|  |  |  |
| --- | --- | --- |
|  | **УНИВЕРЗИТЕТ У НОВОМ САДУ**  **ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА** |  |

|  |
| --- |
| **UNIVERZITET U NOVOM SADU**  **FAKULTET TEHNIČKIH NAUKA**  **NOVI SAD**  **Departman za računarstvo i automatiku**  **Smjer računarstvo i automatika**  **ISPITNI RAD**  **Kandidat: Ognjen Stojisavljević**  **Broj indeksa: RA155/2019**  **Predmet: Sistemska programska podrška u realnom vremenu I**  **Tema rada: MAVN prevodilac** |
| **Novi Sad, jun 2021.** |

**Sadr** **žaj**

**1.Uvod**

**1.1 Mavn prevodilac**

**1.2 Zadatak**

**2.Analiza problema**

**3.Leksička analiza**

**4.Sintaksna analiza**

**5.Zaključak**

**1.Uvod**

**1.1 MAVN Prevodilac**

MAVN (Mips Asembler Visokog Nivoa) je alat koji prevodi program napisan na višem MIPS 32bit asemblerskom jeziku na osnovni asemblerski jezik. Viši MIPS 32bit asemblerski jezik služi lakšem asemblerskom programiranju jer uvodi koncept registarske promjenljive. Registarske promjenljive omogućavaju programerima da prilikom pisanja instrukcija koriste promjenljive umjesto pravih resursa. Ovo znatno olakšava programiranje jer programer ne mora da vodi računa o korišćenim registrima i njihovom sadržaju.

**1.2 Zadatak**

Zadatak MAVN prevodioca je da učitanu ulaznu datoteku pisanu na MAVN jeziku prevede na MIPS 32bit asembler. Ograničiti se na jednu ulaznu datoteku. Koristiti ekstenziju “.mavn” za ulaznu datoteku koja sadrži program na MAVN jeziku.

Prevodilac treba da prilikom prevođenja:

1. Dodeli resurse za registarske promenljive – ograničiti se na 4 registra: t0, t1, t2 i t3 iz

MIPS arhitekture

2. Sve memorijske promenljive generiše u sekciju za podatke - .data vodeći računa o sintaksi asemblerskog jezika

3. Sve instrukcije smesti u programsku sekciju - .text

4. Ime funkcije generiše kao globalni simbol - .globl i kao labelu na prvu njenu

instrukciju

5. Generiše izlaznu datoteku sa ekstenzijom “.s” koja sadrži preveden i korektan MIPS 32 bit asemblerski jezik polaznog programa.

NAPOMENE:

1. Prilikom prevođenja omogućiti detekciju i reagovanje na leksičke, sintaksne i

semantičke greške.

2. Ukoliko se polazni program uspješno prevede na MIPS 32bit asemblerski jezik,

izvršavanje programa je moguće provjeriti korišćenjem QtSpim simulatora.

**2.Analiza problema**

Sam MAVN prevodilac nudi izbor deset podržanih MIPS instrukcija a to su:

* la – (load address) ucitavanje adrese u registar
* bltz – (branch or less than zero) skok ako je registar manji od nule
* jalr – (jump and link register) skok na adresu iz registra i skladistenje povratne adrese u $ra
* lw – (load word) citanje jedne memorijske reci
* sll – (shift left logical) logicki pomeraj ulevo
* jr – (jump register) skok na adresu iz registra(return preko $ra)
* or – (or) bitsko ili
* add – (addition) sabiranje
* abs – (absolute value) apsolutna vrednost
* b – (unconditional branch) bezuslovni skok

Terminalni simboli MAVN jezika su:

: ; , ( )

\_mem, \_reg, \_func, num id, rid, mid, eof, la, bltz, jalr, lw, sll, jr, or, add, abs, b

Deklaracija funkcije:

\_func funcName

funcName – mora početi slovom, u nastavku može biti bilo koji niz slova i brojeva.

Deklaracija memorijske promenljive:

\_mem varName value

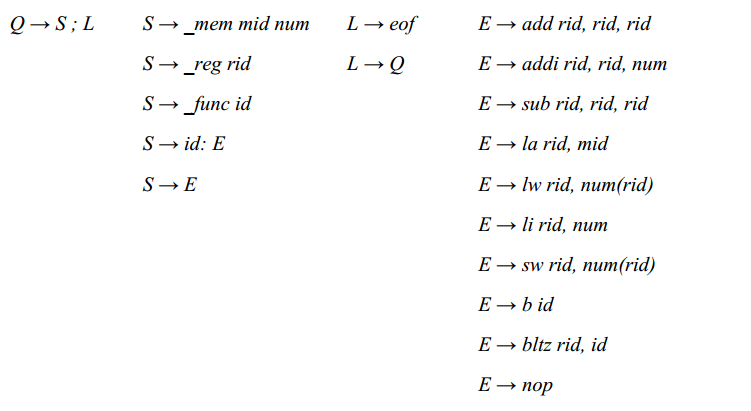
varName – mora početi malim slovom m u nastavku može biti bilo koji broj.

Deklaracija registarske promjenljive:

\_reg varName

varName – mora početi malim slovom r u nastavku može biti bilo koji broj.

Sintaksa MAVN jezika opisana je gramatikom:



Da bi se traženi zahtev programa realizovao potrebno je implementirati odgvarajuće faze u prevođenju izvornog koda i to:

* leksičku analizu
* sintaksnu analizu
* analizu životnog vijeka promjenljivih
* dodela resursa

**3.Leksička analiza**

Klasa vrši leksičku proveru ispravnosti programa, odnosno analizu nad ulaznom datotekom pozivom metoda readInputFile, initialize i Do, čija je povratna vrednost indikator uspešno izvršene leksičke analize.

Metode:

* bool readInputFile(string fileName);
* void initialize();
* bool Do();

**4.Sintaksna analiza**

Klasa implementira metode sintaksne analize programa nakon uspešno izvršene leksičke analize. Takođe vrši i semantičku analizu.Implementiraju se i metode Q S E L koje sluze za sintaksnu analizu.

**5.Zaključak**

Sintaksna analiza ne radi savrseno, program bi trebao da se debaguje medjutim u nedostatku vremena nisam stigao da to uradim kao ni drugi deo zadatka: procena zivotnog veka i dodela resursa. ☺