**DOKUMENTACIJA**

Programski prevodioci - predmetni zadatak

**Osnovni podaci**

|  |  |  |
| --- | --- | --- |
| Broj indeksa | Ime i prezime | Tema |
| SW 40/2019 | Zoran Bukorac | Proširenje miniC-a sa Heap strukturom podataka |

**Korišćeni alati**

|  |  |
| --- | --- |
| Naziv | Verzija |
| Bison, Flex, Hipsim | Verzija sa vežbi |

## Evidencija implementiranog dela

## **Implementacija Heap-a** – leksika, sintaksa, semantika, generisanje koda

## Detalji implementacije

* **Implementacija Heap-a** – U leksici su dodati simboli ’heap’, ’create\_heap’, ’.’, ’size’, ’isEmpty’, ’push’, ’root’, ’pop’. U sintaksi deklaracija heap-a je definisana na slici 1.  
  A screenshot of a computer

  Description automatically generated with medium confidence

Slika 1: Sintaksa deklaracije heap-a

Za tip Heap-a se postavlja int jer je implementiran heap koji će samo da podržava int vrednosti. Zauzimanje prostora za lokalne promenljive je definisan na slici 2.

A screenshot of a computer

Description automatically generated with medium confidence

Slika 2: Zauzimanje prostora za lokalne promenljive

Veličina heap-a je ograničena na 50 elemenata. Implementirano je da pri dodavanju(*push*) ili uklanjanju(*pop*) heap se reorganizuje da bi svaki element bio veće vrednosti od svog roditelja jer je heap implementiran kao binaran *min-heap*.

Push se može videti na slici 3.

Text

Description automatically generated

Slika 3: Push funkcija

Implementirane su još funkcije root() koja vraća element u korenu, size() koja vraća broj elemenata u heap-u i isEmpty() koja vraća 1 ako je prazan heap, a 0 ako nije. U enum *kinds* je dodata vrednost HEAP = 0x40. Primer iz tabele simbola može da se vidi na slici 4.

A picture containing table

Description automatically generated

Slika : Tabela simbola

* **Testiranje** – Postoje 4 test-ok primera, 6 test-semerr primera i 2 test-synerr primera. U test-ok primerima se deklariše više heap-ova i ispitane su sve heap funkcije i u izrazima i iskazima. Na slici 5 se nalazi jedan test-ok primer. Na slici 6 se nalazi asembler kod primera sa slike 5.

Text

Description automatically generated

Slika : Test Ok primer

Graphical user interface

Description automatically generated with low confidence

Slika : Generisan asembler kod slike 5

U test-semerr1 se nalazi semantička greška gde se heap varijabla koristi u izrazu. Test-semerr2 sadrži semantičku grešku gde se koristi pop na prazan heap. U test-semerr3 se koristi root na prazan heap. Kod test-semerr4 dolazi do semantičke greške gde se unsigned varijabli dodeljuje vrednost iz heap-a. Test-semerr5 proizvodi semantičku grešku jer se u heap dodaje unsigned vrednost. Dok test-semerr6 sadrži pogrešnu deklaraciju heap-a. Test-synerr1 sadrži grešku gde se varijabla dodaje u heap. U test-synerr2 se nalaze pozivi funkcija bez obaveznih zagrada.

## Ideje za nastavak

Za sledeći rad bi se mogla implementirati replace funkcija kako bi se eliminisala potreba za push i pop pozivom i tako bi se reorganizacija heap radila samo jednom umesto dva puta. Takođe nije loše implementirati metodu *heapify* koja bi postojeći niz pretvorila u novi heap.

## Literatura

1. <https://cseweb.ucsd.edu//~kube/cls/100/Lectures/lec3/lec3-33.html> - Korišćen za ideju implementacije heap-a u obliku niza i dobijanje indeksa za levo i desno dete i roditelja elementa čvora.