UNIVERSITETI I PRISHTINËS "HASAN PRISHTINA"

FAKULTETI I INXHINIERISE ELEKTRIKE DHE KOMPJUTERIKE

Lënda : Ariktektura e kompjutereve

DETYRA 1

Profesore :

Valon Raca

Punoi:

Shefket Bylygbashi

200714100079

Prishtinë , 2022

HYRJE

Detyra:

```
Opsioni A :
Të shkruhet në MIPS assembly code kodi i mëposhtëm në C++ :
        // A C++ program to demonstrate ëorking of
        // recursion
```

```
#include <bits/stdc++.h>
        using namespace std;
void printFun(int test)
{
        if (test < 1)
                return;
        else {
                cout << test << " ";
                printFun(test - 1); // statement 2
                cout << test << " ";
                return;
        }
}
// Driver Code
int main()
{
        int test = 4;
        printFun(test);
}
```

Ky është një program që demostrone se si punon rekursioni.

Rekursioni është procesi në të cilin një funksion thërret veten drejtpërdrejt ose tërthorazi dhe funksioni përkatës quhet funksion rekurziv. Duke përdorur algoritmin rekurziv, disa probleme zgjidhen lehtë.

Shembuj të problemeve të tilla janë : Kullat e Hanoi (TOH), Inorder/Preorder/Postorder Tree Transals, DFS of Graph, etj.

Realizimi i kodit ne MIPS

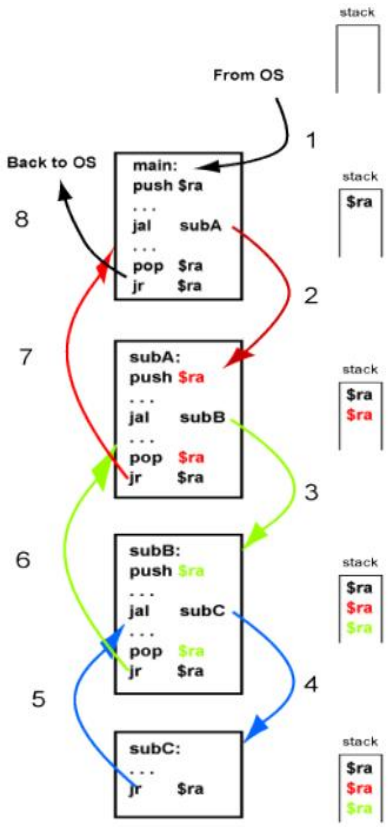
```
1   .text
2   .globl main
3   main:
4   addi $a0, $zero,4
5
```

Në fillim deklarojme funksionin main dhe I japim vlerën e caktuar regjistrit $a0= 4 përmes instruktsionit addi-Iformat.

```
8   printFun:
9
10      addi $sp, $sp, -8
11      sw $ra, 4($sp)
12      sw $a0, 0($sp)
13
```

Pasi kemi me thirr procedure I vendosim në stek regjistrat te cilët kemi me I përdorë. Pasi ka me thirr vazhdimisht vetveten në form recursive , kështu që ne duhet ruajtur secilën here që e thirr vetveten return adresën në stek , për arsye që kur të kthehemi nga procedura të dim se ku duhet vazhduar ekzekutuar më tej.Në stek duhet ruajtur edhe $a0 ,pasi te funksioni kjo zvoglohet për 1.

Pra e vendos return adresen dhe argumentin në stek, I cili punon me parimin LIFO I cili na mundeson rekusionin .



Në programin tone.

$a0	1
$ra	400064
	2
	400064
	3
	400064
	4
	400024

```
16
17          slti $t0, $a0, 1
18          beq $t0, $zero, else
19
```

Shikjome a është a0(n)<1 : vendose regjitstrin $t0 ne 1 nese $a0<1, perndryshe $t0= 0

beq $t0, $zero, else : nese $t0 = 0 ,shko te else

ELSE FUNKSIONI:

```
23      else:
24          li $v0 ,1
25          add $t0 ,$a0 ,$zero
26          syscall
27
28          li $a0, 32
29          li $v0, 11
30          syscall
31
32          addi $a0, $t0, -1
33          jal printFun
```

Tani printon vleren e $a0 , si dhe hapsiren (ascii -code 32 vlera e space) , dhe e zvoglonë vleren e $a0 për një .

jal printFun : shko pa kusht te printFun dhe ruaje adresen e instruktsionit te ardhshem në $ra.

Në momentin kur n<0 fillon të ekzekutohet pjesa tjetër e kodit:

```
19
20          addi $sp, $sp, 8
21          jr $ra
22
```

addi $sp, $sp, 8 : e ngrit vleren në stek për 8 më lart

jr $ra : shko pa kusht te instruktsioni adresa e të cilit është në regjistrin $ra

Fillojmë ti nxjerrim të dhënat nga steku:

```
35
36      lw $a0, 0($sp)
37      lw $ra, 4($sp)
38      addi $sp, $sp, 8
39
40
41      li $v0 ,1
42      add $t0 ,$a0 ,$zero
43      syscall
44
45      li $a0, 32
46      li $v0, 11
47      syscall
48      jr $ra
```

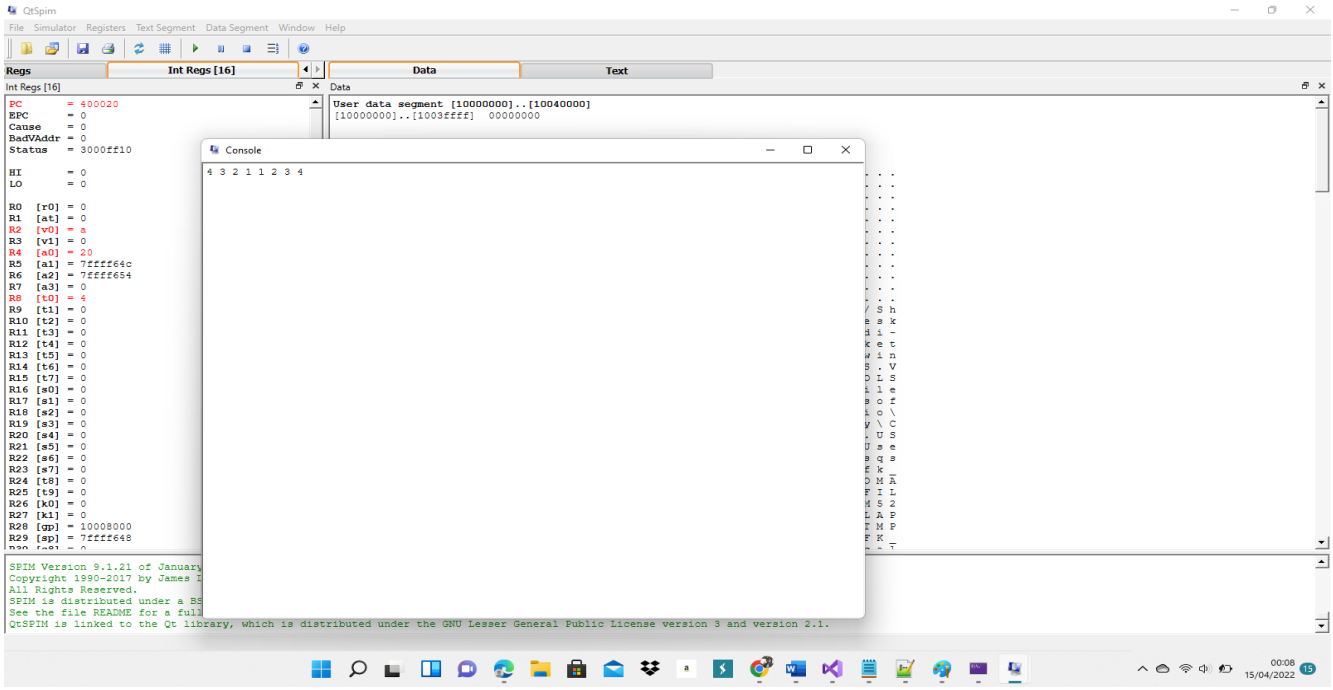
$a0	1
$ra	400064
	2
	400064
	3
	400064
	4
	400024

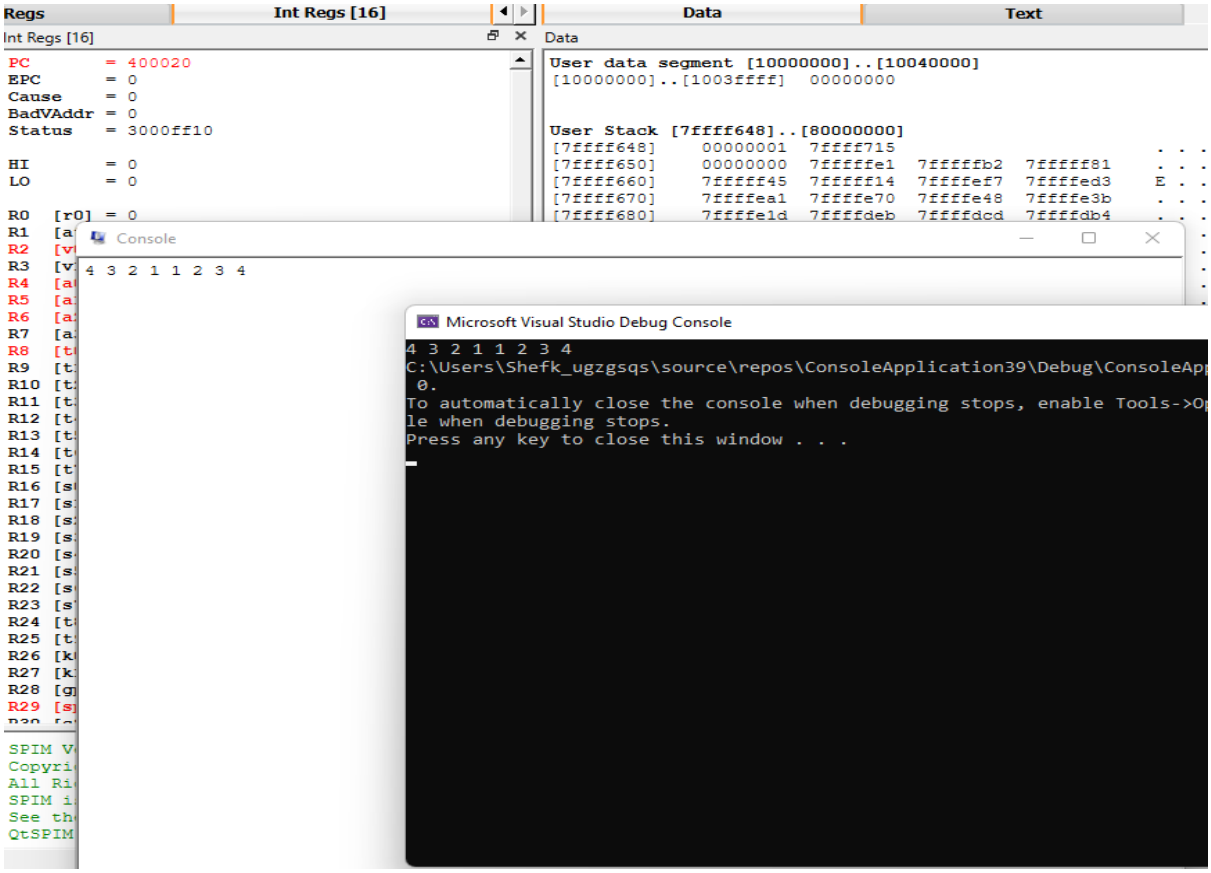
Shtypim vlerat e nxjerra dhe hapsirën pastaj vazhdon përsëritja e kodit deri në nxjerrjen e vlerës se fundit nga steku nmrin 4.

Rezultati : 4 3 2 1 1 2 3 4

Testimet me QtSpim

Testimi I kodit që është shkruar në MIPS asembler që korrespondon me kodin e dhënë në C++ për vleren 4.





Krahasimi I Consolave si rezultet I ekzekutimit te kodit te shkruar ne C++ dhe MIPS assembler per hyrjen 12.