

CSE331 - Computer Organization

HW1 REPORT

Bu programda sayıları ekrana string şeklinde yazmayı anlamak için kurduğum koşullar şekli şu şekildedir:

1. boşluk/sayı/boşluk 2. boşluk/sayı/./boşluk 3. ./boşluk/sayı/boşluk (Büyük harf)

Bu koşullar sağlandığı zaman string şeklinde ekrana yazdırabiliyorum.

Kodda öncelikli olarak .data kısmına sayıların string halini yazdım.

.text kısmında dosya açma dosyadan okuma işlemleri var. Bu işlemten sonra kullandığım \$s0, \$s1, \$s2, \$s3, \$s4 registerları karakter karakter okuma yaparken önceki adreslere gitmek için kullandım.

myLoop döngüsünde \$s0, \$s1, \$s2, \$s3, \$s4 hepsini birer arttırıp lb kullanarak sırasıyla \$t0, \$t1, \$t2, \$t3, \$t4 atadım. \$t0 = anlık adres, \$t1 = anlık adresten iki önceki adres, \$t2 = anlık adresten bir önceki adres, \$t3 = anlık adresten bir sonraki adres, \$t4 = anlık adresten iki sonraki adres anlamına geliyor.

myLoop döngüsünün içerisinde \$t0 ' ı yani anlık adresi beq kullanarak '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' kontrolleri yaptım. **beq \$t0, \$zero, EndLoop** ile dosya sonuna gelip gelmediğini kontrol ettim. '0','1'....'9' ve **beq \$t0, \$zero, EndLoop** sağlanmazsa anlık adreste karakter vardır o yüzden **jal print_character** ile karakter yazma fonksiyonu çağırdım.

EndLoop kısmı programı bitirir.

print_character: li \$v0, 11 ve move \$a0, \$t0 komutu ile \$t0 (anlık adres) daki karakteri yazdırır.

(1. 2. 3. KOŞULLARI '5' SAYI İÇİN NASIL YAPTIĞIMI ANLATICAĞIM '0', '1', '2', '3', '4', '6', '7', '8', '9' İÇİN DE AYNI KOŞULLAR VAR)

beq \$t1, '.', state1_number5 (3. Koşul yani iki önceki adreste '.' mı var)

beq \$t4, '.', state2_1_number5 (2. Koşul yani iki sonraki adreste '' mu var)

beq \$t3, '.', state3_1_number5 (1. Koşul yani bir sonraki adreste '' mu var)

jal print_character (üstteki 3 koşul sağlanmazsa karakter yazma fonksiyonu çağırılır)

j myLoop (myLoop a geri dönme)

1. Koşul (boşluk/sayı/boşluk)

Anlık okuduğum karakteri ile '5' ise **beq \$t0,'5', num_five** ile num_five kısmına gidilir.

beq \$t3, '.', state3_1_number5 ifadesi sağlanırsa state3_1_number5 ' e gidilir. state3_1_number5'te

beq \$t2, '.', state3_2_number5 bir önceki adreste '' var mı kontrolü yapılır. Eğer sağlanırsa

state3_2_number5'a gidilir ve 5 rakamı 'five' şeklinde yazdırılır. num_five , state3_1_number5,

state3_2_number5 larda beq kısmı sağlamadığı zaman karakter yazdırma fonksiyonu çağırılır. Yani

buna şöyle bir örnek vereyim: "Ali Haydar5 Kurban." 5' ten sonra boşluk var state_3_1_number5 ' a

gider fakat 5' ten bir önceki adreste 'boşluk' olmadığı için 5' i rakam olarak yazar.

2. Koşul (boşluk/sayı/./boşluk)

Anlık okuduğum karakteri ile '5' ise **beq \$t0,'5', num_five** ile num_five kısmına gidilir.

beq \$t4, '.', state2_1_number5 ifadesi sağlanırsa state2_1_number5 ' e gidilir. state2_1_number5'te **beq \$t3, '.', state2_2_number5** bir sonraki adreste '.' var mı kontrolü yapılır. Eğer sağlanırsa state2_2_number5' a gidilir. state2_2_number5'te **beq \$t2, '.', state2_3_number5** bir önceki adreste '' var mı kontrolü yapılır. Eğer sağlanırsa state2_3_number5' e gidilir ve 5 rakamı 'five' şeklinde yazdırılır. num_five , state2_1_number5, state2_2_number5, state2_3_number5 larda beq kısmı sağlamadığı zaman karakter yazdırma fonksiyonu çağırılır.

3. Koşul (./boşluk/sayı/boşluk (Büyük harf))

Anlık okuduğum karakteri ile '5' ise **beq \$t0,'5', num_five** ile num_five kısmına gidilir.

beq \$t1, '.', state1_number5 ifadesi sağlanırsa state1_number5' e gidilir. state1_number5' te

beq \$t2, '.', state2_number5 bir önceki adreste '' var mı kontrolü yapılır. Eğer sağlanırsa state2_number5' e gidilir. state2_number5' te **beq \$t3, '.', state3_number5** bir sonraki adreste

'' var mı kontrolü yapılır. Eğer sağlanırsa state3_number5' e gidilir ve 5 rakamı 'Five' şeklinde yazdırılır. num_five , state1_number5, state2_number5, state3_number5 larda beq kısmı sağlamadığı zaman karakter yazdırma fonksiyonu çağırılır.

ALİ HAYDAR KURBAN 151044058