

MID-TERM EXAMINATION

QUESTION 1

1a) It would not be a good idea because the address coded in the instructions would have to be updated anytime a new variable is inserted before existing ones

1b) ~~ELF files produced by the assembler are relocatable files that hold code and/or data.~~

1b) Object .OBJ and listing .LST files

QUESTION 2

a) portability is a characteristic attributed to a computer program if it can be used in an operating system other than the one in which it was created without requiring major rework.

b) No, each assembly language is based on either the processor family or a specific computer

c) ↓ 32-bit registers

	16	15	8	7	0	↓
EAx			AH	AL		Ax Accumulator
EBx			BH	BL		Bx Base
ECx			CH	CL		Cx Counter
EDx			DH	DL		Dx Data

- Ax is the primary accumulator
- Bx is known as the base register
- Cx is known as the count register
- Dx is known as the data register

QUESTIONS

lu

9)

3b) Main proc - This identifies the beginning of the code

MOV AX, 47104 - It tells the program to move "47104" into the register

ADD EAX, 127 - It tells the program ~~to add 127 (in base) to~~ the value ~~already~~ already existing in ~~EAX~~ register

MOV DS, AX - The loader (the part of the OS that reads the program from disk to memory and starts it running) has to set CS, so that it can run the program but may not set DS, so the program copies CS to DS when it starts.

Main ENDP - This is the exit statement that calls a ~~predefined~~ predefined MS-windows function that halts the program

3c) i) value 1 - label

Byte - directive

6Dh - radix, initializer - This tells the system to byte 6Dh under value 1 label (it is an unsigned byte)

ii) value 2 -

DWORD - This is an uninitialized variable and its value will be assigned at runtime.

lp

(iii) Value 3 BYTE -10, -20, -30, -40 - 30 - These are multiple initializers (signed variables). Here, one label is used to declare multiple signed variables

Question 4 ?

TITLE subtraction (sub.as)

; This program subtracts 2 16-bit integers

INCLUDE Irvine32.inc

.data

val1 WORD 5000h

val2 WORD 2000h

val3 WORD 1000h

End val ?

.code

Main proc

mov eax, val1

sub eax, val2

mov End val,

dump regs

exit

Main ENDP

END MAIN