|  |  |  |
| --- | --- | --- |
| 2020/1 Assembly Dersi Ödev 1 | | |
| İlan Tarihi | Son Teslim Tarihi/Saati | Teslim Şekli |
| 28/12/2020 | 31/12/2020 – 23:59 | Cevaplarınızı word dosyası olarak online.yildiz.edu.tr’den yükleyiniz |

(Soru 1-20: 2p, Soru 21: 10p, Soru 22: 50p)

Batuhan LEL 18011024

Soru 1: SP, BX, DI, BP, SI için varsayılan kesim yazmaçları hangileridir?

SI,DI,BX -> DS(Data Segment) SP,BP -> SS(Stack Segment)

Soru 2: 8B07H makine kodu üreten assembly komutunu bulunuz.

MOV AX , [BX]

Soru 3: 8B9E004CH makine kodu üreten assembly komutunu bulunuz.

MOV BX, [BP+4C00]

Soru 4: MOV SI, [BX+2] komutunun makine kodu karşılığını bulunuz.

8B7702H

Soru 5: MOV CS, AL komutundaki hatayı bulunuz.

Code Segment üzerinden doğrudan değişiklik yapamayız. Eğer bir değişiklik yapacaksak JMP veya CALL komutlarını kullanarak istediğimiz yere gitmeliyiz.

Soru 6: MOV DI, NUMBER ve LEA DI, NUMBER komutları arasındaki farkı açıklayınız.

MOV komutunu kullanarak NUMBER değişkeninin tuttuğu değeri DI ya atarız fakat LEA kullanırsak NUMBER değişkenini gösteren adres değerini(efektif adres) DI ya atamış oluruz.

Soru 7: ES’nin gösterdiği kesim alanı içinde BX ile işaret edilen hafıza bölgesinin içeriğini AH yazmacına taşıyan komutu yazınız.

MOV AH, ES:[BX]

Soru 8: BX’i AX’e toplayan ADD komutunu yazınız.

ADD AX, BX

Soru 9: 12H değerini AL’ye toplayan ADD komutunu yazınız.

ADD AL, 12H

Soru 10: DI’yı BP’ye toplayan ADD komutunu yazınız.

ADD BP, DI

Soru 11: 22H değerini CX’e toplayan ADD komutunu yazınız.

ADD CX, 22H

Soru 12: SI ile gösterilen adresin içeriğini AL’ye toplayan ADD komutunu yazınız.

ADD AL, [SI] ;LEA SI, ……

Soru 13: CX’ı DENEME ile gösterilen adres içeriğine toplayan ADD komutunu yazınız.

ADD [DENEME], CX

Soru 14: ADD CX, AH komutundaki hata nedir?

İşlem yapmak istediğimiz registerların boyutları uyuşmamakta.

CX -> WORD

AH -> BYTE

Soru 15: AL, AH, BL ve CL’yi toplayıp sonucu DX’te saklayan komutları yazınız.

XOR DX, DX

MOV BH, AH

CBW

ADD DX, AX

MOV AL, BH

CBW

ADD DX, AX

MOV AL, BL

CBW

ADD DX, AX

MOV AL, CL

CBW

ADD DX, AX

Soru 16: INC [BX] komutundaki hatayı bulunuz.

[BX]’in hangi boyut olarak arttırılacağını belirtmediğimiz için hatalıdır.

Soru 17: SUB ve CMP komutları arasındaki farkı açıklayınız.

İkisi de çıkartma işlemi yapar fakat SUB sonucu bir saklarken CMP komutu sonucu saklamaz.

Soru 18: IMUL ve MUL komutları arasındaki fark nedir?

IMUL çarpma işlemi yaparken işaretli sayılarda işlem yaptığını kabul ederken MUL işaretsiz sayılarda çarpma işlemi yapıldığını varsayarak işlem yapar.

Soru 19: DIV işlemi sonucunda kalan nerede saklanır?

Bölenin boyutu BYTE ise kalan AH üzerinde saklanır.

Bölenin boyutu WORD ise kalan DX üzerinde saklanır.

Soru 20: AX=1001H ve DX=20FFH iken ADD AX, DX komutu çalıştırıldıktan sonra C, A, S, Z ve O bayrakları nasıl değişir?

C : Değişmiyor (0)

Z : DEĞİŞİYOR (1 -> 0 )

S : Değişmiyor (0)

O: Değişmiyor (0)

A : DEĞİŞİYOR (0 -> 1 )

Soru 21: DL’de tutulan sayının küpünü bulan assembly komutları yazınız (10p).

MOV DL,-50 ;DL'YE HERHANGİ BİR SAYİ ATANABİLİR

MOV AL, DL

CBW

MOV BX, AX

IMUL AX

IMUL BX ;SONUC DX:AX İKİLİSİNDE OLUSTU.

Soru 22: Random sayı üretmek için kullanılan yöntemlerden biri de “linear congruential generation” yöntemidir. Bu yöntemde bir sonraki random sayı (xt+1), önceki random sayıya (xt) göre

formülüne göre hesaplanır. x0 değeri ise seed olarak isimlendirilir.

m=8191, a=884 ve c=1 değerleri için verilen yöntemle 0-255 arasında bir random sayı üretmeniz istenmektedir. Kod segmentte 23 ofsetinden itibaren okuyacağınız sayıyı seed olarak kabul ederek bir random sayı üreten kodu tam bir EXE programı olarak assembly dili ile yazın. m, a ve c değerlerini data segmentte uygun şekilde tanımlayın, üretilen random sayıyı yine data segmentte saklayınız (50p).

myds SEGMENT PARA 'veri'

a DW 884

m DW 8191

c DW 1

x DW 0

myds ENDS

myss SEGMENT PARA STACK 'yigin'

DW 20 DUP(?)

myss ENDS

mycs SEGMENT PARA 'kod'

ASSUME CS:mycs, SS:myss, DS:myds

MAIN PROC FAR

PUSH DS

XOR AX, AX

PUSH AX

MOV AX, myds

MOV DS, AX

MOV DX, WORD PTR CS:[23]

don: MOV AX, DX

MUL a

ADD AX, c

ADC DX, 0

DIV M

CMP DX, 256

JA don

MOV x,DX

RETF

MAIN ENDP

mycs ENDS

END MAIN