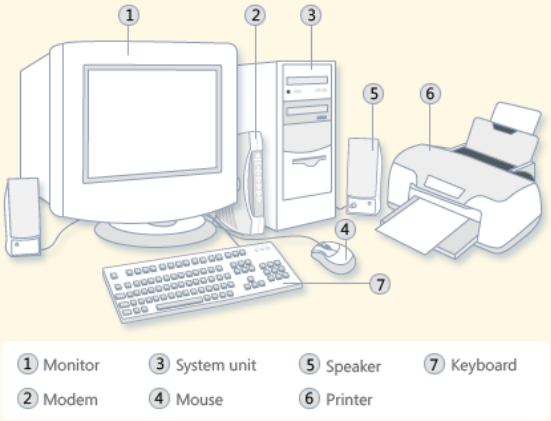
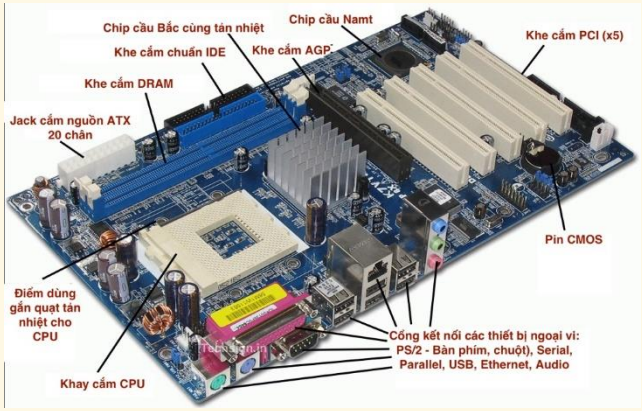
Sơ lược về Lập trình

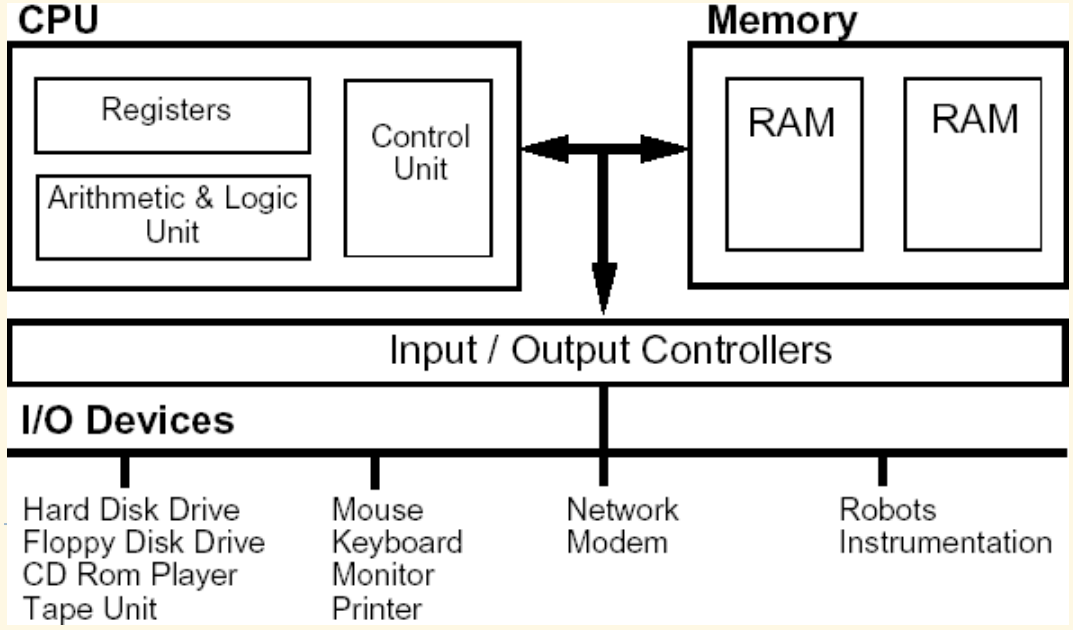
Kiến trúc máy tính

- Phần cứng





https://www.techsignin.com



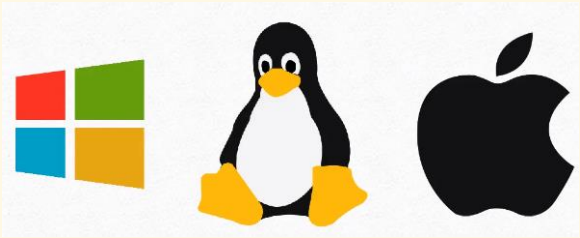
Kiến trúc máy tính

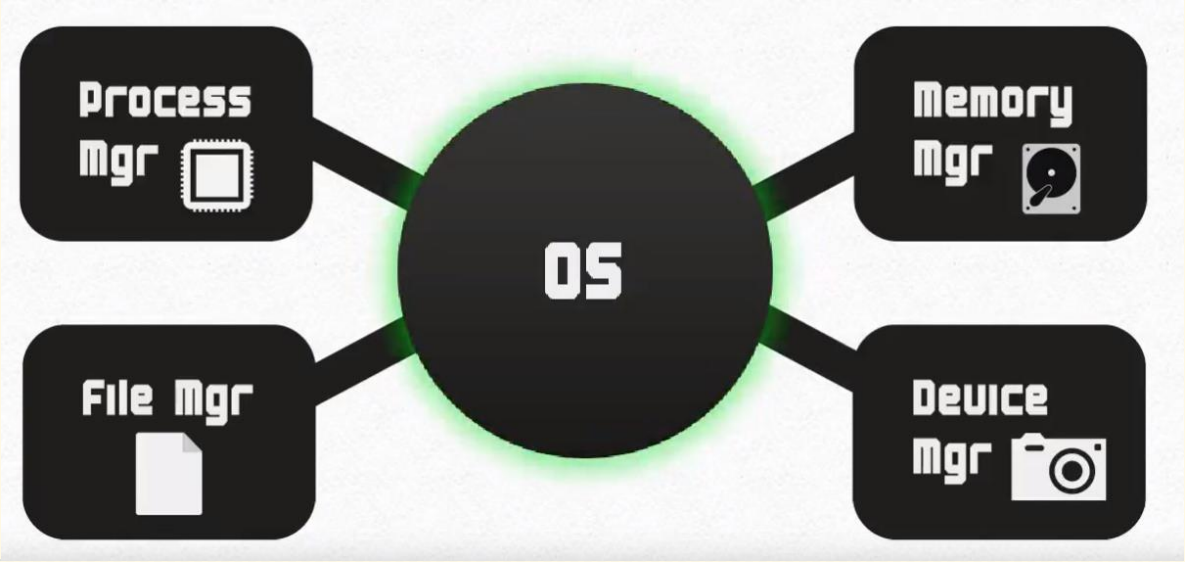
- Phần mềm, Hệ điều hành

https://www.youtube.com/watch?v=5AjReRMoG3Y

Kiến trúc máy tính

- Hệ điều hành:
  - là một phần mềm hệ thống
  - cho phép người dùng thao tác với:
    - phần cứng máy tính
    - thiết bị ngoài
  - chạy phần mềm ứng dụng





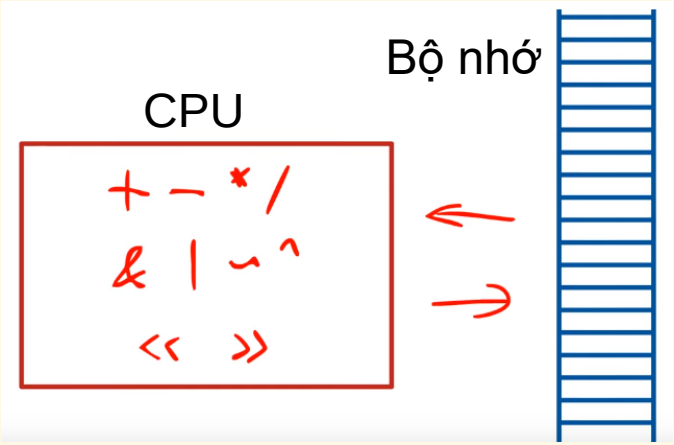
https://www.youtube.com/watch?v=5AjReRMoG3Y
https://sensorstechforum.com

Chương trình

- Phần mềm: tập hợp các chương trình.
- Chương trình: chuỗi lệnh có thứ tự để giải quyết một vấn đề nhất định, được viết bởi 1 *ngôn ngữ lập trình* nhất định.
- Thuật toán/Giải thuật: hệ thống các thao tác, phép toán cần thực hiện chính xác để sau một số hữu hạn các bước ta đạt được kết quả mong muốn.
- Cấu trúc dữ liệu:
  - Biểu diễn các thông tin/ dữ liệu cần thiết cho bài toán dưới cấu trúc phù hợp.

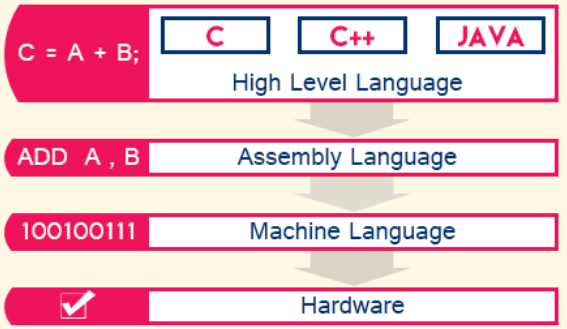
Chương trình chạy – Mã máy

- CPU:
  - Đọc/Ghi vào bộ nhớ
  - Thực hiện các phép toán, v.v.
- Chương trình chạy (exe):
  - Liệt kê các chuỗi lệnh cho máy tính xử lý
  - Máy tính chỉ hiểu mã nhị phân 0 / 1
  - Chương trình chạy viết bằng mã nhị phân -> mã máy
- Bộ nhớ trong của máy tính chứa cả dữ liệu và mã lệnh của chương trình khi chạy.



Ngôn ngữ lập trình

- Quá trình phát triển:
  - Mã máy: dùng trực tiếp mã nhị phân, không cần biên dịch, phụ thuộc vào bộ vi xử lý.
  - Thế hệ thứ 2 (hợp ngữ): cần biên dịch, có thể đọc hiểu được, phụ thuộc và bộ vi xử lý.
  - Thế hệ thứ 3 (cấu trúc): cấu trúc điều khiển, kiểu dữ liệu, đóng gói. VD: Fortran, C, C++, Basic, Pascal, COBOL,…
  - Thế hệ thứ 4: nâng cao hiệu quả như giảm các yếu tố dễ gây lỗi, cú pháp gần gũi hơn với ngôn ngữ nói. VD: SQL, LabVIEW, ColdFusion,…



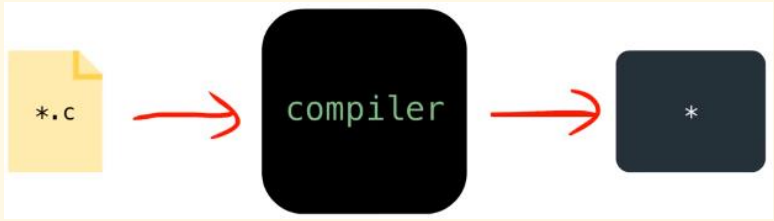
http://btechsmartclass.com

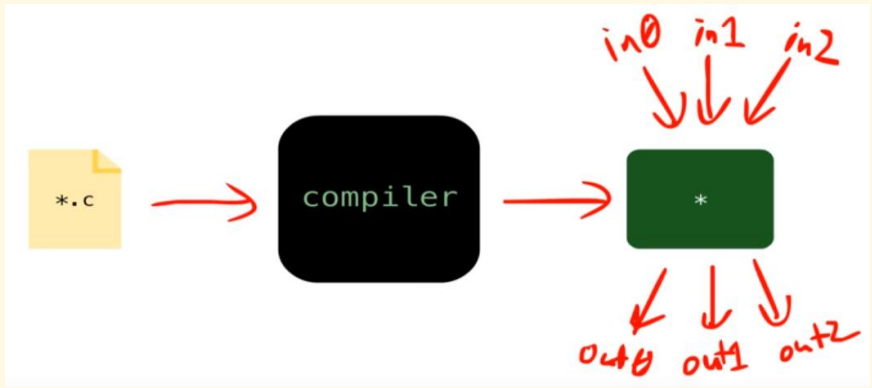
Trình biên dịch

- Là chương trình cho phép chuyển đổi mã nguồn thành mã máy

```
int exp(int a, int b) {
    int result = 1, i;

    for (i = 0; i < b; i++) {
        result *= a;
    }
    return result;
}

int main() {
    int x = 2, y = 5, z;

    if (x < y) {
        x = x + y;
        y = x - y;
        x = x - y;
    }

    z = exp(x, y);
}
```

Source code

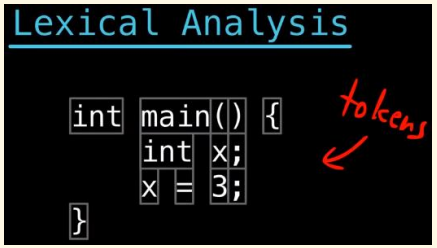


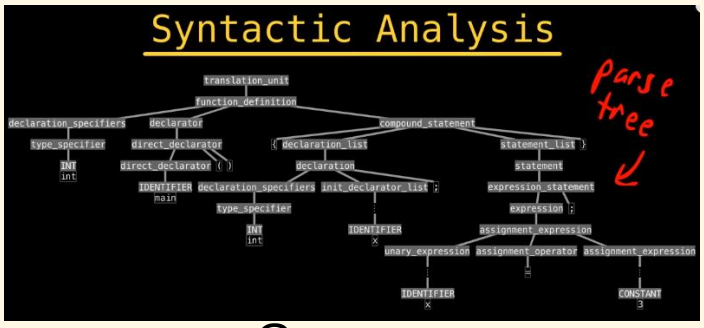


How do computers read code?
Frame of Essence
https://www.youtube.com/watch?v=QXjU9qTsYCc

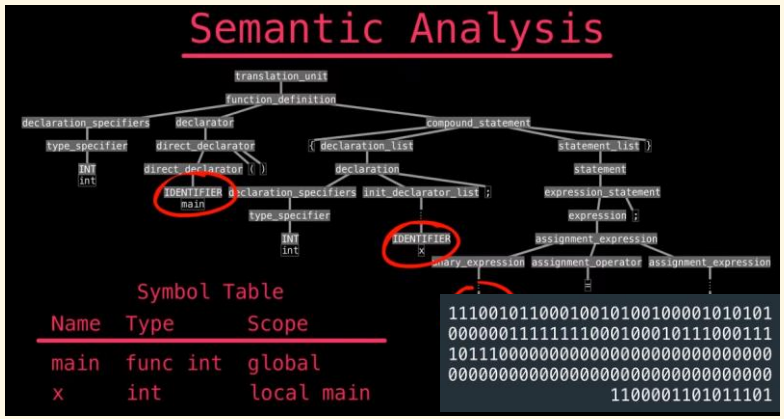
Trình biên dịch

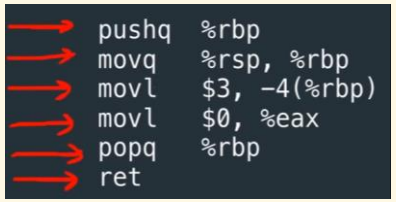
```
int main() {
    int x;
    x = 3;
}
```

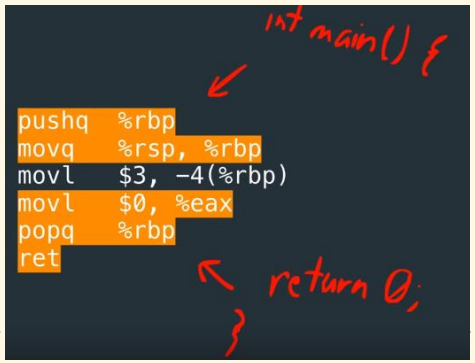


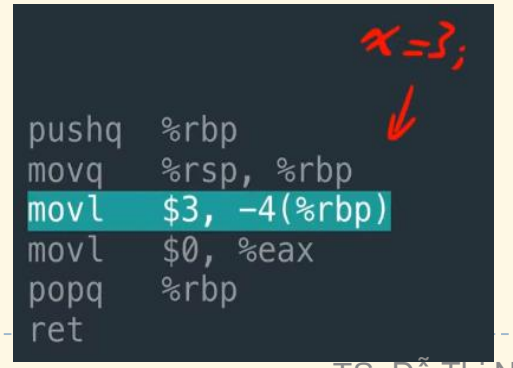


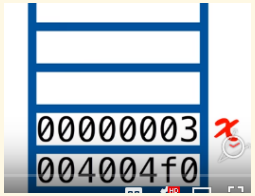
Grammar











Memory location

```
int main() {
    int x;
    x = 3;
    x = x + 1;
}
```

```
x = …   ⟶  mov …
… + …   ⟶  add …
if (…)  ⟶  ?
while (…) ⟶  ?
func(…) ⟶  ?
```

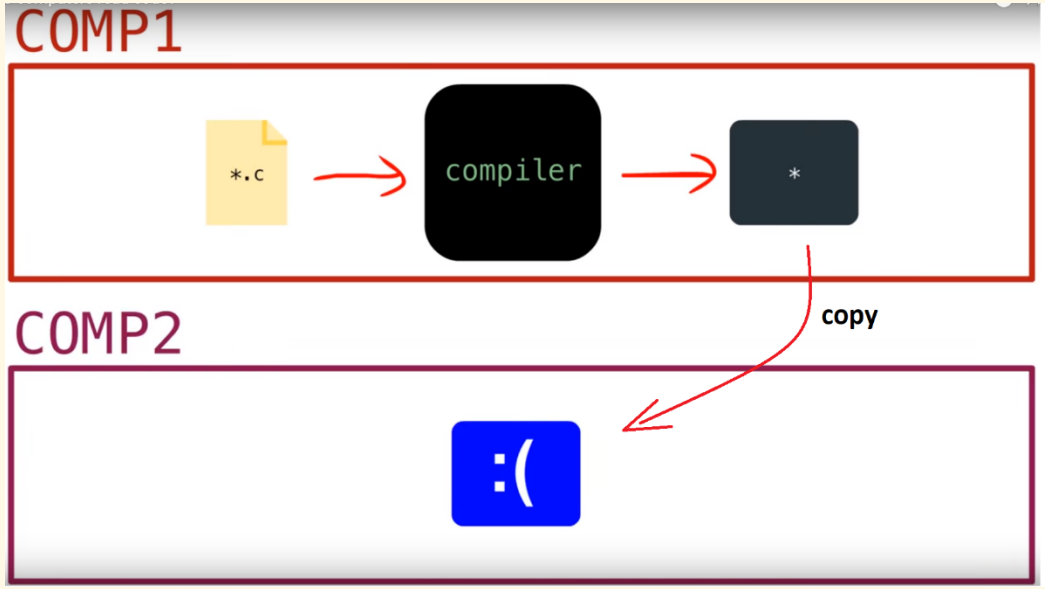
```
int main() {
    int x;
    x = 3;
    if (x < 10) {
        x = x + 1;
    }
}
```

```
        pushq   %rbp
        movq    %rsp, %rbp
        movl    $3, -4(%rbp)
        cmpl    $9, -4(%rbp)
        jg      ENDIF
        addl    $1, -4(%rbp)
ENDIF:  movl    $0, %eax
        popq    %rbp
        ret
```

```
int main() {
    int x;
    x = 3;
    while (x < 10) {
        x = x + 1;
    }
}
```

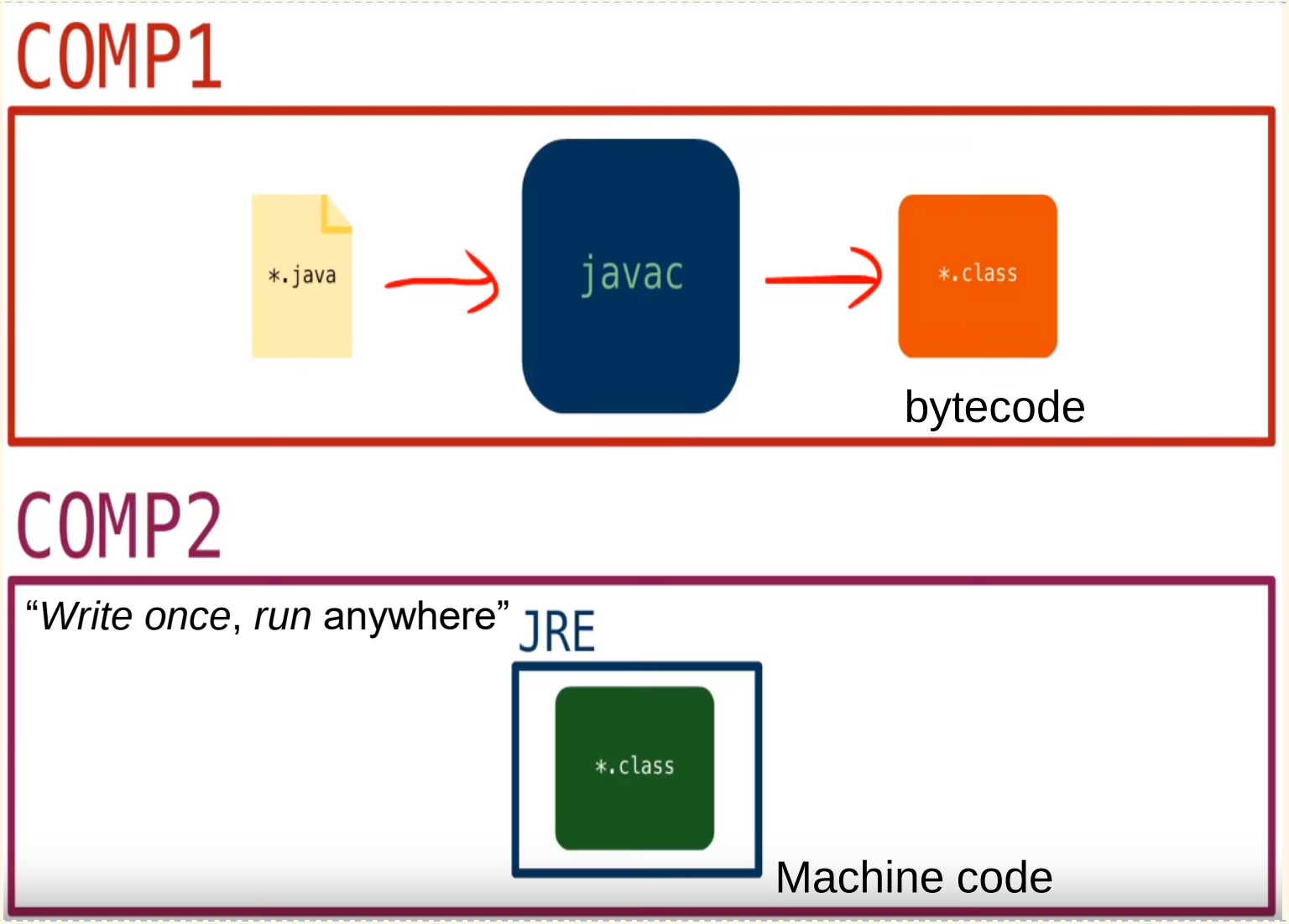
```
        pushq   %rbp
        movq    %rsp, %rbp
        movl    $3, -4(%rbp)
        jmp     WHILE
DO:     addl    $1, -4(%rbp)
WHILE:  cmpl    $9, -4(%rbp)
        jle     DO
        movl    $0, %eax
        popq    %rbp
        ret
```

Trình biên dịch



http://eitguide.net

Interpreter



Ngôn ngữ lập trình

List of programming languages

From Wikipedia, the free encyclopedia

The aim of this list of programming lan

A [edit]
- A# .NET
- A# (Axiom)
- A-0 System
- A+
- A++
- ABAP
- ABC
- ABC ALGOL
- ABSET
- ABSYS

B [edit]
- B
- Babbage
- Bash

C [edit]
- C
- C--
- C++ – ISO/IEC 14882
- C# – ISO/IEC 23270
- C/AL
- Caché ObjectScript
- C Shell (csh)
- Caml
- Calcpad
- Cayenne

D [edit]
- D
- DASL (Datapoint's Advanced System Language)

E [edit]
- E
- Ease

General comparison [edit]

The following table compares general and technical information for a selection of commonly used programming languages. See the individual languages' articles for further information

Language	Intended use	Imperative	Object-oriented	Functional	Procedural	Generic	Reflective	Event-driven
ActionScript 3.0	Application, client-side, web	Yes	Yes	Yes				Yes
Ada	Application, embedded, realtime, system	Yes	Yes[2]		Yes[3]	Yes[4]		
Aldor	Highly domain-specific, symbolic computing	Yes	Yes	Yes				
ALGOL 58	Applic							
ALGOL 60	Applic							
ALGOL 68	Applic							
Ateji PX	Parallel ap							
APL	Application, dat							
Assembly language	Gene							
AutoHotkey	GUI automation (macros),							
AutoIt	GUI automation (macros),							
Bash	Shell, sc							
Ballerina	Integration, agile, se							
BASIC	Application,							
BBj	Application, bu							
BeanShell	Application,							
BitC	Syste							
BLISS	Syste							
BlitzMax	Applicatio							
Blue	Educa							
Boo	Applic							
Bro	domain-specifi							
C	Application, system,[13] ge operat							
C++	Application							
	Application, RAD, business,							

The Top Programming Languages?

According to IEEE Spectrum's interactive ranking, Python is the top programming language of 2017, followed by C, Java and C++. Of course, the choice of which language to use depends on the type of computer the program is to run on, what sort of program it is, and the expertise of the programmer.

Language Rank	Types	Spectrum Ranking
1. Python		100.0
2. C		100.0
3. Java		99.4
4. C++		96.9
5. C#		88.6
6. R		88.1
7. JavaScript		85.3
8. Go		75.7
9. Swift		74.3
10. Ruby		72.0

Image: *IEEE Spectrum Interactive Ranking (2017)*

Quá trình phát triển phần mềm

- Vòng đời của phần mềm:
  - Phân tích yêu cầu của bài toán (analysis)
  - Thiết kế phần mềm (design)
  - Cài đặt thuật toán (development)
  - Kiểm thử (testing)
  - Bảo trì, cập nhật và phát triển tiếp
  - Lỗi thời
- Thử nghiệm: là quá trình kiểm tra sự hoạt động các tính năng của phần mềm
- Gỡ lỗi: là quá trình tìm ra nguyên nhân của lỗi, và sửa nó

Lỗi chương trình

- Lỗi cú pháp (lỗi biên dịch):
  - Do viết chương trình không tuân theo cú pháp quy định
  - Được phát hiện bởi trình biên dịch
  - Chú ý: đôi khi lỗi không được phát hiện vì bị hiểu sai sang cú pháp khác
- Lỗi khi chạy:
  - Khi chương trình chạy vi phạm những điều kiện cho phép
  - Được phát hiện khi chạy
- Lỗi logic:
  - Do thuật toán sai
  - Máy tính không phát hiện

```
>>> print "Hello, world!"
  File "<stdin>", line 1
    print "Hello, world!"
                        ^
SyntaxError: invalid syntax
>>> print("Hello, world!")
Hello, world!
>>>
```