

# HỆ ĐIỀU HÀNH

Phạm Đăng Hải  
haipd@soict.hut.edu.vn

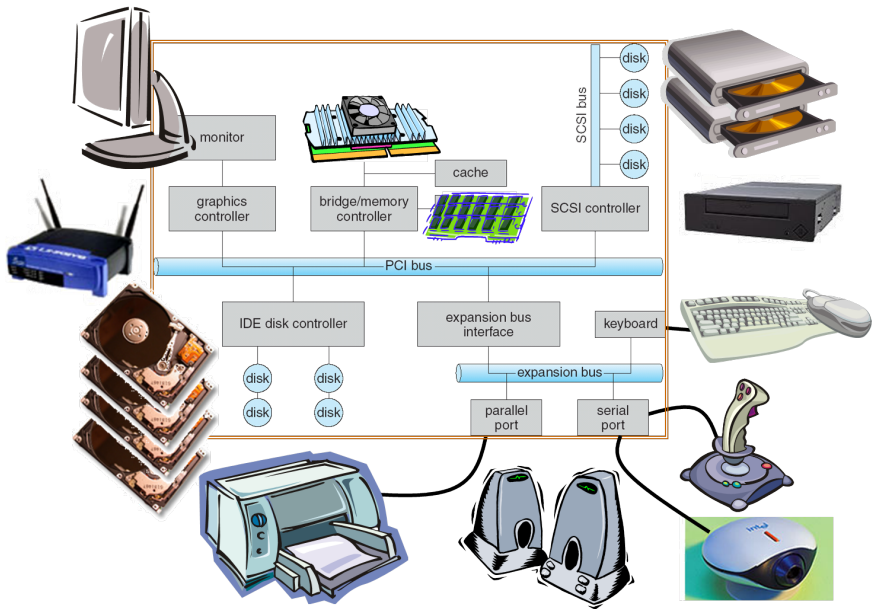
Bộ môn Khoa học Máy tính  
Viện Công nghệ Thông tin & Truyền Thông

Ngày 21 tháng 8 năm 2015



## Chương 5 Quản lý vào ra





(Nguồn: internet)

- 1 Nguyên tắc quản lý chung
- 2 Dịch vụ vào ra của hệ thống
- 3 Hệ thống vào ra đĩa



## Nội dung chính

- 1 Nguyên tắc quản lý chung
- 2 Dịch vụ vào ra của hệ thống
- 3 Hệ thống vào ra đĩa



## 1 Nguyên tắc quản lý chung

- Giới thiệu
- Ngắt và xử lý ngắt



## Thiết bị vào ra

- Đa dạng, nhiều loại thiết bị, mỗi loại có nhiều kiểu khác nhau
  - Quan điểm kỹ thuật: là các thiết bị với bộ VXL, motor, các linh kiện khác
  - Quan điểm lập trình: Giao diện như phần mềm để nhận lệnh, thực hiện và trả kết quả về

### Phân loại thiết bị ngoại vi

- Thiết bị khối (Đĩa từ, băng từ)
  - Thông tin được lưu trữ có kích thước cố định và địa chỉ riêng
  - Có thể đọc ghi một khối độc lập với khối khác
    - Tồn tại thao tác định vị thông tin (*seek*)
- Thiết bị ký tự (Máy in, Bàn phím, chuột,...)
  - Chấp nhận luồng ký tự, không có cấu trúc khối
  - Không có thao tác định vị thông tin
- Loại khác: Đồng hồ



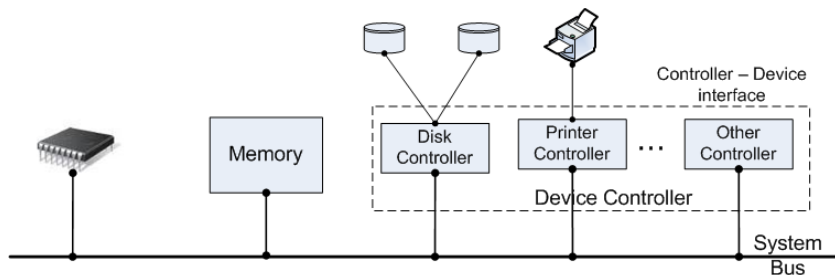
## Thiết bị điều khiển I

- Thiết bị ngoại vi (*Tbnv*) đa dạng và nhiều loại
  - CPU không biết hết  $\Rightarrow$  Không tồn tại tín hiệu riêng cho từng thiết bị
- Processor không điều khiển trực tiếp thiết bị
  - TBNV được nối với hệ thống qua thiết bị điều khiển (*D.C Device controller*)-Bộ điều khiển thiết bị
    - Các mạch điện tử được cắm trên các khe cắm (*slot*) của mainboard máy tính
  - Mỗi thiết bị điều khiển có thể điều khiển được 1,2,4,.. thiết bị ngoại vi
    - Tùy theo số giắc cắm có trên TBĐK
    - Nếu giao diện điều khiển chuẩn (ANSI, IEEE, ISO,...) có thể nối tới nhiều thiết bị khác
  - Mỗi TBĐK có các thanh ghi riêng để làm việc với CPU
    - Dùng các không gian địa chỉ đặc biệt cho các thanh ghi: **cổng vào ra**





## Thiết bị điều khiển II



## Thiết bị điều khiển III

- Giao diện TBĐK và TBNV: Giao diện mức rất thấp
  - Sector = 512bytes = 4096bits
  - Bộ điều khiển đĩa phải đọc/ghi các bit và nhóm lại thành các sector
- HĐH chỉ làm việc với các TBĐK
  - Thông qua các thanh ghi điều khiển của thiết bị
  - Các câu lệnh và tham số sẽ được đưa vào các thanh ghi điều khiển
  - Khi 1 lệnh được bộ điều khiển chấp nhận, CPU sẽ để cho bộ điều khiển hoạt động một mình và nó quay sang làm công việc khác
  - Khi lệnh thực hiện xong, bộ điều khiển sẽ báo cho CPU bằng tín hiệu ngắt
  - CPU sẽ lấy KQ và trạng thái thiết bị thông qua các thanh ghi điều khiển



## Trình điều khiển thiết bị (Device driver)

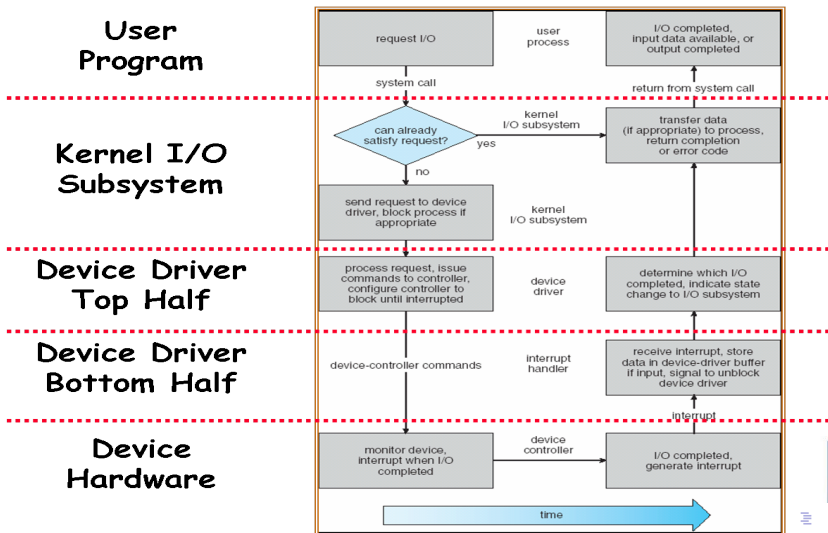
- Là đoạn mã trong nhân của hệ thống cho phép tương tác trực tiếp với phần cứng thiết bị
  - Cung cấp một giao diện chuẩn cho các thiết bị vào ra khác nhau
- Các trình điều khiển thiết bị thường được chia làm 2 mức
  - Mức cao : Được truy nhập qua các lời gọi hệ thống
    - Cài đặt tập lời gọi chuẩn như `open()`, `close()`, `read()`, `write()`...
    - Là giao diện của nhân HĐH với trình điều khiển
    - Luồng mức cao khởi động thiết bị thực hiện vào/ra và sau đó đặt luồng điều khiển tạm nghỉ
  - Mức thấp: Được thực hiện như một thủ tục ngắt
    - Đọc dữ liệu đầu vào, hoặc đưa khối dữ liệu tiếp theo ra ngoài
    - Đánh thức luồng tạm nghỉ mức trên khi vào/ra kết thúc



## 1. Nguyên tắc quản lý chung

## 1.1 Giới thiệu

## Chu kỳ của một yêu cầu vào ra



## Giao tiếp thiết bị ngoại vi với hệ điều hành

- Sau khi hệ điều hành gửi yêu cầu ra *tbnv*, HDH cần phải biết
  - Thiết bị ngoại vi hoàn thành yêu cầu vào ra
  - Thiết bị ngoại vi gặp lỗi
- Có thể thực hiện theo 2 phương pháp *Ngắt* và *thăm dò*



## Giao tiếp thiết bị ngoại vi với hệ điều hành

- Sau khi hệ điều hành gửi yêu cầu ra *tbnv*, HDH cần phải biết
  - Thiết bị ngoại vi hoàn thành yêu cầu vào ra
  - Thiết bị ngoại vi gặp lỗi
- Có thể thực hiện theo 2 phương pháp **Ngắt** và **thăm dò**
- Ngắt (I/O interrupts)
  - Tbnv phát sinh ra một tín hiệu ngắt để báo cho CPU biết
  - IRQ: Đường dẫn vật lý đến bộ quản lý ngắt
    - Ánh xạ các tín hiệu IRQ thành các vector ngắt
    - Gọi tới chương trình xử lý ngắt



## Giao tiếp thiết bị ngoại vi với hệ điều hành

- Sau khi hệ điều hành gửi yêu cầu ra *tbnv*, HDH cần phải biết
  - Thiết bị ngoại vi hoàn thành yêu cầu vào ra
  - Thiết bị ngoại vi gặp lỗi
- Có thể thực hiện theo 2 phương pháp **Ngắt** và **thăm dò**
- Ngắt (I/O interrupts)
  - Tbnv phát sinh ra một tín hiệu ngắt để báo cho CPU biết
  - IRQ: Đường dẫn vật lý đến bộ quản lý ngắt
    - Ánh xạ các tín hiệu IRQ thành các vector ngắt
    - Gọi tới chương trình xử lý ngắt
- Thăm dò (pooling)
  - HDH chu kỳ kiểm tra thanh ghi trạng thái của thiết bị
  - Lãng phí chu kỳ thăm dò nếu thao tác vào ra không thường xuyên



## Giao tiếp thiết bị ngoại vi với hệ điều hành

- Sau khi hệ điều hành gửi yêu cầu ra *tbnv*, HDH cần phải biết
  - Thiết bị ngoại vi hoàn thành yêu cầu vào ra
  - Thiết bị ngoại vi gặp lỗi
- Có thể thực hiện theo 2 phương pháp **Ngắt** và **thăm dò**
- Ngắt (I/O interrupts)
  - Tbnv phát sinh ra một tín hiệu ngắt để báo cho CPU biết
  - IRQ: Đường dẫn vật lý đến bộ quản lý ngắt
    - Ánh xạ các tín hiệu IRQ thành các vector ngắt
    - Gọi tới chương trình xử lý ngắt
- Thăm dò (pooling)
  - HDH chu kỳ kiểm tra thanh ghi trạng thái của thiết bị
  - Lãng phí chu kỳ thăm dò nếu thao tác vào ra không thường xuyên
- Các thiết bị hiện thời có thể kết hợp cả 2 phương pháp (*VD Các thiết bị mạng băng thông cao*)
  - **Ngắt** khi gói tin đầu tiên tới
  - **Thăm dò** với các gói tin tiếp theo cho tới khi vùng đệm rỗng



## 1 Nguyên tắc quản lý chung

- Giới thiệu
- Ngắt và xử lý ngắt



## Khái niệm ngắt

Ngắt là phương tiện để cho các thiết bị khác trong hệ thống báo cho processor biết trạng thái của nó

Ngắt là hiện tượng dừng đột xuất chương trình để chuyển sang thực hiện chương trình khác ứng với một sự kiện nào đó xảy ra

- Ngắt  $><$  chương trình con !?



## Phân loại ngắt

- Theo nguồn gốc
  - Ngắt bên trong
  - Ngắt bên ngoài
- Theo thiết bị
  - Ngắt cứng
  - Ngắt mềm
- Theo khả năng quản lý
  - Ngắt che được
  - Ngắt không che được
- Theo thời điểm ngắt
  - Yêu cầu
  - Báo cáo



## Xử lý ngắt

- ❶ Ghi nhận đặc trưng sự kiện gây ngắt vào ô nhớ cố định
- ❷ Ghi nhận trạng thái của tiến trình bị ngắt
- ❸ Chuyển địa chỉ của chương trình xử lý ngắt vào thanh ghi con trỏ lệnh
  - Sử dụng bảng vector ngắt (**IBM-PC**)
- ❹ Thực hiện chương trình xử lý ngắt
- ❺ Khôi phục lại tiến trình bị ngắt



## Nội dung chính

- 1 Nguyên tắc quản lý chung
- 2 Dịch vụ vào ra của hệ thống**
- 3 Hệ thống vào ra đĩa



## 2 Dịch vụ vào ra của hệ thống

- Vùng đệm (Buffer)
- Quản lý lỗi vào ra
- Kỹ thuật SPOOL

## Khái niệm chung

### Đặc trưng của thiết bị ngoại vi: hoạt động chậm

- Kích hoạt thiết bị
- Chờ đợi thiết bị đạt được trạng thái h/động thích hợp
- Chờ đợi các thao tác vào ra được thực hiện

### Đảm bảo hiệu năng của hệ thống cần

- Giảm số lượng thao tác vào ra, làm việc với từng khối dữ liệu
- Thực hiện song song thao tác vào ra với các thao tác khác
- Thực hiện trước các phép truy nhập

**Vùng đệm:** *Vùng nhớ trung gian, làm nơi lưu trữ thông tin trong các thao tác vào ra*



## Phân loại vùng đệm 1

- Vùng đệm vào
  - Có thể thực hiện ngay phép truy nhập dữ liệu
  - Ví dụ đọc đĩa
- Vùng đệm ra
  - Thông tin được đưa ra vùng đệm, khi nào vùng đệm đầy sẽ đưa ra thiết bị



## Phân loại vùng đệm 2

- Vùng đệm gắn với thiết bị
  - Được xây dựng khi mở thiết bị/file
  - Phục vụ riêng cho thiết bị bị xóa khi đóng thiết bị
  - Thích hợp khi các thiết bị có cấu trúc bản ghi vật lý khác nhau
- Vùng đệm gắn với hệ thống
  - Xây dựng khi khởi tạo hệ thống, không gắn với thiết bị cụ thể
  - Tồn tại trong suốt quá trình hoạt động của hệ thống
  - Mở file/thiết bị  $\Rightarrow$  Gắn với một vùng đệm có sẵn
  - Khi đóng file/thiết bị  $\Rightarrow$  vùng đệm được trả về hệ thống
  - Thích hợp khi các thiết bị có cấu trúc bản ghi vật lý chung
  - Tránh việc tạo xóa vùng đệm nhiều lần
  - Vùng đệm trở thành các tài nguyên găng  $\Rightarrow$  phải điều độ



## Phân loại vùng đệm 3

- Vùng đệm trung chuyển
  - Vùng đệm vào
  - Vùng đệm ra
- Vùng đệm xử lý
- Vùng đệm vòng tròn
  - Vùng đệm vào
  - Vùng đệm ra
  - Vùng đệm xử lý

## 2 Dịch vụ vào ra của hệ thống

- Vùng đệm (Buffer)
- Quản lý lỗi vào ra
- Kỹ thuật SPOOL



## Đặt vấn đề

- Lỗi luôn có thể xảy ra tại mọi bộ phận của hệ thống
  - Phần cứng: Intel 486?
  - Phần mềm: Win 98 ?
  - Thiết bị ngoại vi: Hay gặp lỗi do tác động của môi trường
- Xử lý lỗi  $\Rightarrow$  Trách nhiệm của hệ thống
  - Được quan tâm ngay tại giai đoạn thiết kế, chế tạo
    - Kiểm tra chẵn/lẻ
    - Tổng kiểm tra
  - Các phép kiểm tra do phần cứng/phần mềm đảm nhiệm
    - Yêu cầu sử dụng linh hoạt các phép kiểm tra



## Xử lý lỗi

- Phát hiện lỗi, hệ thống cố gắng khắc phục bằng cách t/hiện lại nhiều lần
  - Nếu lỗi ổn định  $\Rightarrow$  Cố gắng khôi phục lại thông tin ban đầu
- Trường hợp lưu trữ, để đảm bảo chất lượng thông tin
  - Thiết bị điều khiển đọc lại kết quả vừa lưu trữ
    - So sánh với thông tin gốc/So sánh 2 tổng kiểm tra
    - Kết quả báo cho hệ thống để có xử lý tương ứng  
 $\Rightarrow$  Lặp lại thao tác/ Thông báo lỗi
- Thiết bị vào ra thường cung cấp mã trả về (*return code*)
  - Phân tích và đánh giá dựa trên mã nhận được



## 2 Dịch vụ vào ra của hệ thống

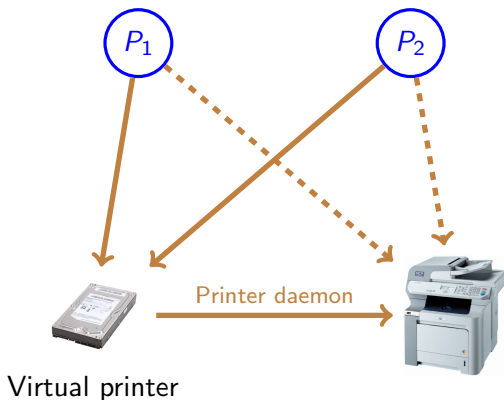
- Vùng đệm (Buffer)
- Quản lý lỗi vào ra
- Kỹ thuật SPOOL

## Kỹ thuật SPOOL (Simultaneous peripheral operation on-line)

- Trên phương diện lập trình, thiết bị vào ra là
  - Trạm nhận các yêu cầu từ chương trình và thực hiện
  - Trả các mã trạng thái để hệ thống phân tích
- Vậy có thể dùng phần mềm mô phỏng các thiết bị vào ra
  - Thiết bị vào ra có thể coi như tiến trình
    - Được điều độ theo quy tắc quản lý tiến trình
- Mục đích
  - Mô phỏng quá trình điều khiển, quản lý thiết bị ngoại vi
    - Kiểm tra hoạt động của các thiết bị đang chế tạo
  - Tạo hiệu ứng sử dụng song song cho các thiết bị chỉ tuần tự



## Kỹ thuật SPOOL: Ví dụ máy in ảo



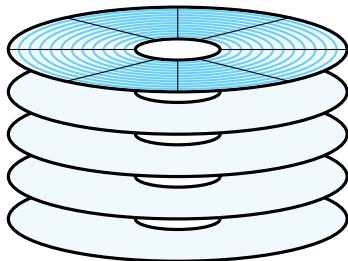
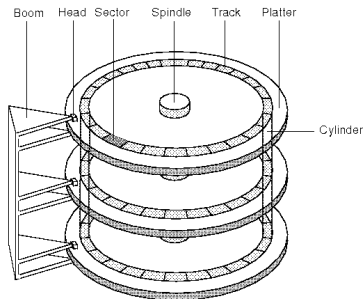


- 1 Nguyên tắc quản lý chung
- 2 Dịch vụ vào ra của hệ thống
- 3 Hệ thống vào ra đĩa**

### 3 Hệ thống vào ra đĩa

- Cấu trúc đĩa từ
- Điều phối truy nhập đĩa

## Cấu trúc



- Mô hình hóa như mảng một chiều các khối logic
  - Khối logic là đơn vị trao đổi nhỏ nhất
- Ánh xạ liên tiếp các khối logic tới các sector của đĩa
  - Khối 0 là sector đầu mặt 0 rãnh/Cylinder ngoài cùng
    - Ánh xạ theo trật tự: Sector → Header → Track/Cylinder
  - Ít phải dịch chuyển đầu từ khi đọc các sector kế tiếp nhau

## Vấn đề truy nhập đĩa

- Hệ điều hành có trách nhiệm sử dụng hiệu quả phần cứng
  - Với đĩa: Thời gian truy nhập nhanh và băng thông cao
- Băng thông được tính dựa trên
  - Tổng số bytes đã trao đổi
  - Khoảng thời gian từ y/cầu dịch vụ đầu cho tới khi hoàn thành
- Thời gian truy nhập gồm 2 phần

**Thời gian định vị (seek time)** Thời gian dịch chuyển đầu từ tới cylinders chứa sector cần truy nhập

**Độ trễ quay (Rotational latency)** Thời gian chờ đợi để đĩa quay tới sector cần truy nhập

- Mục đích: cực tiểu hóa thời gian định vị
  - Thời gian định vị  $\approx$  khoảng cách dịch chuyển
- **Hàng đợi yêu cầu**
  - Đĩa và bộ đ/khiển sẵn sàng, y/cầu truy nhập đc thực hiện ngay
  - Đĩa/bộ đ/khiển chưa sẵn sàng, yêu cầu đc đặt trong hàng đợi
  - Hoàn thành một yêu cầu truy nhập đĩa, lựa chọn y/cầu nào?



### 3 Hệ thống vào ra đĩa

- Cấu trúc đĩa từ
- Điều phối truy nhập đĩa

## Các thuật toán

Tồn tại một số thuật toán điều phối dịch vụ cho yêu cầu vào ra đĩa

- 1 **FCFS**: First Come First Served
- 2 **SSTF**: Shortest Seek Time First
- 3 **SCAN**
- 4 **C-SCAN**: Circular SCAN
- 5 **LOOK/C-LOOK**

### Giả thiết

- Các yêu cầu truy nhập 98, 183, 37, 122, 14, 124, 65, 67
- Đầu đọc đang ở cylinder **53**



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

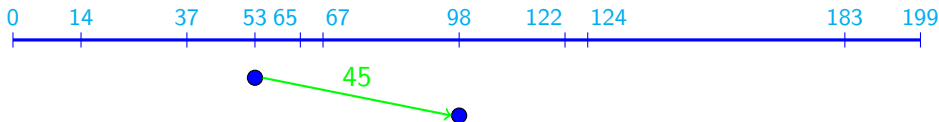
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

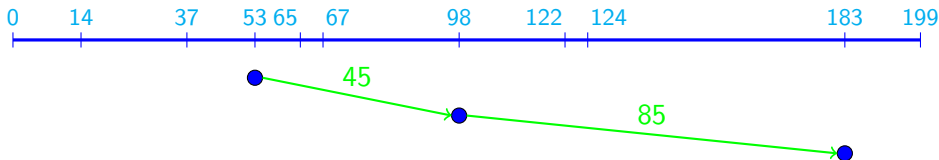




## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

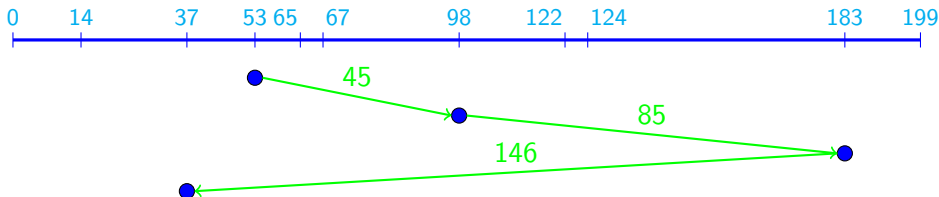
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

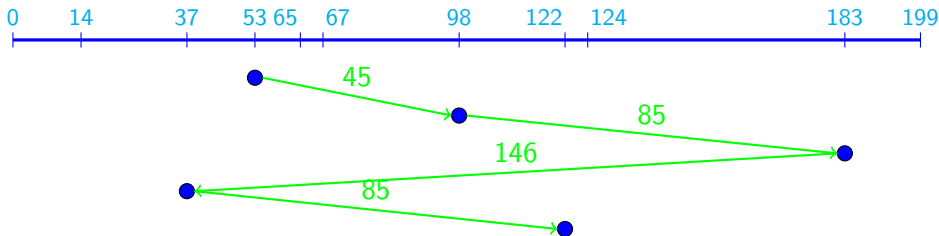
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

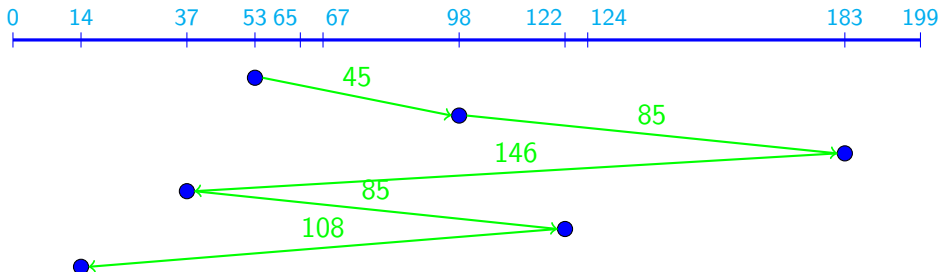
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

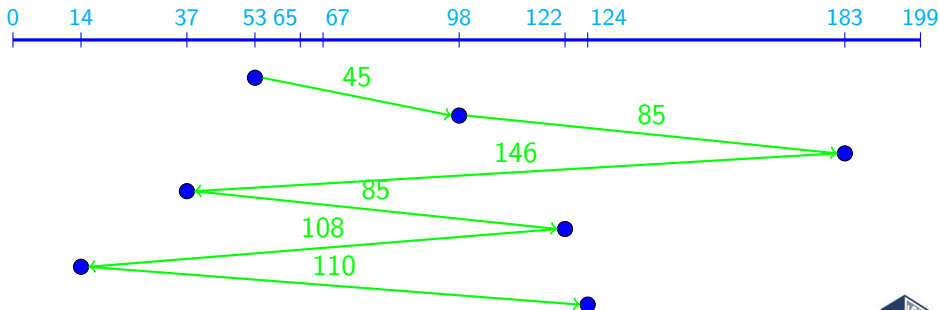
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

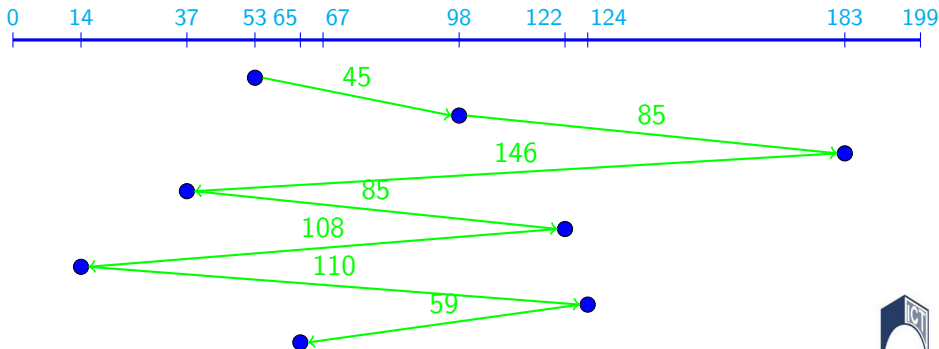
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

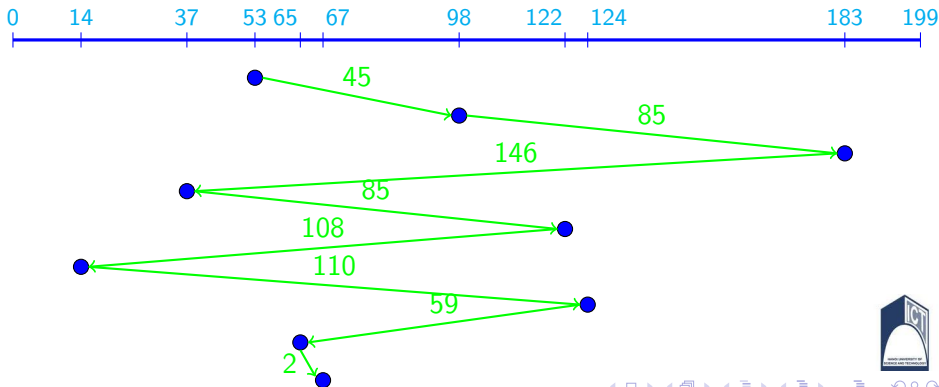
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

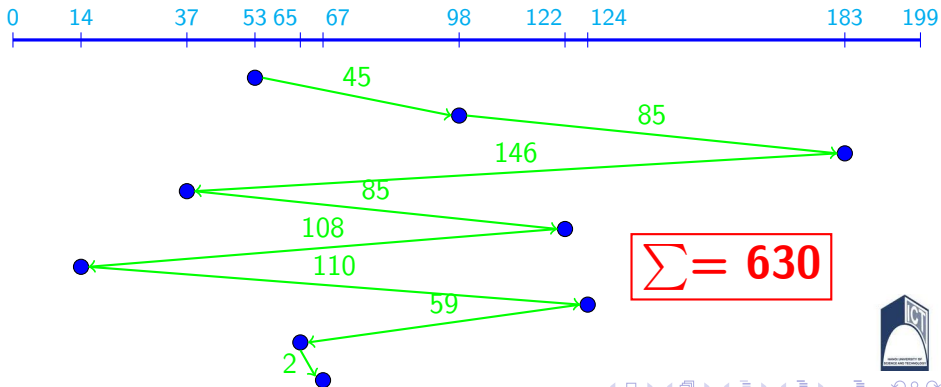
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

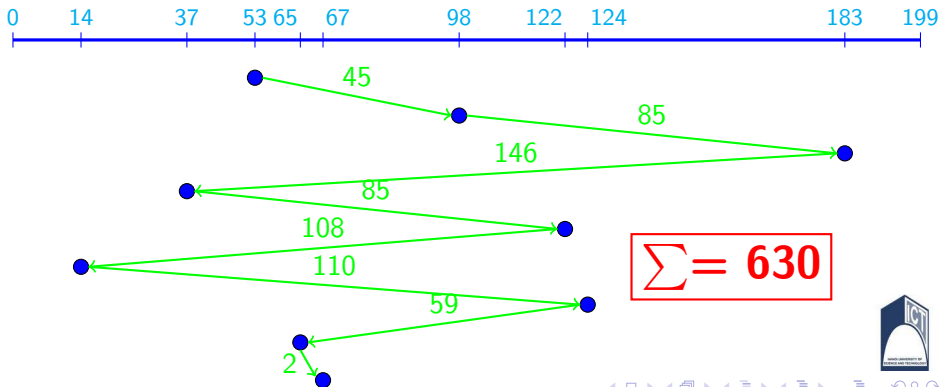




## FCFS

Truy nhập theo trật tự yêu cầu  $\Rightarrow$  Không hiệu quả

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

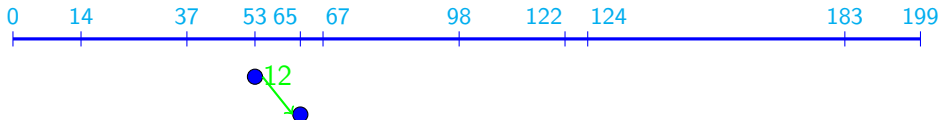
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

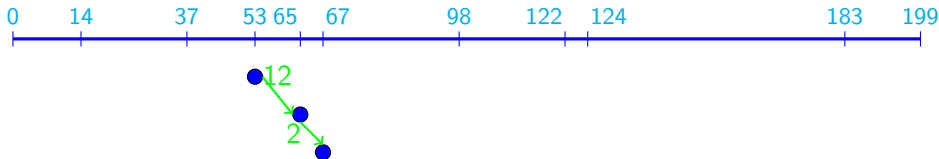
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

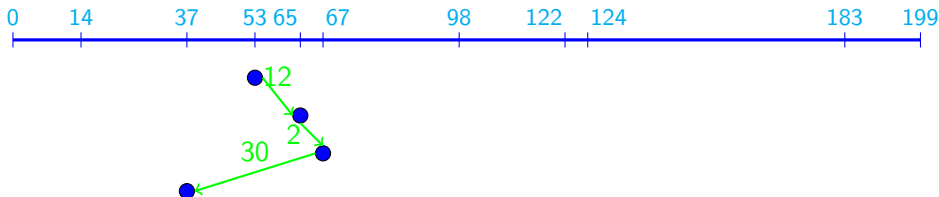
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

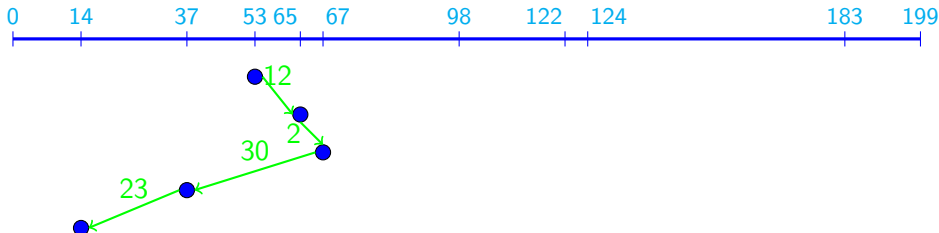
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

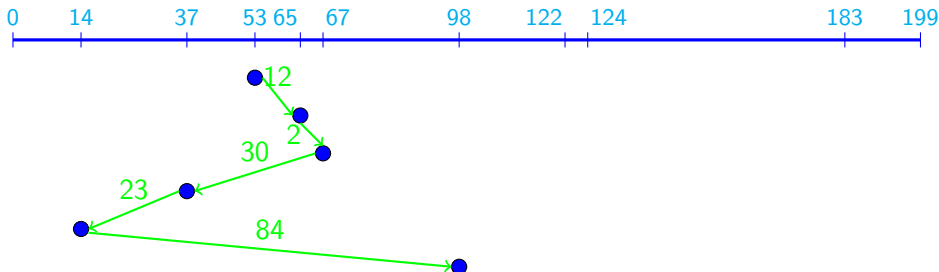
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



# SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

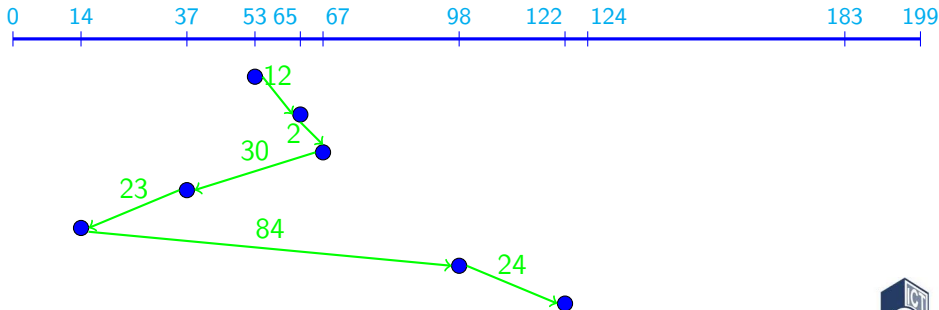
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

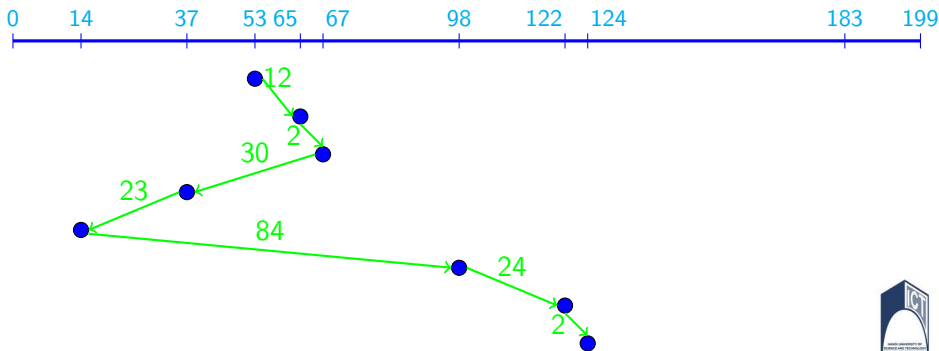




# SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

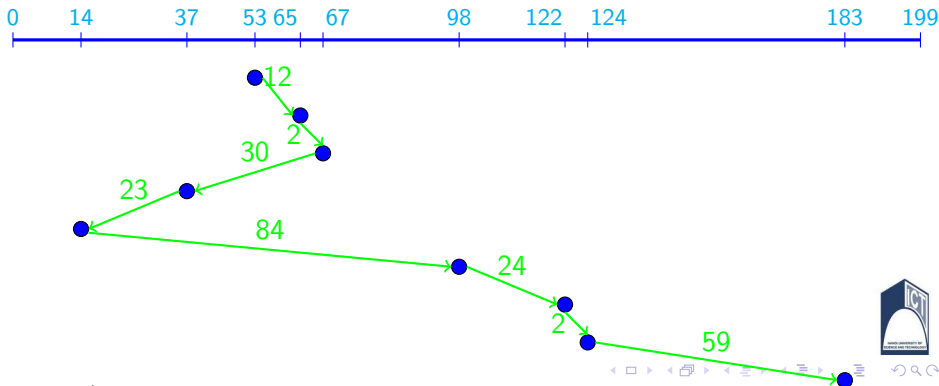
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (*tương tự SJF*)

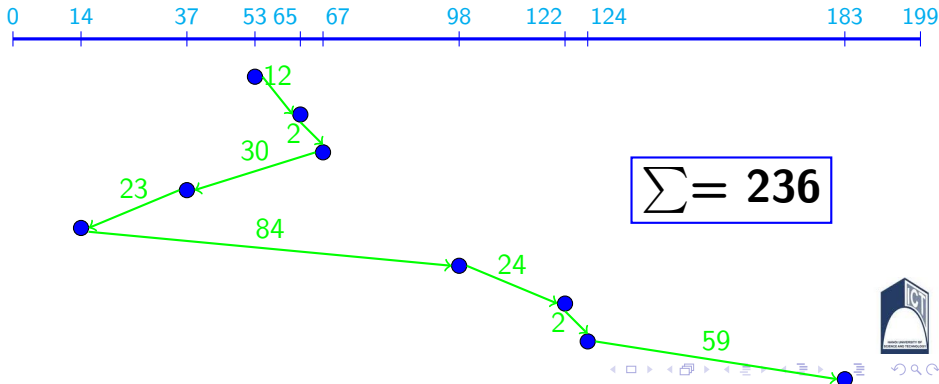
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (tương tự SJF)

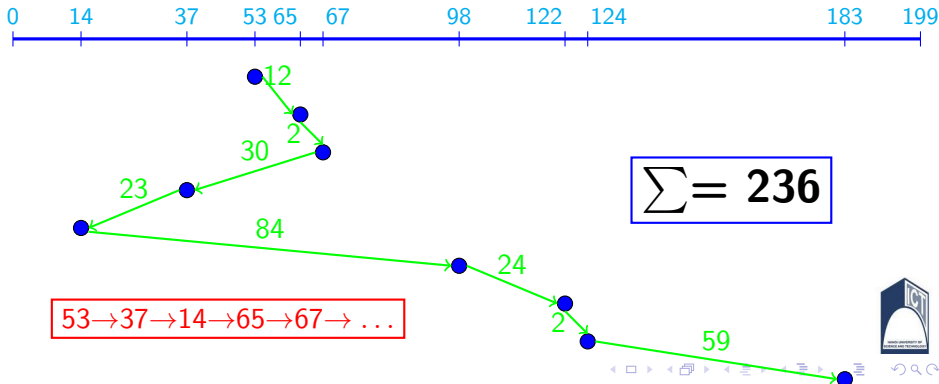
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SSTF

Chọn truy nhập có t/gian định vị từ vị trí hiện tại nhỏ nhất  $\Rightarrow$  Có thể tồn tại y/c phải đợi vô hạn do y/c mới x/hiện gần đầu đọc hơn (tương tự SJF)

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

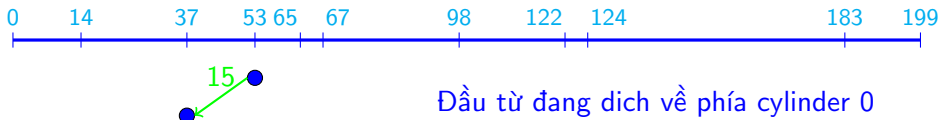


Đầu từ đang dịch về phía cylinder 0

## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

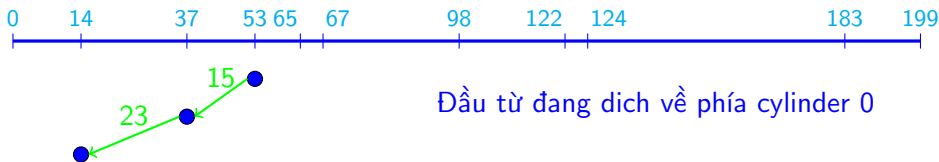
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

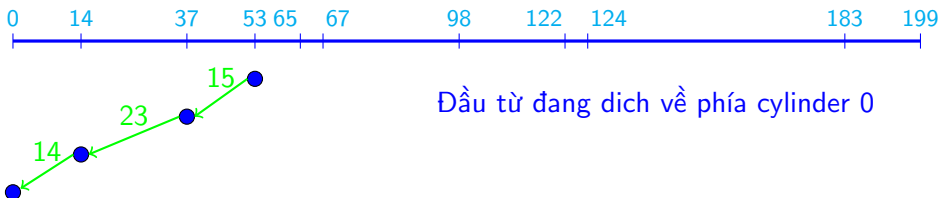
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

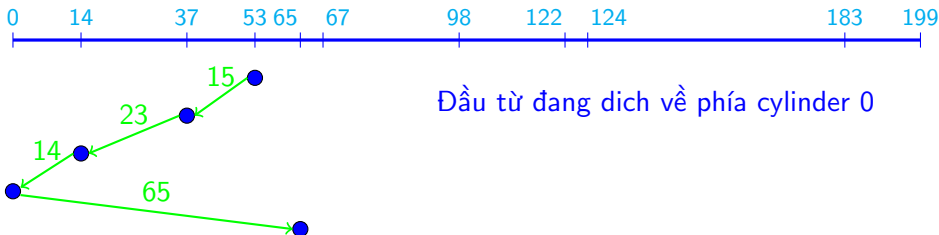




## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

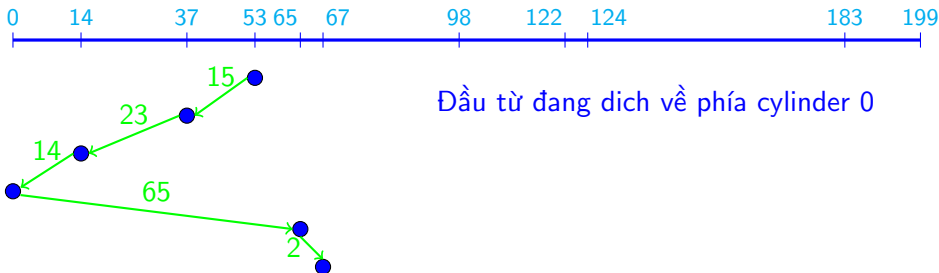
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

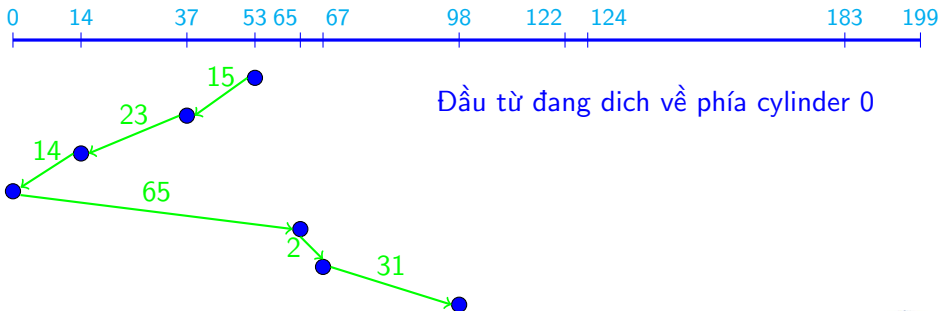
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

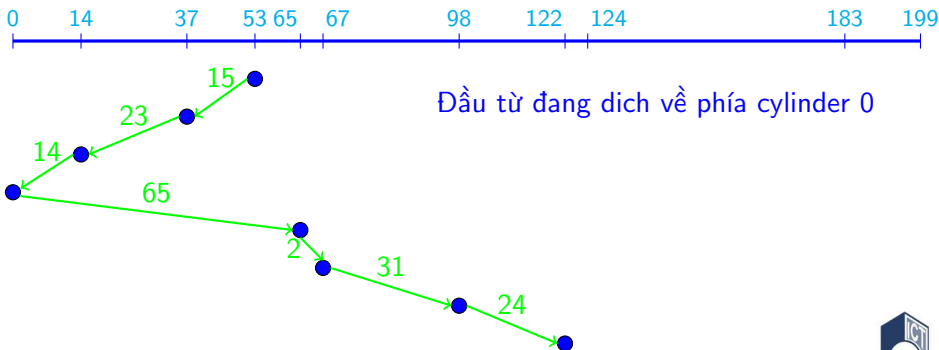
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

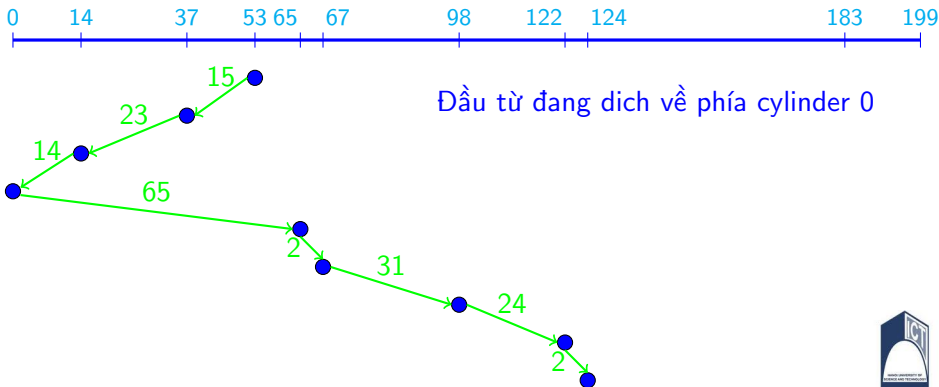
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

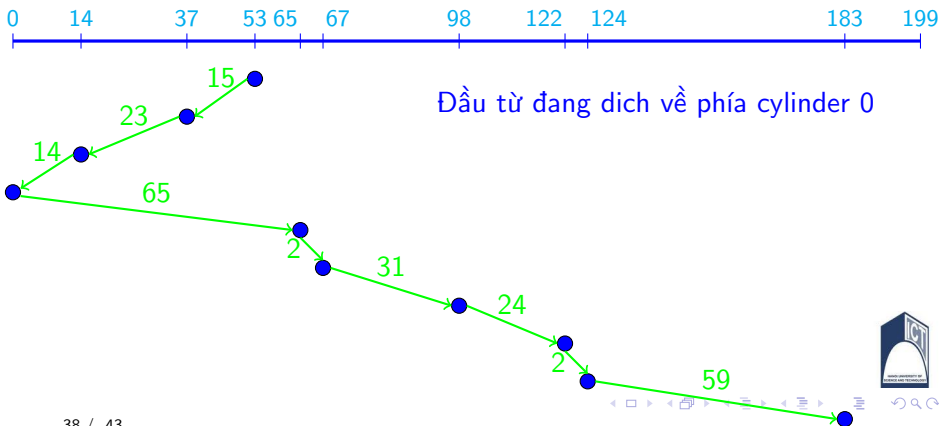
Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## SCAN

Đầu từ dịch chuyển từ cylinder ngoài cùng đến cylinder trong cùng và quay ngược lại. Phục vụ cho các yêu cầu gặp trên đường đi

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## C-SCAN

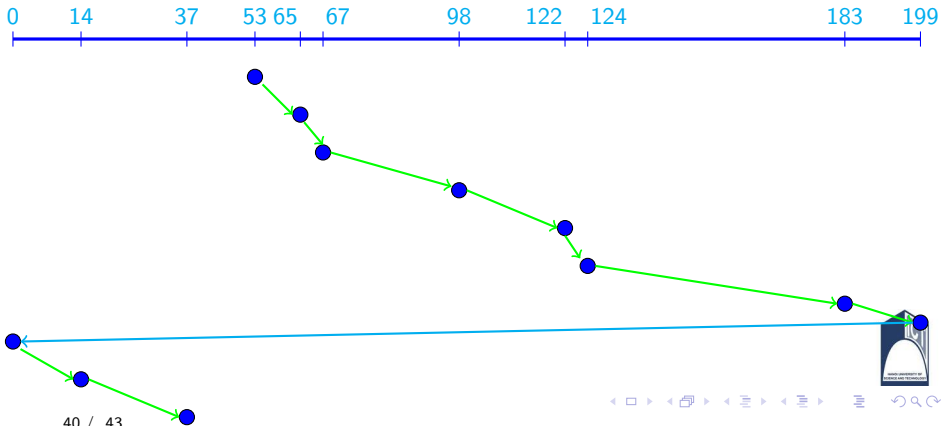
**Nguyên tắc:** *Xử lý các cylinders như một danh sách nối vòng: Cylinder ngoài cùng nối tiếp với cylinder trong cùng*

- Đầu từ d/chuyển từ cylinder ngoài cùng đến cylinder trong cùng
  - Phục vụ cho các yêu cầu gặp trên đường đi
- Khi tới Cylinder trong cùng, quay ngược lại Cylinder ngoài cùng
  - Không phục vụ cho các yêu cầu gặp trên đường đi
- **Nhận xét:** Thu được thời gian đợi đồng nhất hơn thuật toán SCAN
  - Khi đầu đọc đạt tới một phía của đĩa (*cylinders trong cùng, ngoài cùng*), mật độ các yêu cầu xuất hiện ở phía bên kia sẽ lớn hơn so với vị trí hiện tại (*do vừa đi qua*). Các yêu cầu này cũng đợi lâu hơn  $\Rightarrow$  Quay ngay lập tức về phía bên kia



## C-SCAN: Ví dụ

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67

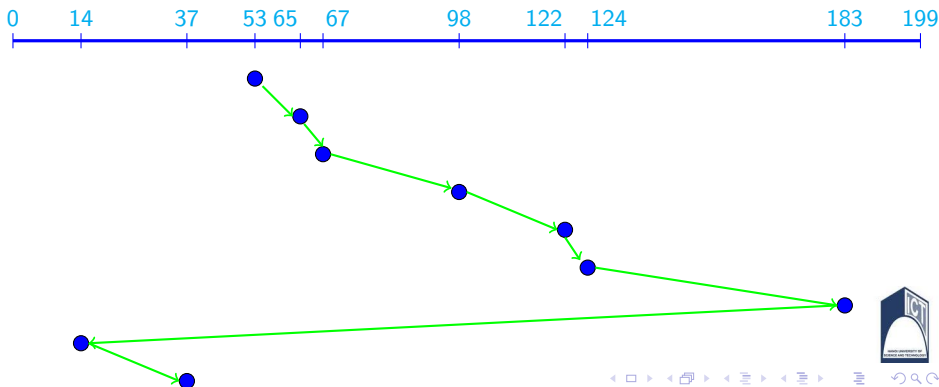




## LOOK/ C-LOOK

Phiên bản của SCAN/C-SCAN: Đầu đọc không di chuyển tới các cylinders ngoài/trong cùng mà chỉ đến các y/cầu xa nhất về 2 phía rồi quay lại ngay

Yêu cầu truy nhập cylinder : 98, 183, 37, 122, 14, 124, 65, 67



## Vấn đề lựa chọn thuật toán

- SSTF: Phổ biến, hiệu quả hơn FCFS
- SCAN/C-SCAN hoạt động tốt hơn cho hệ thống có nhiều yêu cầu truy nhập đĩa
  - Không gặp vấn đề "**starrvation**: chờ đợi quá lâu"
- Hiệu quả của các thuật toán phụ thuộc mạnh về số lượng và kiểu các yêu cầu
- Yêu cầu truy xuất đĩa bị ảnh hưởng mạnh bởi các phương pháp phân phối đĩa cho file
  - Phân phối liên tục: đưa ra các y/cầu truy xuất lân cận nhau
  - Phân phối liên kết/ chỉ mục: có thể gồm các khối được phân bố rộng rãi trên đĩa
- Thuật toán điều phối truy nhập đĩa có thể được viết như những modul riêng biệt của HDH cho phép có thể thay thế bởi các thuật toán khác khi cần thiết
- Cả SSTF và LOOK đều có thể là lựa chọn hợp lý cho thuật toán mặc định



# Kết luận

## 1 Nguyên tắc quản lý chung

- Giới thiệu
- Ngắt và xử lý ngắt

## 2 Dịch vụ vào ra của hệ thống

- Vùng đệm (Buffer)
- Quản lý lỗi vào ra
- Kỹ thuật SPOOL

## 3 Hệ thống vào ra đĩa

- Cấu trúc đĩa từ
- Điều phối truy nhập đĩa

