

Idara Injuning - Etale Fi

IT/Eng02/025

COE 306 Assembly language

25/07/2020

1a.) The addresses would have to be continuously updated whenever new variables are inserted before pre-existing addresses.

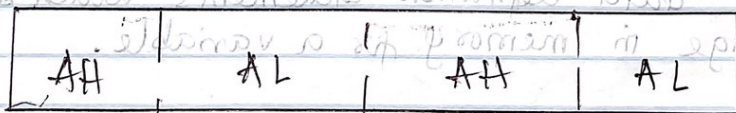
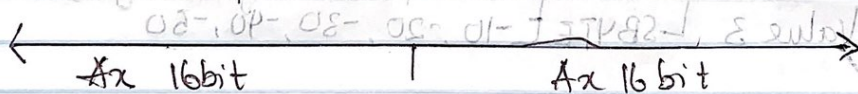
- 1b.) Object file
- ii) Listing file

2a.) A measure of how easily an application can be transferred from one computer environment to another. A computer software application is considered portable to a new environment if the effort required to adapt it to the new environment is within reasonable limits.

2b.) No, they are not because each assembly language is based on either a processor family or a specific computer.

2c.)

32 bit	16 bit	8 bit (High)	8 bit (Lower)
EAX	AX	AH	AL
EBX	BX	BH	BL
ECX	CX	CH	CL
EDX	DX	DH	DL



Higher order of AX | Lower order of AX

Data Initialization

17/Eng02/025

3a) It is achieved by dividing the system memory into groups of independent segments referenced by pointers located in the segment registers. Each segment used to contain specific type of data

```

3b) Main PROC
    MOV AX, 47104 ; move 47104
    ADD EAX, 1270
    MOV DS, AX
main ENDP
    
```

- i) moves 47104 to the ~~AX~~ register
- ii) Add 1270 to the EAX register
- iii) moves DS to AX register

Name	Directive	Initializer
3c) i)	Value 1 - BYTE	6Dh (defined byte)
ii)	Value 2 - DWORD	? (defined signed byte)
iii)	Value 3 - SBYTE	-10, -20, -30, -40, -50

They are data definition statements which reserve storage in memory for a variable.

* The '?' mark in (ii) leaves the variables uninitialized implying it will be signed a value at runtime

~~Answer~~

4.) ~~FILE~~ TITLE Add and subtract, Version 2 (Addsub2.asm)

;
; This program adds and subtracts 16-bit unsigned
; integers and stores the sum in a variable.

INCLUDE Irvine32.inc

.data

val1 DWORD 1000h

val2 DWORD 4000h

val3 DWORD 2000h

finalVal DWORD ?

.code

main PROC

mov ax, val1 ; start with 1000h

add ax, val2 ; add 4000h

sub ax, val3 ; subtract 2000h

mov finalVal, ax ; store the result (3000h)

call DumpRegs ; display the registers

exit

main ENDP

END main