

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Ime:	<i>Luka</i>	Broj indeksa:	<i>18587</i>
Prezime:	<i>Veličković</i>		
LV po redu:	<i>IV</i>	Termin:	<i>3</i>
Datum i vreme početka izrade	<i>22.12.2022 15:52</i>		

Zadatak:

U segmentu podataka su dve skalarne 16b promenljive i niz 32b elemenata, čiji broj elemenata nije nigde zapamćen, već je poznato da se niz završava nulom. Svi podaci se smatraju označenim brojevima.

Treba napisati program na asemblerskom jeziku koji sve elemente jednake drugom elementu uvećava za vrednost druge skalarne promenljive.

- a) Formirati primere početnih vrednosti koji demonstriraju sve osobine zadatog problema, posebno u pogledu graničnih slučajeva, parnosti broja elemenata niza međuprenosa, i za eventualne slučajeve kada nije moguće doći do tačnog rezultata.

Za svaki primer početnih vrednosti:

- navesti po čemu je karakterističan - koju osobinu demonstrira
- ustanoviti vrednosti rezultata, ukoliko je to moguće.

- b) **Zadatak 1:** Napisati program koji rešava dati problem. Za pristup elementima niza koristiti indeksno adresiranje. Primere početnih vrednosti iz a) uneti u segmentu podataka kao kompletne direktive za alokaciju memorije, pri čemu su svi kompleti, osim jednog, podešeni kao komentar.

Zadatak 2: Napisati program koji rešava dati problem, i ima identičan segment podataka kao Zadatak 1. Za pristup elementima niza u memoriji koristiti isključivo instrukcije za rad s nizovima.

U dati šablon izveštaja, deo b) uneti izveštaj za oba zadatka, u redosledu u kome su zadaci dati ovde.

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Rešenje:

a) Primeri vrednosti, šeme operacija

- 1) Početne vrednosti u kojima se isti element pojavljuje više puta, uključujući i drugu poziciju u nizu

skal1 = 15h

skal2 = 255h

niz = [44h, 55h, 12h, 55h, 55h, -18h, 55h, 67h, 0h]

Očekivana vrednost niza nakon transformacije

niz = [44h, 2AAh, 12h, 2AAh, 2AAh, -18h, 2AAh, 67h, 0h]

- 2) Početne vrednosti pri čemu je nula jedini element niza

skal1 = 15h

skal2 = 255h

niz = [0h]

Očekivana vrednost niza nakon transformacije

niz = [0h]

- 3) Početne vrednosti pri čemu je element na drugoj poziciji jedinstven u nizu, a druga skalarna promenljiva negativna

skal1 = 15h

skal2 = -255h

niz = [44h, 55h, 12h, 42h, 94h, -18h, 6h, 67h, 0h]

Očekivana vrednost niza nakon transformacije

niz = [44h, -200h, 12h, 42h, 94h, -18h, 6h, 67h, 0h]

Vrednost -200h se u potpunom komplementu predstavlja kao 0FFFFFFE00h

- 4) Početne vrednosti za koje se ne dobija tačno rešenje

Vrednosti za koje nije moguće dobiti tačno rešenje predstavljaju one slučajeve kada se pri sabiranju druge skalarne vrednosti i drugog elementa niza bit najveće težine menja, što dovodi do promene znaka, i kada se ona ne treba da se javi. Kako bi se ovo sprečilo potrebno je proširiti vrednosti na 64b, međutim kako je niz 32b ovo ostaje kao situacija u kojoj nije moguće dobiti tačno rešenje.

skal1 = 15h

skal2 = EBCDh = (-5171)₁₀

niz = [432h, FFFF4582h, 0] // FFFF4582h = (-47742)₁₀

Očekivana vrednost niza nakon transformacije

niz = [432h, -FFFF314h, 0] // -FFFF314h = (-52913)₁₀

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

b) Kod rešenja, izveštaj o testiranju

- 1) Korišćenje indeksnog adresiranja za pristup elementima niza

```
.386
.model flat, c
.data
    skal1 dw 15h

    skal2 dw 255h
    niz dd 44h, 55h, 12h, 55h, 55h, -18h, 55h, 67h, 0h

; niz dd 0h

; skal2 dw -255
; niz dd 44h, 55h, 12h, 42h, 94h, 18h, 6h, 67h, 0h

; skal2 dw 0EBCDh
; niz dd 432h, 0FFFF4582h, 0
.code
main proc
    mov eax, 0
    mov ax, skal2
    cwd
    shl edx, 16
    or eax, edx

    mov edx, niz[type niz]
    mov ebx, edx
    add ebx, eax
    mov esi, 0

    uslov:
    cmp niz[type niz * esi], 0
    je kraj
    cmp niz[type niz * esi], edx
    jne inkrement
    mov niz[type niz * esi], ebx
    inkrement:
    inc esi
    jmp uslov
    kraj:
    nop
ret
main endp
end main
```

Uvećavanje svih elemenata jednakih drugom elementu za vrednost druge skalarne promenljive ekvivalentno je zameni tih elemenata vrednošću koja se dobija sabiranjem druge vrednosti i druge skalarne promenljive. Stoga se druga skalarna promenljiva (16b) prvo smešta u AX, proširuje u DX:AX, a nakon toga vrednost iz DX smešta u viši deo registra EAX. U registar EDX se potom smešta drugi element niza, a nakon toga se vrši sabiranje (pri čemu se ignoriše eventualni izlazni prenos). Sada se u EBX nalazi element koji će biti umetan na odgovarajuća mesta, a u EDX element sa kojim će se upoređivati.

Osatak problema se rešava tako što se za svaki element proveri da li je terminalni, i da li je jednak drugom element niza. Ukoliko je jednak drugom element niza na njegovo mesto se umeće odgovarajuća vrednost, a ukoliko nije ne dešava se ništa.

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Za primere početnih vrednosti pod 1, 2 i 3 dobija se očekivani rezultat, dok se, kao što je navedeno za primer 4 ne dobija. Slike su prikazane redom od prvog do četvrtog primera.

[illegible]

- 2) Korišćenje instrukcija za rad sa nizovima (.data segment isti kao u prvom delu)

```
.code
```

```
cmp niz, 0
je kraj
```

```
mov eax, 0
mov ax, skal2
cwd
shl edx, 16
or eax, edx
mov edx, eax
```

```
lea esi, niz + type niz
lodsd
add edx, eax
mov ecx, 0
```

```
lea edi, niz
cld
uslov:
```

```
xchg eax, ecx
scasd
je kraj
xchg eax, ecx
sub edi, type niz
```

```
scasd
jne skok
sub edi, type niz
xchg eax, edx
stosd
xchg eax, edx
```

Na početku proveravamo da li je prvi element niza 0, što predstavlja specijalni slučaj. Nakon toga se druga skalarna vrednost pribavlja u EDX. Drugi element niza se pribavlja u akumulator EAX, i ta vrednost se dodaje na vrednost registra EDX. Sada će svi elementi niza koji su jednaki akumulatorom biti zamenjeni vrednošću iz EDX.

Kao što je rečeno, naredbom `scasd` se vrednost akumulatora (u kome se nalazi originalna vrednost drugog elementa niza) upoređuje sa tekućim elementom odredišnog niza. Ukoliko su jednaki, kako bi se na to mesto ubacila vrednost iz EDI, potrebno je vratiti adresu iz EDI za jedan element unazad, što omogućava zamenu uz pomoć naredbe `stosd`.

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

```
skok:
jmp uslov
kraj:
nop
```

Izvršavanje ovog programa dovodi do istih ishoda kao i izvršavanje prethodnog programa. Slike su prikazane redom od prvog do četvrtog primera početnih vrednosti.

[illegible][illegible][illegible][illegible]

Samoevaluacija

Na skali 0-5 (0 - „nikako“, „nimalo“; 5 - „potpuno“), u kom stepenu smatrate da ste:

1) <i>bili savladali gradivo PRE početka rada na vežbi</i>	<i>4</i>
2) <i>razumeli zadatak</i>	<i>5</i>
3) <i>ispunili zahteve zadatka a)</i>	<i>4</i>
4) <i>ispunili zahteve zadatka b)</i>	<i>4</i>
5) <i>istestirali i opisali funkcionisanje svog rešenja</i>	<i>5</i>
6) <i>razumeli ponašanje svog rešenja i pojedinih instrukcija i mehanizama</i>	<i>5</i>
7) <i>imali dovoljno vremena za vežbu</i>	<i>5</i>
8) <i>unapredili svoje znanje u toku vežbe</i>	<i>4</i>

<Luka Veličković, 24.12.2022 11:22 >