

Primeri asemblerskih programa

Primer: Program za računanje NZD

	POČETAK	ulaz
ulaz :	PREBACI_NR	\$12,%0
	PREBACI_NR	\$10,%1
ponovo :	UPOREDI	%1,%0
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće :	ODUZMI	%1,%0
	SKOČI	ponovo
manje :	ODUZMI	%0,%1
	SKOČI	ponovo
kraj :	SKOČI	kraj
	KRAJ	

Dvostruka preciznost – neoznačeni brojevi

	POČETAK	ulaz
a_donji:	NAPUNI	0xFFFF
a_gornji:	NAPUNI	0xFFFF
b_donji:	NAPUNI	0xFFFF
b_gornji:	NAPUNI	0xFFFF
greška:	NAPUNI	0
ulaz:	PREBACI_DR	a_donji,%0
	PREBACI_DR	a_gornji,%1
	PREBACI_DR	b_donji,%2
	PREBACI_DR	b_gornji,%3
	SABERI	%2,%0
	SABERI_P	%3,%1
	SKOČI_ZA_P	van_opsega
	SKOČI	kraj
van_opsega:	PREBACI_NR	\$1,%4
	PREBACI_RD	%4,greška
kraj:	SKOČI	kraj
	KRAJ	

Dvostruka preciznost – označeni brojevi

	POČETAK	ulaz
a_donji:	NAPUNI	0xFFFF
a_gornji:	NAPUNI	0xFFFF
b_donji:	NAPUNI	0xFFFF
b_gornji:	NAPUNI	0xFFFF
greška:	NAPUNI	0
ulaz:	PREBACI_DR	a_donji,%0
	PREBACI_DR	a_gornji,%1
	PREBACI_DR	b_donji,%2
	PREBACI_DR	b_gornji,%3
	SABERI	%2,%0
	SABERI_P	%3,%1
	SKOČI_ZA_V	van_opsega
	SKOČI	kraj
van_opsega:	PREBACI_NR	\$1,%4
	PREBACI_RD	%4,greška
kraj:	SKOČI	kraj
	KRAJ	

Rad sa mašinskom normalizovanom formom

$$\begin{aligned}
 -1.5_{10} &== \\
 -1.1000000_2 \times 2^0 &== \\
 1100000001000000_2
 \end{aligned}$$

realan:	NAPUNI	0xC040
---------	--------	--------

Postavljanje predznaka na 0:

PREBACI_DR	realan,%0
PREBACI_NR	\$0x7FFF,%1
I	%1,%0

Postavljanje predznaka na 1:

PREBACI_NR	\$0x8000,%1
ILI	%1,%0

Rad sa mašinskom normalizovanom formom

Izdvajanje eksponenta:

	PREBACI_DR	realan, %0
	PREBACI_NR	\$0x7F80, %1
	I	%1, %0
	PREBACI_NR	\$7, %2
ponovo:	DESNO	%0
	ODBIJ_1	%2
	SKOČI_ZA_NE_N	ponovo

Rukovanje logičkim vrednostima

$c = (a \neq b)$

$d = c$

a:	ZAUZMI	1
b:	ZAUZMI	1
c:	ZAUZMI	1
d:	ZAUZMI	1
	...	
	PREBACI_DR	a, %0
	PREBACI_DR	b, %1
	PREBACI_NR	\$1, %2
	UPOREDI	%1, %0
	SKOČI_ZA_!=	dalje
	PREBACI_NR	\$0, %2
dalje:	PREBACI_RD	%2, c
	PREBACI_RD	%2, d

Računanje vrednosti celobrojnih izraza

a, b, c – neoznačeni celi brojevi

$$c = (a-b)*2+(a+b)/2$$

a:	ZAUZMI	1	
b:	ZAUZMI	1	
c:	ZAUZMI	1	
	...		
	PREBACI_DR	a, %0	
	PREBACI_DR	b, %1	
	PREBACI_DR	a, %2	
	ODUZMI	%1, %2	
	LEVO	%2	← (a-b)*2
	SABERI	%1, %0	
	DESNO	%0	← (a+b)/2
	SABERI	%0, %2	
	PREBACI_RD	%2, c	

Rukovanje nizovima

- $t[365]$ – niz od 365 vrednosti temperatura
- **prvi, poslednji** – indeks prvog i poslednjeg elementa podintervala
- **donja, gornja** – donja i gornja granica temperature
- **broj = 0** – brojač dana sa temperaturom iz opsega

```
for i in range(prvi, poslednji+1):  
    if (t[i]>=donja) and (t[i]<=gornja):  
        broj += 1
```

t:	ZAUZMI	365	
prvi:	ZAUZMI	1	
poslednji:	ZAUZMI	1	
donja:	ZAUZMI	1	
gornja:	ZAUZMI	1	
broj:	ZAUZMI	1	
	...		
	PREBACI_NR	\$0,%0	← brojač
	PREBACI_DR	prvi,%1	← indeks
	PREBACI_DR	poslednji,%2	
	PREBACI_DR	donja,%3	
	PREBACI_DR	gornja,%4	
provera:	UPOREDI	%2,%1	← uslov za kraj <i>for</i>
	SKOČI_ZA_>	izlaz	
ponovo:	UPOREDI	%3,t(%1)	← $t[i] \geq \text{donja}$
	SKOČI_ZA_±_<	naredni	
	UPOREDI	%4,t(%1)	← $t[i] \leq \text{gornja}$
	SKOČI_ZA_±_>	naredni	
	DODAJ_1	%0	← broj=broj+1
naredni:	DODAJ_1	%1	
	SKOČI	provera	
izlaz:	PREBACI_RD	%0,broj	

Rukovanje slogovima

```
class vreme:
```

```
    sat = 0
```

```
    minut = 0
```

```
    sekund = 0
```

```
a = vreme()
```

```
b = vreme()
```


```
a.sat = 18
```

```
a.minut = 30
```

```
...
```

```
b = copy.deepcopy(a)
```

Rukovanje slogovima

a:		ZAUZMI	3
b:		ZAUZMI	3
		...	
	indeks		
		PREBACI_NR	\$0, %0
		PREBACI_IR	a(%0), %1
		PREBACI_RI	%1, b(%0)
		DODAJ_1	%0
		PREBACI_IR	a(%0), %1
		PREBACI_RI	%1, b(%0)
		DODAJ_1	%0
		PREBACI_IR	a(%0), %1
		PREBACI_RI	%1, b(%0)

Potprogram

Potprogram

- Omogućava izdvajanje često korišćenih sekvenci naredbi
- Omogućava rešavanje problema po funkcionalnim delovima
- Poziv potprograma – prenošenje parametara
 - argumenti – konkretne vrednosti parametara
- Ulazna i izlazna tačka
- Lokalne promenljive – oblast važenja
- Dve vrste potprograma: **funkcije** i **procedure**

Primer: potprogram za NZD

```
def nzd(a, b):  
    while a != b:  
        if a > b:  
            a = a - b  
        else:  
            b = b - a  
    return a  
  
print nzd(10, 12)  
  
x = 15  
y = 115  
z = nzd(x, y)
```

Primer: asemblerski potprogram za NZD

```

nzd:      UPOREDI      %2,%1
          SKOČI_ZA_ =   kraj
          SKOČI_ZA_ <  manje
veće:     ODUZMI      %2,%1
          SKOČI      nzd
manje:    ODUZMI      %1,%2
          SKOČI      nzd
kraj:     PREBACI_RR  %1,%0
          NATRAG
  
```

Funkcija:

%1 - prvi parametar

%2 - drugi parametar

%0 - rezultat

```

x:      NAPUNI      9
y:      NAPUNI      3
z:      ZAUZMI      1
  
```

```

PREBACI_NR    $12,%1
PREBACI_NR    $10,%2
POZOVI       nzd
PREBACI_RD    %0,z
  
```

```

PREBACI_DR    x,%1
PREBACI_DR    y,%2
POZOVI       nzd
PREBACI_RD    %0,z
  
```

Primer: asemblerski potprogram za NZD

```

nzd:      UPOREDI      %2, %1
          SKOČI_ZA_ =   kraj
          SKOČI_ZA_ <  manje
veće:    ODUZMI      %2, %1
          SKOČI      nzd
manje:   ODUZMI      %1, %2
          SKOČI      nzd
kraj:    PREBACI_RP   %1, (%3)
          NATRAG
  
```

Procedura:

%1 - prvi parametar

%2 - drugi parametar

%3 – adresa rezultata

```

x:      NAPUNI      9
y:      NAPUNI      3
z:      ZAUZMI      1
  
```

```

PREBACI_NR    $12, %1
PREBACI_NR    $10, %2
PREBACI_NR    $z, %3
POZOVI      nzd
  
```

```

PREBACI_DR    x, %1
PREBACI_DR    y, %2
PREBACI_NR    $z, %3
POZOVI      nzd
  
```

Broj naredbi pri pozivu potprograma

x:	NAPUNI	9
y:	NAPUNI	3
z:	ZAUZMI	1
	...	
nzd:	UPOREDI	%2,%1
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće:	ODUZMI	%2,%1
	SKOČI	nzd
manje:	ODUZMI	%1,%2
	SKOČI	nzd
kraj:	PREBACI_RP	%1,(%3)
	NATRAG	
	...	
	PREBACI_DR	x,%1
	PREBACI_DR	y,%2
	PREBACI_NR	\$z,%3
	POZOVI	nzd

Opis i poziv

x:	NAPUNI	9
y:	NAPUNI	3
z:	ZAUZMI	1
	...	
nzd:	PREBACI_DR	x,%1
	PREBACI_DR	y,%2
	PREBACI_NR	\$z,%3
nzd1:	UPOREDI	%2,%1
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće:	ODUZMI	%2,%1
	SKOČI	nzd1
manje:	ODUZMI	%1,%2
	SKOČI	nzd1
kraj:	PREBACI_RP	%1,(%3)

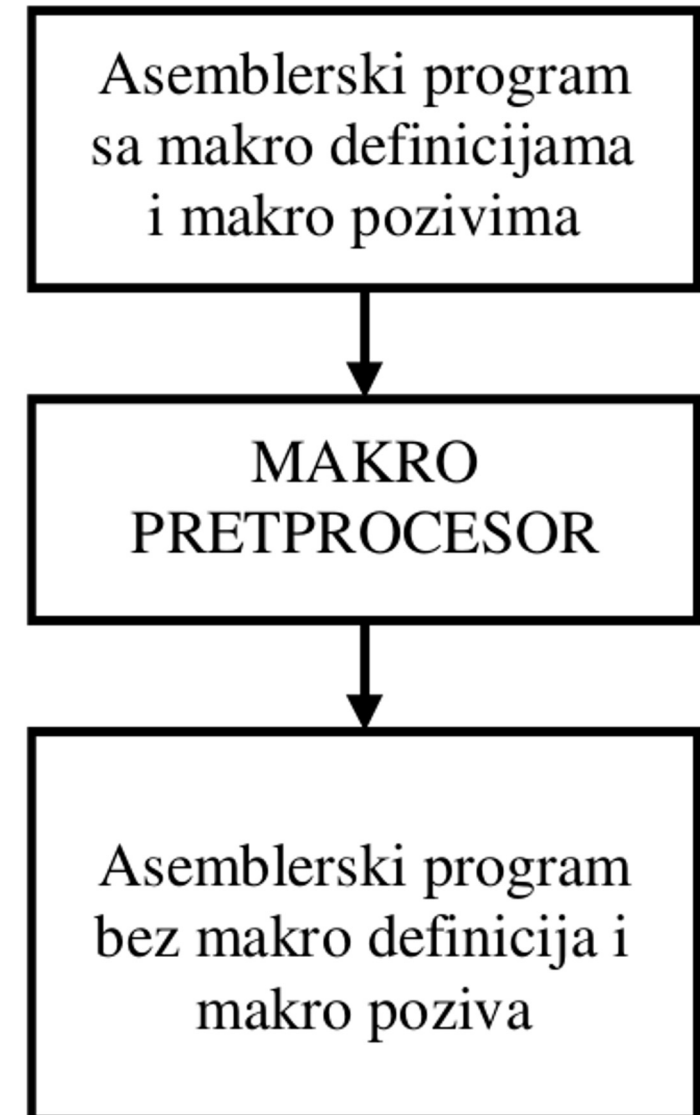
Samo opis

broj poziva	broj naredbi	broj naredbi
	kada istovremeno postoje i opis i poziv potprograma	kada se poziv potprograma zamenjuje opisom potprograma
1	13	11
2	17	22
3	21	33

Makro

Makro

- Zamena poziva definicijom
- makro poziv
- makro definicija
- makro pretprocesor
 - zamena/supstitucija



Opis makro definicija i makro poziva

veliko_slovo -> A|B|C|Č|Ć|D|Đ|E|F|G|H|I|J|K|L|M|N|O|P|R|S|Š|T|U|V|Z|Ž

ime -> veliko_slovo{veliko_slovo|cifra|_}

parametar -> labela|ime

argument -> (broj|parametar) [(+|-|*|/)] (broj|parametar)]
|registar

makro_definicija -> nova_linija ime razmak MAKRO razmak {parametar[,]}
telo nova_linija KRAJ

makro_poziv -> nova_linija [labela:] razmak ime razmak {argument[,]}

Opis makro definicija i makro poziva

Prošireno telo asemblerskog programa:

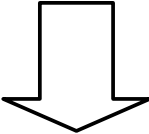
```
telo -> {direktiva  
        |osnovna_naredba  
        |naredba_prebacivanja  
        |upravljačka_naredba  
        |makro_definicija  
        |makro_poziv}
```

Primer: makro poziv - NZD

Makro definicija

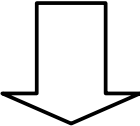
NZD	MAKRO	a, b, c
	PREBACI_DR	a, %1
	PREBACI_DR	b, %2
	PREBACI_NR	\$c, %3
nzd:	UPOREDI	%2, %1
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće:	ODUZMI	%2, %1
	SKOČI	nzd
manje:	ODUZMI	%1, %2
	SKOČI	nzd
kraj:	PREBACI_RP	%1, (%3)
	KRAJ	

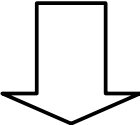
Makro poziv

	NZD	x, y, z
		
	PREBACI_DR	x, %1
	PREBACI_DR	y, %2
	PREBACI_NR	\$z, %3
nzd:	UPOREDI	%2, %1
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće:	ODUZMI	%2, %1
	SKOČI	nzd
manje:	ODUZMI	%1, %2
	SKOČI	nzd
kraj:	PREBACI_RP	%1, (%3)

Makro unutar makroa

DVOSTRUKO NAZIV	MAKRO MAKRO OPERACIJA1 OPERACIJA2 KRAJ KRAJ	NAZIV , OPERACIJA1 , OPERACIJA2 %2 , %0 %3 , %1
--------------------	--	---

	DVOSTRUKO	SABERI_2 , SABERI , SABERI_P
		
SABERI_2	MAKRO SABERI SABERI_P KRAJ	%2 , %0 %3 , %1

	DVOSTRUKO	ODUZMI_2 , ODUZMI , ODUZMI_P
		
ODUZMI_2	MAKRO ODUZMI ODUZMI_P KRAJ	%2 , %0 %3 , %1

Uslovno asembliranje

Uslovna direktiva USLOVNO određuje pod kojim uslovom pretprocesor propušta na asembliranje

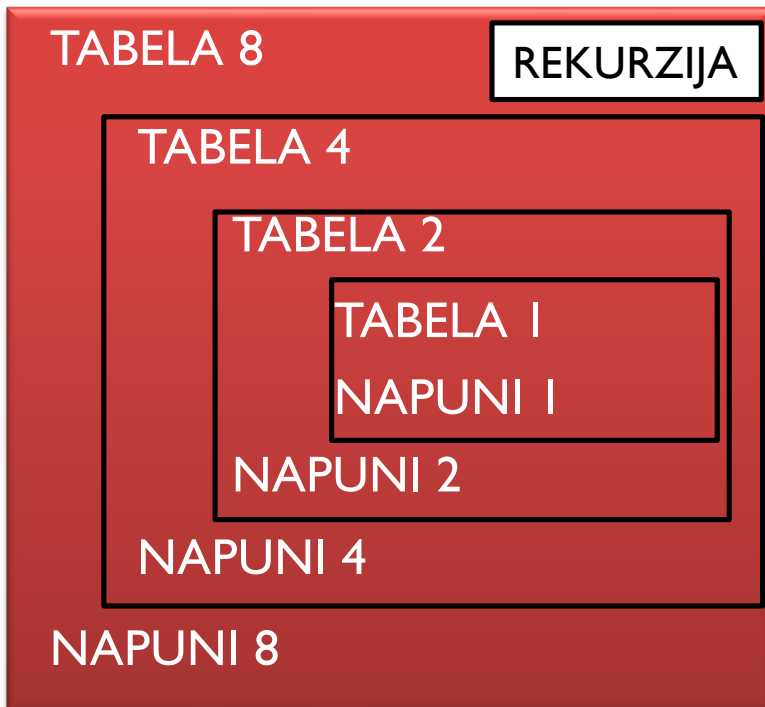
```
uslov -> (broj|parametar) (==|!=|>|<|>=|<=) (broj|parametar)
```

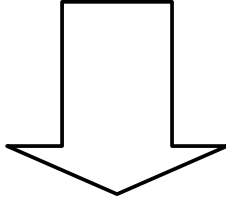
```
uslovna_direktiva -> nova_linija USLOVNO razmak uslov telo nova_linija KRAJ
```

```
telo -> {direktiva  
      |osnovna_naredba  
      |naredba_prebacivanja  
      |upravljačka_naredba  
      |makro_definicija  
      |makro_poziv  
      |uslovna_direktiva }
```

Uslovno asembliranje

TABELA	MAKRO	<code>stepen</code>
	<code>USLOVNO</code>	<code>stepen > 0</code>
	<code>TABELA</code>	<code>stepen/2</code>
	<code>NAPUNI</code>	<code>stepen</code>
	<code>KRAJ</code>	
	<code>KRAJ</code>	

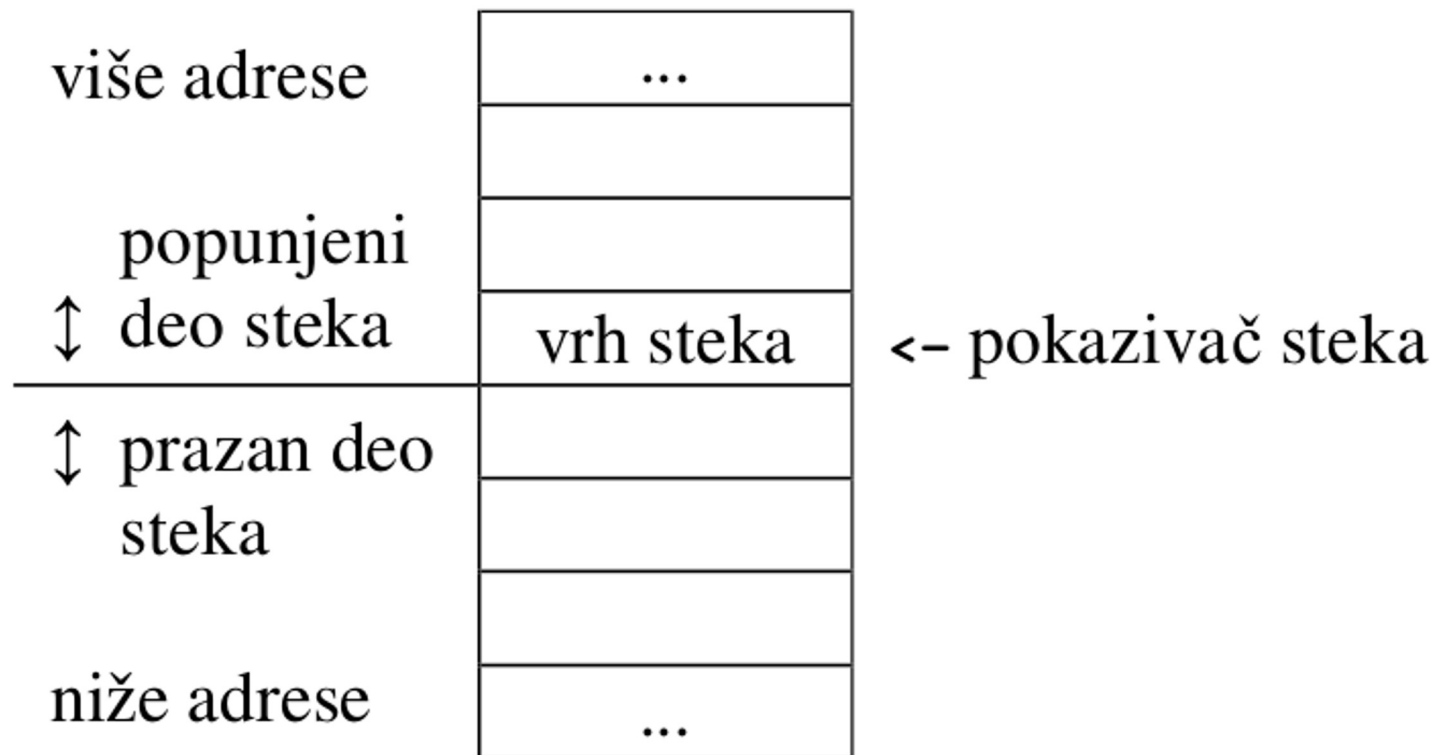


<code>stepeni:</code>	TABELA	8
		
<code>stepeni:</code>	NAPUNI	1
	NAPUNI	2
	NAPUNI	4
	NAPUNI	8

Stek

Stek

- Ugnježdjeni pozivi potprograma
- Prenošnje parametara, lokalne promenljive



Rukovanje stekom

STEK	MAKRO	veličina, pokazivač_steka
stek:	ZAUZMI	veličina
PRIPREMI_STEK	MAKRO	
	PREBACI_NR	\$stek, pokazivač_steka
	PREBACI_NR	\$veličina, %1
	SABERI	%1, pokazivač_steka
	KRAJ	
NA_STEK	MAKRO	registar
	ODBIJ_1	pokazivač_steka
	PREBACI_RP	registar, (pokazivač_steka)
	KRAJ	
SA_STEKA	MAKRO	registar
	PREBACI_PR	(pokazivač_steka), registar
	DODAJ_1	pokazivač_steka
	KRAJ	
	KRAJ	

Rukovanje stekom

STEK 0x100,%12

%12 – pokazivač steka



stek :	ZAUZMI	0x100
PRIPREMI_STEK	MAKRO	
	PREBACI_NR	\$stek,%12
	PREBACI_NR	\$0x100,%1
	SABERI	%1,%12
	KRAJ	
NA_STEK	MAKRO	registar
	ODBIJ_1	%12
	PREBACI_RP	registar, (%12)
	KRAJ	
SA_STEKA	MAKRO	registar
	PREBACI_PR	(%12),registar
	DODAJ_1	%12
	KRAJ	

Frejm

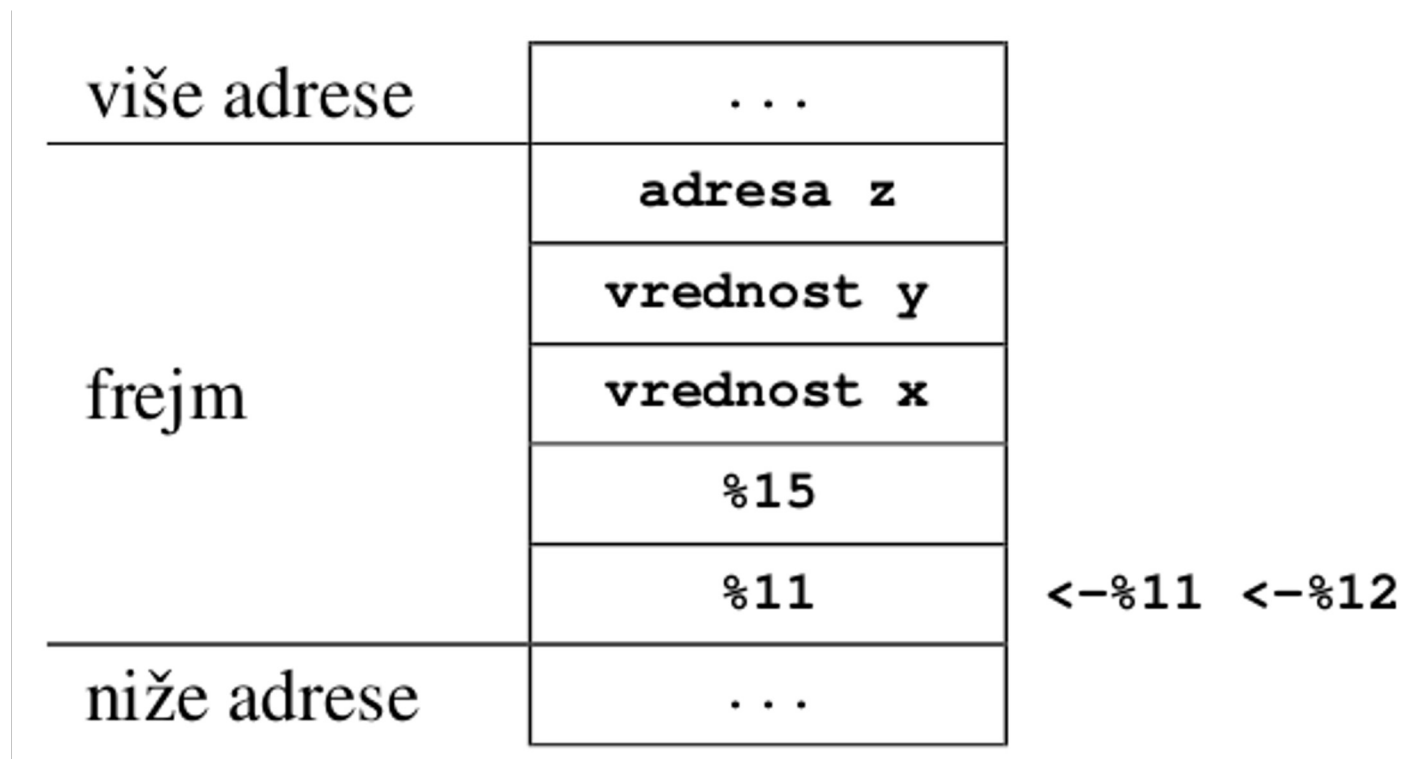
(Stek) Frejm

više adrese	...	
frejm	argument n	
	...	
	argument 2	
	argument 1	
	povratna adresa	
	pokazivač prethodnog frejma	<- pokazivač frejma
	lokalna promenljiva 1	
	...	
	lokalna promenljiva m	<- pokazivač steka
niže adrese	...	

PREBACI_IR	2 (%11) , %0
PREBACI_IR	3 (%11) , %1
UPOREDI	%1 , %0

%11 – pokazivač frejma
%12 – pokazivač steka

Primer: NZD sa stekom



Primer: NZD sa stekom

nzd(x, y, &z)

Stvaranje frejma:

x:	ZAUZMI	1
y:	ZAUZMI	1
z:	ZAUZMI	1
	...	
	PREBACI_NR	\$z, %0
	NA_STEK	%0
	PREBACI_DR	y, %0
	NA_STEK	%0
	PREBACI_DR	x, %0
	NA_STEK	%0
	NA_STEK	%15
	POZOVI	nzd
	SA_STEKA	%15
	PREBACI_NR	\$3, %0
	SABERI	%0, %12

Asemblerska procedura za NZD:

nzd:	NA_STEK	%11
	PREBACI_RR	%12, %11
	PREBACI_IR	2(%11), %0
	PREBACI_IR	3(%11), %1
ponovo:	UPOREDI	%1, %0
	SKOČI_ZA_==	kraj
	SKOČI_ZA_<	manje
veće:	ODUZMI	%1, %0
	SKOČI	ponovo
manje:	ODUZMI	%0, %1
	SKOČI	ponovo
kraj:	PREBACI_IR	4(%11), %1
	PREBACI_RP	%0, (%1)
	SA_STEKA	%11
	NATRAG	