Семинар №6

ФАКИ 2017

Бирюков В. А.

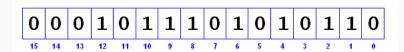
October 3, 2017

Переменные

Переменные

- В языке С все переменные нужно объявить перед использованием
- При объявлении выделяется память под переменную
- Области видимости переменной
- Название переменной может содержать латинские буквы, цифры и __
- Название переменной не может начинаться с цифры

Целочисленные типы



Число бит на тип зависит от компилятора. Обычные значения такие:

Название типа	Число бит	Макс. значения
char	8	0255
short	16	-3276832767
int	32	$-2 \cdot 10^9+ 2 \cdot 10^9$
long	32	$-2 \cdot 10^9+ 2 \cdot 10^9$
long long	64	$-2^{64}+2^{64} - 1$

Беззнаковые целочисленные типы

Число бит на тип зависит от компилятора. Обычные значения такие:

Название типа	Число бит	Макс. значения
unsigned short	16	065535
unsigned int	32	$0 + 4 \cdot 10^9$
unsigned long	32	$0 + 4 \cdot 10^9$
unsigned long long	64	$0+2^{65}-1$

sizeof() – размер файла в байтах

Типы чисел с плавающей точкой



Название типа	Число бит	Макс. значения
float	32	$10^{-38}10^{+38}$
double	64	$10^{-308}10^{+308}$

Обычно используется double, так как float может недостаточно точен

Вывод в stdout. Функция printf.

printf(строка форматирования, пер1, пер2, ...)

Обозначение	Типы	Пример
d или i	Целочисленные типы	392
f	Типы с плавающей точкой	392.5
е	Научная нотация	3.9265e+2
С	Символ	а
S	Строка	HelloMipt!

Приоритет операторов

- (), []
- ++, -, +, -(унарные), sizeof
- *, /, %
- +, -
- >,<,<=,>=
- ==, !=
- &, |, &&, ||
- **③** =, +=, и т.д.

Приоритет операторов С подробнее:

ru.cppreference.com/w/c/language/operator_precedence

Управляющие конструкции

Базовые управляющие конструкции

Цикл while

```
int i = 0;
while (i < 3)
{
    i++;
    printf("%d ", i);
}</pre>
```

Напечатает 1 2 3

Базовые управляющие конструкции

Цикл do while

```
int i = 0;

do
{
    i++;
    printf("%d ", i);
} while (i < 3);</pre>
```

Напечатает 1 2 3

Базовые управляющие конструкции

Цикл for

```
for (int i = 0; i < 3; ++i) {
  printf("%d ", i);
}</pre>
```

Напечатает 1 2 3

Управляющие конструкции Onepatop break

```
for(int i = 0; i < 10; ++i){
  if(i == 6){
    break;
  }
  printf("%d ", i);
}</pre>
```

Управляющие конструкции Oneparop continue

```
for(int i = 0; i < 10; ++i){
  if(i == 6){
    continue;
  }
  printf("%d ", i);
}</pre>
```

Массивы и строки

Массивы Примеры

Объявление:

```
int array[10];
float average_temperature[12];
```

Доступ к элементу:

Нумерация в массиве начинается с 0

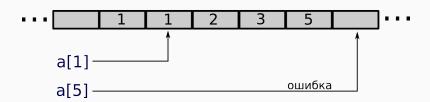
```
printf("%d\n", array[9]);
average_temperature[2] = 5.2;
```

```
int array[10];
for (int i = 0; i < 10; ++i) {
    array[i] = /* something */;
}</pre>
```

Или так:

```
int array[5] = \{1, 1, 2, 3, 5\};
```

int
$$a[5] = \{1, 1, 2, 3, 5\};$$



ASCII TABLE

0	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
2 2 START OF TEXT 34 22 " 66 42 8 98 62 b 3 3 SEND OF TEXT 35 23 # 67 43 C 99 63 c 4 4 4 EURD OF TRAINSUSSION] 36 24 5 68 44 D 100 64 d 5 5 SENDINY 37 25 % 69 45 E 101 65 e 6 6 6 IACKNOWLEDGE 38 26 6 70 46 F 102 66 f 7 7 BELLI 39 27 17 14 74 74 76 103 67 g 8 8 BACKSPACE 40 28 (0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	•
3 SEND OF TEXT 35 23 8 67 43 C 99 63 C 4 LEND OF TRANSMISSION 36 24 5 67 43 D 100 64 d 5 S LENCURY 37 25 % 69 45 E 101 65 e 6 6 ACKNOWLEDGE 38 26 6 70 46 F 102 66 e 7 7 RELL 39 27 71 47 G 103 67 g 8 8 REACKSPACE 40 28 72 48 H 104 68 h 9 9 HORIZONTAL TAB 41 29) 73 49 1 105 69 i 10 A LINE FEED 42 2A 74 4A J 106 6A j 11 B VERTICAL TAB 43 2B + 75 4B K 107 6B k 12 C FORM FEED 44 2C 76 4C L 108 6C I 13 D CARRAIGAE RETURN 45 2D - 77 4D M 109 6D m 14 E SHIFT OUT 46 2E 78 4E N 110 6E n 15 F SHIFT IN 47 2F 7 79 4F 0 111 6F 0 16 10 (DATA LINK ESCAPE 48 30 0 80 50 P 112 70 P 17 11 (DEVICE CONTROL 2) 50 32 2 82 52 R 114 72 r 19 13 (DEVICE CONTROL 2) 51 33 38 35 35 15 73 s 20 14 (DEVICE CONTROL 3) 51 33 35 58 55 V 118 76 V 22 16 (SYNCHRONOUS INCE) 53 36 6 86 56 V 118 76 V 23 17 (EIKO OF TRANS BLOCK) 55 37 7 87 57 W 119 77 V 24 18 (CANCELE) 59 38 88 88 88 88 88 88 X 120 78 X 25 19 (EIKO OF TRANS BLOCK) 56 38 88 88 88 X 120 78 X 26 1A (SUSTITUTE) 58 3A 90 93 50 Y 121 79 Y 26 1A (SUSTITUTE) 58 3A 90 93 50 Y 121 79 Y 26 1A (SUSTITUTE) 58 3A 90 93 50 Y 121 79 Y 27 1B (ESCAPE) 59 38 91 58 [123 78 6 78 78 78 78 78 7	1	1	[START OF HEADING]	33	21	1	65	41	A	97	61	a
4 LEND OF TRANSMISSION 36 24 5 68 44 D 100 64 d d 5 5 EINQUIRY 37 25 % 68 44 5 D 101 65 e 6 6 6 ACKNOWLEDGE 38 26 6 70 46 F 102 66 f 7 7 BELLI 39 27 7 7 47 7 105 10 39 27 7 7 47 7 105 10 39 27 7 7 48 H 104 68 h 9 9 IRONZONTALTAB 41 29 3 49 H 105 69 i 10 64 j 11 8 IVERTICAL TAB 43 28 + 75 48 K 107 68 k 12 C FORM FEED) 44 22 2A + 74 4A J 106 6A j J 11 B IVERTICAL TAB 43 28 + 75 48 K 107 68 k 12 C FORM FEED) 44 20 7 7 4D M 109 6D m 14 E SHIFT OUT] 46 2E 7 77 4D M 109 6D m 15 F ISHIFT IN 47 27 7 79 4F O 111 6F o 17 11 (DEVICE CONTROL I) 49 31 1 81 51 Q 113 71 q 18 12 (DEVICE CONTROL I) 50 32 28 28 25 28 27 27 7 27 7 27 7 28 27 27	2	2	[START OF TEXT]	34	22		66	42	В	98	62	b
5 S [ENQUINY] 37 25 % 69 45 E 101 65 e 6 6 ALAKON/MEDGEJ 38 26 6 70 46 F 102 66 f 7 T [BELL] 39 27 ' 71 47 G 103 67 g 8 8 [BACKSPACE] 40 28 (72 48 H 104 68 h 9 HORIZONTAL TAB] 41 29) 73 49 I 105 69 i 10 A [LINE FEED] 42 2A " 74 4A J 106 6A j 11 B [VERTAL TAB] 43 2B + 75 4B K 107 6B k 12 C [FORM FEED] 44 2C , 76 4C L 108 6C <td< td=""><td>3</td><td>3</td><td>[END OF TEXT]</td><td>35</td><td>23</td><td>#</td><td>67</td><td>43</td><td>C</td><td>99</td><td>63</td><td>c</td></td<>	3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
6 6 ACKNOWLEOGE 38 26 & 70 46 F 102 66 f 77 7 18EL 39 27 71 1 40 68 h 103 67 g 8 8 8 18ACKSPACE 40 28 (72 48 H 104 68 h 105 69 i 100 A 100 1 100	4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
7	5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	е
8 8 [BACKSPACE] 40 28 (72 48 H 104 68 n 9 9 HORDIONIAL TAB] 41 29) 72 48 H 104 68 n 10 A ILINE FEED] 42 2A * 74 4A J 106 6A j 11 B IVERTICAL TAB] 43 28 + 74 4A J 106 6A j 12 C IFORM FEED] 44 2C , 76 4C L 108 6C I 13 D ICARRAGE RETURIJ 45 2D , 76 4C L 108 6C I 14 E ISHIFLOUTI 46 2E , 78 4E N 110 6E n 15 F [SHIFLOUTI 47 2F / 79 4F	6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
9 HORIZONTAL TAB 11 29 73 49 105 69 1 10 A	7	7	[BELL]	39	27		71	47	G	103	67	q
10	8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
	9	9	[HORIZONTAL TAB]	41	29)	73	49	1	105	69	i e
12 C	10	Α	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
13	11	В	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
14 E	12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	1
15	13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
10	14	E	[SHIFT OUT]	46	2E		78	4E	N	110	6E	n
11	15	F	[SHIFT IN]	47	2F	1	79	4F	0	111	6F	0
18	16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	р
19	17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
10	18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r e
15	19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	S
22 16 [SYNCHRONOUS DILE] 54 36 6 86 56 V 118 76 V 23 17 [ENGO FTRANS, BLOCK] 55 37 7 87 57 W 119 77 W 24 18 [CANCEL] 56 38 8 8 8 85 58 X 120 78 X 25 19 [END OF MEDIUM] 57 39 9 89 59 Y 121 79 Y 26 1A [SUBSTRUTE] 58 3A : 90 5A Z 122 7A Z 27 1B [ESCAPE] 59 3B ; 91 5B [123 7B { 124 7C 129 1D [GROUP SEPARATOR] 61 3D = 93 5D 1 125 7D } 30 1E [RECORD SEPARATOR] 61 3D = 93 5D 1 125 7D } 30 1E [RECORD SEPARATOR] 62 3E > 94 5E 126 7E ~	20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
23 17 [ENG OF TRANS, BLOCK] 55 37 7 87 87 57 W 119 77 W 24 18 [CANCEL] 556 38 8 88 58 X 120 78 X 25 19 [END OF MEDIUM] 57 39 9 89 59 Y 121 79 Y 26 1A [SUBSTITUTE] 58 3A : 90 5A Z 122 7A Z 27 1B [ESCAPE] 59 3B ; 91 5B [123 7B { 28 1C [FILE SEPARATOR] 60 3C < 92 5C \ 124 7C 29 1D [GROUP SEPARATOR] 61 3D = 93 5D] 125 7D } 30 1E [RECORD SEPARATOR] 62 3E > 94 5E 6 126 7E ~	21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
24	22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
25	23	17	[ENG OF TRANS. BLOCK]		37	7	87	57	w	119	77	w
26 1A [SUBSTITUTE] 58 3A : 90 5A Z 122 7A Z 27 1B [ESCAPE] 59 3B ; 91 5B [123 7B { 228 1C [FILE SEPARTOR] 60 3C < 92 5C \ 124 7C 29 1D [GROUP SEPARTOR] 61 3D = 93 5D 1 125 7D } 30 1E [RECOR SEPARTOR] 62 3E > 94 5E \ 126 7E \ 7E	24	18	[CANCEL]		38	8	88	58	X	120	78	x
27 1B		19	[END OF MEDIUM]		39	9	89	59		121	79	у
28 1C FILE SEPARATOR 60 3C < 92 5C 124 7C 29 1D GROUP SEPARATOR 61 3D = 93 5D 1 125 7D 30 1E RECORD SEPARATOR 62 3E > 94 5E 126 7E ~	26	1A	[SUBSTITUTE]		3A	:	90	5A	Z	122	7A	z
29 1D [GROUP SEPARATOR] 61 3D = 93 5D 1 125 7D 3 30 1E [RECORD SEPARATOR] 62 3E > 94 5E ^ 126 7E ~	27	1B	[ESCAPE]		3B	;	91	5B	[123	7B	{
30 1E (RECORD SEPARATOR) 62 3E > 94 5E ^ 126 7E ~		1C				<		5C	\			1
	29	1D	[GROUP SEPARATOR]		3D	=	93	5D	1	125	7D	}
31 1F [UNIT SEPARATOR] 63 3F ? 95 5F 127 7F [DEL]	30	1E	[RECORD SEPARATOR]			>		5E	^	126	7E	~
	31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

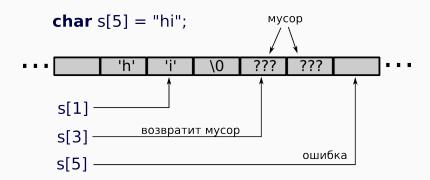
```
int string[10];
for (int i = 0; i < 10; ++i) {
    string[i] = /* something */;
}</pre>
```

Или так:

Или так:

```
int string[10] = "hello";
```

Строки Строки в памяти



Функции для работы со строками

Чтение строки:

```
char string[10];
scanf("%10s", string);
```

Длина строки:

```
#include <string.h>
char string[10] = "hello";
int n = strlen(string);
```

Функции для работы со строками

```
#include <string.h>
char s1[10] = "hi";
char s2[10] = "world";
```

Копирование строки s2 в строку s1:

```
strcpy(s1, s2);
```

Конкатенация строки s2 в строку s1:

```
strcat(s1, s2);
```

Функций

Объявление функций (прототипы функций)

- Как и переменная, функция также должна объявлена перед использованием
- Определение функции называется прототипом функции
- Пример прототипа:

```
int sum (int a, int b)
или:
int sum (int , int)
```

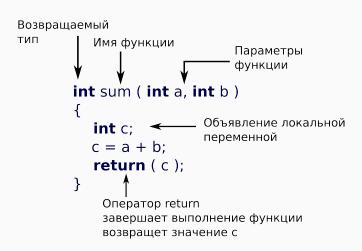
Вызов функций

• Примеры вызова функции:

Функция, определяемая пользователем:

Библиотечные функции:

Определение функции



```
int sum(int a, int b); ← Прототип функции
int main()
                          —— Вызов функции
  printf("%d\n", sum(5, 4));
  return 0;
int sum(int a, int b)
                              Определение
                              функции
  return a + b;
```

```
int sum(int a, int b)
                                Прототип и
{
                                определение
  return a + b;
                                функции
int main()
                                Вызов функции
  printf("%d\n", sum(5, 4));
  return 0;
```

Области видимости переменных

Области видимости переменных

- Область видимости область программы, в пределах которой имя некоторой переменной продолжает быть связанным с этой переменной и возвращать её значение.
- Глобальная переменная объявляются вне всех функций и доступны отовсюду
- Локальная переменная объявляются внутри блока и недоступны вне его

Области видимости переменных

Функция определяет собственную (локальную) область видимости, куда входят:

- Глобальные переменные
- 2 Входные параметры
- Переменные, которые объявляются в теле самой функции

Материалы для подготовки

- Керниган, Ричи. Язык С
- CS50 (перевод лекций есть в vk)
- http://style.vdi.mipt.ru/ тренировка перед контрольной работой