
Khoa
CÔNG NGHỆ THÔNG TIN
ĐH Khoa học Tự nhiên TPHCM

# Giới thiệu hợp ngữ

**Phạm Tuấn Sơn**

ptson@fit.hcmus.edu.vn

# Hợp ngữ (Assembly Language)

- Lệnh máy là dãy bit mà bộ xử lý hiểu để thực thi một công việc nào đó. Ví dụ
  - Lệnh máy MIPS-32bit gán $8 bằng giá trị trong thanh ghi $9 cộng $10
    000000 01001 01010 01000 00000 100000
  - Lệnh máy x86-32bit cộng giá trị trong thanh ghi EAX vào thanh ghi ECX
    000000 0 1 11 000 001

  Như vậy, có nghĩa là muốn yêu cầu bộ xử lý phục vụ thì cần phải cung cấp dãy bit có ý nghĩa tương ứng (nói cách khác là giao tiếp bằng ngôn ngữ máy (machine language))
- Hợp ngữ (assembly language) là ngôn ngữ cấp thấp, cung cấp một cách thể hiện gợi nhớ cho các lệnh máy
  - Để dễ dàng ghi nhớ các mã lệnh, các địa chỉ nơi lưu trữ dữ liệu hoặc lưu trữ các lệnh, người ta đặt tên cho chúng. Đó là mã giả (mnemonic), là tên gọi (label, tên biến, tên chương trình con),…
- Hợp ngữ là cho một bộ xử lý hoặc một dòng bộ xử lý (cùng kiến trúc) nào đó
  - Ví dụ lệnh máy MIPS trên viết lại bằng hợp ngữ cho MIPS-32bit:
    ```
    add $8, $9, $10
    ```
  - Ví dụ lệnh máy x86 trên viết lại bằng hợp ngữ cho x86-32bit:
    ```
    add ECX, EAX
    ```

# Ví dụ chương trình hợp ngữ MIPS-32bit

```
.data      # data segment
  str:
     .asciiz ”hello asm”
.text      # text segment
  .globl main
  main:
  addi $v0, $0, 4    # 4 = print str syscall
  la $a0, str        # load address of string
  syscall            # execute the system call
```

# Ví dụ chương trình hợp ngữ x86-32bit

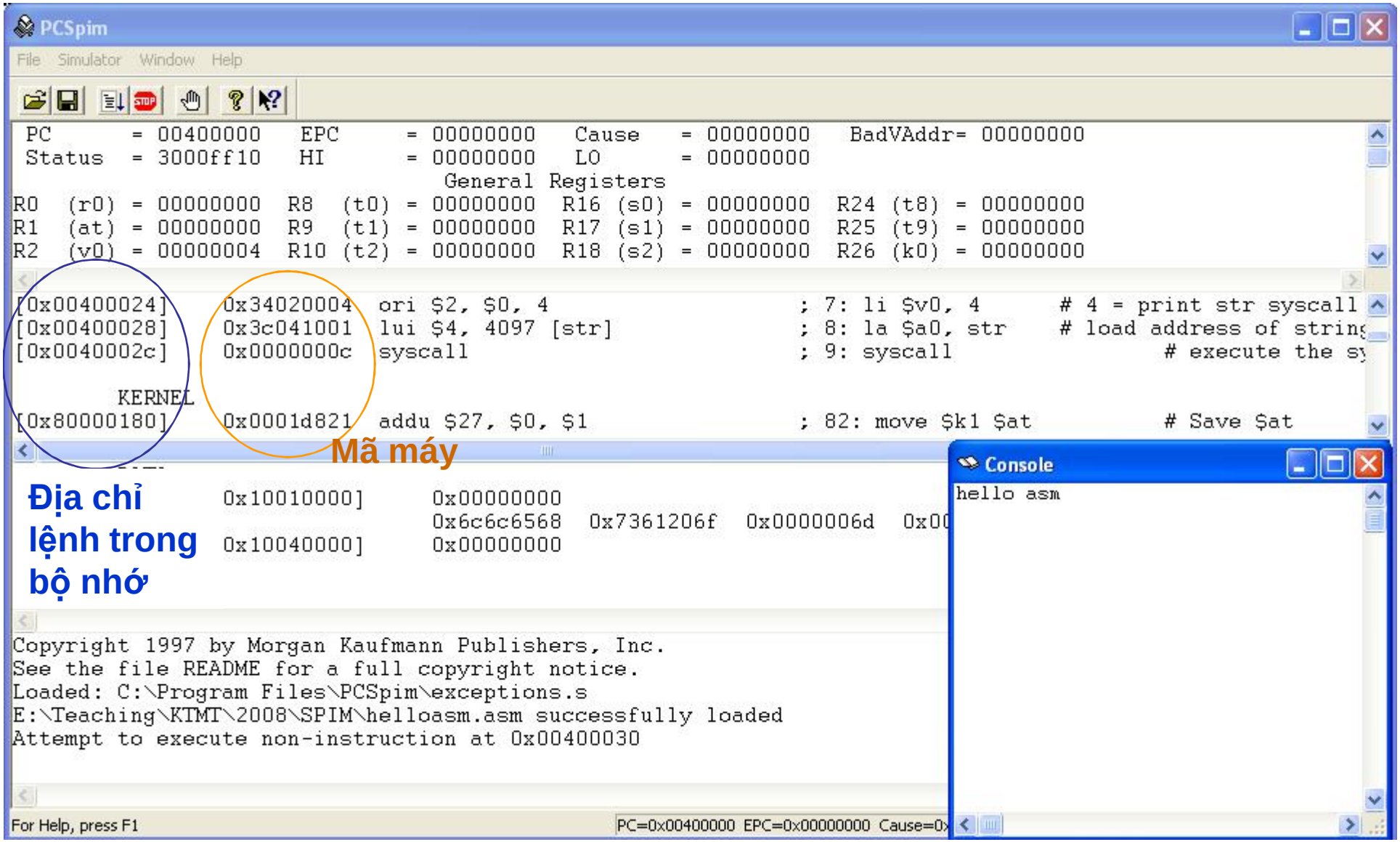
```
global _WinMain@16
extern _MessageBoxA@16

[section .data]
   title db "Message",0
   message db "Hellow World!",0

[section .code]
_WinMain@16:
      push 0
      push title
      push message
      push 0
      call _MessageBoxA@16
      ret 16
```

# Trình biên dịch hợp ngữ

- Chương trình viết bằng hợp ngữ phải được dịch bởi trình biên dịch hợp ngữ (assembler) trước khi máy tính có thể hiểu được nó
- Với một dòng bộ vi xử lí (cùng kiến trúc) – cũng có nghĩa là với một ngôn ngữ máy xác định (tập lệnh máy gần giống nhau) – có thể tồn tại nhiều trình biên dịch hợp ngữ của nhiều nhà cung cấp khác nhau, chạy trên các hệ điều hành khác nhau.
- Ví dụ: cùng là kiến trúc x86, nhưng có thể dùng A86, GAS, TASM, MASM, NASM,…
- Mỗi assembler có thể đưa vào các mở rộng của riêng mình. Vì vậy, một chương trình viết bằng hợp ngữ sẽ mang những đặc trưng riêng phụ thuộc vào trình biên dịch mà tác giả của nó sử dụng

Khoa
CÔNG NGHỆ THÔNG TIN
ĐH Khoa học Tự nhiên TPHCM

# Biên dịch và thực thi chương trình hợp ngữ MIPS bằng PCSpim



# Một số loại assembler

- FASM (Flat Assembler) - for IA32, IA64, open source
- GAS (GNU Assembler) - for many processors, open source
- HLA (High Level Assembler) - for x86, public domain
- HLASM (High Level Assembler) - for mainframes
- Linoleum - for cross platform use
- Lisa - for Apple II 6502
- MAC/65 - for Atari 800 6502
- MACRO-11 - for DEC PDP-11
- MACRO-32 - for DEC VAX
- MASM (Microsoft Macro Assembler) - for x86, from Microsoft
- MI (Machine Interface) - for AS/400, compile-time intermediate language has many fe
- NASM (Netwide Assembler) - for x86, open source
- PAL-III - for DEC PDP-8
- Pass32 powerful DOS/Win16/Win32 assembler
- RosASM - 32 bit Assembler; The Bottom Up Assembler, open source GPL
- Sphinx C-- - mix of Assembly and C, allows combining Assembly commands with C-
- SSK (*Sistema Simvolicheskogo Kodirovaniya*, or "System of symbolic coding") - for
- SynAssembler - for Atari 800 6502 from Synapse Software
- TASM (Turbo Assembler) - for x86 from Borland
- Terse (Algebraic Assembly Language) - for x86 from Jim Neil
- VASM - a NASM compatible assembler with AMD64 support; open source

**http://en.wikipedia.org/wiki/List_of_assemblers**

# Khái niệm Thứ bậc của các ngôn ngữ

- Một máy tính chỉ có thể hiểu một số ít mệnh lệnh, một vài kiểu dữ liệu. Nghĩa là, máy tính chỉ hiểu được một loại ngôn ngữ rất hạn chế, đó là ngôn ngữ máy.
- Để gợi nhớ ngôn ngữ máy, ta có hợp ngữ **à** ngôn ngữ cấp thấp
- Các bài toán thực tế rất phức tạp, sử dụng hợp ngữ để biểu diễn rất khó. Do đó, cần xây dựng các ngôn ngữ dễ sử dụng hơn **à** ngôn ngữ cấp cao, như C, java, …
- Dĩ nhiên, bộ xử lý không thể hiểu được các lệnh và kiểu dữ liệu được xây dựng trên một ngôn ngữ cấp cao. Do đó, đi kèm với một ngôn ngữ cấp cao, luôn có một thành phần làm nhiệm vụ diễn giải thành ngôn ngữ máy tương ứng (trên một bộ xử lý), đó là trình biên dịch

# Mô hình phân tầng các ngôn ngữ trên máy tính

**High Level Language Program (e.g., C)**

*Compiler*

**Assembly Language Program (e.g.,MIPS)**

*Assembler*

**Machine Language Program (MIPS)**

*Machine Interpretation*

**Hardware Architecture Description (e.g. block diagrams)**

*Architecture Implementation*

**Logic Circuit Description (Circuit Schematic Diagram)**

```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw   $t0, 0($2)
lw   $t1, 4($2)
sw   $t1, 0($2)
sw   $t0, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```