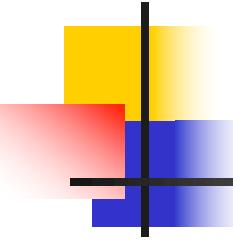


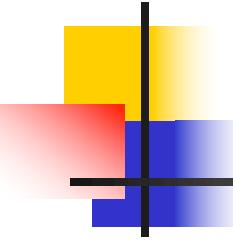
Bài 1: Cơ bản về lập trình

TS. Trịnh Tuấn Đạt
Viện CNTT-TT, ĐHBK Hà Nội



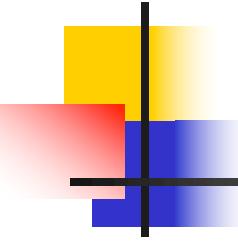
Nội dung

1. Chức năng của hệ thống máy tính
2. Cấu trúc của hệ thống máy tính
3. Bộ xử lý trung tâm (CPU)
4. Hoạt động của chương trình máy tính
5. Ngôn ngữ lập trình



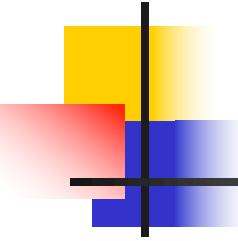
Nội dung

1. **Chức năng của hệ thống máy tính**
2. Cấu trúc của hệ thống máy tính
3. Bộ xử lý trung tâm (CPU)
4. Hoạt động của chương trình máy tính
5. Ngôn ngữ lập trình



1. Chức năng của hệ thống máy tính

- Xử lý dữ liệu
- Lưu trữ dữ liệu
- Trao đổi dữ liệu
- Điều khiển



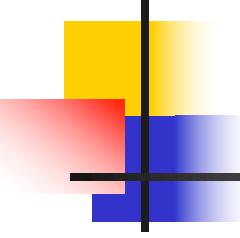
1. Chức năng của hệ thống máy tính

- Xử lý dữ liệu:

- Chức năng quan trọng nhất của máy tính
- Dữ liệu có thể có rất nhiều dạng khác nhau và có yêu cầu xử lý khác nhau.

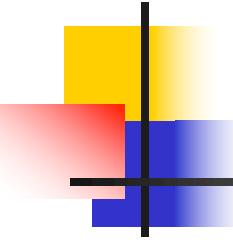
- Lưu trữ dữ liệu:

- Dữ liệu đưa vào máy tính được xử lý ngay hoặc có thể được lưu trong bộ nhớ.
- Khi cần chúng sẽ được lấy ra xử lý.



1. Chức năng của hệ thống máy tính

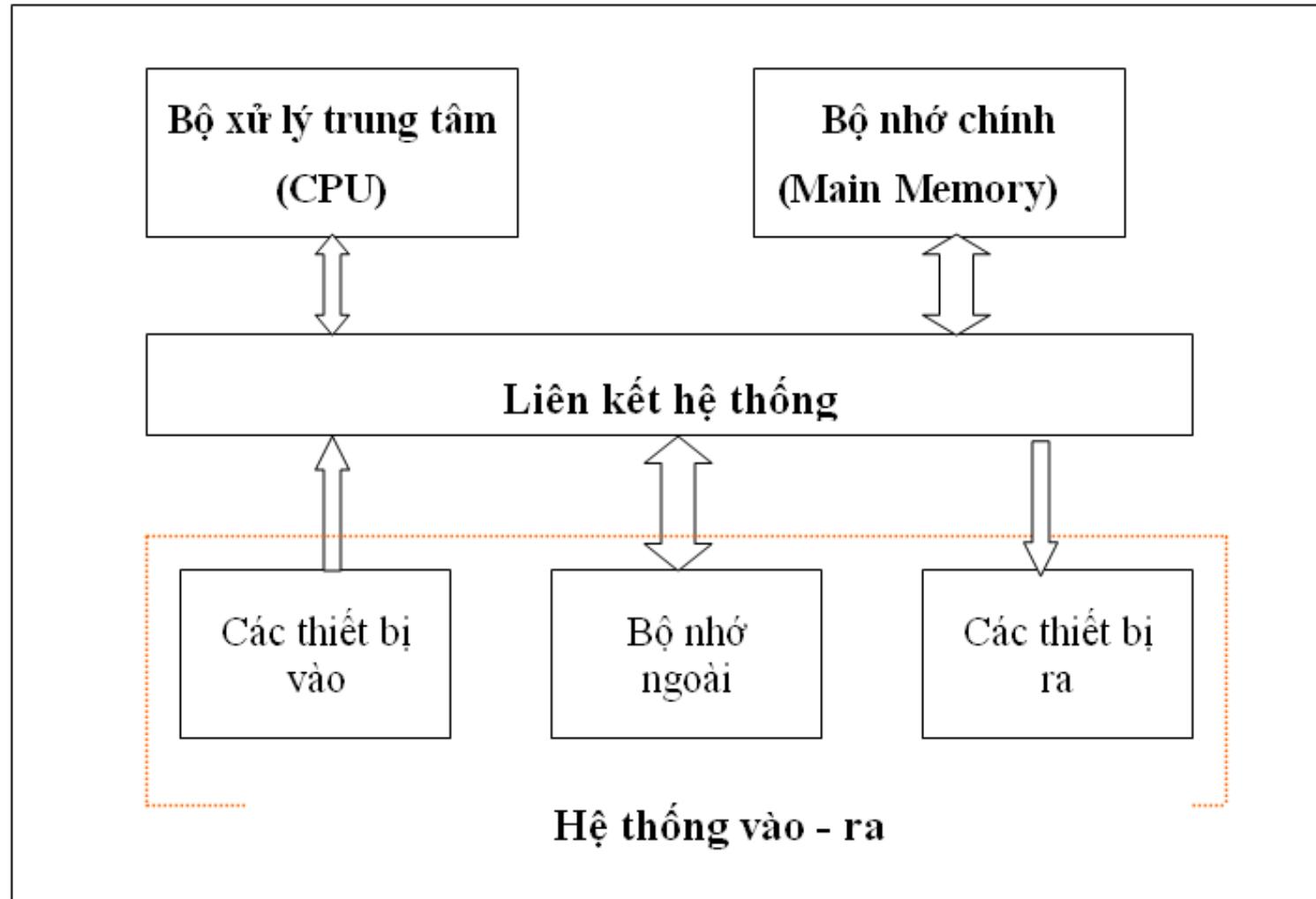
- Trao đổi dữ liệu:
 - Trao đổi dữ liệu giữa các thành phần bên trong và bên ngoài máy tính → Quá trình vào ra (input-output)
 - Các thiết bị vào-ra: nguồn cung cấp dữ liệu hoặc nơi tiếp nhận dữ liệu.
 - Dữ liệu được vận chuyển trên khoảng cách xa gọi là *truyền dữ liệu* (data communication).
- Điều khiển:
 - Máy tính cần phải điều khiển ba chức năng trên

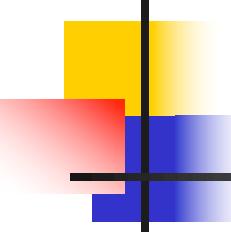


Nội dung

1. Chức năng của hệ thống máy tính
2. **Cấu trúc của hệ thống máy tính**
3. Bộ xử lý trung tâm (CPU)
4. Hoạt động của chương trình máy tính
5. Ngôn ngữ lập trình

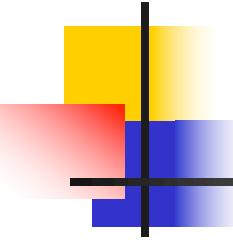
2. Cấu trúc của hệ thống máy tính





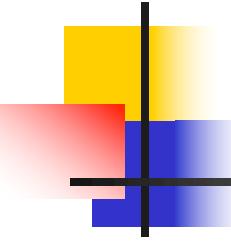
2. Cấu trúc của hệ thống máy tính

- Bộ xử lý trung tâm – CPU (Central Processor Unit)
 - Điều khiển các hoạt động của máy tính và thực hiện xử lý dữ liệu.
- Bộ nhớ chính (Main Memory)
 - Lưu trữ chương trình và dữ liệu.
- Hệ thống vào ra (Input-Output System):
 - Trao đổi thông tin giữa máy tính và thế giới bên ngoài
- Liên kết hệ thống (System Interconnection):
 - Kết nối và vận chuyển thông tin giữa CPU, bộ nhớ chính và hệ thống vào ra của máy tính với nhau.



Nội dung

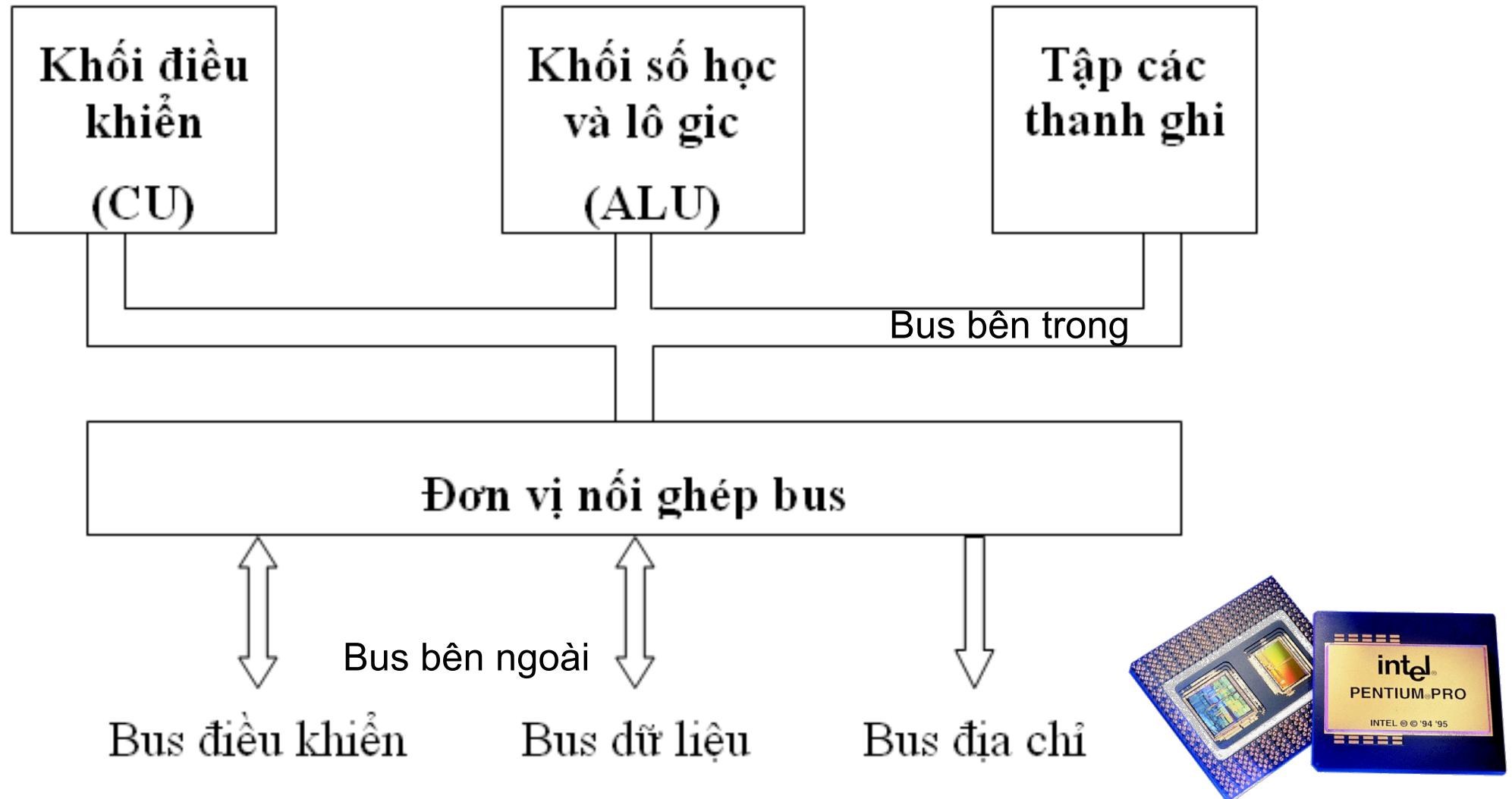
1. Chức năng của hệ thống máy tính
2. Cấu trúc của hệ thống máy tính
3. **Bộ xử lý trung tâm (CPU)**
4. Hoạt động của chương trình máy tính
5. Ngôn ngữ lập trình

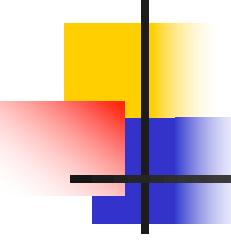


3. Bộ xử lý trung tâm - CPU

- Chức năng
 - Điều khiển hoạt động của toàn bộ hệ thống máy tính
 - Xử lý dữ liệu
- Nguyên tắc hoạt động: CPU hoạt động theo chương trình nằm trong bộ nhớ chính, bằng cách:
 - Nhận lệnh từ bộ nhớ chính
 - Giải mã lệnh và phát các tín hiệu điều khiển thực thi lệnh
 - CPU có thể trao đổi dữ liệu với bộ nhớ chính hay hệ thống vào-ra.
 - Thực hiện lệnh
 - Ghi kết quả

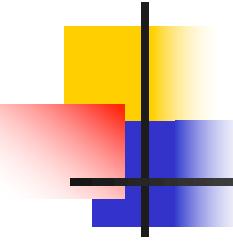
3. Bộ xử lý trung tâm (CPU)





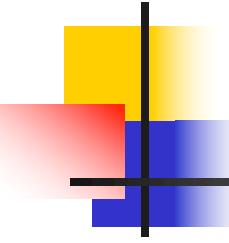
3. Bộ xử lý trung tâm (CPU)

- Control Unit (CU): Điều khiển hoạt động của máy tính theo chương trình đã định sẵn
- Arithmetic Logic Unit (ALU): Thực hiện các phép toán số học và các phép toán logic trên các dữ liệu cụ thể
- Register File (RF): Lưu trữ các thông tin tạm thời phục vụ cho hoạt động của CPU
- Internal bus: Kết nối các thành phần bên trong CPU với nhau
- Bus Interface Unit: Kết nối và trao đổi thông tin giữa các bus bên trong với các bus bên ngoài với nhau



Nội dung

1. Chức năng của hệ thống máy tính
2. Cấu trúc của hệ thống máy tính
3. Bộ xử lý trung tâm (CPU)
4. **Hoạt động của chương trình máy tính**
5. Ngôn ngữ lập trình

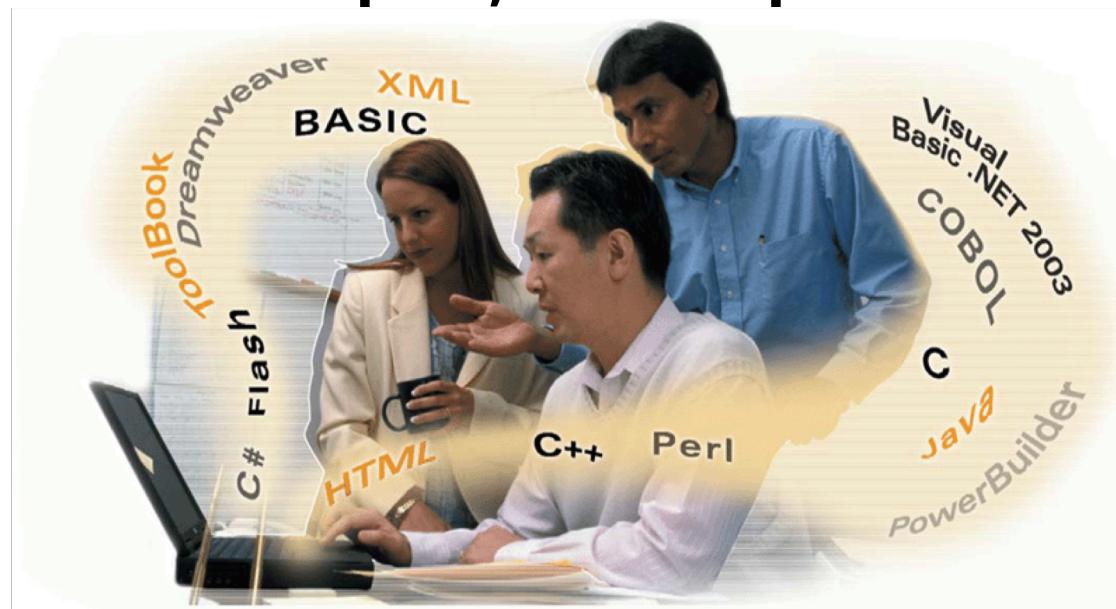


4. Hoạt động của chương trình máy tính

- Máy tính: phần cứng (thiết bị) + phần mềm
- Phần mềm điều khiển phần cứng. Phần mềm cơ bản nhất là hệ điều hành
- Mỗi phần mềm giải quyết bài toán (nhiệm vụ) nào đó
- Lập trình: là viết chương trình/phần mềm

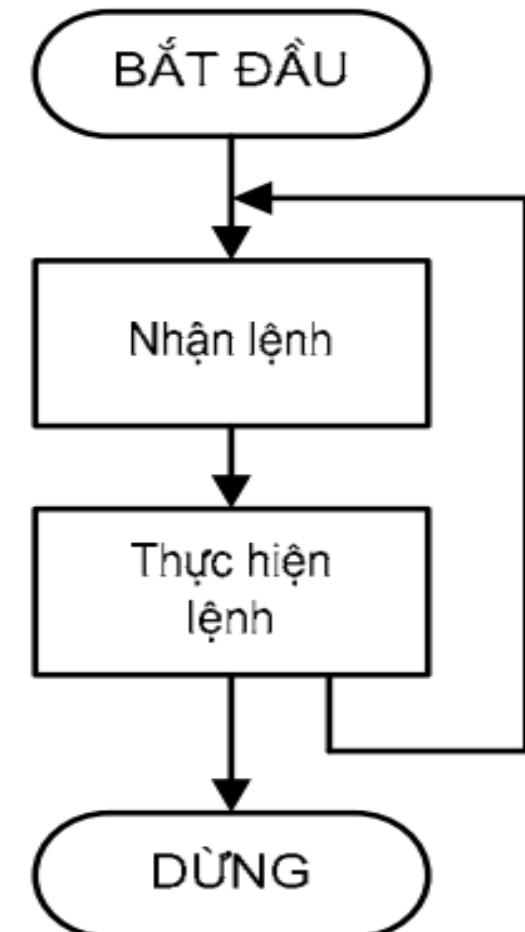
4. Hoạt động của chương trình máy tính

- Chương trình máy tính (computer program):
Tập hợp các lệnh chỉ dẫn cho máy tính thực hiện nhiệm vụ
- Ngôn ngữ lập trình (programming language):
Dùng để viết các lệnh, chỉ thị



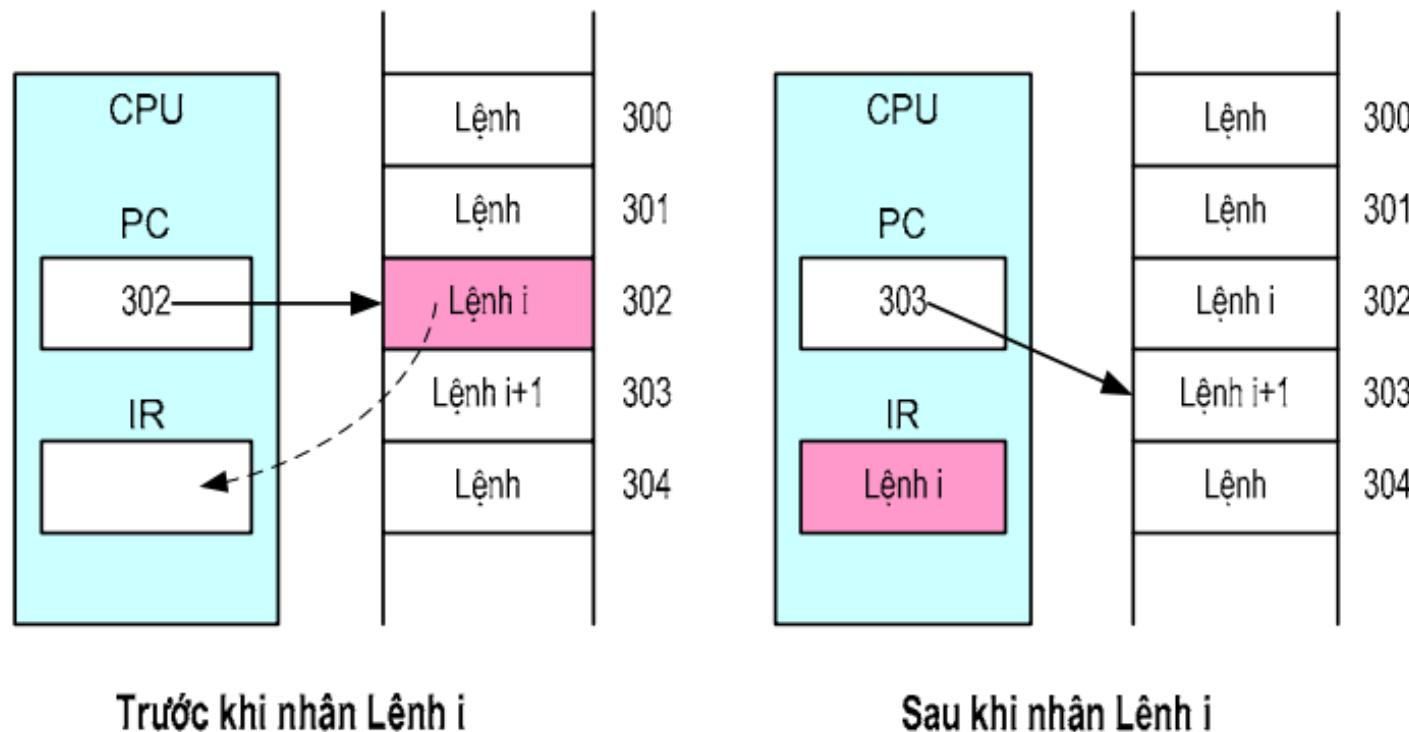
4. Hoạt động của chương trình máy tính

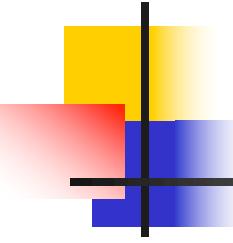
- Chương trình máy tính được nạp vào bộ nhớ chính (primary memory) như là một tập các lệnh viết bằng ngôn ngữ mà máy tính hiểu được, tức là một dãy tuần tự các số nhị phân (binary digits).



4. Hoạt động của chương trình máy tính

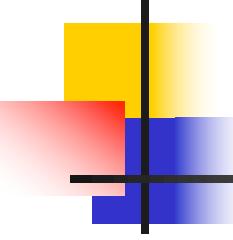
- Bắt đầu mỗi chu trình lệnh, CPU nhận lệnh từ bộ nhớ chính.
 - PC (Program Counter): thanh ghi giữ địa chỉ của lệnh sẽ được nhận
 - Lệnh được nạp vào thanh ghi lệnh IR (Instruction Register)
- Sau khi lệnh được nhận vào, nội dung PC tự động tăng để trỏ sang lệnh kế tiếp





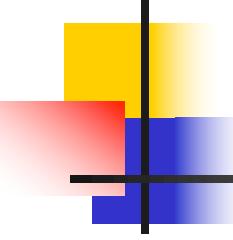
Nội dung

1. Chức năng của hệ thống máy tính
2. Cấu trúc của hệ thống máy tính
3. Bộ xử lý trung tâm (CPU)
4. Hoạt động của chương trình máy tính
5. **Ngôn ngữ lập trình**



5. Ngôn ngữ lập trình (NNLT)

- Một NNLT là 1 hệ thống các ký hiệu dùng để liên lạc, trao đổi với máy tính nhằm thực thi một nhiệm vụ tính toán.
- Các thành phần căn bản của 1 NNLT:
 - Cú pháp (syntax): luật dùng để ghép các ký hiệu thành câu lệnh, thành chương trình hợp lệ về mặt cấu trúc
 - Ngữ nghĩa (semantic): luật dùng để ghép các ký hiệu thành câu lệnh, thành chương trình có ý nghĩa
- Có rất nhiều NNLT, khoảng 1000 ngôn ngữ (60's đã có hơn 700)
 - phần lớn là các ngôn ngữ hàn lâm, có mục đích riêng hay phạm vi ứng dụng hạn chế
 - Ngôn ngữ máy
 - Ngôn ngữ assembly
 - Các ngôn ngữ khác



Ngôn ngữ máy

- Máy tính chỉ nhận các tín hiệu điện tử - có, không có - tương ứng với các dòng bits.
- Một chương trình ở dạng đó gọi là mã máy (machine code).
- Ban đầu chúng ta phải dùng machine code để viết chương trình:
→ Quá phức tạp, giải quyết các bài toán lớn là không tưởng

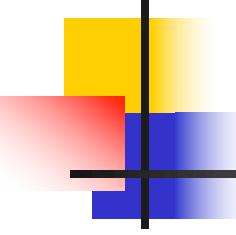
```
23fc 0000 0001 0000 0040
```

```
0cb9 0000 000a 0000 0040
```

```
6e0c
```

```
06b9 0000 0001 0000 0040
```

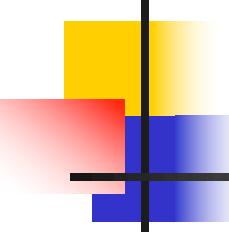
```
60e8
```



Ngôn ngữ ASSEMBLY

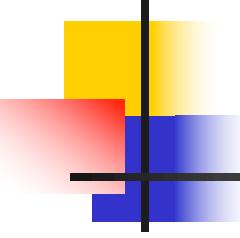
- Là bước đầu tiên của việc xây dựng cơ chế viết chương trình tiện lợi hơn – thông qua các ký hiệu, từ khóa và cả mã máy.
- Tất nhiên, để chạy được các chương trình này thì phải chuyển thành thành machine code.
- Vẫn còn phức tạp, cải thiện không đáng kể

```
movl    #0x1, n
compare:
    cmpl    #oxa, n
    cgt     end_of_loop
    acddl   #0x1, n
    bra     compare
end_of_loop:
```



Các loại ngôn ngữ lập trình - Theo mức độ trừu tượng

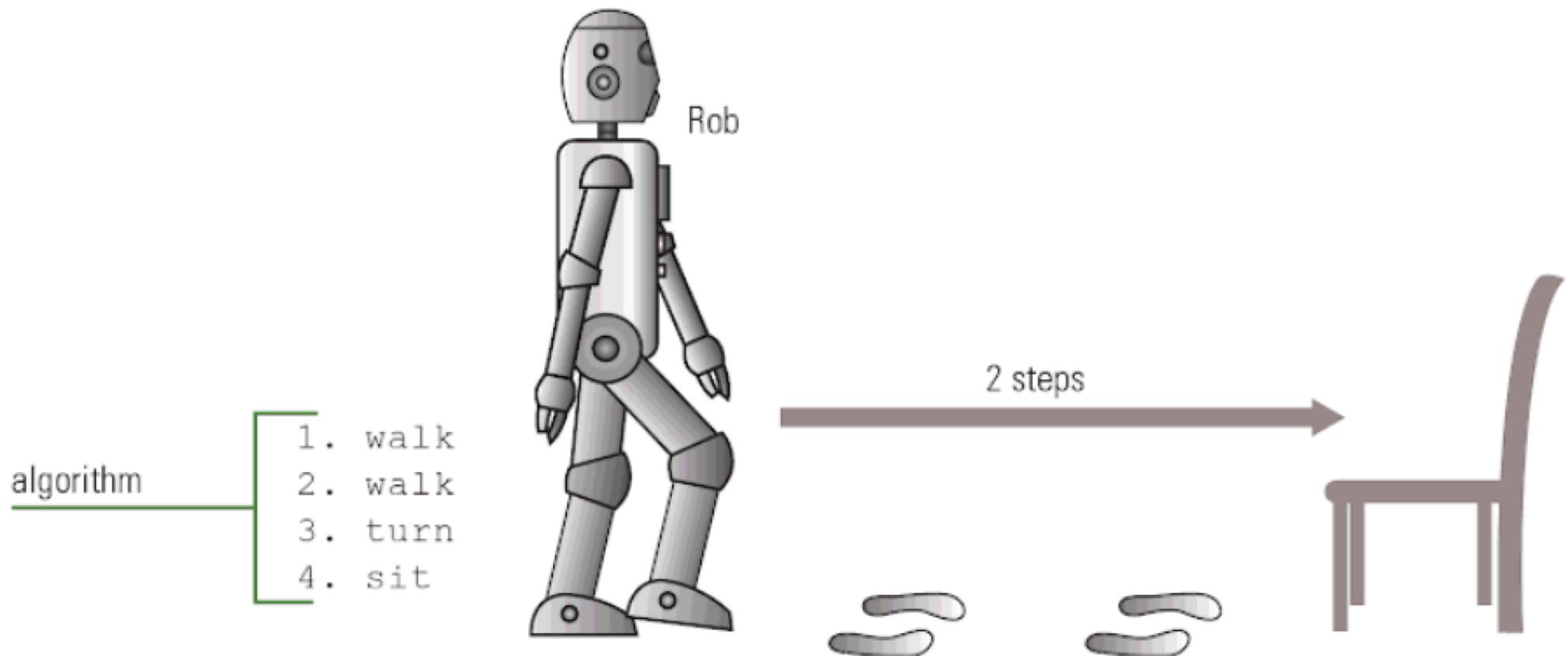
Level	Instructions	Memory handling
Low level languages	Dạng bits – giống các lệnh machine	Truy cập và cấp phát trực tiếp bộ nhớ
High level languages	Dùng các biểu thức và các dòng điều khiển xác định	Truy cập và cấp phát bộ nhớ qua các lệnh, toán tử - operators
Very high level languages	Hoàn toàn trừu tượng, độc lập phần cứng	Che dấu hoàn toàn việc truy cập và tự động cấp phát bộ nhớ



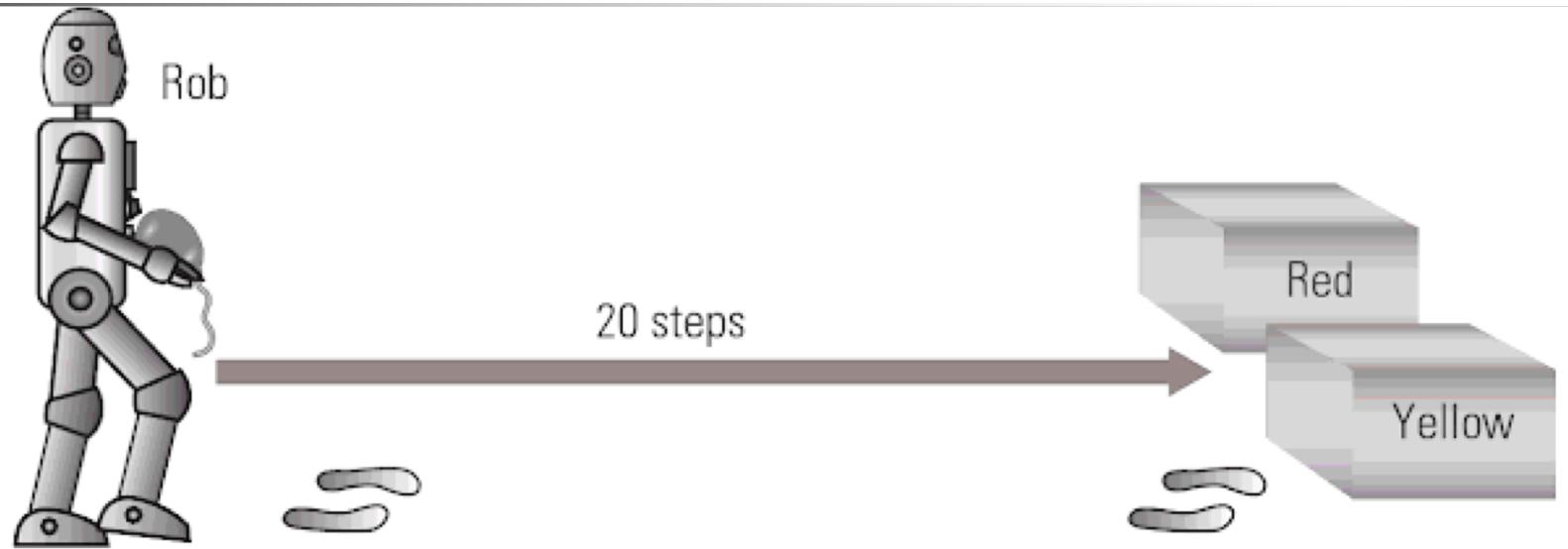
Compiler vs. Interpreter

- Trình dịch (Compiler): là chương trình thực hiện **biên dịch toàn bộ** chương trình nguồn thành mã máy **trước khi thực hiện**
- Trình thông dịch (Interpreter):
 - là chương trình **dịch và thực hiện từng dòng lệnh** của chương trình cùng lúc
 - Dịch từ ngôn ngữ này sang ngôn ngữ khác, không tạo ra chương trình dạng mã máy hay assembly

Ví dụ lập trình



Ví dụ lập trình



algorithm

indent the
instructions
within the
if and
otherwise
sections of a
selection
structure

1. repeat 20 times:
 walk
2. if the balloon is red, do this:
 drop the balloon in the red box
- otherwise, do this:
 drop the balloon in the yellow box
3. turn
4. repeat 20 times:
 walk
5. turn