

ARHITEKTURA RAČUNARA

DRUGI KOLOKVIJUM, I GRUPA

NAPOMENE:

- UKOLIKO STUDENT RADI ZADATAK NA VIŠE NAČINA, DUŽAN JE DA OZNAČI KOJI JE NAČIN IZRADE POTREBNO PREGLEDATI. U SUPROTNOM ZADATAK NEĆE BITI PREGLEDAN I BIĆE OCIJENJEN SA 0 BODOVA.
- PRI OCJENJIVANJU ZADATAKA I PITANJA ĆE SE BODOVATI SAMO NJIHOVA POTPUNA IZRADA, T.J. NEĆE SE BODOVATI REZULTATI BEZ POSTUPKA KOJIM SE DOŠLO DO TIH REZULTATA.

<p>1. a) Objasniti ulogu registara \$2 i \$4. (2 poena)</p> <p>b) Odrediti sadržaj registra \$10 nakon izvršenja instrukcije slt \$10,\$15,\$16. Sadržaji registara \$15 i \$16 su 4 i 5, respektivno. (2 poena)</p> <p>c) Sadržaji registara \$10 i \$11 su 10 i 11, respektivno. Da li će nakon izvršenja instrukcije beq \$10,\$11,Lab doći do skoka na labelu Lab? Objasniti. (2 poena)</p> <p><i>preporucem</i></p>	<p>2. Dat je dio koda u programskom jeziku C:</p> <pre>int a=0; int b=9; while(b>=a) { if(b<5) b=b-2; b--; a++; }</pre> <p>Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim a i b dodijeljeni registri \$15 i \$16, respektivno. (9 poena)</p>
<p>3. Dat je dio koda u MIPS asemblerskom jeziku:</p> <pre>addi \$8,\$0,1 addi \$9,\$0,0 petlja: slti \$10,\$8,4 beq \$10,\$0,Izlaz add \$9,\$9,\$8 addi \$8,\$8,1 j petlja Izlaz: addi \$9,\$9,-3</pre> <p>Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8, \$9 i \$10 nakon njegovog izvršenja. (8 poena)</p>	<p>4. Data je procedura u programskom jeziku C koja elemente niza cijelih brojeva koji su jednaki koeficijentu k zamjenjuje dvostrukom negativnom vrijednošću tog koeficijenta. Procedura kao rezultat vraća broj izvršenih zamjena. Niz array, broj elemenata niza N i koeficijent k su argumenti procedure. Napisati odgovarajući MIPS kod. (12 poena)</p> <pre>int procedura(int array[], int N, int k) { int i, br=0; for(i=0;i<N;i++) if(array[i]==k) { array[i]=-2*k; br++; } return br; }</pre>

ARHITEKTURA RAČUNARA

DRUGI KOLOKVIJUM, II GRUPA

NAPOMENE:

- UKOLIKO STUDENT RADI ZADATAK NA VIŠE NAČINA, DUŽAN JE DA OZNAČI KOJI JE NAČIN IZRADE POTREBNO PREGLEDATI. U SUPROTNOM ZADATAK NEĆE BITI PREGLEDAN I BIĆE OCIJENJEN SA 0 BODOVA.
- PRI OCJENJIVANJU ZADATAKA I PITANJA ĆE SE BODOVATI SAMO NJIHOVA POTPUNA IZRADA, T.J. NEĆE SE BODOVATI REZULTATI BEZ POSTUPKA KOJIM SE DOŠLO DO TIH REZULTATA.

<p>1. a) Objasniti ulogu registara \$3 i \$6. (2 poena)</p> <p>b) Odrediti sadržaj registra \$10 nakon izvršenja instrukcije slt \$10,\$11,\$12. Sadržaji registara \$11 i \$12 su isti. (2 poena)</p> <p>c) Sadržaji registara \$8 i \$9 su 8 i 9, respektivno. Da li će nakon izvršenja instrukcije bne \$8,\$9,Lab doći do skoka na labelu Lab? Objasniti. (2 poena)</p>	<p>2. Dat je dio koda u programskom jeziku C:</p> <pre>int m=1; int n=5; while(n<7) { if(n>=3) m=m+n; n+=1; }</pre> <p>Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim m i n dodijeljeni registri \$10 i \$11, respektivno. (9 poena)</p>
<p>3. Dat je dio koda u MIPS asemblerskom jeziku:</p> <pre>addi \$8,\$0,6 addi \$9,\$0,0 petlja: slti \$10,\$8,4 bne \$10,\$0,Izlaz add \$9,\$9,\$8 addi \$8,\$8,-1 j petlja Izlaz: addi \$9,\$9,-3</pre> <p>Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8, \$9 i \$10 nakon njegovog izvršenja. (8 poena)</p>	<p>4. Data je procedura u programskom jeziku C koja elemente niza cijelih brojeva koji su različiti od koeficijenta k zamjenjuje dvostrukom negativnom vrijednošću tog koeficijenta. Procedura kao rezultat vraća broj izvršenih zamjena. Niz array, broj elemenata niza N i koeficijent k su argumenti procedure. Napisati odgovarajući MIPS kod. (12 poena)</p> <pre>int change(int array[], int N, int k) { int i, br=0; for(i=0;i<N;i++) if(array[i]!=k) { array[i]=-2*k; br++; } return br; }</pre>

ARHITEKTURA RAČUNARA

DRUGI KOLOKVIJUM, III GRUPA

NAPOMENE:

- UKOLIKO STUDENT RADI ZADATAK NA VIŠE NAČINA, DUŽAN JE DA OZNAČI KOJI JE NAČIN IZRADE POTREBNO PREGLEDATI. U SUPROTNOM ZADATAK NEĆE BITI PREGLEDAN I BIĆE OCIJENJEN SA 0 BODOVA.
- PRI OCJENJIVANJU ZADATAKA I PITANJA ĆE SE BODOVATI SAMO NJIHOVA POTPUNA IZRADA, T.J. NEĆE SE BODOVATI REZULTATI BEZ POSTUPKA KOJIM SE DOŠLO DO TIH REZULTATA.

<p>1. a) Objasniti ulogu registara \$0 i \$5. (2 poena)</p> <p>b) Odrediti sadržaj registra \$20 nakon izvršenja instrukcije slt \$20,\$15,\$16. Sadržaji registara \$15 i \$16 su 8 i 6, respektivno. (2 poena)</p> <p>c) Sadržaji registara \$13 i \$14 su 13 i 14, respektivno. Da li će nakon izvršenja instrukcije beq \$13,\$14,Lab doći do skoka na labelu Lab? Objasniti. (2 poena)</p>	<p>2. Dat je dio koda u programskom jeziku C:</p> <pre>int x=0; int y=13; while(y>=x) { if(y<6) y=y-3; y--; x++; }</pre> <p>Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim x i y dodijeljeni registri \$13 i \$14, respektivno. (9 poena)</p>
<p>3. Dat je dio koda u MIPS asemblerskom jeziku:</p> <pre> addi \$8,\$0,2 petlja: addi \$9,\$0,0 slti \$10,\$8,6 beq \$10,\$0,Izlaz add \$9,\$9,\$8 addi \$8,\$8,1 j petlja Izlaz: addi \$9,\$9,-2</pre> <p>Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8, \$9 i \$10 nakon njegovog izvršenja. (8 poena)</p>	<p>4. Data je procedura u programskom jeziku C koja elemente niza cijelih brojeva koji su veći od koeficijenta k zamjenjuje negativnom vrijednošću tog koeficijenta. Procedura kao rezultat vraća broj izvršenih zamjena. Niz array, broj elemenata niza N i koeficijent k su argumenti procedure. Napisati odgovarajući MIPS kod. (12 poena)</p> <pre>int procedura(int array[], int N, int k) { int i, br=0; for(i=0;i<N;i++) if(array[i]>k) { array[i]=-k; br++; } return br; }</pre>

ARHITEKTURA RAČUNARA

DRUGI KOLOKVIJUM, IV GRUPA

NAPOMENE:

- UKOLIKO STUDENT RADI ZADATAK NA VIŠE NAČINA, DUŽAN JE DA OZNAČI KOJI JE NAČIN IZRADE POTREBNO PREGLEDATI. U SUPROTNOM ZADATAK NEĆE BITI PREGLEDAN I BIĆE OCIJENJEN SA 0 BODOVA.
- PRI OCJENJIVANJU ZADATAKA I PITANJA ĆE SE BODOVATI SAMO NJIHOVA POTPUNA IZRADA, T.J. NEĆE SE BODOVATI REZULTATI BEZ POSTUPKA KOJIM SE DOŠLO DO TIH REZULTATA.

<p>1. a) Objasniti ulogu registara \$29 i \$31. (2 poena)</p> <p>b) Odrediti sadržaj registra \$9 nakon izvršenja instrukcije slt \$9,\$11,\$12. Sadržaji registara \$11 i \$12 su isti. (2 poena)</p> <p>c) Sadržaji registara \$18 i \$19 su 8 i 9, respektivno. Da li će nakon izvršenja instrukcije bne \$18,\$19,Lab doći do skoka na labelu Lab? Objasniti. (2 poena)</p>	<p>2. Dat je dio koda u programskom jeziku C:</p> <pre>int m=1; int n=5; while(n<7) { if(n>=3) m=m+n; n+=1; }</pre> <p>Napisati ovaj dio koda u MIPS asemblerskom jeziku pod pretpostavkom da su promjenljivim m i n dodijeljeni registri \$10 i \$11, respektivno. (9 poena)</p>
<p>3. Dat je dio koda u MIPS asemblerskom jeziku:</p> <pre> addi \$8,\$0,6 petlja: addi \$9,\$0,0 slti \$10,\$8,4 bne \$10,\$0,Izlaz add \$9,\$9,\$8 addi \$8,\$8,-1 j petlja Izlaz: addi \$9,\$9,-3</pre> <p>Kako izgleda kod u programskom jeziku C koji obavlja ekvivalentnu radnju datom MIPS kodu? Odrediti sadržaj registara \$8, \$9 i \$10 nakon njegovog izvršenja. (8 poena)</p>	<p>4. Data je procedura u programskom jeziku C koja elemente niza cijelih brojeva koji su manji ili jednaki od koeficijenta k zamjenjuje negativnom vrijednošću tog koeficijenta. Procedura kao rezultat vraća broj izvršenih zamjena. Niz array, broj elemenata niza N i koeficijent k su argumenti procedure. Napisati odgovarajući MIPS kod. (12 poena)</p> <pre>int change(int array[], int N, int k) { int i, br=0; for(i=0;i<N;i++) if(array[i]<=k) { array[i]=-k; br++; } return br; }</pre>

Prisprema za II kolokvijum, 05.11.2014.

II kolokvijum 2013. god.

I grupa!

1) a) \$2 se pojavljuje kao rezultat pravo ugrađivanje ^{irritacije} procedure preratna rezultat svog iritiranja
\$4 - I argument povara procedure

b) sli \$10, \$15, \$16, \$15 ← 4, \$16 ← 5

\$10 = $\begin{cases} 1, & \$15 < \$16 \\ 0, & \$15 \geq \$16 \end{cases}$, \$15 = 4
\$16 = 5

→ \$10 ← 1 nam iritiranje postupke

c) \$10 ← 10
\$11 ← 11

log \$10, \$11, Lab # ne proutrokuje sva na postupke u onim lab, jer je \$10 = 10 ≠ \$11 = 11

2)

```
int a=0;
int b=9;
while (b >= a)
{
```

```
    if (b < 5)
        b = b - 2;
        b--;
    a++;
```

```
}
```

a ↔ \$15, b ↔ \$16

Assembly:

~~addi~~

move \$15, \$0 # a=0

addi \$16, \$0, 9 # b=9

Loop:

slt \$8, \$16, \$15 # \$8=1 a < b / \$8=0, a >= b

bne \$8, \$0, Exit # itni it jete a < b / \$8=1 (b < a)

slti \$8, \$16, 5 # \$8=1 a < 5 / \$8=0, a >= 5

beg \$8, \$0, Hext # itni a < 5 / \$8=0 (b >= 5)

addi \$16, \$16, -2 # b = b - 2

addi \$16, \$16, -1 # b = b - 1

Hext: addi \$15, \$15, 1 # a = a + 1

j Loop

Exit: — —

3

addi \$8, \$0, 1 # \$8 ← 1

addi \$9, \$0, 0 # \$9 ← 0

potinja: stbi \$10, \$8, 4 # \$10 = 1 and je \$8 < 4 (\$10 = 0 and je)

bej \$10, \$0, islet # itadi and je \$10 = 0 (\$8 > 4)

add \$9, \$9, \$8 # \$9 = \$9 + \$8

addi \$8, \$8, 1 # \$8 = \$8 + 1

j potinja

itert: addi \$9, \$9, -3 # \$9 = \$9 - 3

C kod ekvivalenti?

\$8, \$9, \$10 = ?

```

int i = 1;
int j = 0;
while (i < 4)
{
    j = j + i;
    i++;
}
j = j - 3;

```

```

int i = 0;
for (i = 1; i < 4; i++)
    j = j + i;
j = j - 3;

```

Ovo sam nalazio u C kodu, a ne u MIPS kodu. Da li je to OK? Jeste!

\$8 ← i, \$9 ← j

U ovom insajnu je pravi kod:

\$8 = 4, \$9 = 6 - 3 = 3
\$10 = 0

```

(4) int procedura(int array[], int N, int k)
{
    int i, lr=0;
    for(i=0; i<N; i++)
        if(array[i]==k)
            array[i]=-2*k;
    lr++;
}
return lr;
}

```

(array &)

$\$4 \leftrightarrow \text{array}$
 $\$5 \leftrightarrow N$
 $\$6 \leftrightarrow k$
 $\$15 \leftrightarrow i$
 $\$16 \leftrightarrow lr$

← (1)

Loop: $\$9$
 Exit: $\$10$

Next: $\$9$
 Next: $\$10$

```

move $16, $0 # lr=0
move $15, $0 # i=0
Loop: slt $8, $15, $5 # $8 = { 1, i < N
           0, i >= N
        }
        beq $8, $0, Exit # iradi ako je $8=0 (i >= N)
        muli $8, $15, 4 # $8 = 4 * i
        add $8, $8, $4 # $8 ← adr. array[i]
        lw $9, 0($8) # $9 ← array[i]
        bne $9, $6, Next # idi u sledeci ako je array[i] != k
        muli $9, $9, -2 # $9 = -2 * k
        sw $9, 0($8) # array[i] ← -2 * k
        addi $16, $16, 1 # lr = lr + 1
        addi $15, $15, 1 # i = i + 1
        j Loop
Exit: move $2, $16 # $2 ← lr

```

← (2)

Penyelesaian se sederhana: registera \$16, \$15, \$8, \$9
→ 4 registera!

addi \$29, \$29, -16
sw \$16, 0(\$29)
sw \$15, 4(\$29)
sw \$8, 8(\$29)
sw \$9, 12(\$29)

1

lw \$16, 0(\$29)
lw \$15, 4(\$29)
lw \$8, 8(\$29)
lw \$9, 12(\$29)
addi \$29, \$29, 16
jr \$31

2

Prisrue za II kolokvijum, 12.11.2017.

II kolokvijum 2013.

II grupa

1) a) Reg. §3 štiti sa ~~načinu~~ supertivne rezultate da uop se
elobni iritacijem procedre (potpisan)
u Reg. §6 se supita u argument formare procedure

2) b) št $\phi_{10}, \phi_{11}, \phi_{12}$ } $\phi_{10} = ?$
 $\phi_{11} = \phi_{12}$

$$\phi_{10} = \begin{cases} 1, & \phi_{11} < \phi_{12} \\ 0, & \phi_{11} \geq \phi_{12} \end{cases} \Rightarrow \phi_{10} = \phi$$

c) $\phi_8 < 8$

$\phi_9 < 9$

luc ϕ_8, ϕ_9, Lab

← one procedure (injecija) sraz na
karakteru sa dazgajem Lab, paito je
 $\phi_8 = 8 \neq \phi_9 = 9$

2

```

int u=1;
int n=5;
while (u<7)

```

{

```

if (u>=3)

```

```

    u=u+u;

```

```

    u+=1;

```

}

```

u ↔ $10

```

```

n ↔ $11

```

Program:

```

addi $10, $0, 1 # u=1

```

```

addi $11, $0, 5 # u=5

```

Loop:

```

slti $8, $11, 7 # $8 = { 1, u < 7
                        0, u >= 7

```

```

bge $8, $0, Exit # in case u >= 7

```

Case: \$8

```

slti $9, $11, 3 # $9 = { 1, u < 3
                        0, u >= 3

```

```

bne $9, $0, Next # Idhar wacha u-a, aur je u < 3

```

```

add $10, $10, $11 # u = u + u

```

```

Next: addi $11, $11, 1 # u = u + 1

```

j Loop

Exit: _____

③ `addi $8, $0, 6` # $\$8 = 6$
`addi $9, $0, 0` # $\$9 = 0$
`seteq $10, $8, 4` # $\$10 = \begin{cases} 1, & \$8 < 4 \\ 0, & \$8 \geq 4 \end{cases}$
`lwe $10, $0, $10` # ~~indikator je \$10 = 0 (\$8 < 4)~~
`add $9, $9, $8` # $\$9 = \$9 + \$8$
`addi $8, $8, -1` # $\$8 = \$8 - 1$
`j $10, $9, $8` # ~~svaki u naredbu se odnosi na ftepa~~
`lwe $9, $9, -3` # $\$9 = \$9 - 3$

Privedeni C-код?
 $\$8, \$9, \$10 = ?$

$i \leftrightarrow \$8, j \leftrightarrow \9

```

int i = 6;
int j = 0;
while (i >= 4)
{
    j = j + i;
    i = i - 1;
}
j = j - 3;
    
```

ili

```

int j = 0;
for (i = 6; i >= 4; i--)
    j = j + i;
j = j - 3;
    
```

$\$8 = 3$ (izvan se iz ftepa kada je $i < 4$)
 $\$9 = 15 - 3 = 12$
 $\$10 = 1$ (da bi se izvršio program - da bi se izvršio \$10
 navedeno)

```

(4) int change (int array [N], int H, int k)
{
    int i, lr = 0;
    for (i = 0; i < N; i++)
        if (array[i] != k)
            array[i] = -2 * k;
            lr++;
    return lr;
}

```

$\$4 \leftrightarrow \text{array}(\text{start})$
 $\$5 \leftrightarrow N$
 $\$6 \leftrightarrow k$
 $\$15 \leftrightarrow i$
 $\$16 \leftrightarrow \text{lr}$

← (1)

```

Loop:
    move $15, $0      # i = 0
    move $16, $0      # lr = 0
    slt $8, $15, $5   # $8 = { 1, i < N
                        #       0, i ≥ N
    beq $8, $0, Exit  # i not in range i ≥ N
    muli $8, $15, 4    # $8 = 4 * i
    addi $8, $8, $4    # $8 = addr. (array[i])
    lw $9, 0($8)      # $9 = array[i]
    beq $9, $6, Hext  # i is not equal with array[i] ≠ k
    muli $10, $6, -2   # $10 = -2 * k
    sw $10, 0($8)     # array[i] ← -2 * k
    addi $16, $16, 1   # lr = lr + 1
Hext:
    addi $15, $15, 1   # i = i + 1
    j Loop
Exit:
    move $2, $16      # $2 ← lr

```

← (2)

Učienyji se saartji regitara \$15, \$16, \$8, \$9, \$10; kabe ih
dovati se stacenu!

addi \$29, \$29, 20
su \$15, 4 (\$29)
su \$16, 4 (\$29)
su \$8, 8 (\$29)
su \$9, 12 (\$29)
su \$10, 16 (\$29)

17

na noven ih vaditi se stacenu unad u regitru

lu \$15, 4 (\$29)
lu \$16, 4 (\$29)
lu \$8, 8 (\$29)
lu \$9, 12 (\$29)
lu \$10, 16 (\$29)
addi \$29, \$29, 20
jr \$31

2