

## CSE331 - Computer Organization HW1

Bu projeyi geliştirme adımlarım aşağıdaki gibidir:

- Kullanıcıdan dosya adını girdi olarak konsoldan alıyorum ve oluşturduğum ibuffer isimli diziye yerleştiriyorum.
  - Dosya adını konsoldan aldığım için dosya ismi+ '\n' karakteri birlikte ibuffer dizisine yerleştiği için ilk olarak bu '\n' karakterinden kurtulmam gerekiyordu. Bu nedenle kurduğum döngü yapısı ile bu karakteri yok sayarak oluşturmuş olduğum fbuffer isimli yeni diziye yerleştiriyorum.
  - Daha sonra dosyayı açıp okuma işlemi gerçekleştiriyorum.
  - Dosyadan okuduğum veri buffer isimli diziye atama yapıyorum.
  - Sonraki adımda tek tek karakter kontrolü yaparak gelen karakterin 0 ile 9 arasında bir rakam olup olmadığını kontrol ediyorum. Ve eğer karakter bu rakamlardan biri ise yazı karşılığı olarak değiştirip newContent isimli diziye yeni hali ile yerleştiriyorum.
  - İki basamaklı ve daha fazla basamaklı sayıları algoritmam(asıl döngüde kontrol yapılarak ) ödevde istenilen gibi yazıya çevirmemektedir.
  - Algoritmam gereği cümle sonu kontrolleri için noktadan sonra mutlaka bir boşluk bulunmalıdır.
  - Algoritmam ondalıklı sayılar ile karşılaşır (4.13 gibi) ödevde istenildiği gibi bu ondalıklı sayıları yazıya çevirmemektedir.
  - Noktadan sonra gelen karakter rakam ise ilk harfi büyük olarak çevrilir.
  - Karşılaştırma yapıldıktan sonra yazıya çevirme işlemini yapan döngü convert number procedure ile yapılmaktadır.
  - Buffer dizisi içindeki(dosyadaki tüm karakterler) tek tek gezildikten sonra newContent adlı dizi içinde bu adımlar sonucunda güncel hali ile toplanmış bulunuyor
  - Tüm işlemler bittikten sonra newContent ekrana bastırılır.
- Aşağıda deneme dosyamı ve çıktısını görmekteyiz.

```
Dosya Düzen Biçim Görünüm Yardım
0 1 2 3 4 5 6 7 8 9. Bugün pazardan 12 elma aldım, 2 tanesini yedim. 7 tane
elma çürük çıktı. Bugün günlerden ekim 3. 4 ekim yarının tarihidir.
Pi sayısının değeri 3.14.
```

```
Mars Messages Run I/O
Enter file name: hw1.txt
zero one two three four five six seven eight nine. Bugün pazardan 12 elma aldım, two tanesini yedim. Seven tane elma çürük çıktı. Bugün günlerden ekim three. Four ekim yarının tarihidir. Pi sayısının değeri 3.14.
-- program is finished running --
Clear
```

Enter file name: hw1.txt

zero one two three four five six seven eight nine. Bugün pazardan 12 elma aldım, two tanesini yedim. Seven tane elma çürük çıktı. Bugün günlerden ekim three. Four ekim yarının tarihidir. Pi sayısının değeri 3.14.

-- program is finished running --

Get filename from user and save filename in ibuffer.

```

removeSpace:
    lb $t3, 0($a0)
    beq $t3, $t2, ifremove
        sb $t3, 0($a1)
        addi $a0, $a0, 1
        addi $a1, $a1, 1
    j doneremove
ifremove:
    sb $t3, 0($a1)
    addi $a0, $a0, 1
    addi $a1, $a1, 1
    lb $t3, 0($a0)
    sb $t3, 0($a1)
    addi $a0, $a0, 1
    addi $a1, $a1, 1
    lb $t3, 0($a0)
    sb $t3, 0($a1)
    addi $a0, $a0, 1
    addi $a1, $a1, 1
    lb $t3, 0($a0)
    sb $t3, 0($a1)
    j finish
    doneremove:
        j removeSpace

```

Kullanıcıdan alınan dosya ismi içerisindeki '\n' i kaldıran kod bu şekildedir. t2 registerında noktanın ascii değerinin decimal değeri vardır. Bu değerle karşılaştırmaya kadar tüm dosya ismi fbuffera yazılır noktadan sonra uzandı txt oldu 3 karakter daha okunarak döngü bitirilir.

**readArrayloop:** →

```
lb $a0, 0($t2)
```

readArrayLoop arrayden alınan karakter kontrolü bu döngü ile yapılır.

Arrayde gezinerek 0 ile 9 arasında karakter olup olmadığı beq instructionı ile kontrol edilir.

# 0 kontrolü

```
li $t1, 48
```

```
beq $a0, $t1, ifZero
```

# 1 kontrolü

```
li $t1, 49
```

```
beq $a0, $t1, ifOne
```

# 2 kontrolü

```
li $t1, 50
```

```
beq $a0, $t1, ifTwo
```

# 3 kontrolü

```
li $t1, 51
```

```
beq $a0, $t1, ifThree
```

·  
·  
·  
·

# 8 kontrolü

```
li $t1, 56
```

```
beq $a0, $t1, ifEight
```

# 9 kontrolü

```
li $t1, 57
```

```
beq $a0, $t1, ifNine
```

```
sb $a0, 0($a3)
```

Dizideki karakterin sonuna gelindiğinde döngünün sonlanması burada kontrol edilir ← beq \$a0, \$zero, exitLoop

##### convert number function #####

```
convert_number:
```

```
numberArray
```

```
lb $a1, 0($a0)
```

```
sb $a1, 0($a3)
```

```
beq $a1, $zero, exit
```

```
addi $a0, $a0, 1
```

```
addi $a3, $a3, 1
```

```
j numberArray
```

```
exit:
```

```
addi $t2, $t2, 1
```

```
jr $ra
```

Bu fonksiyon a0 registerı ile parametresini alır. Parametresinde karşılaşılan rakamın yazı olarak data kısmında tanımlanan stringin başlangıç adresini alır. Döngü ile rakamın karakter arrayi olarak tutulan yazı şeklini karakter karakter olarak newContente yerleştirir.

