

---

# Osnove rada u Unix okruženju

- `$gcc program.c` (**GNU c compiler**)
  - `$a.out`
  
  - `$gcc program.c -o novo_ime`
  - `$program.exe`
  
  - `$gcc program1.c program2.c ...`
  - `gcc` → `cc`
-

---

# Osnove rada u Unix okruženju, Osnove rada u Win okruženju: Dev-C++

## Primjer:

```
/******
```

Ovo je komentar

```
*****/
```

```
int main ()  
{  
    printf ("Dobrodosli na predavanje!\n");  
    return 0;  
}
```

---

---

# Osnovni tipovi podataka

- `char` – znakovni tip, `sizeof(char)=1` (bajt)
  - `int` – cjelobrojni tip, `sizeof(int)=4`
  - `float` – realni tip u jednostrukoj preciznosti, `sizeof(float)=4`
  - `double` – realni tip u dvostrukoj preciznosti, `sizeof(double)=8`
-

---

# Osnovni tipovi podataka (2)

- kvalifikatori: `short`, `long` i `unsigned`

## Primjer:

`short int ili short`

`unsigned int ili unsigned`

`unsigned long int ili unsigned long`

`sizeof(short) <= sizeof(int) <= sizeof(long)`

---

---

# Operandi

Operandi:       konstante

                  varijable

Atributi:       ime

                  vrijednost

                  tip

---

# Konstante

- cjelobrojne: 1, 200, -3589, 23u, 19999L
- oktalne: 01, 0235, -02456
- heksadecimalne: 0x1, 0X1, 0X1FFF, -0x23aa
- znakovne: 'a', 'A', '\n', '\b', '\\'
- konstantni znakovni nizovi (string): "Zagreb", "a"
- realne: 3., 337., -23., 3.e4, 3.E4, -3.e+4, .3e-4L  
3.f, 337.f, -342.f, 3.e4f, -3.039E+4f
- simboličke: `#define TRUE 1,`  
`#define Pi 3.141593`

# Konstante (2)

Primjer:

```
#include<stdio.h>

int main( ){
    const x=010;

    printf("x= %d",x);
    return 0;
}
```

---

# Varijable

Varijable su simbolička imena za lokacije u koje pohranjujemo vrijednosti.

Osnovni tipovi:

- numerički (cjelobrojni, realni,...)
  - znakovni
  - pokazivači – varijable koje sadrže adrese drugih varijabli
-



---

# Varijable (2)

Simboličko ime:

- počinje slovom
- maksimalno 255 znakova
- $a \neq A$
- slova, znamenke, \_
- ne smije biti rezervirana riječ

Primjer:

x, y2, rezultati\_mjerenja, Program\_V03

“x”, ad-c, extern, 3x

---

# Ključne riječi

- Pišu se malim slovima.
- Za prevodioca imaju posebno značenje.
- Prema ANSI standardu, C ima sljedeće 32 ključne riječi:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

---

# Deklaracija i inicijalizacija varijabli

```
tip ime; tip ime = konstantni_izraz;
```

## Primjer:

```
int a;
```

```
short x,y,z;
```

```
char d, x='\t';
```

```
int e=23+45, a,b;
```

```
double p, q=3.14;
```

---

---

# Enumeracija

```
enum ime {clan_1, clan_2, ..., clan_n};
```

## Primjer:

```
enum logical {false, true};
```

```
enum logical flag;
```

```
enum dan_u_tjednu {po, ut, sr, ce, pe} dan;
```

```
enum boje {plavo=-1, zuto, crveno, zeleno=0, sivo};
```

```
flag=true;
```

```
dan=ce;
```

---

---

# Promjena imena nekog tipa

```
typedef stari_tip novi_tip;
```

## Primjer:

```
typedef float real;
```

```
typedef enum logical boolean;
```

```
real x;
```

```
boolean flag;
```

---

---

# Operatori i izrazi - pregled

## Operatori:

- pridruživanja
- aritmetički
- relacijski
- logički
- nad bitovima

## Izrazi:

- imaju tip i vrijednost
  - kombinacija operandada i operatora
-

---

# Operatori pridruživanja: =

```
varijabla = izraz;
```

Primjer:

```
x=3;
```

```
y=3.+5.342;
```

```
a=a+1;
```

```
b=c=10; /* D→L */
```



---

# Pretvaranje tipova

Kada u aritmetičkom izrazu sudjeluju operandi različitih tipova, onda prije izračunavanja izraza dolazi do pretvaranja operanada u najprecizniji tip prisutan u izrazu; rezultat aritmetičkog izraza bit će tog tipa.

- Ako su operandi tipa `float` i `double`, onda će `float` biti pretvoren u `double` prije izračunavanja.
-



---

# Pretvaranje tipova (2)

- Ako je jedan od operanada realnog tipa (`float`, `double`) a drugi cjelobrojnog (`char`, `short`, `int`, ...), onda se cjelobrojni tip konvertira u realni.
  - Ako nema realnih operanada, onda se operandi tipa `char` i `short` pretvaraju u tip `int` i zatim se (ako je potrebno) pretvaraju u najprecizniji prisutni cjelobrojni tip.
-

---

# Pretvaranje tipova (3)

Kada u aritmetičkom izrazu sudjeluju operandi različitih tipova, onda prije izračunavanja izraza dolazi do pretvaranja operanada u najprecizniji tip prisutan u izrazu; rezultat aritmetičkog izraza bit će tog tipa.

- Ako su operandi tipa `float` i `double`, onda će `float` biti pretvoren u `double` prije izračunavanja.
-

# Aritmetički operatori

Operator	Značenje	Primjer:
-	unarni minus	$x = -y;$
*	množenje	$c = a * b;$
/	dijeljenje	$a = a / b;$
%	modulo	$ostatak = a \% b;$
+	zbrajanje	$b = a + b;$
-	oduzimanje	$x = b - a;$

---

# Prioriteti

- unarni:  $D \rightarrow L$
- multiplikativni (\*, /, %):  $L \rightarrow D$
- aditivni (+, -):  $L \rightarrow D$
- pridruživanje:  $D \rightarrow L$

## Primjer:

```
int x,y,z;
```

```
...
```

```
z=x/y*y+x%y;
```

---

# Operatori inkrementiranja i dekrementiranja

`x++ x=x+1`

```
int x,z;
```

```
x=3;
```

```
z=x++-2;
```



postfix

`x=4, z=1`

`x-- x=x-1`

```
int x,z;
```

```
x=3;
```

```
z=++x-2;
```



prefix

`x=4, z=2`

Nakon izvršavanja:



---

# Operatori inkrementiranja i dekrementiranja (2)

Primjer:

```
i=7;  
printf("i= %d\n", --i);  
printf("i= %d\n", i--);  
printf("i= %d\n", i);
```

