ELMU 4088 - Bilgisayar Mimarisi Çalışma 3

Dizi elemanlarını bulma:

DİZİ adlı oniki (12) elemanlı bir dizi aşağıdaki şekilde verilmiştir;

2 0 -4 16 0 1 -8 -10 0 2 -3

Bu dizide yer alan değeri sıfırdan (0) küçük elemanları bulan MIPS birleştirici dil (assembly language) kodunu yazınız. Bulunan dizi elemanlarını *YENİ_DİZİ* adlı yeni bir diziye depolayınız. Oluşturulan yeni dizi konsola basılacaktır.

Bir dizi için yer tahsis etme;

Dizi: .space 400 (byte cinsindendir, bu alana 100 eleman yerleştirilebilir).

Elemanları 32-bit (word) boyutlu bir diziyi .data segmentinde tanımlama;

ARRAY: .word 2, 0, -4, 16, 0, 1, -8, -10, 0, 2, -3, 11

- .data ve .text segmentlerini tanımlamayı unutmayınız.
- Kullanıcı tanımlı dizi için yer tahsis edilmelidir.
- Dizinin taban adresi belirlenmelidir.
- Çıkış için sistem çağrısı (exit system call) yapmayı unutmayınız.

Bu durum programcının kontrolü bir sonraki komut yerine programdaki bir başka konuma dallandırmasına [branch] (atlamasına [jump]) izin verir. Bu tip komutlar birleştirici (assembler) dilde koşullu işlemleri yapabilen program yürütülebilmesine olanak tanır.

Eğer \$rs kaydedicisi \$rt kaydedicisinden küçük ise, \$rd kaydedicisi set (buna 1 yazılır) edilir. Aksi halde \$rd kaydedicisine sıfır yazılır;

```
SLT RD, RS, RT # RD = (RS\pm < RT\pm) ? 1 : 0
```

İki kaydedicinin içeriği eşit ise dallanma olur;

```
BEQ RS, RT, Label #eğer RS = RT, Label'a dallan
```

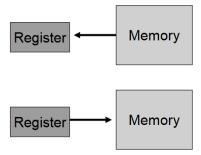
İki kaydedicinin içeriği eşit değil ise dallanma olur;

```
BNE RS, RT, Label #eğer RS ≠ RT, Label'a dallan
```

İşlem yapabilmek için değerler bellek transfer konutları kullanılarak bellekten alınmalıdır.

Load word ← lw \$t0, bellek-adresinden Store word → sw \$t0, bellek-adresine

ELMU 4088 - Bilgisayar Mimarisi Çalışma 3



MIPS komut kümesi belleğe erişim için tahsis edilmiş load ve store komutlarını içerir.

1b	Load Byte	Rt = (byte) Mem [Rs+offset]
sw	Store Word	Mem [Rs+offset] = Rt
j	Jump	döngü

> Dizi indisi 0 değerinden başlar.