



MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



2020-2021 Öğretim Yılı Bahar Yarıyılı, Bilgisayar Organizasyonu Dersi

Final Sınav Ödevi

Öğretim Elemanı Unvanı / Adı Soyadı: Prof.Dr. İsmail KADAYIF

Veriliş Tarihi: 08/06/2021

Öğrenci Adı Soyadı: Aras Taşdemir

Öğrenci No: 190401031

Teslim Tarihi: 11/06/2021

Not: Soruların cevapları ilgili boşluklara yazılacaktır (type edilecek, elle yazılmayacak). Sonra dosyanız pdf dosyasına dönüştürerek UBYS sistemine yüklenmelidir.

1) Aşağıdaki soruları cevaplayınız.

a) Tek duyarlı (single precision) kayan noktalı sayı gösterimi kullanılarak ifade edilebilecek en büyük sayıyı hexadecimal

(onaltılık) düzende yazınız? (5 puan)

0x7f7fffff

b) Bir önceki şıkta bulduğunuz bu en büyük sayıyı decimal (onluk) düzende yazınız? (5 puan) $3.402823466 \times 10^{38}$

c) Tek duyarlı (single precision) kayan noktalı sayı gösterimi kullanılarak ifade edilebilecek en küçük sayıyı hexadecimal (onaltılık) düzende yazınız? (5 puan)

0xff7fffff

d) Bir önceki şıkta bulduğunuz bu en küçük sayıyı decimal (onluk) düzende yazınız? (5 puan) $-3.402823466 \times 10^{38}$

2)

dizi: .space 2000

Yukarıdaki tanımlamanın 25x20 boyutlarında iki boyutlu bir tam sayı dizisine ait olduğunu düşünelim. Ayrıca bu dizinin belleğe column-major yöntemine göre yerleştirildiği ve dizi[5][5] elemanının adresinin 0x10010AB0 olduğu bilinmektedir. Buna göre;

a) dizi[12][8] elemanının adresini bulunuz. (5 puan)

Adres: 0x10010BF8

b) Bu dizinin başlangıç adresini (dizi[0][0] ın adresi) bulunuz. **(5 puan)**

Başlangıç adresi: 0x100108A8

c) Bu dizinin en son elemanının adresini bulunuz. **(5 puan)**

Başlangıç adresi + 0x7cc = 0x10011074 **7** en son elemanın adresi

3) Aşağıdaki soruları cevaplayınız?

a) jal (jump and link) ve jr (jump register) komutlarının temel farkını hangi durumlarda kullanıldıklarını dikkate alarak kısaca açıklayınız. **(5 puan)**

Jal komutu kendini izleyen komutun adresini \$31'e (\$ra)yazar. Etiket procedure_label olan komuta dallanmayı sağlar. jr\$31: \$31'in içeriği pc (program counter) a yazılır (\$31'in işaret ettiği komuta jump edilir (sıçranır)) jr dönüş adresini içeren register'ı belirten bir tek işleneni vardır jal ise basit atlama talimatı gibi davranır ancak aynı zamanda kayıt 31(\$ra) de bir dönüş adresi saklar.

b) lw \$5,40(\$6) komutunun makine dilinde kodlanmasını hexadecimal düzende gösteriniz. **(5 puan)**

100011 00110 00101 00000 00000 101000

opcode \$6 \$5 hexadecimala

çevirirsek 0x8cc50028

c) addi \$8,\$8,-60 komutunun makine dilinde kodlanmasını hexadecimal düzende gösteriniz. **(5 puan)**

addi işleminde elde ettiğimiz binary kodu: 001000 01000 01000 1111111111000100 Opcode

r8 r8 -60(two complements)

hexadecimale çevirirsek 0x2108FFC4



MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
2020-2021 Öğretim Yılı Bahar Yarıyılı, Bilgisayar Organizasyonu Dersi
Final Sınav Ödevi



Öğretim Elemanı Unvanı / Adı Soyadı: Prof.Dr. İsmail KADAYIF

Veriliş Tarihi: 08/06/2021

Öğrenci Adı Soyadı: Aras Taşdemir

Öğrenci No: 190401031

Teslim Tarihi: 11/06/2021

- d) Fonksiyon çağrılarında genellikle \$31 yazmacındaki geri dönüş adresi çağrılan prosedür kodunda stacke itilir ve prosedürden dönülmeden önce de stackten çekilerek çağırılan prosedüre geri dönülmesi sağlanır. Hangi tip prosedür çağrılarında \$31 yazmacının stacke itilmesine gerek olmadığını belirtiniz? **(5 puan)**

Tek bir fonksiyon çağrılacağı zaman \$31(return adres) yazılabilir ancak çağırdığımız fonksiyon da bir fonksiyon çağırıyor ise kodun devam edeceği adresi kaybederiz yani çağrılan işlem bir tane ise \$31 tek kullanmak yeterli olacaktır, birden fazla fonksiyon iç içe yada recursive bir fonksiyon varsa stack'e itmemiz gerekecektir.

- 4) Tamsayı kare bir matrisin asal köşegen üzerindeki elemanlarını (satır numarası sütun numarasına eşit olan elemanlar) 1, diğer elemanlarını ise -1 yapan bir programı MAL dilinde yazınız. Bu programın veri tanımlamaları sizin için **soru4.s** dosyasında yapılmıştır. Sizden tanımlı veriler haricinde herhangi başka bir veri tanımlaması yapmamanız istenmektedir. Yazdığınız kodu **soru4.s** dosyasına kaydederek ve QtSpim simülatöründe doğru çalıştığından emin olduktan sonra bu dosya ile birlikte sisteme yükleyiniz. Not: Programdaki **ar** ve **satir_sutun_sayisi** veri tanımlamaları değiştirildiğinde de (örneğin ar: .space 400 satir_sutun_sayisi: 10) programınızın doğru çalışması gerekmektedir. **(25 puan)**

- 5) C dilinde yazılı aşağıdaki kod parçacığı verilmektedir. Amaç, bu kod parçacığına karşı gelen kodu QtSpim simülatöründe çalıştırılmaktır.

```
int x,y;
```

```
int *ptr1,*ptr2;
```

```
x = 88,
```

```
ptr1 = &x;
```

```
ptr2 = &y;
```

```
*ptr2 = *ptr1;
```

- a) Yukarıdaki veri tanımlamalarına karşı gelen veri tanımlamalarını QtSpim için yazınız. **(5 puan)**

```
.data
```

```
x: .word
```

```
y: .word      ptr1:
```

```
.word      ptr2:
```

```
.word
```

- b) Yukarıdaki C koduna karşı gelen kodu QtSpim için yazınız. **(15 puan)** .data x: .word 0 y: .word 0 ptr1: .word 0 ptr2: .word 0

```
.text
main:

    la $s0,x    li
    $t0,88      sw $t0,
    0($s0)

    la    $s1,ptr1
    sw $s0,0($s1)

    la $s2,ptr2
    la $s3,y      sw
    $s3, 0($s2)

    lw $s2, 0($s2)
    lw  $s1, 0($s1)
    lw  $s1, 0($s1)
    sw $s1, 0($s2)
    li $v0, 10

    syscall
```

