

## II kolokvijum iz Programskih prevodilaca

1. Definirati klasu za predstavljanje uslovne naredbe u apstraktnom sintaksnom stablu. Uslovna naredba je definisana sledećim skupom smena:

$CondStatement \rightarrow Statement \text{ if } Expression \mid Statement \text{ unless } Expression$

Smisao strukture je da se naredba navedena na početku strukture izvršava ukoliko je uslov naveden na kraju zadovoljen kada ispred uslova stoji ključna reč **if**. Ukoliko ispred uslova stoji ključna reč **unless**, naredba treba da se izvrši ako uslov nije zadovoljen.

Definirati zapis ovako definisane uslovne naredbe u medjukodu niskog nivoa, i u klasi za predstavljanje ove naredbe u apstraktnom sintaksnom stablu implementirati funkciju *translate* za generisanje takvog medjukoda.

2. Data je funkcija u C-u za nalaženje zadatog elementa u lančanoj listi.

```
struct list                struct list* find( struct list* head, int x )
{
    int info;               {
    struct list* next;       if ( head == 0 ) return 0;
};                           if ( head->info == x ) return head;
                             return find( head->next, x );
                             }

```

- Definirati sadržaj aktivacionog sloga funkcije *find*.
- Definirati 8086 asemblerski kod za datu funkciju. Pretpostaviti da se rezultat funkcije prenosi kao dodatni parametar.

## II kolokvijum iz Programskih prevodilaca

1. Definirati klasu za predstavljanje uslovne naredbe u apstraktnom sintaksnom stablu. Uslovna naredba je definisana sledećim skupom smena:

$CondStatement \rightarrow Statement \text{ if } Expresion \mid Statement \text{ unless } Expresion$

Smisao strkture je da se naredba navedena na početku strukture izvršava ukoliko je uslov naveden na kraju zadovoljen kada ispred uslova stoji ključna reč **if**. Ukoliko ispred uslova stoji ključna reč **unless**, naredba treba da se izvrši ako uslov nije zadovoljen.

Definirati zapis ovako definisane uslovne naredbe u medjukodu niskog nivoa, i u klasi za predstavljanje ove naredbe u apstraktnom sintaksnom stablu implementirati funkciju *translate* za generisanje takvog medjukoda.

2. Data je funkcija u C-u za nalaženje zadatog elementa u lančanoj listi.

```
struct list                struct list* find( struct list* head, int x )
{
    int info;               {
    struct list* next;       if ( head == 0 ) return 0;
};                           if ( head->info == x ) return head;
                             return find( head->next );
                             }

```

- Definirati sadržaj aktivacionog sloga funkcije *find*.
- Definirati 8086 asemblerski kod za datu funkciju. Pretpostaviti da se rezultat funkcije prenosi kao dodatni parametar.

## 1. Zadatak

### IF slučaj

IMC<exp>

Load\_Mem R1, result<exp>

JumpIfZero R1, kraj

IMC<s>

kraj:

### UNLESS slučaj

IMC<exp>

Load\_Mem R1, result<exp>

JumpIfNotZero R1, kraj

IMC<s>

kraj:

### Implementacija

```
import java.io.BufferedWriter;
```

```
import java.io.IOException;
```

```
public class CondStatement extends Statement {  
    private Statement s;  
    private Expression exp;  
    private boolean ifUnless; // true za if, false za unless
```

```

public CondStatement(Statement s, Expression exp, boolean ifUnless) {

    this.s = s;

    this.exp = exp;

    this.ifUnless = ifUnless;
}

void translate(BufferedWriter out) throws IOException {

    String kraj = ASTNode.genLab();

    exp.translate(out);

    exp.genLoad("R1", out);

    if (ifUnless) {

        // za if naredbu ako se evaluacijom exp dobije vrednost false (0 u međukodu)

        // skače se na labelu za kraj i preskače se izvršenje naredbe s

        out.write("JumpIfZero R1, " + kraj + "\n");

    } else {

        // za else naredbu ako se evaluacijom exp dobije vrednost true (1 u međukodu)

        // skače se na labelu za kraj i preskače se izvršenje naredbe s

        out.write("JumpIfNotZero R1, " + kraj + "\n");

    }

    s.translate(out);

    out.write(kraj + ":\n");

}
}

```