



Fenerbahçe Üniversitesi

BLM 101 – Bilgisayar Mühendisliğine Giriş

LAB 4: Von Neumann Modeli

Önemli Not: Aşamaları tamamladıkça, dersin hocası veya asistanı yanınıza çağırarak, tamamladığınız aşamayı gösterdikten sonra diğer aşamaya geçiniz.

LAB Hakkında:

Bilgisayar mimarisinin temel modellerinden olan Von Neumann Modeli üzerinde deneyler yapılacaktır. Von Neumann simülatörü ile basit bir işlemci mekanizmasının nasıl çalıştığı gösterilecektir. İşlemci içi verinin bellekten alımı, hesaplamanın yapılıp, tekrar belleğe yazımı gösterilecektir. Bu işlemcide Accumulator (ACC) denen bir saklayıcı bulunmaktadır. Hesaplama sonuçlarını tutmaktadır. LAB'da yazmanız istenen kodların amacı, bir işlemcinin içerisinde komutlar nasıl çözülüp, işlenip, belleğe yazıldığı öğrenilmesidir. Simülatör aracı bu aşamaları görselleştirmektedir.

Von Neumann Simülatörü adresi: <http://levent.tc/araclar/vonneumann/>

Ekranın sağ tarafında bulunan Komutlar ve Değişkenler belleğin (RAM) içerisinde bulunan sayılardır. Komutları 0-1'lar halinde yazmak yerine, assembly denen bir dil ile ifade ediyoruz.

Komutlar bölümüne, bu işlemci için geçerli komut seti ile komutlar yazabilirsiniz.

Komut Adı	Görevi	Örnek Kullanım
LOD ADDR	Yükleme (Load), Bellekteki verilen adresin içerisinde değeri alıp, ACC saklayıcısına yerleştirir. $ACC = *(ADDR)$	LOD X LOD Y
STO ADDR	Kaydetme (Store), ACC'nin içerisindeki değeri alıp, bellekte verilen adrese yazar. $*(ADDR) = ACC$	STO X STO Y
ADD ADDR	Bellekteki verilen adresteki değeri alır, ACC ile toplayıp, ACC'nin üzerine yazar. $ACC = ACC + *(ADDR)$	ADD X ADD Y
SUB ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çıkartıp, ACC'nin üzerine yazar. $ACC = ACC - *(ADDR)$	SUB X SUB Y
MUL ADDR	Bellekteki verilen adresteki değeri alır, ACC ile çarpıp, ACC'nin üzerine yazar. $ACC = ACC * *(ADDR)$	MUL X MUL Y
DIV ADDR	Bellekteki verilen adresteki değeri alır, ACC ile bölüp, ACC'nin üzerine yazar.	DIV X DIV Y

	ACC = ACC / (*(ADDR))	
JMZ SAYI	ACC'ın değeri 0 ise, verilen sayı değerini PC'e atar, değilse işlem yapmaz.	JMZ 5 // Yani ACC == 0 ise, program 5 nolu satıra atlar
JMP SAYI	PC = Sayı olur.	JMP 5 // Program doğrudan 5. satırdan devam eder
NOP	No Operation, hiçbir işlem yapılmaz.	NOP
HLT	Uygulama durur	HLT

Değişkenler bölümünde ise bellekteki ilk değerlerinin ne olacağını ayarlayabilirsiniz.

Başlat tuşu, programı çalıştırmaktadır. Uygulamayı durdurup başa dönmek için, iki defa durdur tuşuna basınız (Programın başa döndüğünü PC'nin 0 olduğundan anlayabilirsiniz).

Daha önceden yazılmış bir assembly kodunu açmak için sağ üstteki aç tuşuna, mevcut kodu kaydetmek için kaydet tuşuna basabilirsiniz.

LAB'ın aşamaları ve puanlar:

Algoritmanızı kurarken, bu işlemcinin çalıştırabileceği komut setine göre kurmalısınız. Komutları düşünerek, algoritmanızı nasıl ifade edebileceğinizi bulmalısınız.

1- BMI (Body Mass Index) Hesaplayan Uygulama (0 Puan):

BMI (Body Mass Index) hesaplayan uygulama geliştirin. Bellekten iki sayı (X ve Y adreslerini kullanın) alın, ilk sayı kilogramı (Örn 75), ikinci sayı ise kullanıcının boyu (Örn 180) cm cinsinden olmalıdır. Girilen kiloyu, 10000 ile çarpın, girilmiş boyun karesine bölüp (kg * 10000/boy/boy) T1 bellek adresine yazıp, uygulamayı sonlandırınız.

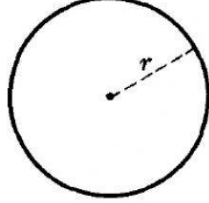
2- Bellekte Yer Değiştirme (Swap) (0 Puan):

Bellekte X ve Y adreslerinin içerisinde başlangıçta 10 ve 20 değeri giriniz. Yazacağınız kod parçacığı çalıştığı zaman X'in içerisindeki değer Y'de, Ynin içerisindeki değer X'te olsun. Program çalıştıktan sonra kendini sonlandırmalıdır.

3- Çemberin Çevresi ve Alanı (25 Puan):

Bellekte X adresine, çemberin yarı çapını, Y adresine ise π olarak 3 sayısını yazınız. Uygulama çalıştıktan sonra Z adresine çevre bilgisi, W adresine ise alan değeri yazılmalıdır. Diğer adresleri istediğiniz gibi kullanabilirsiniz. Program çalıştıktan sonra kendini sonlandırmalıdır.

$$\text{Alan} = \pi r^2$$
$$\text{Çevre} = 2\pi r$$



4- Not Hesaplama (50 Puan):

Bellekte X adresinde 1. vize, Y adresinde 2. vize ve Z adresinde ise final notu bulunmaktadır. Vize notlarının dönem notuna etkisi %30, final notunun etkisi ise %40'tır. Buna göre program çalıştıktan sonra W adresine dönem notunu yazan bir uygulama geliştirin. Diğer adresleri istediğiniz gibi kullanabilirsiniz. Program çalıştıktan sonra kendini sonlandırmalıdır.

Örnek giriş:

X: 90

Y: 50

Z: 80

Uygulama çalıştıktan sonra:

W: 74

bellekte kaydedilmiş olmalıdır.

5- Aritmetik İşlem (25 Puan):

$X = ((Y * 15) + (Z / 2)) * 5$ işlemini gerçekleştiren kod parçacığını yazınız. Diğer adresleri istediğiniz gibi kullanabilirsiniz. Program çalıştıktan sonra kendini sonlandırmalıdır.

Örnek giriş olarak

Y: 10

Z: 8

Uygulama çalıştıktan sonra:

X = 770

bellekte kaydedilmiş olmalıdır.