



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μάθημα: Αρχιτεκτονική Υπολογιστών

Ονοματεπώνυμο: Ειρήνη Δόντη

A.M.: 03119839

1^η Σειρά Γραπτών Ασκήσεων

5^ο Εξάμηνο

Τμήμα 1^ο

Αθήνα

2021 - 2022

Άσκηση 1

Τα μεταφρασμένα προγράμματα από κώδικα MIPS σε C είναι τα ακόλουθα:

A)

```
int pow = 1;

int x = 0;

while (pow != 128) {

    pow = 2*pow;

    x +=1;

}
```

B)

```
for( int i =0; i < 1000; i++) {

    arr[i] =8*arr[i];

}
```

Άσκηση 2

Οι δοσμένες συναρτήσεις γλώσσας C, μετατράπηκαν σε κώδικα MIPS και εκτελέστηκαν μέσω του προσομοιωτή QtSpim. Παρακάτω, παρουσιάζονται τα ζητούμενα προγράμματα σε μορφή κειμένου με τα αντίστοιχα στιγμιότυπα οθόνης τους από τον text editor.

A)

```
# Eirini Donti 03119839
```

```
.data
```

```
input: .asciiz "teststring"
```

```
output: .asciiz "String Length: "
```

```
.text
```

```
main:
```

```
    la $a0, input    # load address of input string
```

```
    jal strlen       # call strlen function
```

```
    li $v0, 4        # call to print output message
```

```
    la $a0, output
```

```
    syscall
```

```
li $v0, 1      # call to print length of the string
```

```
move $a0, $t0
```

```
syscall
```

```
addi $v0, $0, 10 # exit program
```

```
syscall
```

strlen:

```
li $t0, 0      # int i = 0;
```

l1:

```
lb $t1, 0($a0) # load character to $t1
```

```
beqz $t1, exit # if null character => exit
```

```
addi $a0, $a0, 1 # increase string pointer
```

```
addi $t0, $t0, 1 # i = i + 1;
```

```
j l1          # jump l1
```

exit:

```
jr $ra
```

```

# Eirini Donti 03119839

.data

input: .asciiz "teststring"

output: .asciiz "String Length: "

.text

main:
    la $a0, input      # load address of input string
    jal strlen         # call strlen function

    li $v0, 4          # call to print output message
    la $a0, output
    syscall

    li $v0, 1          # call to print length of the string
    move $a0, $t0
    syscall

    addi $v0, $0, 10    # exit program
    syscall

strlen:
    li $t0, 0          # int i = 0;
l1:
    lb $t1, 0($a0)     # load character to $t1
    beqz $t1, exit      # if null character => exit
    addi $a0, $a0, 1    # increase string pointer
    addi $t0, $t0, 1    # i = i + 1;
    j l1               # jump l1

exit:
    jr $ra

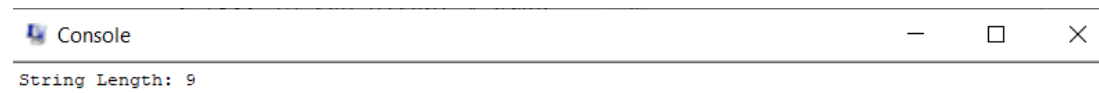
```

Το παραπάνω στιγμιότυπο οθόνης απεικονίζει τον κώδικα που δακτυλογραφήθηκε στον text editor και προσομοιώθηκε στο πρόγραμμα QtSpim.

Έπειτα, θα ανεβάσουμε το αρχείο στον προσομοιωτή και θα εκτελέσουμε το παραπάνω πρόγραμμα για συγκεκριμένες συμβολοσειρές:

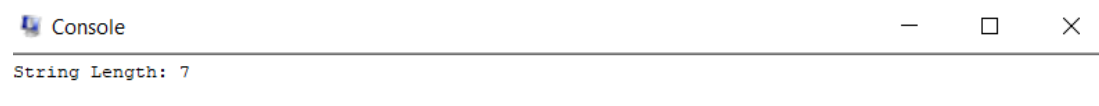
1^η εκτέλεση με τη συμβολοσειρά “structure”:

Η κονσόλα εμφάνισε ότι η συμβολοσειρά έχει 9 χαρακτήρες.



2^η εκτέλεση με τη συμβολοσειρά “example”:

Η κονσόλα εμφάνισε ότι η συμβολοσειρά έχει 7 χαρακτήρες.



3^η εκτέλεση με τη συμβολοσειρά “electronical”:

Η κονσόλα εμφάνισε ότι η συμβολοσειρά έχει 12 χαρακτήρες.



Παρατηρούμε ότι το πρόγραμμα τύπωσε, σε όλες τις περιπτώσεις, σωστά αποτελέσματα.

B)

Eirini Donti 03119839

.data

input: .asciiz "n = "

output: .asciiz "\nfib(n) = "

n: .word 0

result: .word 0

.text

main:

li \$v0, 4 # call to print input message

la \$a0, input

syscall

li \$v0, 5 # call to read integer n

syscall

sw \$v0, n

```
lw $a0, n
```

```
jal fib      # call fib function
```

```
sw $v0, result
```

```
li $v0, 4      # call to print output message
```

```
la $a0, output
```

```
syscall
```

```
li $v0, 1      # call to print integer n
```

```
lw $a0, result
```

```
syscall
```

```
li $v0, 10     # call to end program
```

```
syscall
```

```
fib:
```

```
subu $sp, $sp, 8  # push registers $ra, $s0
```

```
sw  $ra, ($sp)
```

```
sw  $s0, 4($sp)
```

```
move $v0, $a0
```

```
beq $a0, 0, exit  # if n = 0 exit
```



```
beq $a0, 1, exit # else if n = 1 exit
```

```
    #else
```

```
    move $s0, $a0
```

```
    sub $a0, $a0, 1 # n = n - 1;
```

```
    jal fib        # call fib(n-1)
```

```
    move $a0, $s0
```

```
    sub $a0, $a0, 2 # n = n - 2;
```

```
    move $s0, $v0 # save fib(n-1)
```

```
    jal fib        # call fib(n-2)
```

```
    add $v0, $s0, $v0 # fib(n-1) + fib(n-2)
```

```
exit:
```

```
    lw $ra, ($sp)
```

```
    lw $s0, 4($sp)
```

```
    addu $sp, $sp, 8 # clear stack
```

```
    jr $ra
```

Το παρακάτω στιγμιότυπο οθόνης απεικονίζει τον κώδικα που δακτυλογραφήθηκε στον text editor και προσομοιώθηκε στο πρόγραμμα QtSpim.

```
# Eirini Donti 03119839
```

```
.data
```

```
input: .asciiz "n = "
```

```
output: .asciiz "\nfib(n) = "
```

```
n: .word 0
```

```
result: .word 0
```

```
.text
```

```
main:
```

```
li $v0, 4          # call to print input message
la $a0, input
syscall
```

```
li $v0, 5          # call to read integer n
syscall
```

```
sw $v0, n
lw $a0, n
```

```
jal fib            # call fib function
```

```
sw $v0, result
```

```
li $v0, 4          # call to print output message
la $a0, output
syscall
```

```
li $v0, 1          # call to print integer n
lw $a0, result
syscall
```

```
li $v0, 10         # call to end program
syscall
```

```
fib:
```

```
subu $sp, $sp, 8   # push registers $ra, $s0
sw $ra, ($sp)
sw $s0, 4($sp)
```

```
move $v0, $a0
beq $a0, 0, exit    # if n = 0 exit
beq $a0, 1, exit    # else if n = 1 exit
```

```
    #else
```

```
move $s0, $a0
sub $a0, $a0, 1     # n = n - 1;
jal fib             # call fib(n-1)
```

```
move $a0, $s0
sub $a0, $a0, 2     # n = n - 2;
move $s0, $v0       # save fib(n-1)
jal fib             # call fib(n-2)
```

```
add $v0, $s0, $v0   # fib(n-1) + fib(n-2)
```

```
exit:
```

```
lw $ra, ($sp)
lw $s0, 4($sp)
addu $sp, $sp, 8    # clear stack
```

```
jr $ra
```

Έπειτα, θα ανεβάσουμε το αρχείο στον προσομοιωτή και θα εκτελέσουμε το παραπάνω πρόγραμμα για συγκεκριμένες τιμές του φυσικού αριθμού n (εισάγοντας στην κονσόλα τον αριθμό n που θέλουμε):

1^η εκτέλεση για $n = 0$:

Η κονσόλα εμφάνισε ότι $\text{fib}(n) = 0$.



```
Console
n = 0
fib(n) = 0
```

2^η εκτέλεση για $n = 1$:

Η κονσόλα εμφάνισε ότι $\text{fib}(n) = 1$.



```
Console
n = 1
fib(n) = 1
```

3^η εκτέλεση για $n = 15$:

Η κονσόλα εμφάνισε ότι $\text{fib}(n) = 610$.



```
Console
n = 15
fib(n) = 610
```

Παρατηρούμε ότι το πρόγραμμα τύπωσε, σε όλες τις περιπτώσεις, σωστά αποτελέσματα.