
ARKITETURA E KOMPJUTERËVE



DETYRA 1

OSMAN BYTYQI
190714100081
April 2022

Kodi në gjuhën C++

Kodi në vazhdim paraqet një funksion i cili si vlere pranon parametrin test ku është hardcoduar me vleren 4, më pas funksioni do të shtyp të gjithë numrat me rënditje prej 4 deri në 1 dhe prej 1 deri ne 4.

Kusht i nevojshëm që ky funksion të vazhdoj me ekzekutim është që vlera e variables test apo parametrin në këtë rast të jetë më e madhe se 1 përndryshe programi do të ndaloj ekzekutuar, ku këtë kusht e kemi arritur përmes një if bloku duke kontrolluar që test variabla të jetë më e madhe se 1, dhe në fund e kemi thirrur në funksionin main.

```
1
2  #include<iostream>
3  using namespace std;
4  void printFun(int test){
5      if (test<1){
6          return ;
7      }
8      else {
9          cout<<test<<" ";
10         printFun(test-1);
11         cout<<test<<" ";
12         return ;
13     }
14 }
15 int main(){
16     int test=4;
17     printFun(test);
18 }
```

Realizimi i Kodit në Mips

```
.text
.globl main

main:
addi $s0,$zero,4
add $a0,$s0,$zero      # vendosja e vleres se regjistrit s0 ne argument
jal printFun           #thirrja e funkstionit
printFun:
addi $sp,$sp,-8         # rezervimi i hapsires ne stack
sw $ra,4($sp)           # rezervimi i vleres return ne stack
sw $s0,0($sp)           # rezervimi i vleres se parametrut ne stack
slti $t0,$s0,1          #test for n<1
beq $t0,$zero,else      #if test>=1 go to else
beq $t0,1 else2
addi $sp,$sp,8          #lirimi i hapsires se stackut

printfun2:
addi $sp,$sp,-8         # rezervimi i hapsires ne stack
sw $ra,4($sp)           # rezervimi i vleres return ne stack
sw $s0,0($sp)           # rezervimi i vleres se parametrut ne stack
slti $t0,$s0,4          #test for n<1
beq $t0,1 else2
addi $sp,$sp,8          #lirimi i hapsires se stackut
li $v0,10
syscall

else:
li $v0,1                # print test
move $a0,$s0
syscall

li $v0,4                #print space
```

```
la $a0,space           #load address of space
syscall
```

```
addi $s0,$s0,-1        #t-1
sw $s0,0($sp)          #ruajta ne stack
j printFun              #kcimi ne printfun
addi $sp,$sp,8          #zbrasja e stackut
```

```
li $v0,1                # print test
move $a0,$a0
syscall
```

```
li $v0,4                #print space
la $a0,space            #load address of space
syscall
addi $v0,$zero,0        #return
jr $ra
```

```
else2:
addi $s0,$s0,1
li $v0,1                # print test
move $a0,$s0
syscall
```

```
li $v0,4                #print space
la $a0,space            #load address of space
syscall
```

```
sw $s0,0($sp)          #ruajta ne stack
j printfun2             #kcimi ne printfun
addi $sp,$sp,8          #zbrasja e stackut
```

```
.data
space: .asciiz " "
```

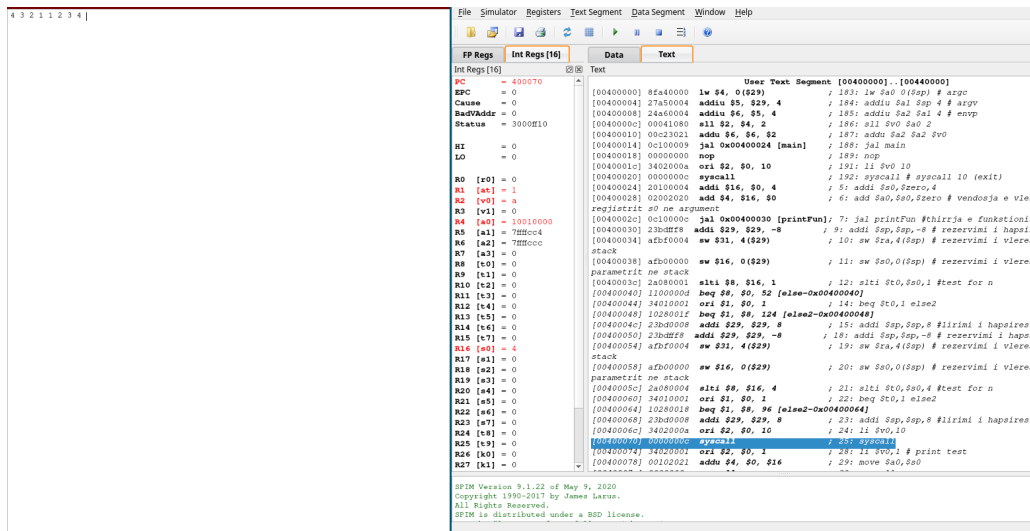


Figura 1: Testimi

Përpos implementimit të logjikës së menyrës së zbritjes së parametrimit një pjesë tjetër interesante ka qenë puna me stack e cila ka dashur një koncentrim më të madh.

Përfundimi

Arkitektura e Kompjuterëve është ndër lëndët e vetme deri tash ku nuk kam pasur aq njohuri paraprake, andaj edhe zgjodha opsionin e parë në mënyrë që të hulumtoj më shumë dhe të mësoj më mirë.

Përfundimi i detyrës nuk ishte edhe aq i lehtë, mirëpo kjo ndikoi që të përvetësoj ato që janë shpjeguar në ligjërata dhe ushtrime dhe t'i kuptoj ato që nuk i kam pasur edhe aq të qarta, çka edhe është qëllim i detyrave të tilla. Në ndihmë kanë ardhur artikuj të ndryshëm në Google dhe inqizime të llojllojshme në YouTube. Me kombinim të të gjitha veglave ndihmëse, zgjidhja e detyrës mori punë disa orëshe brenda disa ditëve