

Primer 1: Računanje zbira članova niza koji se unosi sa tastature

```
; Zadatak: Ucitati niz od 10 clanova i odrediti njihovu sumu
;
; kompajliranje i linkovanje
; nasm -f elf niz.asm
; gcc -m32 -o niz niz.o ../asm_io.o ../driver.c
;
#include <stdio.h>
#define DUZINA_NIZA 10
;
int main()
{
    unsigned int i, suma;
    unsigned int niz1[DUZINA_NIZA];
    ;
    for (i=0; i<DUZINA_NIZA; i++)
    {
        printf("Unesi clan niza: ");
        scanf("%u", &niz1[i]);
    }
    ;
    suma=0;
    for (i=0; i<DUZINA_NIZA; i++)
        suma+= niz1[i];
    ;
    printf("Suma clanova niza je %d\n", suma);
    ;
    return 0;
;}

#include "../asm_io.inc"
#define DUZINA_NIZA 10

segment .data
    poruka1      db      "Suma clanova niza je ", 0
    poruka2      db      "Clan niza: ", 0

segment .bss
    niz1         resd    10

segment .text
    global  asm_main
asm_main:
    enter 0,0          ; rutina za inicijalizaciju
    pusha

    ; Ovde pocinje koristan kod
    mov ecx, 0         ; u ecx registru je brojac i
    mov esi, 0         ; indeksni registar esi postavi na nulu

petlja_unos:
    cmp ecx, DUZINA_NIZA ; uporedi ecx na DUZINA_NIZA
    je kraj_petlja_unos  ; ako je jednako, izađi iz petlje

    mov eax, poruka2     ; stampaj poruku i učitaj clan niza
    call print_string
    call read_int        ; u eax se nalazi učitana vrednost

    mov [niz1+esi], eax   ; postavi clan na odgovarajuće mesto u nizu
    add esi, 4            ; povecaj esi za 4 (4-double word)
    inc ecx               ; povecaj brojac za 1
    jmp petlja_unos

kraj_petlja_unos:

; Niz je ucitan, u sledecim linijama se racuna suma njegovih clanova
    mov edx, 0           ; u edx registru se cuva suma
    mov ecx, DUZINA_NIZA ; brojac postavljen na DUZINA_NIZA jer ide unazad
    mov esi, 0           ; index registar postavljen na nulu

petlja:
    add edx, [niz1+esi]   ; edx += niz1[i]
    add esi, 4            ; pomeri se za 4 bajta (jedan clan niza je duzine 4 bajta)
    loop petlja           ; kraj petlje
```

```
mov    eax, poruka1      ; stampanje rezultata
call   print_string

mov    eax, edx          ; rezultat se nalazi u edx, kopiraj u eax
call   print_int
call   print_nl

popa
mov    eax, 0            ; vrati se nazad u C
leave
ret
```