

# Postupci optimiranja

- **Preuređivanje naredbi**

- **Cilj**

- skratiti vrijeme izvođenja ciljnog programa
    - smanjiti veličinu ciljnog programa

- **Način**

- učinkovita uporaba svih sredstava računala
      - memorijska hijerarhija, višestruke aritmetičko-logičke jedinke, procesorski cjevovod

- **Širok spektar postupaka optimiranja prilagođenih različitim razinama međukôda**

- **Uporaba odgovarajuće strukture međukôda**

- pojednostavljuje analizu izvođenja programa
  - omogućava skupljanje veće količine podataka
  - povećava učinkovitost postupaka optimiranja



## Postupci optimiranja međukôda niže razine i ciljnog programa

- **Optimiranje se provodi**
  - tijekom generiranja ciljnog programa
  - tijekom pripreme ciljnog programa za izvođenje
- **Preuređivanjem naredbi optimira se**
  - uporaba raznih sredstava procesora
    - registara, aritmetičko-logičkih jedinki, cjevovoda, priručne memorije

## Postupci optimiranja međukôda niže razine i ciljnog programa

- **Jednostavni postupci**
  - koriste pojednostavljenu analizu izvođenja programa zasnovanu na *prozorčiću*
- **Složeni postupci**
  - koriste potpunu analizu izvođenja programa
  - Upravljanje načinom dohvata podataka
  - Optimiranje naredbi grananja
  - Optimiranje petlji
  - Optimiranje procedura
  - Optimiranje osnovnih blokova



## Jednostavni postupci optimiranja zasnovani na prozorčiću

```
MOVE #20,D0
CMP Količina,D0
BGT L1
BRA L2
```

```
L1:   MOVE #1000,D0
      MOVE D0,Cijena
      BRA L3
L2:   MOVE #1200,D0
      MOVE D0,Cijena
L3:   MOVE #8,D0
      MOVE D0,PosPor
      MOVE Cijena,D0
      MOVE D0,NovaCijena
      MOVE PosPor, D0
      DIVU #2,D0
      MOVE D0,PosPor
```



## Jednostavni postupci optimiranja zasnovani na prozorčiću

- Izuzimaju prekomjerne naredbe
- Izuzimaju nedohvatljive naredbe
- Preuređuju tijek izvođenja programa
- Algebarski pojednostavljaju izraze
- Koriste posebne strojne naredbe karakteristične za zadani procesor
- Koriste strojne naredbe koje brže računaju zadane operacije



## Jednostavni postupci optimiranja zasnovani na prozorčiću

- Izuzimanje prekomjernih naredbi

MOVE D0, Var1

MOVE Var1, D0

← Prekomjerna naredba

- Smanjuje se veličina ciljnog programa
- Ubrzava se brzina izvođenja ciljnog programa
  - izbjegava se nepotreban pristup memoriji

## Jednostavni postupci optimiranja zasnovani na prozorčiću

- Izuzimanje nedohvatljivih naredbi

JMP Ozn37

MOVE D3, Var45

Nedohvatljiva naredba

- Smanjuje se veličina ciljnog programa
- Ubrzava se brzina izvođenja ciljnog programa
  - nepotrebno se ne opterećuju sredstva procesora
    - registri, programski cjevovod, aritmetičko-logičke jedinke, priručna memorija



## Jednostavni postupci optimiranja zasnovani na prozorčiću

- Optimiranje tijeka izvođenja programa

**JMP Ozn44**

— — —

**Ozn37 : JMP Ozn44**

- Ubrzava se rad ciljnog programa
  - izvodi se samo jedna naredba bezuvjetnog grananja
  - druga naredba bezuvjetnog grananja – nedohvatljiva
- Optimiranje naredbi uvjetnog grananja
  - uvjetno grananje na naredbu bezuvjetnog grananja zamjenjuje se uvjetnim grananjem na odredišnu oznaku naredbe bezuvjetnog grananja
  - bezuvjetno grananje na naredbu uvjetog grananja zamjenjuje se uvjetnim grananjem
  - zasebno se optimira uvjetno grananje na naredbu uvjetnog grananja



## Jednostavni postupci optimiranja zasnovani na prozorčiću

- **Algebarska pojednostavljenja**

**a := a+0**

**a := a\*1**

- Smanjuje se veličina ciljnog programa
- Ubrzava se rad ciljnog programa

## Jednostavni postupci optimiranja zasnovani na prozorčiću

- **Uporaba posebnih strojnih naredbi karakterističnih za zadani procesor**

**CMP D1, D2**

**JMPPOS Ozn14**

**MOVE D2, D0**

**JMP Ozn15**

**Ozn14:**      **MOVE D1, D0**

**Ozn15:**      **- - -**

**CMP D1, D2**

**MOVE D2, D0**

**MOVEPOS D1, D0**

## Složeni postupci optimiranja

### *Upravljanje načinom dohvata podataka*

- **Unaprijedno dohvaćanje podataka**

- podaci se spremaju u priručnu memoriju ili registar
- postiže se istodobnost obrade i dohvata podataka
  - procesor obrađuje jedne podatke
  - komunikacijski sklopovi dohvaćaju druge podatke
  - primjer
    - sumira se niz vrijednosti spremljenih u radnoj memoriji računala
    - procesor pribraja jednu vrijednost
    - komunikacijski sklopovi istodobno dohvaćaju sljedeću vrijednost iz radne u priručnu memoriju



# Složeni postupci optimiranja

## Optimiranje naredbi grananja

- **Predviđanje načina izvođenja naredbi grananja**
  - naredbe koje se izvode istodobno
    - istodobnost izvođenja postiže se procesorskim cjevovodom i višestrukim aritmetičko-logičkim jedinkama
    - pogreška u predviđanju ispunjena uvjeta grananja
      - prekida se izvođenje naredbe
      - naredbe se više ne izvode istodobno, već slijedno
- **Preuređivanje strukture naredbi grananja**

## Složeni postupci optimiranja

### Optimiranje naredbi grananja

- Složena naredba grananja izvornog jezika  
**if-then-else**

```
ako (!Uvjet) skoči OznElse
- - -
ThenGrana
- - -
skoči OznZavrši
OznElse: - - -
ElseGrana
- - -
skoči OznZavrši
```



## Složeni postupci optimiranja

### Optimiranje naredbi grananja

- **Složena naredba grananja if-then-else**
  - izuzimaju se prazne grane
  - izuzimaju se nedohvatljive grane
    - uvjet *if-then-else* naredbe je konstantne vrijednosti
  - preuređuju se ugniježđeni uvjeti grananja
  - **primjer**
    - obje grane *then* i *else* su prazne
    - u dalnjem se programu ne koristi popratni rezultat
      - iz programa izuzmu se sve naredbe koje čine naredbu grananja *if-then-else*

## Složeni postupci optimiranja

### *Optimiranje naredbi grananja*

- **Prazna grana**

ako (!Uvjet) skoči OznElse

- - -

**ThenGrana**

- - -

skoči OznZavrši

OznElse:

- - -

**ElseGrana**

- - -

skoči OznZavrši



## Složeni postupci optimiranja

### *Optimiranje naredbi grananja*

- **Prazna grana**

ako (*!Uvjet*) skoči *OznZavrši*

- - -

*ThenGrana*

- - -

skoči *OznZavrši*



## Složeni postupci optimiranja

### Optimiranje naredbi grananja

- Ugniježđene **if-then-else** naredbe

```
ako (a>b) skoči OznZavrši  
c := 1
```

```
ako (a>b) skoči Ozn1  
d := 2
```

```
skoči OznZavrši
```

```
Ozn1:      d := 3
```

```
OznZavrši:- - -
```



## Složeni postupci optimiranja

### Optimiranje naredbi grananja

- Ugniježđene **if-then-else** naredbe

```
ako (a>b) skoči OznZavrši  
c := 1  
d := 2
```

OznZavrši: - - -



# Složeni postupci optimiranja

## Optimiranje petlji

- **Struktura petlje**
  - naredba uvjetnog grananja
  - naredbe tijela petlje
  - naredba bezuvjetnog grananja
- **Optimiranje naredbi petlji**
  - izuzimanje praznih petlji
    - tijelo prazne petlje nema ni jedne naredbe
  - izravnavanje petlji
    - izuzmu se naredbe grananja i ponove se naredbe njezinog tijela onoliko puta koliko se puta izvodi petlja

## Složeni postupci optimiranja

### Optimiranje petlji

- **Invertiranje petlji**

- postupak zamjene naredbe uvjetnog i bezuvjetnog grananja samo jednom naredbom grananja na kraju petlje

Ozn1 :      ako (*i>1000*) skoči Ozn2

*i* := *i+1*

*a* := *a+i*

skoči Ozn1

Ozn2 :      - - -



## Složeni postupci optimiranja

### Optimiranje petlji

- **Invertiranje petlji**

- postupak zamjene naredbe uvjetnog i bezuvjetnog grananja samo jednom naredbom grananja na kraju petlje

Ozn1 :

*i := i+1*

*a := a+i*

ako (*i<=1000*) skoči Ozn1

Ozn2 :

- - -



## Složeni postupci optimiranja

### Optimiranje petlji

- **Isključivanje petlje**

- postupak izlučivanja naredbe uvjetnog grananja izvan petlje
  - naredba uvjetnog grananja je naredba tijela petlje
  - ostale naredbe tijela petlje ne mijenjaju vrijednost izraza kojim je zadan uvjet petlje

Ozn1:

ako ( $i < 1000$ ) skoči Ozn2

$i := i + 1$

ako ( $b < c$ ) skoči Ozn3

$a := a + i$

Ozn3:

skoči Ozn1

Ozn2:

— — —



## Složeni postupci optimiranja

### Optimiranje petlji

- **Isključivanje petlje**

- postupak izlučivanja naredbe uvjetnog grananja izvan petlje
  - naredba uvjetnog grananja je naredba tijela petlje
  - ostale naredbe tijela petlje ne mijenjaju vrijednost izraza kojim je zadan uvjet petlje

Ozn1 :                    ako ( $b < c$ ) skoči Ozn2  
                              ako ( $i < 1000$ ) skoči Ozn2  
                               $i := i + 1$   
                               $a := a + i$   
Ozn3 :                    skoči Ozn1  
Ozn2 :                    - - -



## Složeni postupci optimiranja

### Optimiranje procedura

- **Postupci optimiranja**

- Zamjena naredbe poziva procedure naredbama tijela procedure koje je korisnik sam izgradio primjenom strojnog jezika
  - korisnik sam izgrađuje proceduru primjenom strojnog jezika
    - procedura učinkovito koristi sva sredstva procesora
  - procedura se ugrađuje u ciljni program tijekom njegovog generiranja
- Zamjena naredbe poziva potprograma naredbama njegovog tijela
  - izbjegava se vrijeme potrebno za izvođenje naredbe poziva procedure i naredbe povratka iz procedure
  - veliki broj poziva malih procedura
    - zamjena znatno ubrzava izvođenje ciljnog programa
    - povećava veličinu ciljnog programa



## Složeni postupci optimiranja

### Optimiranje procedura

- Maksimalno približavanje naredbi koje spremaju i osvježavaju stanje računala
  - procedura započinje naredbama koje spremaju stanje izvođenja programa
  - procedura završava naredbama koje ga osvježavaju
  - naredbe je moguće izuzeti iz ciljnog programa
    - ako ih je moguće približiti na način da one postanu susjedne
  - izuzimanjem naredbi
    - ubrzava se izvođenja ciljnog programa
    - smanjuje se veličina ciljnog programa

## Složeni postupci optimiranja

### Optimiranje osnovnih blokova

- **Preuređivanje osnovnih blokova**

- **Udruživanje osnovnih blokova**

- blokovi A i B udružuju u jedinstveni osnovni blok

- blok A jedini prethodnik bloka B
    - blok B je njegov jedini sljedbenik

- udruživanjem osnovnih blokova

- izuzima se naredba bezuvjetnog skoka na kraju bloka A
    - povećava se veličina bloka
      - » pojednostavljuje se analiza izvođenja programa
      - » povećavaju učinkovitost postupaka optimiranja
      - » dulji niz sljednih naredbi bez naredbi grananja omogućuje bolju iskoristivost procesorskog cjevovoda



## Složeni postupci optimiranja

### Optimiranje osnovnih blokova

- Povezivanjem zajedničkih završetaka osnovnih blokova
  - više osnovnih blokova završava jednakim naredbama
  - izvođenje svih blokova nastavlja se istim blokom
    - zajedničke naredbe svih blokova osim jednog zamijene naredbom bezuvjetnog grananja
    - naredba bezuvjetnog grananja usmjeruje izvođenje programa na zajedničke naredbe u bloku u kojem one nisu izuzete
  - smanjuje se veličina ciljnog programa
  - moguće je ubrzanje izvođenja ciljnog programa
    - manji ciljni program stane čitav u naredbenu priručnu memoriju bez potrebe njezinog dinamičkog punjenja tijekom izvođenja ciljnog programa



## Postupci optimiranja međukôda više razine i izvornog programa

- **Cilj**

- skratiti vrijeme izvođenja ciljnog programa
- smanjiti veličinu ciljnog programa

- **Način**

- učinkovitom uporabom memorijske hijerarhije
  - registara
  - priručne memorije

- **Postupci optimiranja**

- Upravljanje načinom pristupa podacima
- Optimiranje petlji

## Upravljanje načinom pristupa podacima

- **Upravljanje načinom pristupa podacima**

- Skalarne varijable

- stavci polja pristupa se primjenom skalarne varijable
    - skalarna varijabla omogućuje spremanje vrijednosti stavke polja u registar

```
za (i=2; i<n; i++)  
    a[i] = a[i-1]+11;
```

```
s = a[1];  
za (i=2; i<n; i++)  
{  
    s = s+11;  
    a[i] = s;  
}
```



## Optimiranje petlji

- **Vremenska bliskost pristupa podacima**

- naredbe koje se izvode jedne iza drugih koriste iste podatke
- omogućava učinkovitu uporabu memorijske hijerarhije
- jednom dohvaćeni podatak u registar ili priručnu memoriju koristi više naredbi koje se slijedno izvode jedna iza druge
- naredbe koje koriste podatke u registrima manje su veličine i brže od naredbi koje koriste podatke u memoriji

- **Prostorna bliskost pristupa podacima**

- naredbe koje se izvode jedne iza drugih koriste susjedne podatke u memoriji
- osiguran je unaprijedni dohvati podataka
  - iz radne memorije u priručnu memoriju prebacuju se blokovi veličine 32 i više okteta
- ne isprazne jedan drugome priručnu memoriju



- **Preuređivanje petlji**

- mijenja se redoslijed i smjer izvođenja petlji
- preoblikuje se prostor indeksa petlji
- više petlji udružuje se u jedinstvenu petlju
- petlje se raspodijeljuju u više petlji
- stvaraju se manji blokovi u prostoru indeksa petlje

## Optimiranje petlji

- **Radna memorija**

- redak po redak

$a[1][1], a[1][2], a[1][3], \dots$

```
za (i=1; i<101; i++)  
    za (j=1; j<101; j++)  
    {  
        Dohvati stavku a[i][j];  
    }
```

- **Pristup stavkama polja**

- stupac po stupac

$a[1][1], a[2][1], a[3][1], \dots$

**Nisu prostorno  
bliski**



## Optimiranje petlji

- **Radna memorija**

- redak po redak

$a[1][1], a[1][2], a[1][3], \dots$

```
za (i=1; i<101; i++)  
    za (j=1; j<101; j++)  
    {  
        Dohvati stavku a[i][j];  
    }
```

- **Pristup stavkama polja**

- redak po redak

$a[1][1], a[1][2], a[1][3], \dots$

**Prostorno  
bliski**

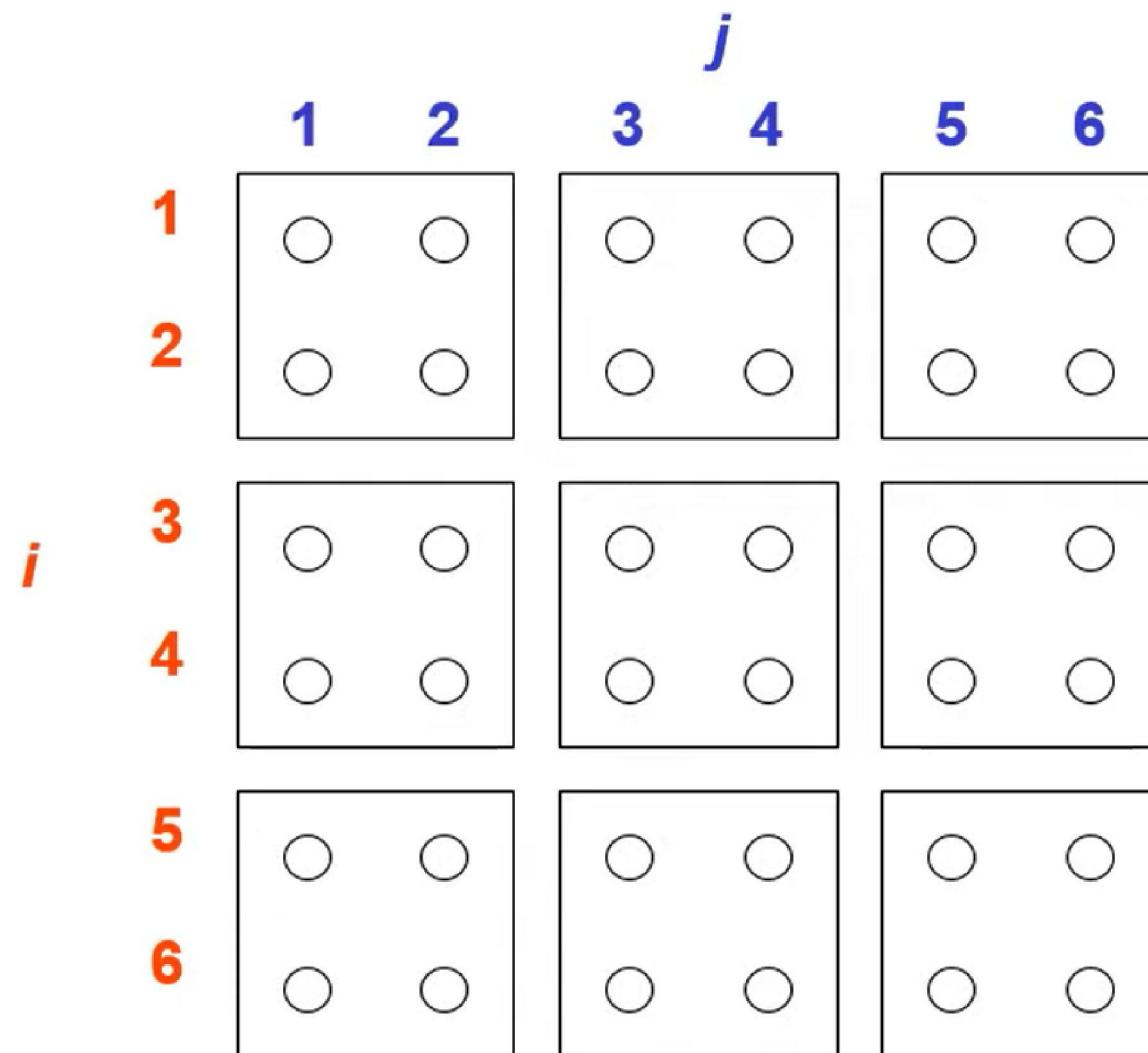


## Optimiranje petlji

```
za (i=1; i<=6; i++)
    za (j=1; j<=6; j++)
    {
        ako (i == Paran broj)
            m = i-1;
        inače
            m = i+1
        ako (j == Paran broj)
            n = j-1;
        inače
            n = j+1
        Obradi stavku a[i] [j];
        Obradi stavku a[m] [j];
        Obradi stavku a[i] [n];
        Obradi stavku a[m] [n];
    }
```



## Optimiranje petlji



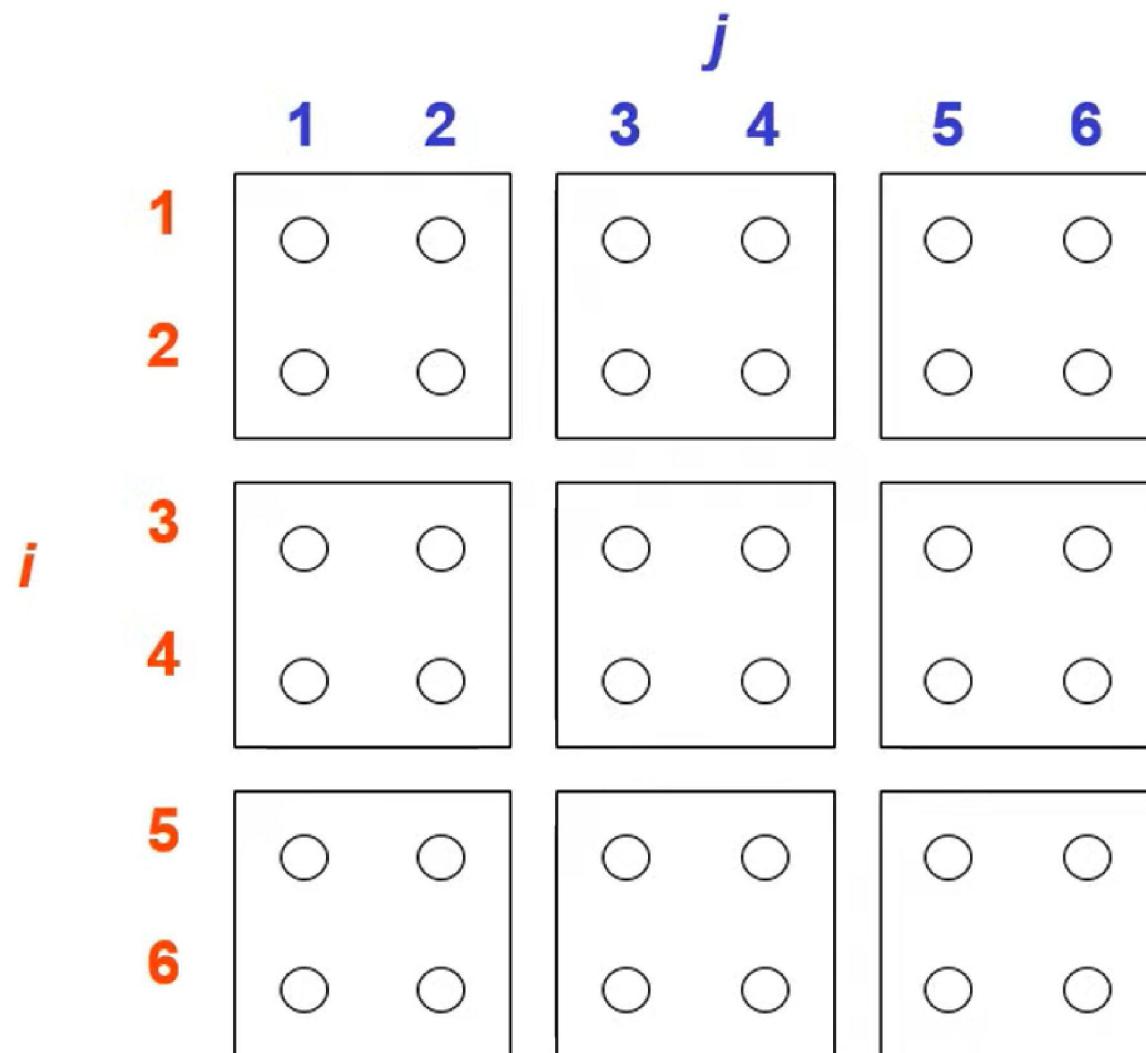
Nisu vremenski  
bliski

## Optimiranje petlji

```
za (i=1; i<=6; i=i+2)
    za (j=1; j<=6; j=j+2)
        za (i1=i; i1<=min(i+1,6); i1++)
            za (j1=j; j1<=min(j+1,6); j1++)
{
    ako (i1 == Parni broj)
        m = i1-1;
    inače
        m = i1+1
    ako (j1== Parni broj)
        n = j1-1;
    inače
        n = j1+1
    Obradi stavku a[i1][j1];
    Obradi stavku a[m][j1];
    Obradi stavku a[i1][n];
    Obradi stavku a[m][n];
}
```



## Optimiranje petlji



Vremenski  
bliski

## Postupci optimiranja međukôda srednje razine

- **Međukôd srednje razine**
  - pojednostavljene naredbe izvornog jezika
  - sliče strojnim naredbama
  - moguće ga je optimirati primjenom različitih postupaka
    - uključujući i prethodno opisane postupke koji su prilagođeni optimiranju međukôda više i niže razine



## Postupci optimiranja međukôda srednje razine

- **Postupci optimiranja**
  - Optimiranje procedura
  - Optimiranje petlji
  - Optimiranje izraza
  - Prenošenje jediničnih naredbi
- **Optimiranje procedura: cilj**
  - smanjiti utjecaj izvođenja naredbi poziva i povratka iz procedure
  - smanjiti utjecaj naredbi koje spremaju i osvježavaju stanje izvođenja programa

## Optimiranje procedura

- prethodno opisani postupci optimiranja procedura
- pretvorba rekurzivnih poziva procedura u petlje
  - skraćuje se vrijeme izvođenja programa
    - ne izvode se naredbe poziva i povratka iz procedure
    - ne izvode se naredbe koje spremaju i osvježavaju stanje izvođenja programa



## Optimiranje procedura

- **uvišestručavanje procedura**
  - procedura se poziva na više različitih mesta
  - više različitih poziva koristi jednake vrijednosti ulaznih parametra
  - umjesto jedne, definira se više procedura različitih imena
    - sve izgrađene procedure iste su funkcionalnosti
    - procedure koriste različite konstantne vrijednosti ulaznih parametara
    - konstantne vrijednosti ulaznih parametara jednoznačno su određene imenom procedure
    - konstantne vrijednosti ulaznih parametara se ne razmjenjuju pozivima procedura
  - ne izvode se naredbe mehanizma razmjene vrijednosti parametara
    - skraćuje se vrijeme izvođenja programa



## Optimiranje petlji

- provjera granica indeksa polja
  - dohvat stavke polja u petlji
    - višekratna provjera indeksa polja
    - zamjenjuje se jedinstvenom provjerom indeksa petlje
- spregnute varijable
  - vrijednosti varijable se tijekom izvođenja petlje postupno smanjuje ili povećava za konstantnu vrijednost

```
Ozn: i := i-1
      a := 5*i
      ako (i>10) skoči Ozn
```



## Optimiranje petlji

- provjera granica indeksa polja
  - dohvat stavke polja u petlji
    - višekratna provjera indeksa polja
    - zamjenjuje se jedinstvenom provjerom indeksa petlje
- spregnute varijable
  - vrijednosti varijable se tijekom izvođenja petlje postupno smanjuje ili povećava za konstantnu vrijednost

**a := 5\*i**

**Ozn: i := i-1**

**a := a-5**

**ako (i>10) skoči Ozn**

**a – spregnuta varijabla**

## Optimiranje izraza

- **unaprijedno računanje izraza konstantne vrijednosti**
  - ne generiraju se naredbe koje računaju konstantnu vrijednost izraza
  - u generirani ciljni program izravno se zapiše rezultat izračunatog konstantnog izraza
- **izlučivanje izraza izvan petlje**
  - naredbe petlje ne mijenjaju vrijednost izraza
  - naredbe koje računaju izraz izluče se izvan petlje
- **izuzimanje istovjetnih izraza**
  - izraz se računa dva ili više puta
  - ne mijenjaju se vrijednosti varijabli izraza između dva računanja

## Optimiranje izraza

- **izuzimanje mrtvih izraza**
  - vrijednost izraza se ne koristi u nastavku programa
- **razlaganje izraza, pojednostavljenje izraza**
  - primjenom svojstava asocijativnosti, komutativnosti, distributivnosti
  - omogućuje primjenu raznih postupaka optimiranja
    - algebarska pojednostavljenja, uporaba učinkovitih strojnih operacija

## Prenošenje jediničnih naredbi

- **Jedinična naredba *Lijevo:=Desno***
  - **Lijevo** - ime varijable
  - **Desno** - vrijednost konstante ili ime isključivo jedne varijable
  - jediničnu naredbu slijedi **Rezultat:=Izraz(Lijevo, itd.)**
  - naredbe koje se izvode između jedinične naredbe **Lijevo:=Desno** i naredbe **Rezultat:=Izraz(Lijevo, itd.)** ne mijenjaju vrijednost varijable **Lijevo** i vrijednost desne strane **Desno**
    - umjesto imena varijable **Lijevo** koristi se vrijednost konstante ili ime varijable **Desno**
    - nakon zamjene - **Rezultat:=Izraz(Desno, itd.)**



## Prenošenje jediničnih naredbi

- opisano preuređivanje omogućuje
  - izuzimanje mrtve naredbe
    - ako nakon zamjene nijedna naredba u nastavku programa ne koristi varijablu **Lijevo**, onda je naredba **Lijevo:=Desno** mrtva
  - unaprijedno računanje konstantnih izraza, izlučivanje konstantnih izraza izvan petlje, izuzimanje nedohvatljivih grana
    - moguće je da izraz **Izraz(Desno,itd.)** poprimi konstantnu vrijednost nakon zamjene imena varijable **Lijevo** konstantom **Desno**