

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

Ime:	<i>Aleksandar Gospavić</i>	Broj indeksa:	<i>19095</i>
Prezime:			
LV po redu:	<i>IV</i>	Termin:	<i>3</i>
Datum i vreme početka izrade	<i>06.12.2023. 21:41</i>		

Zadatak:

U segmentu podataka su dve skalarne 8b promenljive i niz 32b elemenata, čiji broj elemenata je zapamćen u prvoj skalarnoj promenljivoj. Svi podaci se smatraju označenim brojevima.

Treba napisati program na asemblerskom jeziku koji sve elemente jednake prvom 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:

$$n = 8$$

$$x = 25h$$

$$niz = [55h, 44h, 1Bh, 55h, 55h, -18h, 55h, F7h]$$

Očekivana vrednost niza nakon transformacije:

$$niz = [7Ah, 44h, 1Bh, 7Ah, 7Ah, -18h, 7Ah, F7h]$$

- 2) Početne vrednosti pri čemu je druga skalarna promenljiva negativna:

$$n = 8$$

$$x = -12h (=) EEh \text{ u PK } (=) -18_{10}$$

$$niz = [55h, 10000044h, abc043h, 55h, 55h, -18h, 55h, F7h]$$

Očekivana vrednost niza nakon transformacije:

$$niz = [43h, 10000044h, abc043h, 43h, 43h, -18h, 43h, F7h]$$

- 3) 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 prvog elementa niza bit najveće težine menja, što dovodi do promene znaka. 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.

$$n = 4$$

$$x = 7Fh$$

$$niz = [7FFFFFFAh, 640FABh, 7FFFFFFAh, 65Dh]$$

Očekivana vrednost niza nakon transformacije:

$$niz = [80000079h, 640FABh, 80000079h, 65Dh]$$

80000079h u PK je $(-2147483527)_{10}$

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

b) Kod rešenja, izveštaj o testiranju

Zadatak 1: Korišćenje indeksnog adresiranja za pristup elementima niza

```
.386
.model flat, c
.data
    n db 8
    x db 25h
    niz dd 55h, 44h, 1Bh, 55h, 55h, -18h, 55h, 0F7h

    ;n db 8
    ;x db -12h
    ;niz dd 55h, 10000044h, 0abc043h, 55h, 55h, -18h, 55h, 0F7h

    ;n db 4
    ;x db 7Fh
    ;niz dd 7FFFFFFAh, 640FABh, 7FFFFFFAh,

65Dh
.code
main proc
    mov eax, niz
    mov edx, eax

    movsx ebx, x
    add ebx, eax

    mov esi, 0
    movzx ecx, n
    dec esi
gore:
    inc esi
    mov eax, niz[type niz * esi]
    cmp eax, edx
    jne preskoci
    mov niz[type niz * esi], ebx
    preskoci:
    loop gore

    ret
main endp
end main
```

Uvećavanje svih elemenata jedankih prvom elementu za vrednost druge skalarne promenljive ekvivalentno je zameni tih elemenata vrednošću koja se dobija sabiranjem vrednosti prvog elementa i druge skalarne promenljive.

Stoga se rezultat sabiranja prvog elementa i druge skalarne promenljive smešta u registar EBX, dok se prvi element pamti u registar EAX. Registar ESI služi kao brojač, u svakom prolazu se poveća za 1. Prolaskom kroz sve elementa niza, proverava se da li je trenutni element jednak prvom (zapamćen u EAX) i ako jeste u memoriju na adresi gde se nalazi trenutni element se upisuje zapamćena vrednost zbira prvog elementa i druge skalarne promenljive (zapamćena u EBX).

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

Memory 1	
Address:	&niz
0x009F4002	0000007a
0x009F4006	00000044
0x009F400A	0000001b
0x009F400E	0000007a
0x009F4012	0000007a
0x009F4016	ffffffe8
0x009F401A	0000007a
0x009F401E	000000f7

Memory 1	
Address:	&niz
0x00D84002	00000043
0x00D84006	10000044
0x00D8400A	00abc043
0x00D8400E	00000043
0x00D84012	00000043
0x00D84016	ffffffe8
0x00D8401A	00000043
0x00D8401E	000000f7

Memory 1	
Address:	&niz
0x007C4002	80000079
0x007C4006	00640fab
0x007C400A	80000079
0x007C400E	00000065d

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

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

```
.code
main proc

    movzx ecx, n
    mov eax, niz           ; u EAX prvi element sa kojim poredimo
    movsx edx, x           ; u EDX povecan prvi element koji setujemo
    add edx, eax

    lea edi, niz
petlja:
    scasd                 ; cmp eax, [edi] i edi += 4
    jne preskoci
    xchg eax, edx
    sub edi, type niz     ; vrati u libelu edi -= 4
    stosd                 ; mov [edi], eax
    xchg eax, edx
    preskoci:
    loop petlja

    nop
    ret
main endp
end main
```

Zbog instrukcija sa rad sa nizovima moramo pre petlje da postavimo određene registre na početno stanje. Tako u EAX smeštamo prvi element niza sa kojim se svi ostali elementi porede, a u EDX rezultat zbira prvog elementa i druge skalarne promenljive. U registar EDI se smešta adresa početna niza, a u registar ECX broj elemenata niza. Instrukcija SCASD će uporediti sadržaj akumulatora (prvi element niza) i podatka sa adrese koja je smeštena u EDI. Ako nisu jednaki nastavi se dalje sa pretragom. Primetimo da nije potrebno da inkrementiramo EDI jer će to instrukcija SCASD uraditi sama. U suprotnom, ako jesu jednaki, moramo eksplicitno vratiti EDI na prethodnu adresu. Zatim u EAX smestimo element prvog niza uvećan za drugu skalarnu promenljivu (zapamćena u EDX). Ovo je potrebno jer i instrukcija STOSD radi sa akumulatorom (EAX). Ovim je novi podatak smešten u memoriju i EDI povećan pa je ovime spreman za sledeći element niza. Zatim vratimo stanje akumulatora još jednom instrunkcijom XCHG. Ovaj postupak se ponavlja n puta, jer nam je u prvoj skalarnoj promenljivoj broj elemenata niza. Izvršavanje ovog programa dovodi do istih ishoda kao i izvršavanje prethodnog programa. Slike su prikazane redom od prvog do trećeg primera početnih vrednosti.

Memory 1		
Address: &niz		
0x009F4002		0000007a
0x009F4006		00000044
0x009F400A		0000001b
0x009F400E		0000007a
0x009F4012		0000007a
0x009F4016		ffffffe8
0x009F401A		0000007a
0x009F401E		000000f7

Memory 1		
Address: &niz		
0x00D84002		00000043
0x00D84006		10000044
0x00D8400A		00abc043
0x00D8400E		00000043
0x00D84012		00000043
0x00D84016		ffffffe8
0x00D8401A		00000043
0x00D8401E		000000f7

Memory 1		
Address: &niz		
0x007C4002		80000079
0x007C4006		00640fab
0x007C400A		80000079
0x007C400E		00000065d

Arhitektura i organizacija računara 1 – laboratorijska vežba - Izveštaj

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

Aleksandar Gospavić, 07.12.2023. 23:04