# Bài thực hành số 1

# Nhập môn

### Mục đích

* Làm quen với ngôn ngữ lập trình Assembly
* Biết cách viết, dịch, chạy và chẩn lỗi (debug) một vài chương trình đơn giản

### Tóm tắt lý thuyết

Hợp ngữ (assembler) là ngôn ngữ bậc thấp, giúp cho người lập trình không phải ghi nhớ mã máy (opcode) mà sử dụng các từ ngữ gợi nhớ (pseudo-code) gần với ngôn ngữ tự nhiên để miêu tả công việc cần thực hiện. Tuy vậy, assembler rất gần với ngôn ngữ máy, đòi hỏi người lập trình phải hiểu biết tương đối đầy đủ về cấu trúc phần cứng máy tính.

Với mỗi kiểu kiến trúc của bộ vi xử lý, có một bộ lệnh riêng, do đó, có một ngôn ngữ assembler riêng cho nó. Ở đây, chúng ta nghiên cứu assembler cho các bộ vi xử lý Intel thuộc họ x86. Các chương trình sẽ được viết cho chế độ thực (real mode) trong DOS và được biên dịch bằng Turbo Assembler.

**Cấu trúc thông thường của một chương trình hợp ngũ**

**.model** <Khai báo kiểu chương trình>

**.stack** <Khai báo kích thước ngăn xếp>

**.data**

<Khai báo dữ liệu>

**.code**

<Các lệnh>

**end**

*Ví dụ:* Chương trình sau in ra màn hình dòng chữ “Hello !”

**.model** small

**.stack** 100h

**.data**

s DB “Hello !$” ; khai báo xâu kí tự cần in

**.code**

mov AX,@data ; lấy địa chỉ data segment ghi vào DS

mov DS,AX ; Vì model small, đây cũng là địa chỉ

; segment của xâu s

; xuất chuỗi

mov DX, OFFSET s ; lấy địa chỉ offset ghi vào DX

mov AH , 9

int 21h ; gọi hàm 9, ngắt 21h để in

mov AH, 4Ch ; Thoát khỏi chương trình

int 21h

**end**

Lưu ý:

* Mọi chương trình đều phải có đoạn code thoát khỏi chương trình, nếu không chương trình sẽ không dừng khi hết chương trình của mình.

**Khai báo biến trong hợp ngữ**

Cú pháp:

**<tên biến> D<Kiểu DL> <giá trị khởi tạo>**

hoặc

**<tên biến> D<Kiểu DL> <số phần tử> dup(<giá trị khởi tạo>)**

Các kiểu dữ liệu: B (1 byte), W (2 bytes), D (4 bytes)

Nếu không khởi tạo, dùng dấu hỏi “**?**”

*Ví dụ:*

|  |  |
| --- | --- |
| **Khai báo trong C** | **Khai báo trong hợp ngữ** |
| char ch; | ch DB ? |
| char ch = ‘a’; | ch DB ‘a’ |
| char ch = 5; | ch DB 5 |
| char s[]=”\nhello world!” | s DB 10,13,”hello world!$” |
| int i=100; | i DW 100 |
| long l; | l DD ? |
| char a[] = {1,2,3}; | a DB 1,2,3 |
| char a[100]; | a DB 100 dup(?) |
| char a[100][50]; | a DB 100 dup(50 dup(?)) |

**Dịch, liên kết, chạy và chẩn lỗi chương trình từ dấu nhắc DOS**

Cần có các file: tasm.exe (dịch), tlink.exe (liên kết), td.exe (chẩn lỗi). Các bước như sau:

* + B1. Thiết lập đường dẫn

**path = %path%;**<đường dẫn đến thư mục chứa các file kể trên>

* + B2. Biên dịch từ file .ASM sang file .OBJ

**Tasm** <tên file chương trình>.ASM

* + B3. Biên dịch từ file .OBJ sang file .EXE

**Tlink** <tên file>.OBJ

* + B4: chạy chương trình:

<tên file>.EXE

* + B5: chẩn lỗi (nếu cần thiết)

**Td** <tên file>.EXE

Để tự động hóa, ta có thể tạo file .BAT chứa các lệnh trên.

*Ví dụ:*

Tạo file RunASM.bat trong cùng thư mục với tập tin .ASM với nội dung như sau :

**tasm %1**

**tlink %1**

**%1**

(%1 là lấy tham số thứ nhất trong command line)

Sau đó để biên dịch, liên kết và thực thi chương trình hello.ASM ta chỉ cần gõ :

**RunASM hello**

**Công cụ EditPlus**

Đây là công cụ soạn thảo văn bản tiện dụng, cho phép tự động đổi màu chữ theo cú pháp. Ngoài ra còn có thể thiết đặt phím tắt để gọi các tiện ích khác. Để dùng cho soạn thảo chương trình assembler, cần copy file định nghĩa cú pháp vào thư mục cài đặt và đăng kí sử dụng nó cho những file có tên mở rộng “.asm”.

* B0. Cấu hình Edit Plus: xem trong [file hướng dẫn](file:///E:\OneDrive\Thang\HOC%20TAP\GIAO%20TRINH\Kien%20Truc%20May%20Tinh\Tai%20lieu%20Hop%20Ngu\PhuLuc1_CauHinhEditPlus.doc)
* B1. Biên dịch file .ASM : nhấn Ctrl + 1 sẽ biên dịch file đang soạn thảo thành .OBJ
* B2. Liên kết : nhấn Ctrl + 2 sẽ biên dịch file .OBJ thành .EXE
* B3. Chạy chương trình : nhấn Ctrl + 3 sẽ chạy chương trình .EXE
* B4. Chẩn lỗi chương trình : nhấn Ctrl + 4 sẽ debug chương trình .EXE

Lưu ý:

* Để tránh phiền phức khi làm việc với Turbo Assembler, tránh đặt tên thư mục có chứa khoảng trắng.
* Trước khi nhấn Ctrl + 1 lần đầu tiên, nhớ lưu chương trình với tên cần thiết.

**Một số lệnh cơ bản**

MOV des,src : chép dữ liệu từ src sang des

INC des : tăng des một đơn vị

DEC des : giảm des một đơn vị

ADD des,src : des = des + src

SUB des,src : des = des – src

INT num : gọi ngắt

### Tài liệu tham khảo

1. Nguyễn Minh Tuấn, Giáo trình hợp ngữ - Chương 1, ĐHKHTN, 2002
2. Randal Hyde, The art of assembly language programming – Chapter 1.
3. Norton Guide
4. Dan Rollins, TechHelp v.6.0

### Bài tập

Bài 1. Viết CT nhập vào 1 ký tự, xuất ra ký tự đó

*Ví dụ:*

Moi ban nhap 1 ky tu: b

Ky tu vừa nhập: b

Bài 2. Viết chương trình xuất ra màn hình một số dòng.

*Ví dụ:*

De chay duoc 1 CT hop ngu ban can thuc hien cac buoc sau:

Dich file ASM thanh file OBJ

Lien ket file OBJ thanh file EXE

Chay file EXE

Bài 3. Viết CT nhập vào 1 ký tự, xuất ra ký tự liền trước và liền sau.

*Ví dụ:*

Moi ban nhap 1 ky tu: b

Ky tu lien truoc: a

Ky tu lien sau: c

Bài 4. Viết CT nhập vào 1 ký tự thường. In ra ký tự Hoa

*Ví dụ:*

Moi ban nhap 1 ky tu: b

Ky tu Hoa: B

Bài 5. Viết CT nhập vào 1 ký tự hoa. In ra ký tự thường

*Ví dụ:*

Moi ban nhap 1 ky tu: B

Ky tu thường: b

Bài 6. Viết chương trình nhập vào 2 số nguyên dương x1, x2 (1 ≤ x2 < x1 < 9). Xuất ra kết quả các phép tính: x1-1, x1 +2, x1+x2, x1-x2

*Ví dụ:*

x1 = 5

x2 = 3

x1 – 1 = 4

x1 + 1 = 6

x1 + x2 = 8

x1 – x2 = 7

### Mở rộng

1. Tự tìm hiểu xem hàm nào trong ngắt 21h dùng để nhập một xâu kí tự ? Ngoài ngắt 21h, còn ngắt nào có thể dùng để nhập xuất từ bàn phím ? (dùng NortonGuide hoặc TechHelp).
2. Viết chương trình nhập tên và in ra màn hình câu “Hello ” + tên đã nhập.
3. Tìm hiểu xem tại sao không có lệnh MOV x1, x2 (x1,x2 là hai biến trong bộ nhớ)
4. Hai lệnh “INC AX” và “ADD AX, 1” khác nhau chỗ nào ?

### Hướng dẫn

**Bài 1.** Để nhập 1 một ký tự sử dụng hàm 1 của ngắt 21h, để xuất, sử dụng hàm 2.

*Ví dụ:*

mov AH,1

int 21h ; kết quả trong AL

mov DL,AL ; kí tự cần xuất trong DL

mov AH,2

int 21h

**Bài 2.**Cặp kí tự xuống dòng là 10,13. Có thể khai báo nhiều xâu kí tự hoặc chung một xâu.

*Ví dụ:*

Msg3 DB 10,13,9,“1. Dich file ASM thanh file OBJ.$”

Msg4 DB 10,13,9,“2. Lien ket file OBJ thanh file EXE.$”

Hoặc

Msg34 DB 10,13,9,“1. Dich file ASM thanh file OBJ.”

DB 10,13,9,“2. Lien ket file OBJ thanh file EXE.$”

**Bài 3, 4.** Kí tự hoa và kí tự thường của cùng một chữ cái tiếng Anh cách nhau 20h. Do đó, để chuyển đổi chữ hoa thành chữ thường và ngược lại, chỉ cần dùng lệnh ADD, SUB.

**Bài 5.** Để chuyển đổi các kí tự ‘0’ – ‘9’ thành số 0 – 9 chỉ cần thực hiện phép trừ đi 48 (mã của ‘0’). Sau khi thực hiện phép tính, chuyển đổi thành kí tự và in ra màn hình (có thể dùng biểu diễn Hex).