



UNIVERZITET CRNE GORE
ELEKTROTEHNIČKI FAKULTET PODGORICA
STUDIJSKI PROGRAM: PRIMJENJENO RAČUNARSTVO

Seminarski rad

REALIZACIJA PROJEKTA VI (PROJEKAT)

Profesor:

Dr. Nevena Radović

Student:

Radule Bulatović 23/17

Nemanja Nedić 35/17

Veselin Lazarević 94/17

Podgorica, 2020

SADRŽAJ:

1.	UVOD	1
2.	PLAN RADA	2
3.	KOD	4
4.	REZULTAT	6

1. UVOD

U ovom radu ćemo napisati i objasniti glavni program čiji je ulazni podatak nulom terminisani string dužine do 30 karaktera. Program treba da poziva proceduru *podstring* koja kreira podstring od svakog karaktera početnog stringa koji se nalazi na parnim pozicijama (paran indeks). Glavni program će trebati da pita korisnika da unese proizvoljni string, a potom (za zadati string) da vrati poruke sljedeće sadržine: "*Podstring od karaktera sa parnih pozicija je: X*" i "*Dužina unijetog stringa je: Y*", gdje će X i Y prikazivati sadržaj dobijenog podstringa, kao i dužinu inicijalnog stringa, respektivno). Program ćemo testirati na primjeru stringa: „Kolokvijum“ i prikazati rezultate.

Radi jednostavnosti podrazumijevaćemo rad sa engleskim alfabetom. Takođe ćemo voditi računa da napisani program radi za sve stringove dužine do 30 karaktera (dužine mogu biti parni ili neparni brojevi).

2. PLAN RADA

U prvom koraku u okviru *.data* direktive deklariramo stringove (**korak1**, **korak2** i **korak3**) kao i polje za upis stringa (**input**).

```
.data

input: .space 30
korak1: .ascii "Unesite proizvoljan string\n"
korak2: .ascii "\nPodstring od karaktera sa parnih
pozicija je: "
korak3: .ascii "\nNjegova duzina je: "
```

U main funkciji započinjemo sa štampanjem poruke koja se nalazi deklarirana kao **korak1** (Unesite proizvoljan string), zatim korisnik unosi string koji se premješta u odgovarajući registar (uneseni string ne može biti duži od 30 karaktera).

```
# PRINT: Unesite proizvoljan string:
    li $v0, 4
    la $a0, korak1
    syscall

# Citanje ulaznog podatka i premještanje u $t0:
    li $v0, 8
    la $a0, input
    la $a1, 30
    move $t0, $a0
    syscall
```

Nakon toga se na ekranu štampa poruka deklarirana kao **korak2** (Podstring od karaktera sa parnih pozicija je:), nakon čega se poziva procedura **podstring** koja će stampati podstring.

```
# PRINT: Podstring od karaktera sa parnih pozicija je:
    li $v0, 4
    la $a0, korak2
    syscall

# Funkcija stampa podstring, broj članova je smješten u $t7
    jal podstring
```

U okviru ove labele ćemo imati dva brojača, jedan za elemente niza, dok će drugi biti za elemente podniza (setovani na vrijednost 0). Takođe ćemo u neki registar (**\$t6**) smjestiti vrijednost 2 koji će nam služiti za ispitivanje parnosti niza.

podstring:

```
li $t5, 0 #brojač elemenata
niza li $t6, 2 #ispitivanje
parnosti
li $t7, 0 #brojac elemenata podniza
```

Zatim idemo na labelu **counter** u okviru koje imamo registar (**\$t1**) u kojem smiještamo članove niza. Zatim ispitujemo da li je član niza (**\$t1**) terminacioni karakter (0), ako jeste skočimo na labelu **L1**, a ako nije inkrementiramo brojač, prelazimo na adresu sledećeg člana pa povratak na labelu **counter** dok imamo članova niza. U labeli **L1** se vraćemo na početak niza.

counter:

```
lb $t1, 0($t0)      # smjestanje clana niza u $t1
beqz $t1, L1        # u koliko se u $t1 nalazi terminacioni karakter(0) skoci
                    # na adresu L1
addi $t5, $t5, 1    # inkrementiranje brojaca
addi $t0, $t0, 1    # prelazak na adresu sledeceg clana niza
j counter           # povratak na labelu counter
```

L1:

```
addi $t0, $t0, -1   # $t0 ima vrijednost za jednu vecu od adrese
                    # poslednjeg clana niza pa je smanjujemo za 1,
                    # $t0 sada pokazuje na poslednji clan niza
sub $t1, $t0, $t5    # (poslednji_Clan) - (broj_Clanova) = (pocetak_niza)
```

Po završetku ovog dijela koda stanje u registrima je sljedeće:

\$t5 – broj karatera u stringu

\$t1 – adresa prvog karaktera stringa

\$t0 – adresa kraja stringa

Realizujemo labelu **printer** u kojoj ćemo iterirati kroz string registrom **\$t1** pa je potrebno na početku ispitati da li on još uvijek pokazuje na string. Ako bi vrijednost **\$t1** se izjednačila sa vrijednošću terminacionog karaktera procedura bi završila sa radom skokom na labelu **exit**.

printer:

```
slt $t2, $t1, $t0    # ako smo dosli do poslednjeg clana niza $t2 = 0
beqz $t2, exit        # ako $t2 = 0 skacemo na labelu exit
```

Parnost indeksa određujemo dijeljenjem indeksa niza (**\$t1**) sa brojem 2 (**\$t6**) prilikom čega će ostatak pri dijeljenju biti sačuvan u registru **\$hi**. Komanda **mfhi** premješta vrijednost **\$hi** registra u željeni registar. Ispitivanjem da li je taj ostatak jednak nuli znat ćemo da li se radi o parnom ili neparnom broju.

```
div $t1, $t6          # dijeljenje indeksa clana niza sa 2
mfhi $t2              # HI cuva ostatak pri dijeljenju, smjestamo ga u $t2
```

Neparan broj dovodi do skoka na labelu **increment** čime se zaobilazi štampanje karaktera stringa i inkrementiranje brojača elemenata podstringa (**\$t7**).

bne \$t2, \$zero, increment # l % 2 == 0--> broj je paran

Po završetku rada procedura će prikazati karaktere na parnim pozicijama a njihov broj će se proslijediti glavnom programu preko registra **\$t7**.

lb \$a0, 0(\$t1)	#	prosledjivanje clana niza proceduri
li \$v0, 11	#	pozivanje procedure za štampanje karaktera
syscall		
addi \$t7, \$t7, 1	#	inkrementiranje brojača podniza
increment:		
addi \$t1, \$t1, 1	#	inkrementiranjem \$t1 prelazimo na sledeci clan
niza		
j printer	#	Vracanje na labelu printer
exit:		
jr \$ra		

U nastavku rada glavni program će prikazati poruku deklarisanu kao **korak3** (Njegova dužina je) a nakon nje i vrijednost registra **\$t7** čime se rad programa završava.

#	PRINT: Njegova dužina je
	la \$a0, korak3
	li \$v0, 4
	syscall
#	PRINT: \$t7
	li \$v0, 1
	move \$a0, \$t7
	syscall
	li \$v0, 10
	syscall
	.end main

3. KOD

.data

input: .space 30

korak1: .asciiz "Unesite proizvoljan string\n"

korak2: .asciiz "\nPodstring od karaktera sa parnih pozicija je: "

korak3: .asciiz "\nNjegova duzina je: "

.text

.globl main

main:

#-----

```
#    PRINT: Unesite proizvoljan string:
    li $v0, 4
    la $a0, korak1
    syscall
```

```
#    Citanje ulaznog podatka i premjestanje u $t0
    li $v0, 8
    la $a0, input
    la $a1, 30
    move $t0, $a0
    syscall
```

```
#    PRINT: Podstring od karaktera sa parnih pozicija je:
    li $v0, 4
    la $a0, korak2
    syscall
```

```
#    Funkcija stampa podstring, broj clanova je smjesten u $t7
    jal podstring
```

```
#    PRINT: Njegova duzina je
    la $a0, korak3
    li $v0, 4
    syscall
```

```
#    PRINT: $t7
    li $v0, 1
    move $a0, $t7
    syscall
```

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

.end main

#-----

podstring:

```
    li $t5, 0 #brojac elemenata niza
    li $t6, 2 #ispitivanje parnosti
    li $t7, 0 #brojac elemenata podniza
```

Odredjivanje broja clanova niza

counter:

lb \$t1, 0(\$t0)	#	smjestanje clana niza u \$t1
beqz \$t1, L1	#	u koliko se u \$t1 nalazi terminacioni karakter skoci na adresu L1
addi \$t5, \$t5, 1	#	inkrementiranje brojaca
addi \$t0, \$t0, 1	#	prelazak na adresu sledeceg clana niza
j counter	#	povratak na labelu counter

L1:

addi \$t0, \$t0, -1	#	\$t0 ima vrijednost za jednu vecu od adrese poslednjeg clana niza pa je smanjujemo za 1,
	#	\$t0 sada pokazuje na poslednji clan niza
sub \$t1, \$t0, \$t5	#	(poslednji_Clan) - (broj_Clanova) = (pocetak_niza)
	#	(X[0] = X[I] - I)

printer:

slt \$t2, \$t1, \$t0	#	ako smo dosli do poslednjeg clana niza \$t2 = 0
beqz \$t2, exit	#	rezultat ove dvije linije koda je identican ako \$t2 = 0 skacemo na labelu exit while(\$t1<\$t0)
div \$t1, \$t6	#	dijeljenje indeksa clana niza sa 2
mfhi \$t2	#	HI cuva ostatak pri dijeljenju, smjestamo ga u \$t2
bne \$t2, \$zero, increment	#	I % 2 == 0--> broj je paran
	#	I % 2 == 1--> broj je neparan
	#	skokom na labelu increment zaobilazimo stampanje
	#	i brojanje clana niza i prelazimo na sledeci clan
lb \$a0, 0(\$t1)	#	prosledjivanje clana niza proceduri
li \$v0, 11	#	pozivanje procedure za stampanje karaktera
syscall		
addi \$t7, \$t7, 1	#	inkrementiranje brojaca podniza

increment:

addi \$t1, \$t1, 1	#	inkrementiranjem \$t1 prelazimo na sledeci clan niza
j printer	#	Vracanje na labelu printer

exit:

jr \$ra

4. REZULTAT

Za testiranje projekta koristimo riječ “Kolokvijum”, a rezultati su prikazani u nastavku. Za realizaciju je koriscen MIPS simulator QtSpim.



```
Console
Unesite proizvoljan string
Kolokvijum

Podstring od karaktera sa parnih pozicija je: Klkiu
Njegova duzina je: 5
```

Slika 1 – Rezultat iz QtSpim simulatora