DOKUMENTACIJA

Programski prevodioci - predmetni zadatak

**Osnovni podaci**

|  |  |  |
| --- | --- | --- |
| Broj indeksa | Ime i prezime | Tema |
| SW1/2019 | Jovan Jovančević | Strukture podataka niz i stek |

**Korišćeni alati**

|  |  |
| --- | --- |
| Naziv | Verzija |
| GCC compiler, Flex, Bison, Hipsim |  |

## Evidencija implementiranog dela

## *Ovde navedite svaki pojedinačan zadatak koji ste radili u projektu. Navesti da li je odrađena sintaksa, semantika i generisanje koda za taj deo (ukoliko to ima smisla za taj zadatak).*

Prenos niza po referenci – sintaksa i semantika

Niz kao povratna vrednost funkcija – sintaksa i semantika

Inicijalizacija niza pri deklaraciji – sintaksa, semantika i generisanje koda

Indeksiranje nizova – sintaksa, semantika i generisanje koda

Dodela vrednosti nizu na određenom indeksu – sintaksa, semantika i generisanje koda

*Push* operacija za stek - sintaksa, semantika i generisanje koda

*Pop* operacija za stek – sintaksa, semantika i generisanje koda

*Top* operacija za stek – sintaksa, semantika i generisanje koda

## Detalji implementacije

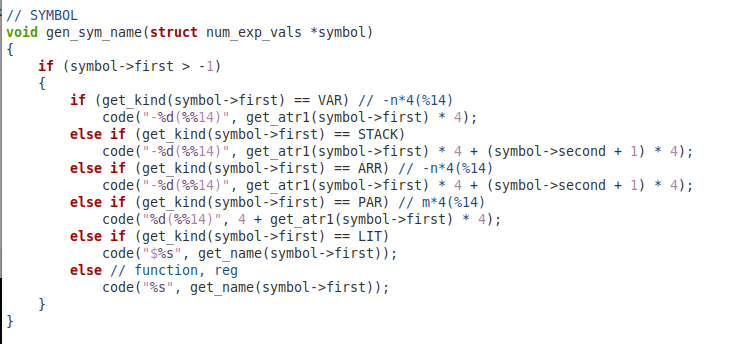
*Detaljnije objasniti na koji način su implementirani zadaci. Delovi koda -može, ukoliko ima smisla. Slike - može (prikaz izgenerisanog stabla, izgled izmenjene tabele simbola,... ). Obavezno numerisati tabele, slike, grafike i referencirati ih pravilno u dokumentaciji.*

*Navesti koji test fajlovi su relevantni.*

Svi relevantni testovi nalaze se u korenskom direktorijumu projekta u folderima *stek* i *niz*.

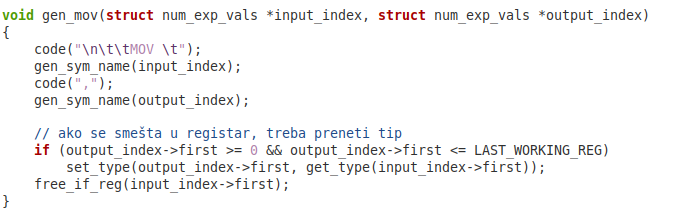
Osnovna ideja za ove strukture podataka bila je sledeća:

* Ostavljanje mesta na steku (gde su elementi smeštani kada se radi generisanje koda)
* U tabeli simbola se čuva samo referenca (head) na niz/stek, var\_num se povećava za onoliko koliko elemenata ima u određenoj strukturi podataka plus jedan. Da posle pri generisanju koda ne bi došlo do “gaženja” nekih informacija.
* Vrednost koju nosi *exp* ili *num\_exp* više nije samo jedan integer, već struktura koja sadrži dva integer-a. Ovo proširenje bilo je potrebno zbog offset-a koje uvode nizovi i stek (pokazivač na vrh steka).
* U *gen\_sym\_name* funkciji dodata su još dva grananja za niz i za stek. Gde se generisanje „imena“ radi slično kao i za obične varijable s tim da se doda još i offset pomnožen sa 4 koliko zauzima svaki element. Na [slici 1](#slika1) je prikazan kod funkcije.



Slika 1

* Nakon promene ove funkcije nije bilo potrebno praviti velike izmene u *gen\_mov* funkciji jer većinu posla obavlja prethodno navedena funkcija kao i sponuta struktura koja je nazvana *num\_exp\_vals*. Na [slici 2](#slika2) je prikazan kod funkcije.



Slika 2

**Niz**

Primeri korišćenja:

* Inicijalizacija pri deklaraciji – *int niz = {1, 2 , 3};*
* Deklaracija niza određene dužine – *int niz[10];*
* Indeksiranje niza (koristimo niz kao izraz) – *var\_a = niz[5];*
* Indeksiranje niza (iskaz) – *niz[1] = 11; niz[1] = a; niz[1] = niz2[1];*
* Element niza kao povratna vrednost – *return niz[1];*
* Niz kao povratna vrednost
  + Potpis funkcije: *int\* func() {...}*
  + Return naredba: *return niz;*

**Stek**

Primeri korišćenja:

* Deklaracija steka određene dužine: *stack::int stek[5];*
* Push operacija: *stek.push(10); stek.push(a); stek.push(niz[1]);*
* Pop operacija*: stek.pop();*
* Top operacija*: stek.top();*
* Vrh steka kao povratna vrednost: *return* *stek.pop();*

## Ideje za nastavak

Ideja za proširivanje projekta bilo bi generisanje koda za prenos niza po referenci.

Kada bi se radilo o diplomskom radu, pokušao bih da implementiram agregatne funkcije za niz kao što su *Max, Min, Sum, Avg*...

Takođe dodatan izazov bi bio da se omogući dinamičko proširivanje nizova.

## Literatura

<https://stackoverflow.com>

<https://www.programiz.com>