



SWE4001 – System Programming

Module 3: Assembler

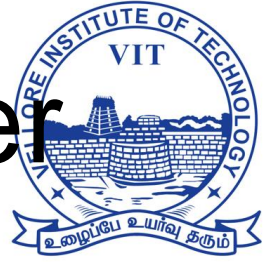
Lesson 9 of 9: Implementation Examples

Implementation Examples



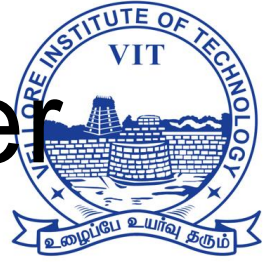
- Microsoft MASM Assembler
- Sun Sparc Assembler
- IBM AIX Assembler

Microsoft MASM Assembler



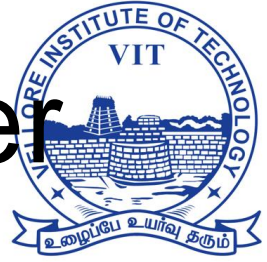
- SEGMENT
 - a collection segments, each segment is defined as belonging to a particular class, CODE, DATA, CONST, STACK
 - registers: CS (code), SS (stack), DS (data), ES, FS, GS
 - similar to program blocks in SIC

Microsoft MASM Assembler



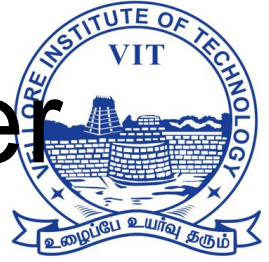
- Default data segment is DS
- ASSUME
 - e.g. ASSUME ES:DATASEG2
 - e.g. MOVE AX, DATASEG2
 - MOVE ES,AX
 - similar to BASE in SIC

Microsoft MASM Assembler



- JUMP with forward reference
 - near jump:
 - Same code segment
 - 2 or 3 bytes
 - far jump:
 - Different code segment
 - 5 bytes
 - e.g. JMP TARGET
 - Default – forward reference is a near jump.

Microsoft MASM Assembler

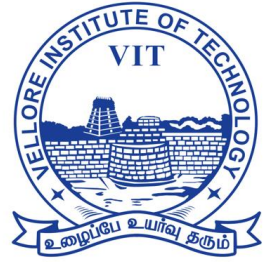


- Warning: JMP FAR PTR TARGET
- Warning: JMP SHORT TARGET
- Pass 1: reserves 3 bytes for jump instruction
- phase error

Microsoft MASM Assembler



- SEGMENT directive specifies the same name as a previously defined segment, it is considered to be a continuation of that segment.
- PUBLIC, EXTRN
 - similar to EXTDEF, EXTREF in SIC



.model small

.stack

.data

Message db "Press Y or N:\$" ;Prompt for user

Uyes db "You pressed Y!\$" ;Pressed y

Uno db "You pressed N!\$" ;Pressed n

.code

_start:

mov ax,03h ;clears screen (function 3h)

int 10h ;interrupt 10h

mov ax, SEG Message ;put segment of message into AX

mov ds, ax ;put this into DS

mov dx, OFFSET Message ;put offset of message into AX

mov ah,09h ;Function 9h of

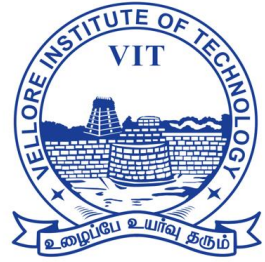
int 21h ;Interrupt 21h

mov ah,01h ;function 01h of int21h,

int 21h ;get char from keyboard

cmp al, "Y" ;if ah Y then

je Yes ;Goto Yes label



```
cmp al, "N" ;if ah N then  
je No ;Goto No Label  
jne _start ;if not Y or N then goto start
```

Yes:

```
mov ax,03h ;clears screen (function 3h)  
int 10h ;interrupt 10h
```

```
mov ax, SEG Uyes ;segment of Uyes  
mov ds, ax ;put segment into DS  
mov dx, OFFSET Uyes ;put offset of Uyes into  
DX
```

```
mov ah,09h ;function 9h print string at DS:DX  
int 21h ;call interrupt 21h  
jmp close
```

No:

```
mov ax,03h ;clears screen (function 3h)  
int 10h ;interrupt 10h
```

```
mov ax, SEG Uno ;segment of Uno  
mov ds, ax ;put segment into DS  
mov dx, OFFSET Uno ;put offset of Uno into DX  
mov ah,09h ;function 9h print string at DS:DX  
int 21h ;call interrupt 21h  
jmp close  
  
close:  
mov ax,4c00h ;put 4c00h into ax, closing back to DOS  
int 21h ;INT 21h, return to DOS  
  
end _start
```