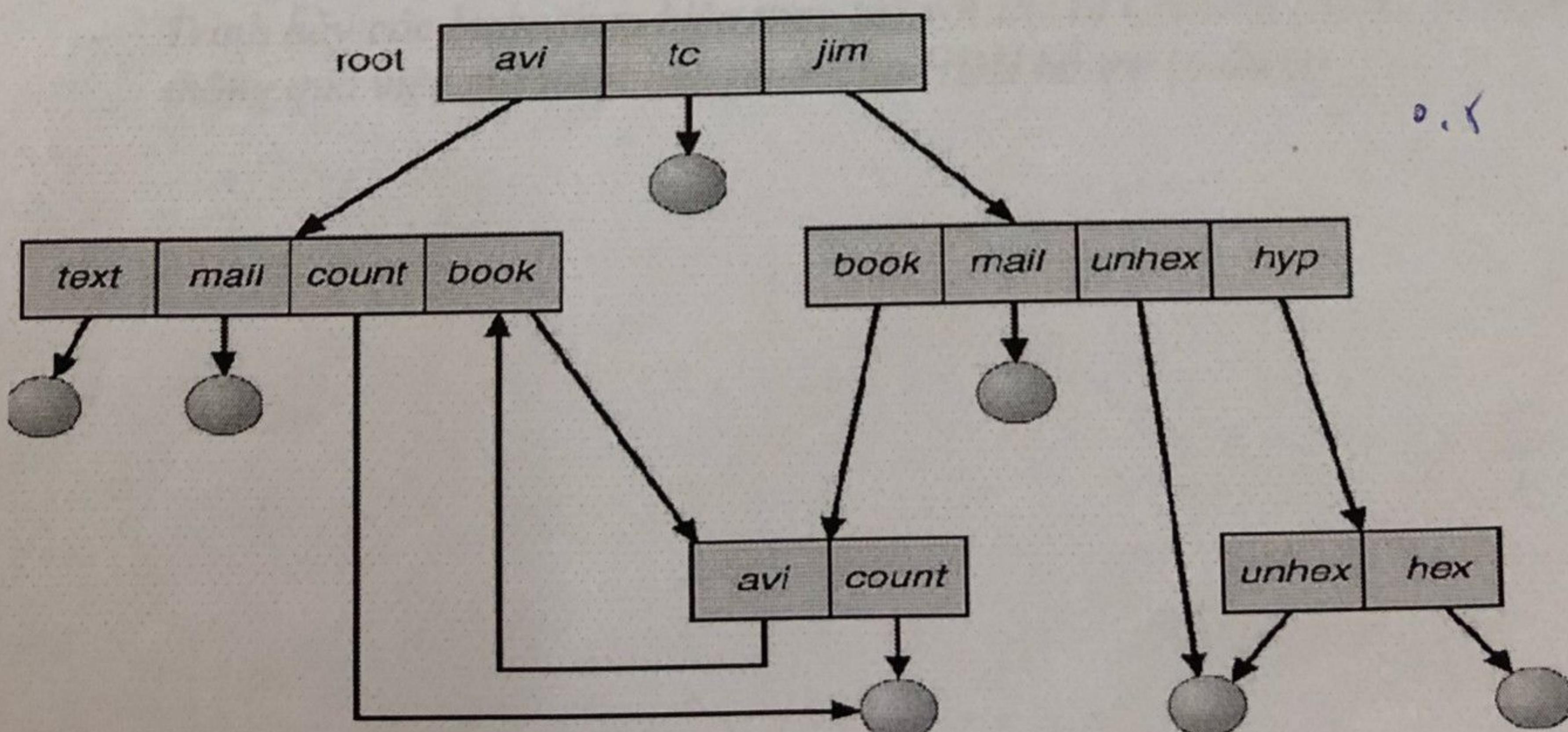
- Thư mục cấu trúc cây là trường hợp tổng quát của thư mục hai cấp. Cây có thư mục gốc. Mỗi tập tin trong hệ thống có tên đường dẫn duy nhất. Tên đường dẫn là đường dẫn từ gốc xuống tất cả thư mục con tới tập tin xác định



- Thư mục chia sẻ thư mục con và tập tin



- Cấu trúc thư mục dạng đồ thị tổng quát

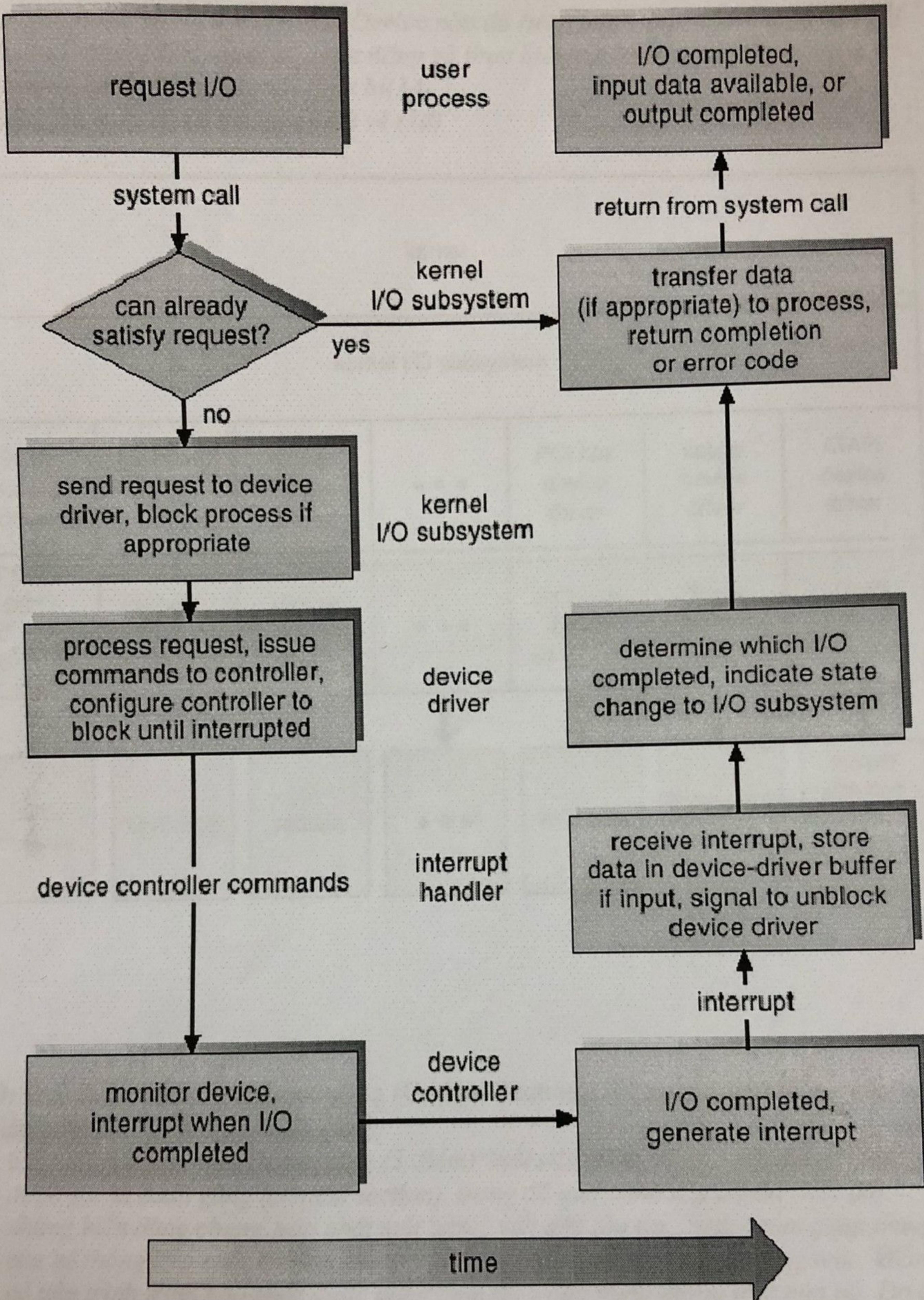


Câu 47: Các phương pháp được sử dụng để quản lí, cấp phát các block đĩa cho các File.

- *Nêu được nguyên tắc chung của việc tổ chức đĩa thành các Block (5 điểm):* Đĩa được tổ chức thành các cylinder và trên mỗi cylinder chia thành các sector và block (block có thể đồng nhất với sector hoặc không nhất thiết). Cấp phát/lưu trữ tệp tin trên đĩa tức là cấp phát các block này cho các tệp tin giống như Trang của bộ nhớ trong cho các tiến trình. Dữ liệu không nhất thiết được lưu trữ trên các block liền nhau, ngược lại sẽ có bảng liên kết các block đó lại (như FAT của MS DOS)
- *Trình bày các cách tổ chức cấp phát Block cho các File để lưu trữ (5đ):* 3 cách là cấp phát kè:
 1. Phương pháp cấp phát kè yêu cầu mỗi tập tin chiếm một tập hợp các khối kề nhau trên đĩa;
 2. Cấp phát liên kết: Với cấp phát liên kết, mỗi tập tin là một danh sách các khối đĩa được liên kết; các khối đĩa có thể được phân tán khắp nơi trên đĩa. Thư mục chứa một con trỏ chỉ tới khối đầu tiên và các khối cuối cùng của tập tin
 3. Cấp phát được lập chỉ mục: Mỗi tập tin có khối chỉ mục của chính nó, khối này là một mảng các địa chỉ khối đĩa. Mục từ thứ i trong khối chỉ mục chỉ tới khối i của tập tin.

Câu 48: Tại sao ta có thể nói “Hệ điều hành thực hiện Vào/Ra”. Hãy giải thích cơ chế Chương trình ứng dụng thực hiện I/O như thế nào?

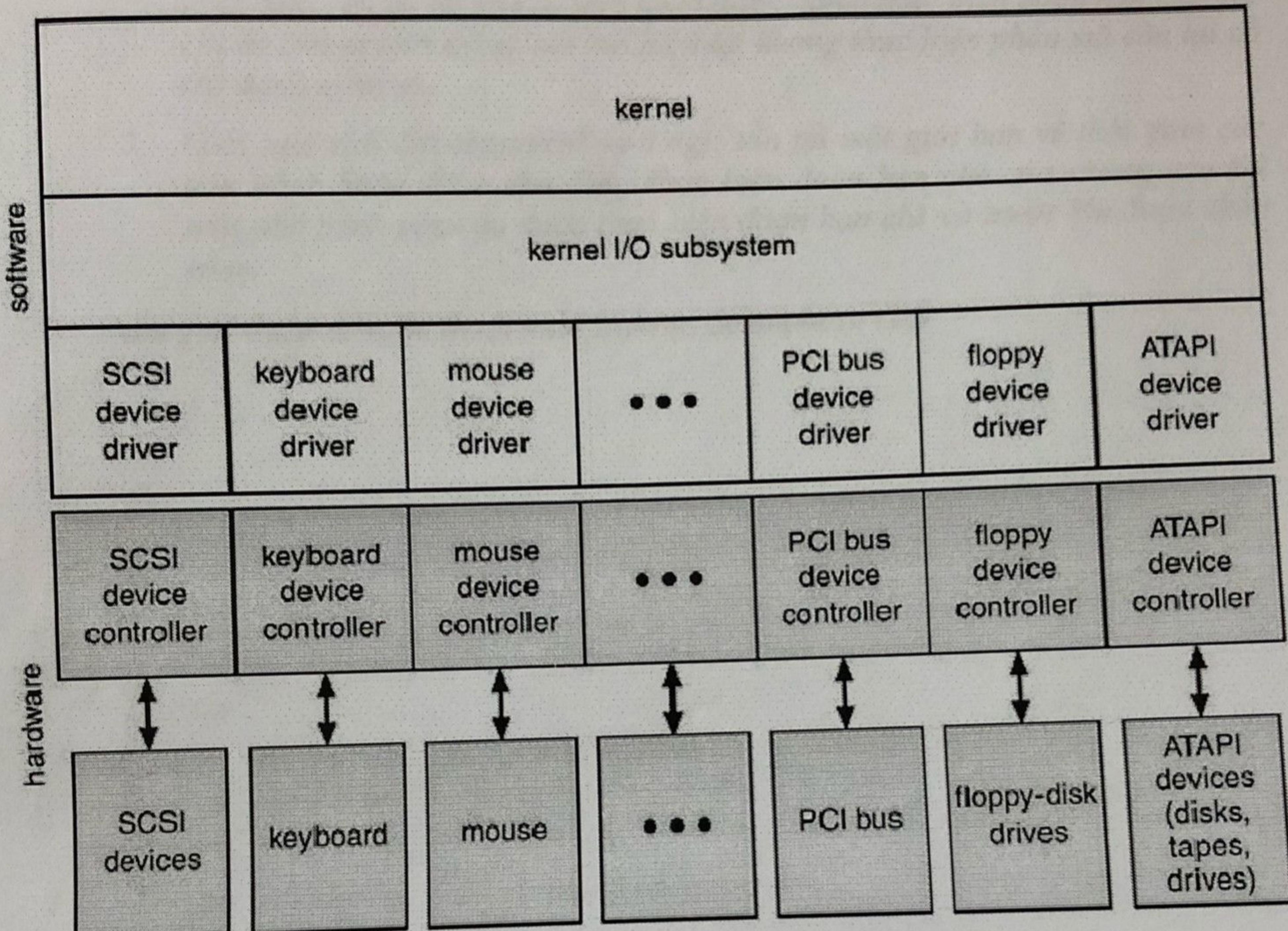
- *Nêu được nguyên tắc chính của HĐH quản lý I/O (3 điểm):* Thông qua việc che đậy sự khác biệt giữa các bộ điều khiển thiết bị từ hệ thống nhập/xuất của nhân, nhiều lời gọi hệ thống nhập/xuất đóng gói các hành vi của thiết bị trong một vài lớp phát sinh để che đậy sự khác biệt từ các ứng dụng. Thực hiện hệ thống con nhập/xuất độc lập với phần cứng đơn giản hóa công việc của người phát triển hệ điều hành. Cung cấp cho người sử dụng các tiện nghi và đồng nhất hóa phần cứng khác nhau dưới góc nhìn người lập trình (4đ).
- *Trình bày các bước thực hiện thao tác với I/O từ Chương trình của người sử dụng thông qua việc gọi tới thành phần của HĐH hỗ trợ (6 điểm)*



Câu 49: Phân biệt Device, Driver và Device Controller.

- **Trình bày và phân biệt được Device:** Là thành phần phần cứng cấu thành nên thiết bị cho phép các phần mềm có thể hoạt động tức là bộ phận chấp hành (3đ)
- **Device Controller:** Phần điều khiển ngoại vi là một thành phần của hệ thống máy tính và/hoặc của thiết bị - tức là nằm giữa – cũng là dạng phần cứng cho phép các tín hiệu đến/đi từ Ngoại vi với CPU. Thông thường chia bộ đệm và các thanh ghi điều khiển (3đ)

- Driver: Là trình điều khiển một Device nào đó (trên một Controller có thể kết nối nhiều Device khác nhau có chức năng và thực hiện rất khác nhau) cho phép một chương trình tương tác với thiết bị (3đ).
- Giải thích đủ rõ có thể qua hình vẽ (1đ)



Câu 50: Giải thích khái niệm Đoạn găng (CriticalSection). Nêu ngắn gọn biện pháp giải pháp điều phối tiến trình qua đoạn găng.

- Giải thích khái niệm Đoạn găng (3 điểm): Mỗi tiến trình có một phân đoạn mã, được gọi là **đoạn găng** (critical section), trong đó quá trình này có thể thay đổi những biến dùng chung, cập nhật một bảng, viết đến tập tin... Đặc điểm quan trọng của hệ thống là ở chỗ, khi một tiến trình đang thực thi trong vùng tương trục, không có tiến trình nào khác được phép thực thi trong vùng tương trục của nó. Do đó, việc thực thi của các đoạn găng bởi các tiến trình là sự loại trừ hỗn loạn. Vấn đề Critical Section là thiết kế một giao thức mà các tiến trình có thể dùng để cộng tác. Mỗi tiến trình phải yêu cầu quyền để đi vào đoạn găng của nó.
- Giới thiệu 3 điều kiện giải quyết vấn đề (2đ x 3): phải thỏa mãn các 3 điều kiện:

1. *Loại trừ lẫn nhau (Mutual exclusion):* Nếu tiến trình Pi đang thực hiện trong đoạn hạn chế, thì không có tiến trình nào có thể thực hiện trong đoạn hạn chế.
 2. *Tiến triển (progress):* Nếu không có tiến trình nào đang thực hiện trong đoạn hạn chế và có một số tiến trình đang muốn thực hiện đoạn hạn chế, thì chỉ có những tiến trình nào mà hiện tại không thực hiện phần mã còn lại có thể được xem xét.
 3. *Giới hạn chờ đợi (bounded waiting):* tồn tại một giới hạn về thời gian các tiến trình khác được cho phép thực hiện đoạn hạn chế của chúng sau khi một tiến trình yêu cầu được thực hiện đoạn hạn chế và trước khi được chấp nhận.
- Nếu giới thiệu sơ lược thuật toán Bakery, Semaphore (1đ)

HOANG BAO KHANH

Date of birth 11/09/1998
Gender Female
Phone 01628849522
Email khanhh.tmu@gmail.com
Address 105 , D2 , Ngã Tân , Cầu Giấy , Hà Nội
Website <https://www.facebook.com/xinh.jen>



EDUCATION HISTORY :

- 2013-2016:High School Graduate Trung Van High School
- 2016-up to present 2nd year student : thuongmai University
International Training Department
Major : Marketing and Sales
Grades A for English Business Management

SKILLS AND ABILITIES

- Language : English can communicate fluently in daily situations
Computer: Can use word, excel, powerpoint

INTERESTS

- Make up : I usually research style make-up in the free time and I often make- up for myself and my friends
- Cooking : In the weekend , I often cook meals for my family

REFERENCES

Provided upon request