Konštanty, operátory, príkazy vetvenia

Základy procedurálneho programovania 1, 2020

Ing. Marek Galinski, PhD.

Opakovanie

2

Hello world!

```c
#include <stdio.h>

int main(int argc, const char * argv[]) {
    printf("Zjavne sa mi to kompiluje spravne - funguje to!\n");
    return 0;
}
```


3

Hello world!

- **Funkcia `main()`**
  - **Vždy musí byť v programe, je volaná ako prvá pri spustení program**
  - Má spravidla celočíselnú návratovú hodnotu (ale nie je to nutné)
  - Nemusí mať argumenty, ale môže mať
  - Tu nám vyplýva, že program v C pozostávajú z funkcií (min. 1)

4

Aritmetické výrazy

- **Priorita vyhodnocovania aritmetických výrazov**

Operátor(y)	Operácia(e)	Priorita
( )	zátvorky	Vyhodnotené ako prvé Vnorené zátvorky – najvnútornejšia najskôr Na rovnakej úrovni – zľava doprava
* / %	Násobenie, delenie, zvyšok po delení	Vyhodnotené ako druhé Na rovnakej úrovni – zľava doprava
* -	Pripočítanie, odpočítanie	Vyhodnotené ako tretie Na rovnakej úrovni – zľava doprava
=	priradenie	Vyhodnotené ako posledné

Terminálový vstup a výstup

- **Formátovaný vstup a výstup**

```
printf("Zadajte strany obdlznika: ");
scanf("%f %f", &a, &b);
printf("Stvorec so stranami: %f a %f ma: \n", a, b);
o = 2 * a + 2 * b;
s = a * b;
printf("- obvod %f\n- obsah %f\n", o, s);
```

Zapisovanie číselných konštánt

7

Celočíselné konštanty

- **Môžeme ich zapísať v rôznych číselných sústavách**

- **Desiatková** (dekadická) `15` `0` `1`
  - Zapisuje sa ako postupnosť číslic, na začiatku však nesmie byť nula (okrem hodnoty nula)
- **Osmičková** (oktalová) `017` `00` `01`
  - Číslica 0 nasledovaná osmičkovými číslicami (0-7)
- **Šestnástková** (hexadecimálna) `0xF` `0XF` `0x0` `0X1`
  - Reťazec 0x alebo 0X (znak nula) nasledovaný hexadecimálnymi číslicami (0-9,a-f,A-F)

Celočíselné konštanty

- **Typ konštantny môže byť určený implicitne alebo explicitne**

```
int hodnota = 52;
vyraz = hodnota + 14542;
vyraz = hodnota + 14542L;
vyraz = hodnota + 14542U;
vyraz = hodnota + 14542lu;
vyraz = hodnota + 14542LU;

vyraz = hodnota + -14542u;
```

<- TOTO JE NEZMYSEL

Reálne konštanty

- **Bodka môže byť viac menej kdekoľvek, no mení to význam**

```
float zlomok;

zlomok = 9 / 5;
zlomok = 9.0 / 5.0;

zlomok = 9. / 5.;

zlomok = .9 / .5;
```

```
float hodnota;

hodnota = 56.8;
hodnota = 568E-1;

hodnota = 5000000;
hodnota = 5e6;
```

Znakové konštanty

- **Znak uzatvorený v apostrofoch – `‘a’, ‘*’, ‘8’`**

- Hodnota zodpovedá ASCII hodnote
- Veľkosť konštanty je celočíselná, nie char

- Neviditeľné znaky
  - Zložené z troch oktalových číslic – **`‘\012’`**
  - **Escape character ‘\’, mení význam**
    - **`‘012’`** – nie je jeden znak
    - **`‘\012’`** – je jeden znak

ASCII tabuľka

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII tabuľka

riadiace znaky	0	-	31	
medzera	32	' '		
pomocné znaky	33	'!'	- 47	'/'
číslice	48	'0'	- 57	'9'
pomocné znaky	58	':'	- 64	'@'
veľké písmená	65	'A'	- 90	'Z'
malé písmená	97	'a'	- 122	'z'
pomocné znaky	123	'{'	- 126	'-'

Escape sekvencie

- **Niektoré escape sekvencie sa dajú zapísať aj znakovým ekvivalentom, nie len numerickým kódom.**

\n	0x0A	nový riadok (new line, line feed)
\r	0x0D	návrat na začiatok riadku (carriage return)
\f	0x0C	nová stránka (formfeed)
\t	0x09	tabulátor (tab)
\b	0x08	posun doľava (backspace)
\a	0x07	písknutie (allert)
\\	0x5C	spätné lomítko (backslash)
\'	0x2C	apostrof (single quote)
\0	0x00	nulový znak (null character)

Reťazcové konštanty

- **Reťazec znakov je uzatvorený do klasických úvodzoviek “ ”**
- **Napr: “Som FIITkar, toto je retazcova konstantna”**
- **Dajú sa spájať: “Som” “FIITkar, ” “toto je retazcova konstanta”**

```
printf("Dame si %s alebo rum?", "whisky");
```

Špeciálne znaky

- Dajú sa zapísať znaky \, “ a % keď vieme, že sú to špeciálne formátovacie znaky? Ako ich zapíšeme?

Špeciálne znaky

- **Dajú sa zapísať znaky \, “ a % keď vieme, že sú to špeciálne formátovacie znaky? Ako ich zapíšeme?**

```
char znak;

znak = '\\';
znak = '%%';
znak = '\"';
```

Formátovací reťazec

18

Formátovanie: výpis čísel

```
float pi = 3.1415;

printf = "Hodnota Pi je: %.2f", pi);
>> Hodnota Pi je: 3.14

printf = "Hodnota Pi je: %g", pi);
>> Hodnota Pi je: 3.1415
```


19

Formátovanie: výpis čísel

```
float  f = 3.14159265358979323384;
double d = 3.14159265358979323384;

printf("Pi: %.16f\n", f);
printf("Pi: %3.16lf\n", d);
printf("Pi: %5.2lf\n", d);
printf("Pi: %10.5lf\n", d);
printf("Pi: %g\n", d);
```


20

Konverzia malé/veľké písmeno

```
#include <stdio.h>

int main()
{
    int c;

    printf("Zadajte male pismeno: ");
    c = getchar();
    c = c - 'a' + 'A';

    printf("Velke pismeno: %c\n", c);
    return 0;
}
```


21

Špeciálne unárne operátory

- **Inkrement ++**
- **Dekrement –**

```
i++;
j--;
```

- **Prefix zápis - ++hodnota**
  - **Inkrementuje sa pred použitím**

```
++i;
--j;
```

- **Postfix zápis – hodnota--**
  - **Dekrementuje sa až po použití**

```
i--;
j++;
```


22

Operátory priradenia

```
55
56      x = x + vyraz;
57      x += vyraz;
58
59      x = x - vyraz;
60      x -= vyraz;
61
62      x *= vyraz;
63      x /= vyraz;
64      x %= vyraz;
65
```


23

Kombinované využitie

```
66
67    int i = 4, j = 2, k = 0;
68
69    ++i;
70    i++;
71    k = 5 * (j+1);
72    k = 5 * ++j;
73    k = 5 * j++;
74    i *= 5 * j--;
75    k += --j;
76
```


24

Príkazy vetvenia

`if, if-else`

25

Logické (Boolove) výrazy

- **Výrazy, kde sú aplikované logické operácie na relačné alebo aritmetické výrazy**
  - **Porovnávanie výrazov, hodnôt, …**
- **Výsledok je TRUE alebo FALSE**
  - **Lenže C ich nepozná, takže 0 alebo 1 (najčastejšie, nie nutne)**
  - **Používa sa typ int**

26

Logické (Boolove) výrazy

- **Operátory**
  - **Aritmetické a logické**

==	rovnosť
!=	nerovnosť
<	menší
<=	menší alebo rovný
>	väčší
>=	väčší alebo rovný

&&	logický súčin
||	logický súčet
!	negácia

27

Logické operátory

- && - logický súčin (AND, konjunkcia)
- ll - logický súčet (OR, disjunkcia)
- ! – negácia (NOT)

```
A && B
A || B
!A
```

A	B	!A	A && B	A || B
1	1	0	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	0

28

Logické operátory

```
int x = 10, y = 5;

(x == 10) // 1 (TRUE)
!(y < x)  // 0 (FALSE)
(x != 10) // 0 (FALSE)

(y <= x) && (y > 2)   // 1 (TRUE)
(x < 10) || (y == 20) // 0 (FALSE)
```


29

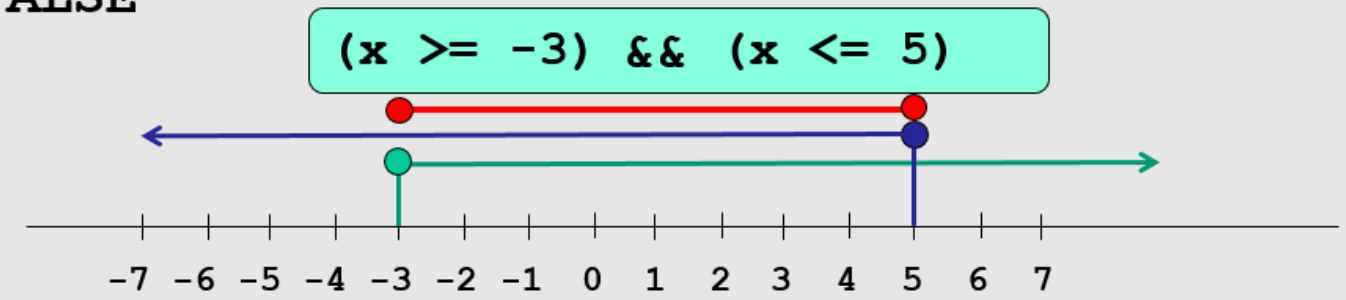
Logické operátory

```
// POZOR NA TOTO!

x = 10  //TOTO JE PRIRADENIE

x == 10 //TOTO JE POROVNANIE
```

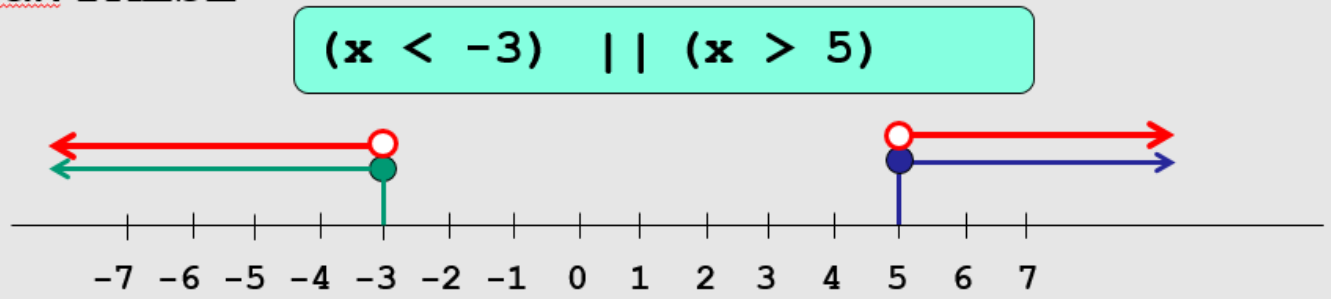

30

Zložené logické výrazy

- Podmienka je **TRUE**, ak číslo **x** je z intervalu <-3, 5>, inak **FALSE**



- Podmienka je **TRUE**, ak číslo **x nie** je z intervalu <-3, 5>, inak **FALSE**



31

Negácia výrazu

Negácia otáča logické operátory `&&` a `||`

```
!(A && B) = !A || !B
```

- `!((x>=-3) && (x<=5)) = (x<-3) || (x>5)`

```
!(A || B) = !A && !B
```

- `!((x%2) || (x>5)) = !(x%2) && (x<=5)`

32

Vyhodnocovanie logických výrazov

- **Výrazy sa vyhodnocujú zľava doprava**

```
A && B
```

  - **Ak A je TRUE, musíme zistiť či aj B je TRUE**
  - **Ak A je FALSE, nie je potrebné ďalej vyhodnocovať**

```
A || B
```

  - **Ak A je FALSE, musíme zistiť, či B je TRUE**
  - **Ak A je TRUE, nie je potrebné ďalej vyhodnocovať**

33

Vyhodnocovanie logických výrazov

- **Vyhodnocovanie v jazyku C končí v momente, kedy je známy výsledok vyhodnocovania**

```
(a != 0) && ((b / a) < 50)

(a != 0) || (z % 7)
```

Delenie nulou nenastane, program nespadne

34

Skrátené vyhodnocovanie logických výrazov

- **Ak je napr. nejaký výpočet príliš drahý, alebo by mohol ohroziť beh programu**

```
int toggle = 0;

if (toggle && kontrola(n)) {
    ...
}
```


35

Priority vyhodnocovania logických výrazov

- **Aritmetické operátory a operátory porovnania majú väčšiu prioritu ako logické operátory**

```
( (c >= 'A') && (c<='Z') )
(  c >= 'A'  &&  c<='Z'  )
```

- **Zátvorkami nikdy nič nepokazíte (ak ich date správne)**

36

Upozornenie

- **Nezamieňajte && za & a || za |**
  - **&& a || sú logické výrazy**
  - **& a | sú bitové operácie**

37

Príkaz `if`

- **Jeden z najčastejšie používaných príkazov**

```
if (podmienka)
    prikaz;

if (podmienka) {
    prikaz;
    prikaz;
}
```

```
if (x > 7)
    y = a * b;

if (b == 3 && a == 2) {
    c = a++;
    y -= 7;
}
```

- **Zátvorky sú nutné**

38

Príkaz `if-else`

- **Rozšírenie klasického `if` príkazu**

```
if (podmienka)
    prikaz;
else
    iny_prikaz;
```

```
if (podmienka) {
    prikaz;
    prikaz;
} else {
    iny_prikaz;
    iny_prikaz;
}
```


39

Príkaz `if-else`

Príkazy je možné vnárať do seba

```
if (a > 10) {
    if (b == 0) {
        if (pismeno == 'r') {
            printf("pismeno je r");
        } else {
            printf("Pozri kam som sa az dostal...");
        }
    }
} else {
    if (b == 5) {
        printf("Ahoj!");
    }
}
```


40

Príklady

41

Príklady

- **Zdrojové kódy príkladov sú dostupné aj na dokumentovom serveri v AIS**
  - Priečinok **Príklady k prednáškam**
    - **p02_ascii.c**
    - **p02_floatdouble.c**
    - **p02_specoperatory.c**
    - **p02_malevelke.c**
    - **p02_ifchar.c**

Footnotes

- **Prednáška je dostupná na YouTube:**

https://www.youtube.com/watch?v=QZ0ZxwskI5Q

V prednáške boli použité materiály zo slidov prednášok ZPrPr1 od Gabriely Grmanovej.

Q?

44