

Programski prevodioci: Vežbe 3

Sadržaj

1. Uvod.....	1
2. Kompajliranje, pokretanje i testiranje.....	1
3. Napomena za .l datoteku	1
4. Zadaci	2
4.1. Zadatak 1.....	2
4.2. Zadatak 2.....	2
4.3. Zadatak 3.....	3
4.4. Zadatak 4.....	3
4.5. Zadatak 5.....	3
4.6. Zadatak 6.....	4

1. Uvod

Na ovim vežbama biće prikazana gramatika programskog jezika miniC.

2. Kompajliranje, pokretanje i testiranje

Kao i prethodne nedelje, rešenje se kompajlira i pokreće na sledeći način:

```
make
./syntax
```

Kako bismo izbegli da prilikom testiranja svaki put ručno kucamo test primer, moguće je test primer iskucati u nekoj datoteci, a zatim upotrebom redirekcije proslediti tu datoteku na standardni ulaz programa:

```
./syntax < ulazna_datoteka
```

3. Napomena za **.l** datoteku

Obratiti pažnju da pojedini regularni izrazi vraćaju identične tokene:



```
"int"           { return _TYPE; }  
"unsigned"      { return _TYPE; }
```



```
"+"            { return _AROP; }  
"-"            { return _AROP; }
```

```
"<"            { return _RELOP; }  
">"            { return _RELOP; }  
"<="           { return _RELOP; }  
">="           { return _RELOP; }  
"=="           { return _RELOP; }  
"!="           { return _RELOP; }
```



Skener je implementiran na ovakav način, kako bi se pojednostavila gramatika jezika.

Na svakom mestu u gramatici gde može da se pojavi operator **+**, sigurno je uvek validno i da se na tom mestu pojavi operator **-**. Prema tome, za potrebe sintaksne analize, nema potrebe praviti distinkciju između operatora **+** i operatora **-**, pa se zato u oba slučaja vraća isti token. Isto važi i za relacione operatore i za tipove.

4. Zadaci

4.1. Zadatak 1

Deklaracija miniC promenljive omogućuje deklaraciju samo jedne promenljive:

```
int a;
```

Proširiti miniC gramatiku tako da se u jednoj deklaraciji može deklarirati više promenljivih, odvojenih zarezom:

```
int a, b, c;
```

4.2. Zadatak 2

Proširiti gramatiku **select** iskazom.

Sintaksa **select** iskaza ima oblik:

```
"select" <vars> ①  
"from" id "where" "(" <condition> ")" ";" ②
```

① Pojam <vars> predstavlja jednu ili više promenljivih, odvojenih zarezom.

② Pojam <condition> predstavlja relacioni izraz.

```
select a, b, c from details where (a > 4);
```

4.3. Zadatak 3

Proširiti miniC gramatiku **do while** iskazom.

Sintaksa **do while** iskaza ima oblik:

```
"do"  
    <statement> ①  
"while" "(" <condition> ")" ";" ②
```

① Pojam <statement> predstavlja jednu naredbu ili blok naredbi.

② Pojam <condition> predstavlja relacioni izraz.

4.4. Zadatak 4

Proširiti miniC gramatiku postinkrement operatorom, koji se može primeniti samo na identifikator.

Postinkrement operator se može pojaviti:

Unutar izraza

```
a = b + c++ - 5;
```

Kao poseban iskaz

```
a++;
```

4.5. Zadatak 5

Proširiti miniC gramatiku BASIC **for** petljom:

```
"for" <id1> "=" <const1> <direction> <const2> [ "step" <const3> ]  
    <statement>  
"next" <id2>
```

gde su <id1> i <id2> identifikatori, <const1> do <const3> su konstante, <direction> je jedna od ključnih reči "to" ili "downto", dok je deo ["step" <const3>] opcion.

Primeri:

```
for i = 1 to 9
  a = a + 1;
next i
```

```
for j = 10 downto 0 step 2
  a = a - 1;
next j
```

4.6. Zadatak 6

Proširiti miniC gramatiku funkcijama sa više parametara

```
int foo(int p, unsigned b, unsigned c){
  return p;
}
```

Poziv funkcije

```
a = foo(2, 3u, a+b);
```

Ne zaboraviti da se funkcije moraju i pozivati.