Aula 5 e meio: Explorando os tipos básicos

Professor(a): João Eduardo Montandon (103)
Virgínia Fernandes Mota (106)
jemaf.github.io
http://www.dcc.ufmg.br/∼virginiaferm

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA

Explorando os tipos básicos

- Tipos: int, char, float, double
- Modificadores: long, short, unsigned

Tipo Inteiro

- Comumente varia de 16 a 32 bits.
- Caso um valor maior do que o suportado seja assinalado à uma variável, ocorre **overflow** $\rightarrow$ o valor fica aparentemente sem sentido.
- Caso um valor menor do que o suportado seja assinalado à uma variável, ocorre **underflow** $\rightarrow$ o valor fica aparentemente sem sentido.
    - Em ambos os casos: Não há bits suficientes para representar o valor.

Tipo inteiro

- Dois modificadores opcionais: signed (-n a n) e unsigned (0 a 2n).
- Dois modificadores opcionais: short (8 bits) e long (32 bits) $\rightarrow$ a quantidade de bits depende do tamanho da palavra da máquina.
- Como um inteiro é armazenado no computador?
  - Notação binária: com sinal e sem sinal
  - Sinal e Magnitude , Complemento de 2

Tipo ponto flutuante

- Comumente varia de 32 a 80 bits.
- Também pode ocorrer overflow e underflow: Não há bits suficientes para representar o expoente.
- Eles não são armazenados na notação binária simples, mas sim em notação binária científica.
  - $(-1)^{sinal} \times mantissa \times 2^{expoente}$
- Tanto a mantissa quanto o expoente tem limites!
- float $<$ double $<$ long double.

Trabalhando com números - Funções Matemáticas

- Com a biblioteca **math.h** (cmath) podemos encontrar facilmente funções para calcular potências, raiz quadrada, funções trigonométricas para cálculos que envolvem seno, cosseno e tangente, além de constantes para números irracionais como, por exemplo, π (M_PI) e $\sqrt{2}$ (M_SQRT2).
- Trigonométricas:
  - *sin*(): Retorna o valor do seno. Recebe como argumento o valor dos graus em double.
  - *cos*(): Retorna o valor do co-seno. Recebe como argumento o valor dos graus em double.
  - *tan*(): Retorna o valor da tangente. Recebe como argumento o valor dos graus em double.

Trabalhando com números - Funções Matemáticas

- Logarítmicas:
  - *log*(): Retorna o valor do logaritmo na base 2. Exige um argumento do tipo double.
  - *log*10(): Retorna o valor do logaritmo na base 10. Exige um argumento do tipo double.
- Potências:
  - *pow*(): Retorna o valor da base elevada ao expoente. Recebe dois argumentos do tipo double, o primeiro é a base e o segundo o expoente. Por exemplo: se quisermos saber o resultado da operação 2^{10}, faríamos pow(2, 10).
  - *sqrt*(): Retorna o valor da raiz quadrada. Recebe como argumento um double do qual ele deve extrair a raiz.

Trabalhando com números - Funções Matemáticas

- Arredondamento:
  - *ceil*(): Retorna o primeiro float sem casas decimais acima. Recebe um float como argumento. Exemplo: ceil (45.98561) resultaria em 46.
  - *floor*(): Retorna o primeiro float sem casas decimais abaixo. Recebe um float como argumento. Exemplo: floor (45.98561) resultaria em 45.

Tipo caracter

- Cada variável possui 8 bits.
- Cada combinacão de bits representa um caracter.

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

Trabalhando com caracteres

- Dentre as funções encontradas da biblioteca **ctype.h** (cctype) há aquelas que modificam o estado da letra (maiúsculas e minúsculas) e até mesmo funções que servem para descobrir se o que foi digitado é um ponto, vírgula, número, espaço, etc.
- Não é necessário colocá-la como arquivo de cabeçalho.
- Algumas funções:
  - toupper(): esta função recebe um argumento que deve ser um caractere e retorna o caractere correspondente em formato maiúsculo, se o caractere já for maiúsculo, a função não o modifica.
  - tolower(): esta função recebe um argumento que deve ser um caractere e retorna o caractere correspondente em formato minúsculo, se o caractere já for minúsculo, a função não o modifica.

Trabalhando com caracteres

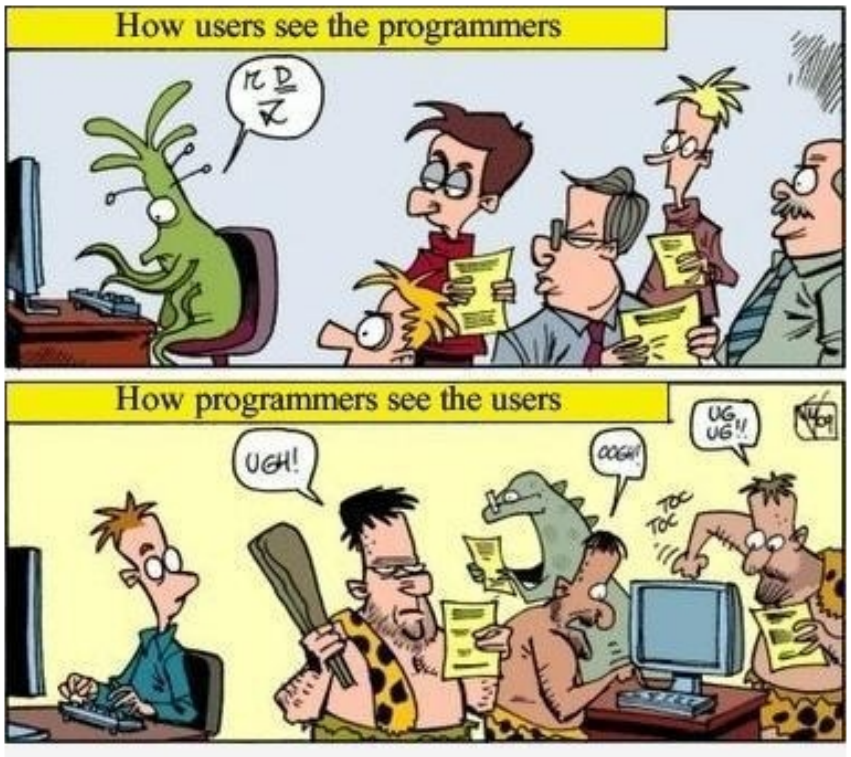
- isalnum(): verifica se o caractere ou inteiro passado como parâmetro é alfanumérico. Isso inclui todos os números e as letras do alfabeto, tanto maiúsculas quanto minúsculas.
- isalpha(): verifica se o caractere ou inteiro passado como parâmetro é alfabético. Isso inclui todas as letras do alfabeto, tanto maiúsculas quanto minúsculas.
- isdigit(): verifica se o caractere ou inteiro passado como parâmetro é um digito. Isso inclui todos os números.
- ispunct(): verifica se o caractere ou inteiro passado como parâmetro é uma pontuação. Isso inclui qualquer tipo de pontuação. Porém, não é capaz de verificar se uma letra é acentuada.
- isspace(): verifica se o caractere ou inteiro passado como parâmetro é um espaço em branco.

Trabalhando com caracteres

- islower(): verifica se o caractere ou inteiro passado como parâmetro é uma letra minúscula
- isupper(): verifica se o caractere ou inteiro passado como parâmetro é uma letra maiúscula
- iscntrl(): verifica se o caractere ou inteiro passado como parâmetro é um caractere de comando. Isso inclui CTRL, ALT, ENTER, BACKSPACE, etc.
- isxdigit(): verifica se o caractere ou inteiro passado como parâmetro é compatível com um número hexadecimal. Isso inclui todos os número (0 - 9) e qualquer letra entre A e F (não importa se minúsculo ou maiúsculo).

Sites interessantes!

- Dúvida sobre alguma função ou biblioteca do C?
  http://www.tiexpert.net/
  http://www.cplusplus.com/



Resolução da prova

E o que vocês acharam da prova?

Na próxima aula...

Subrotinas