

Part B 組合語言的基本概念

第四章 組合語言程式的基本架構

第五章 程式的組譯連結和執行

第六章 指令和定地址法介紹

第七章 程式流程控制和邏輯運算

第四章 組合語言程式的基本架構

程式架構

語法

資料定義

高階語言與低階語言的比較

高階語言

- 具生產力: 容易寫(快完成), 容易讀(維護容易)

低階語言

- 硬體控制容易
- 程式碼較精簡
- 執行較快

組合語言的特性

程式說明 (Comments)

保留字 (Reserved words)

名稱 (Identifiers)

敘述 (Statements)

控制命令 (Directives)

Comments

一行中分號(不在引號 [單引號 ‘ ’ 或 雙引號 “ ”]中)和其後的文字

組譯程式不處理

作為程式的說明

`;Calculate productivity ratio`

`ADD AX,BX ;Accumulate total quantity`

Reserved words

指令: ADD, MOV, ...

控制命令: SEGMENT, END, ...

運算: FAR, SIZE, ...

預先定義的符號: @Data, ...

列於附錄D

不能用來作為名稱

Identifiers

變數的名字: 代表一個記憶體位置(地址)或常數

標記(Label): 代表指令的位置

程式中以名稱代表記憶體位置或常數, 寫作者不必算位置, 容易改變常數值

Identifiers

可用的文字:

英文字母 A-Z

阿拉伯數字: 1-9

特殊符號: ? _ \$ @ .

可有31個字(MASM6.0後可有247字)

第一個字不能是數字或句點(.)

Statements

[identifier] operation [operand(s)] [;comment]

指令:

L30: MOV AX,0 ;Label, op, ..

指示命令:

COUNT DB 1 ;name, directive, ...

Directives

PAGE, TITLE

SEGMENT

PROC

END

ASSUME

processor: .286, .386, .486

.EXE 程式的分段定義

```
page 60,80
title a04asm1 segments for an .exe program
;-----
stacksg segment para stack 'Stack'
...
stacksg ends
;-----
datasg segment para 'Data'
...
datasg ends
;-----
codesg segment para 'Code'
main proc far
...
main endp ;end of a procedure
codesg ends ;end of a segment
end main ;end of program
```

堆疊段

資料段

程式碼段

.EXE 程式的骨架

```
page 60,80
title a04asm1 skeleton of an .exe program
;-----
stacksg segment para stack 'Stack'
...
stacksg ends
;-----
datasg segment para 'Data'
...
datasg ends
;-----
codesg segment para 'Code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
mov ax,datasg ;set address of data
mov ds,ax ;segment in DS
...
mov ax,4c00h ;end processing
int 21h
main endp ;end of a procedure
codesg ends ;end of a segment
end main ;end of program
```

程式載入時, DS指到PSP

回到作業系統

.EXE 程式的例子

```
page 60,80
title a04asm1 (exe) move and add operations
;-----
stacksg segment para 'Stack'
dw 32 dup(0)
stacksg ends
;-----
datasg segment para 'Data'
fldd dw 215
flde dw 125
fldf dw ?
datasg ends
;-----
codesg segment para 'Code'
main proc far
assume ss:stacksg,ds:datasg,cs:codesg
mov ax,datasg ;set address of data
mov ds,ax ; segment in DS
mov ax,fldd ;move 215 to ax
add ax,flde ;add 125 to ax
mov fldf,ax ;store sum in fldf
mov ax,4c00h ;end processing
int 21h
main endp ;end of a procedure
codesg ends ;end of a segment
end main ;end of program
```

簡化的分段宣告

```
page 60,80
title a04asm2 (exe) move and add operations
;-----
.model small
.stack 64
.data
fldd dw 215
flde dw 125
fldf dw ?
;-----
main .code
proc far
mov ax,@data ;set address of data
mov ds,ax ; segment in DS
mov ax,fldd ;move 215 to ax
add ax,flde ;add 125 to ax
mov fldf,ax ;store sum in fldf
mov ax,4c00h ;end processing
int 21h
main endp ;end of a procedure
end main ;end of program
```

堆疊段

資料段

程式碼段

定義資料

```
page 60,80
title a04defin (EXE) define data directives
.model small
.data
; db - define bytes:
;
;-----
byte1 db ? ;uninitialized
byte2 db 48 ;decimal constant
byte3 db 30h ;hex constant
byte4 db 01111010b ;binary constant
byte5 db 10 dup (0) ;ten zeros
byte6 db 'PC Emporium' ;character string
byte7 db '12345' ;number as character
byte8 db 01,'Jan',02,'Feb',03,'Mar' ;table of month
; dw - define words:
;
;-----
word1 dw 0fff0h ;hex constant
word2 dw 01111010b ;binary constant
word3 dw byte8 ;address constant
word4 dw 2,4,6,7,9 ;table of 5 constants
word5 dw 8 dup (0) ;eight zeros
```

定義資料

```
; dd - define doublewords:
;
;-----
dword1 dd ? ;uninitialized
dword2 dd 41562 ;decimal value
dword3 dd 24,48 ;two constants
dword4 dd byte3 - byte2 ;difference betw addresses
; dq - define quadwords:
;
;-----
qword1 dq 0 ;zero constant
qword2 dq 05e39h ;hex constant
qword3 dq 41562 ;decimal constant
end
```


EQU Directive

組譯時, 右邊的文字取代左邊的文字

MPY EQU MUL

通常用以定義常數

FACTOR EQU 12

TABLEX DB FACTOR DUP (?)

相當於

TABLEX DB 12 DUP (?)