Programski prevodioci (vežbe)

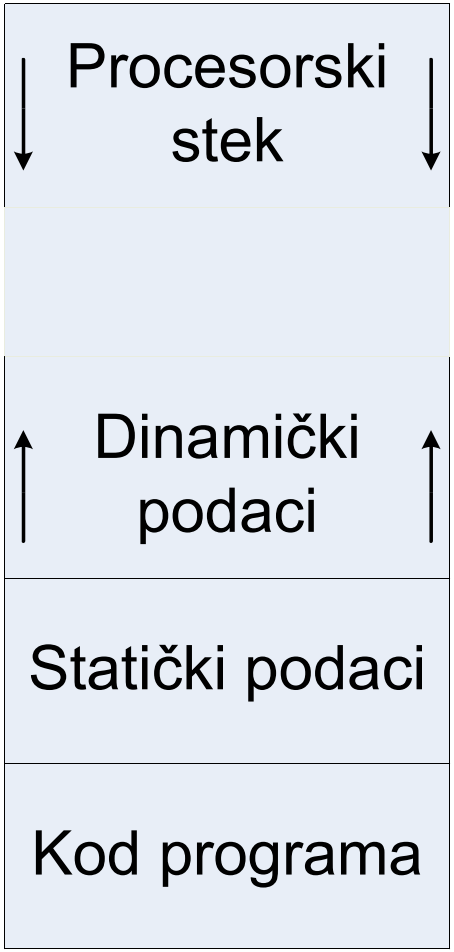
Upravljanje memorijom u toku izvršenja programa

Prevođenje poziva potprograma

Upravljanje memorijom (Runtime memory management)

- Organizacija memorije u toku izvršenja programa
- Pristup podacima u različitim delovima memorije
- Promene sadržaja memorije u trenutku poziva potprograma i u trenutku povratka na pozivajući modul

Uobičajena organizacija memorije



Primer: Smeštanje podataka u različite delove memorije u C programima

- U oblasti statičkih podataka:
  - Globalne promenljive
  - Lokalne promenljive koje imaju klasu memorije static.
- U oblasti steka:
  - Parametri funkcija
  - Lokalne promenljive koje imaju klasu memorije auto.
- U oblasti dinamičkih podataka:
  - Neimenovane promenljive za koje je dinamički alociran memorijski prostor u toku izvršenja programa (malloc ili calloc funkcijom). Ovim promenljivama se pristupa pomoću pokazivača.
- U registrima procesora:
  - Lokalni podaci koji imaju klasu memorije register.

Načini adresiranja kod mikroprocesora 8086

Način adresiranja	Format	Primer	Značenje
neposredno	`CONST`	100	Vrednost operanda je navedena konstanta.
direktno memorijsko	`NAME` ili `[CONST]`	X ili [0100]	Vrednost operanda je sadržaj memorijske lokacije čija je adresa data simboličkim imenom ili konstantom.
direktno registarsko	`REG`	AX	Vrednost operanda je sadržaj datog registra.
indirektno registarsko	`[REG]`	[BX]	Vrednost operanda je u memorijskoj lokaciji čija se adresa nalazi u datom registru.
bazno	`[REG+CONST]`	[BP + 4]	Bazni registar čuva početnu adresu nekog bloka, a stvarna adresa operanda se dobija tako što se na sadržaj datog baznog registra doda vrednost navedene konstante.
indeksno	`NAME[REG]` ili `CONST[REG]`	X[SI] ili 25h[DI]	Konstanta ili simboličko ime predstavlja početnu adresu bloka, a u registru je zapamćen pomeraj u odnosu na tu početnu adresu. Adresa operanda se opet dobija sabiranjem početne adrese i pomeraja.

Pristup podacima koji su smešteni u različitim delovima memorije

- U registrima procesora:
  - Direktnim registarskim adresiranjem
- U delu za statičke podatke:
  - Direktnim memorijskim adresiranjem
- U steku:
  - Baznim adresiranjem
- U delu za dinamičke podatke:
  - Pristup adresi (na jedan od prethodnih načina) – u neki od registara se dovede adresa podatka
  - Indirektnim registarskim adresiranjem

Primer:

- Napisati asemblerski kod za izvršenje sledećih C-naredbi:

```
register int i;
static int j;
int a;
static int *b;
i=j;
a=*b;
```

  - Promenljiva i je smeštana u nekom registru (neka je to registar SI).
  - Promenljiva j je u delu za statičke podatke čija je adresa određena u fazi prevođenja i zapamćena u tabeli simbola. Neka je to adresa 100.
  - Promenljiva a je zapamćena u steku (ako je to jedina lokalna promenljiva sa klasom meorije auto, njen pomeraj u odnosu na registar BP je -2)
  - Pokazivačka promebljiva b je, takođe, u statičkoj zoni memorije i neka je ona zapisana na adresi 102.

Rešenje:

- Naredbi

```
i=j;
```

odgovara kod:

```
MOV    SI, [0100]
```

– registarsko adresiranje za pristup prom i, direktno mem. adresiranje za pristup prom. j

- Naredbi

```
a=*b;
```

odgovara kod:

```
MOV    BX, [0102]
```

- direktno mem. adresiranje za pristup pokazivaču b

```
MOV    AX, [BX]
```

- indirektno registarsko adresiranje za pristup neimenovanoj prom. na koju ukazuje b

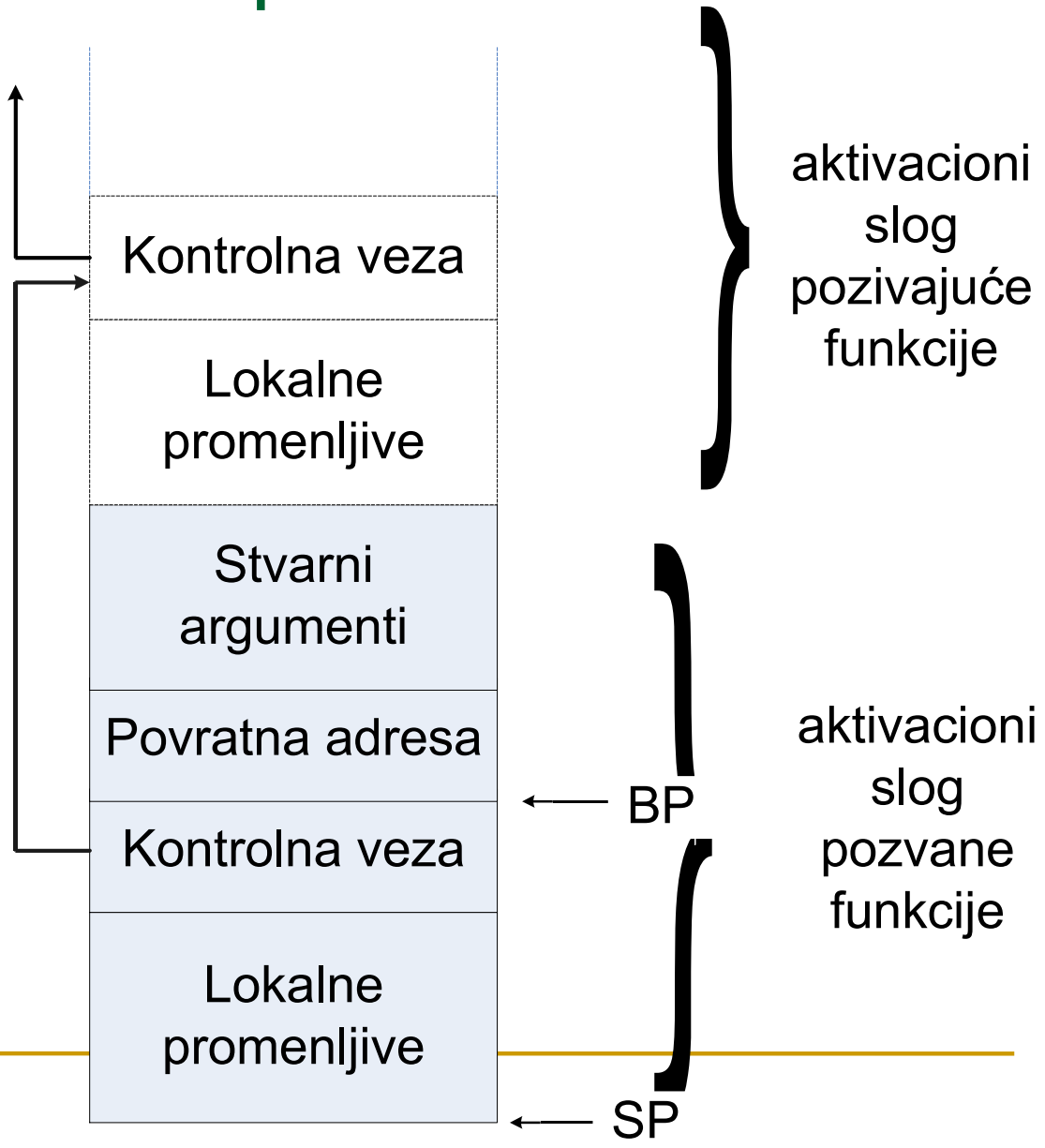
```
MOV    [BP-02], AX
```

- indeksno adresiranje za pristup prom. a

Promena sadržaja steka prilikom poziva potprograma

- Prilikom poziva potprograma kreira se tzv. aktivacioni slog (aktivacioni zapis) potprograma i smešta se u stek.
- U aktivacionom slogu pamte se:
  - Stvarni parametri,
  - Povratna adresa,
  - Kontrolna veza (prethodni sadržaj baznog registra BP)
  - Lokalne promenljive

Sadržaj steka nakon poziva potprograma



Sekvenca pozivanja

- **Sekvenca pozivanja** je skup aktivnosti koje izvršavaju pozivajuća i pozvana funkcija da bi se kreirao aktivacioni zapis pozvane funkcije.
- Aktivnosti pozivajuće funkcije:
  - Na steku rezerviše prostor za smeštanje rezultata pozvane funkcije
  - Na stek stavlja stvarne parametre pozvane funkcije.
  - Izvršava CALL naredbu koja na stek stavlja povratnu adresu i prenosi kontrolu pozvanoj funkciji.
- Aktivnosti pozvane funkcije:
  - Na stek stavlja sadržaj registra BP (kontrolna veza).
  - Sadržaj registra SP, koji u tom trenutku ukazuje na kontrolnu vezu, kopira u BP.
  - Od sadržaja registra SP oduzima broj memorijskih lokacija potrebnih za smeštaj lokalnih promenljivih. Na ovaj način se na steku alocira potreban prostor za lokalne promenljive.

Sekvenca povratka

- **Sekvenca povratka** je skup aktivnosti koje izvršavaju pozvana i pozivajuća funkcija da bi se aktivacioni zapis pozvane funkcije izbacio iz steka.
- Aktivnosti pozvane funkcije:
  - Sadržaj registra BP kopira u registar SP čime se sa steka oslobađa prostor za lokalne promenljive i one efektivno prestaju da postoje.
  - Sa vrha steka (koji u tom trenutku ukazuje na polje kontrolne veze pozvane procedure) skida se vrednost kontrolne veze i smešta u registar BP.
  - Izvršava naredbu RET koja skida sa vrha steka povratnu adresu i smešta u brojač naredbi IP.
- Aktivnosti pozivajuće funkcije:
  - Dodavanjem odgovarajuće vrednosti na SP sa steka uklanja stvarne parametre i rezultat pozvane funkcije i normalno nastavlja rad.

Primer:

- Prikazati izgled generisanog koda za funkciju f, aktivacioni slog i deo koda koji odgovaraju sledećem pozivu funkcije f:

```
f(1,2)

void f( int a, int b )
{
  int c;
  c=a+b;
}
```

Generisani kod poziva funkcije f

- deo sekvence pozivanja
  - smeštanje stvarnih parametara na stek

```
MOV     AX,2
PUSH    AX
MOV     AX,1
PUSH    AX
```

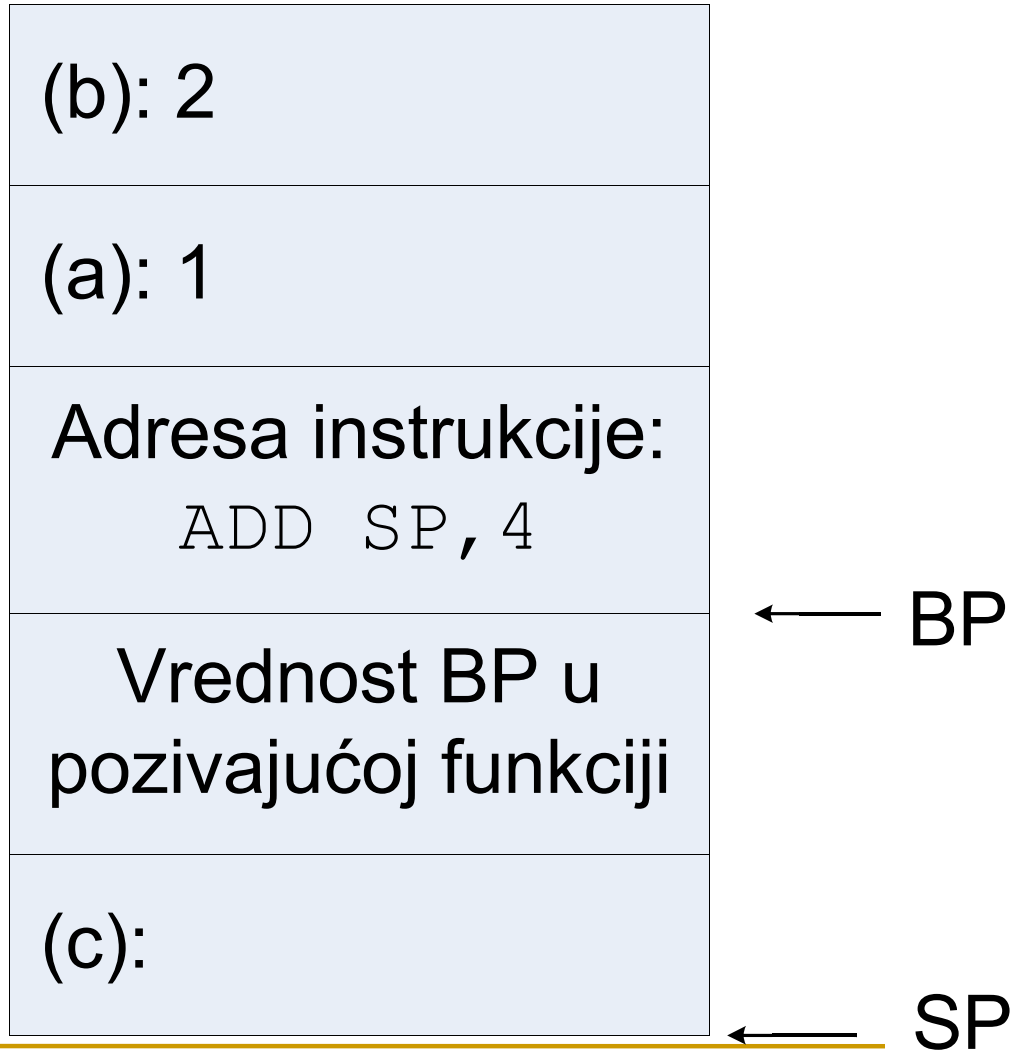
  - poziv funkcije f

```
CALL    f
```

- deo sekvence povratka
  - skidanje stvarnih parametara sa steka

```
ADD  SP,4
```

Aktivacioni slog funkcije f



Generisani kod funkcije f

- deo sekvence pozivanja
  - smeštanje sadržaja BP u stek

```
PUSH  BP
```

  - definisanje novog sadržaja BP

```
MOV   BP, SP
```

  - alociranje prostora za lokalnu promenljivu c

```
SUB   SP,2
```

- telo funkcije

```
MOV   AX, [BP+04]
ADD   AX, [BP+06]
MOV   [BP-02], AX
```

- deo sekvence povratka
  - oslobađanje prostora rezervisanog za lokalne promenljive

```
MOV   SP, BP
```

  - rekonstrukcija sadržaja registra BP

```
POP   BP
```

  - povratak u pozivajući modul

```
RET
```