

UNIVERSITETI I PRISHTINËS
“HASAN PRISHTINA”
FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE
DEGA: INXHINIERI KOMPJUTERIKE



Lënda: Arkitektura e kompjutëreve

Prof. Valon Raca

Detyra 1

Punoi: Egzon Gashi.

Nr. ID: 180715100044.

Hyrje

Detyra në C++.

```
#include <iostream>
#include <string>
using namespace std;

int operacioniMeVektore(int v1[], int v2[], int n){
    int temp1, temp2, sum = 0;

    for(int i=0; i < n; i++){
        temp1 = v1[i];
        temp2 = v2[n-1-i];
        v1[i] = temp1+temp2;
        v2[n-1-i] = temp1-temp2;
        sum = sum + v1[i] + v2[n-1-i];
    }
    return sum;
}

int main(){
    int n = 6;
    int vektori1[] = {5, 4, 9, 17, 31, 8};
    int vektori2[] = {-5, 2, 3, 4, 6, -3};
    int rezultati = operacioniMeVektore(vektori1, vektori2, n);

    cout << "Rezultati i operacionit: " << rezultati;
}
```

Në këtë detyrë të dhënë në gjuhën programuese C++, kemi një funksion jasht main dhe funksionin main, ka të deklaruar një integer dhe dy vektor me vlera të caktuara, pastaj në main e therrasim funksionin e dhënë jashta main-it, dhe atë rezultat që e kthen e paraqesim në console, funksioni bënë disa veprime të cilat I kemi hasur edhe më parë.

Realizimi i kodit ne MIPS

```
.data
Vektori1: .word 5, 4, 9, 17, 31, 8
Vektori2: .word -5, 2, 3, 4, 6, -3
rezultati: .ascii "Rezultati i operacionit: "
```

Në këtë pjesë e kam bërë deklarinimin e vektorëve dhe cout-in final.

```
.text
operacioniMeVektore:
la $t5, Vektori1
la $t6, Vektori2
li $t1, 0
```

Në këtë pjesë kam ngarkuar adresën e vektorit 1 dhe 2 në regjistrat \$t5 dhe \$t6. Dhe me vlerën 0 kam ngarkuar regjistrin \$t1.

```
main:
li $t2, 6
la $a1, Vektori1
la $a2, Vektori2
addi $a3, $t2, 0
```

Tek main-l në regjistrin \$t2 e vendosim vlerën 6, pra int n = 6 në C++, dhe l vendosim adresat e vektorit1 dhe vektorit2 në regjistrat \$a1 dhe \$a2.

```
jal operacioniMeVektore
```

Bëjmë një jump and link tek operacioniMeVektor.

```

loop_:
    bge $t1, $t2, loop_exit

    sll $s4, $t1, 2
    add $s4, $s4, $t5
    lw  $t7, 0($s4)
    addi $s6, $t7, 0

    addi $t0, $t2, -1
    sub $s2, $t0, $t1

    addi $s5, $s2, 0
    mul $s5, $s5, 4
    add $s3, $s5, $t6
    lw  $t3, 0($s3)
    addi $s7, $t3, 0

    add $t7, $s6, $s7
    sub $t3, $s6, $s7

    add $s1, $s1, $t7
    add $s1, $s1, $t3

    addi $t1, $t1, 1
    j loop_

```

Në këtë pjesë e kam bërë pjesën e loop-ës në C++, fillimisht me anë të bge (branch on greater than or equal) e kam caktuar se kur duhet të dalim nga loop, e të shkojmë tek loop_exit.

Në vazhdim e kam bërë temp1 = v1[i] dhe temp2 = v2[n-1-i], pastaj shumën e këtyre dyave, pra temp1 dhe temp2, I kam dërguar në v1[i], kurse zbritjen e tyre në v2[n-1-i], dhe në fund e gjejmë shumën, kjo ecuri vazhdon deri sa të vlejë kushti.

```

loop_exit:
    addi $v1, $s1, 0
    jr $ra

```

Posa të dalim nga loop-a atëherë I brazojmë \$s1 me \$v1 dhe me anë të jump register-it shkojmë të adresa stored në \$ra.

```
li $v0, 4  
la $a0, rezultati  
syscall
```

E shfaqim pjesën e string-ut “Rezultati i operacionit: ”.

```
li $v0, 1  
la $a0, ($v1)  
syscall
```

E nxjerrim sum-in, dhe e shfaqim në console.

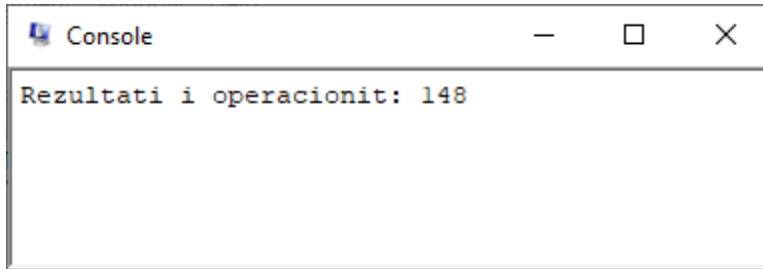
```
li $v0, 10  
syscall
```

Dhe mbyllet programi.

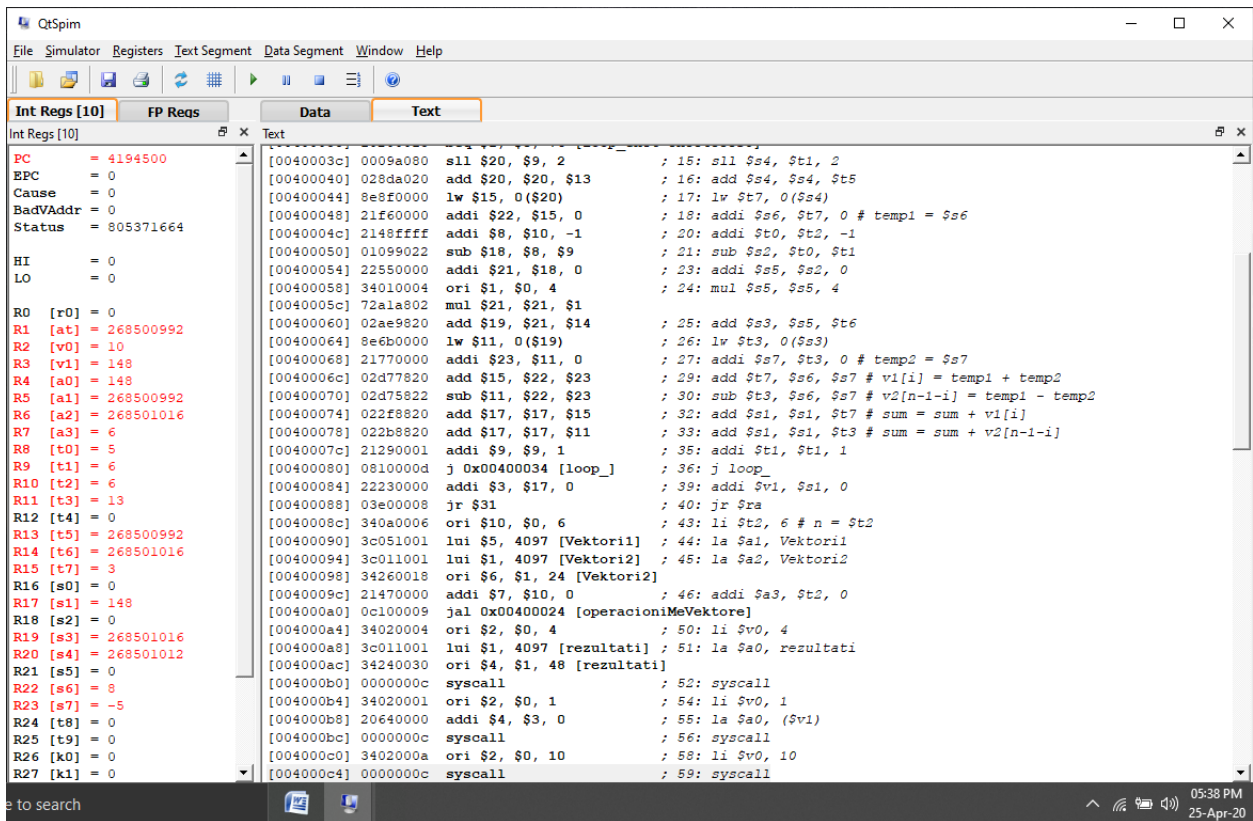
Testimet me QtSpim

Int Regs [10]		x
PC	=	4194500
EPC	=	0
Cause	=	0
BadVAddr	=	0
Status	=	805371664
HI	=	0
LO	=	0
R0 [r0]	=	0
R1 [at]	=	268500992
R2 [v0]	=	10
R3 [v1]	=	148
R4 [a0]	=	148
R5 [a1]	=	268500992
R6 [a2]	=	268501016
R7 [a3]	=	6
R8 [t0]	=	5
R9 [t1]	=	6
R10 [t2]	=	6
R11 [t3]	=	13
R12 [t4]	=	0
R13 [t5]	=	268500992
R14 [t6]	=	268501016
R15 [t7]	=	3
R16 [s0]	=	0
R17 [s1]	=	148
R18 [s2]	=	0
R19 [s3]	=	268501016
R20 [s4]	=	268501012
R21 [s5]	=	0
R22 [s6]	=	8
R23 [s7]	=	-5
R24 [t8]	=	0
R25 [t9]	=	0
R26 [k0]	=	0
R27 [k1]	=	0
R28 [gp]	=	268468224
R29 [sp]	=	2147480916
R30 [s8]	=	0
R31 [ra]	=	4194468

Popullimi i regjistrave në QtSpim.



Rezultati në QtSpim.



E gjithë dritarja e QtSpim.