

# Общие Сведения о языке СИ

лекция 2

# План лекции

- Общие сведения
- Лексика языка Си
- Форма Бекуса-Наура описания синтаксиса

# Общие сведения о языке Си

- Dennis Ritchie (1941-2011)
  - Язык для разработки ОС UNIX
  - 1969-1973, Bell Laboratories, США
- Стандарты
  - ANSI (C89)
  - C99
    - //, описание переменных не в начале блока, массивы переменной длины
  - C11
    - Параллелизм, полиморфизм





# Лексика языка Си

- Программа на Си -- одна или несколько *единиц компиляции* (файлов)
- Стадии работы компилятора
  - Формирование *лексем* (в том числе работа препроцессора)
  - Синтаксический анализ
  - Семантический анализ
  - Оптимизация
  - Генерация кода

# Лексемы языка Си

- Символы-разделители
- Идентификаторы
- Ключевые слова
- Константы, строковые литералы
- Символы операций и скобки

# Символы-разделители языка Си

- Пробелы
- Символы табуляции
- Переводы строк
- Комментарии
  - C89: от /\* до \*/
  - C99: C89 и от // до конца строки
  - Эквивалентно одному пробелу

# Идентификаторы языка Си

- Последовательность букв и цифр, начинающаяся с буквы
  - Знак подчеркивания `_` является буквой
- Идентификатор функции (переменной), которую можно вызвать (использовать) из другой единицы компиляции, называется *внешним* идентификатором
  - Значимыми являются не менее 6 первых символов
  - Верхний и нижний регистр могут не различаться
- Остальные идентификаторы называются *внутренними*
  - Значимыми являются не менее 31 символа
  - Верхний и нижний регистр различаются



# Ключевые слова языка Си

- ANSI:
  - auto break case char const continue default do
  - double else enum extern float for goto if
  - int long register return short signed sizeof static
  - struct switch typedef union unsigned void volatile while
- C99:
  - `_Bool` `_Complex` inline restrict
- C11:
  - `_Alignas` `alignof` `_Atomic` `_Generic` `_Noreturn`  
`_Thread_local`

# Константы языка Си

- Целые
- Символьные
- С плавающей точкой
- Константы перечислимых типов
- Строковые литералы

# Целые константы

- Константа записывается в 8-, 10- или 16-ричной системе счисления и может иметь суффиксы `u` (или `U`) и/или `l` (или `L`)
- 8-ричная запись состоит из цифр и начинается с цифры `0`
- 10-тичная запись состоит из цифр и начинается не с цифры `0`
- 16-ричная запись состоит из префикса `0x` и послед. цифр `0-9` и букв `a-f`
  - 16-ричные цифры со значения от `10` до `15` обозначаются буквами от `a-f`
  - Регистр не учитывается
- Константа получает тип с наименьшим диапазоном, содержащим значение константы
  - 10-тичная без суффикса – первый из `int`, `long int`, `unsigned long int`
  - 8- и 16-ричная без суффикса – первый из `int`, `unsigned int`, `long int`, `unsigned long int`
  - С суффиксом `u` или `U` -- первый из `unsigned int`, `unsigned long int`
  - С суффиксом `l` или `L` -- первый из `long int`, `unsigned long int`
  - С суффиксом `ul` или `UL` имеет тип `unsigned long int`

# СИМВОЛЬНЫЕ КОНСТАНТЫ

- Необязательный префикс L и один или нескольких символов в кавычках ' (например 'x' или L'x')
  - В кавычки ' нельзя брать одну кавычку ' или конец строки
  - Значением константы с одним символом внутри является код этого символа в кодировке, принятой на данной машине
  - Значение константы с несколькими символами может зависеть от реализации
- Константа без префикса имеет тип char
- Константа с префиксом L имеет тип wchar\_t (описан в stddef.h)

Escape-последовательность	Запись	Escape-последовательность	Запись
новая строка	<code>\n</code>	обратная наклонная черта	<code>\\</code>
горизонтальная табуляция	<code>\t</code>	Знак вопроса	<code>\?</code>
вертикальная табуляция	<code>\v</code>	одионочная кавычка (single quote) \'	<code>\'</code>
возврат на шаг	<code>\b</code>	двойная кавычка (double quote) \"	<code>\"</code>
возврат каретки	<code>\r</code>	восьмеричный код ooo	<code>\ooo</code>
перевод страницы	<code>\f</code>	шестнадцатеричный код hh	<code>\xhh</code>
сигнал "звонок"	<code>\a</code>		

# Константы с плавающей точкой

- Целая часть, десятичная точка, дробная часть, e или E, и порядок (возможно, со знаком), и, возможно, суффикс f, F, l или L
- Целая, дробная часть и порядок -- последовательности цифр
- Целая часть или дробная часть (но не обе вместе) могут отсутствовать
- Десятичная точка или E с порядком (но не обе вместе) могут отсутствовать
- Тип определяется суффиксом
  - F или f -- тип float
  - L или l -- тип long double
  - Без суффикса – тип double

# Константы перечислимых ТИПОВ

- Идентификаторы, объявленные как элементы перечисления enum
- Имеют тип int
- Значения определяются внутри enum

# Строковые литералы

- Необязательный префикс L и последовательность символов, в двойных кавычках (например, "... " или L"... " )
  - В двойные кавычки нельзя брать одну двойную кавычку или конец строки
  - В строках можно использовать те же escape-последовательности, что и в символьных константах
- Константа без префикса имеет тип массив char
- Константа с префиксом L имеет тип массив wchar\_t
- Значение строки хранится в памяти глобальных переменных (static) и инициализируется заданными символами, за которыми идет '\0'
- Поведение программы, пытающейся изменить строковый литерал, не определено
- Написанные рядом строковые литералы объединяются в одну строку
- После любой конкатенации к строке добавляется символ '\0'
- Конкатенация строк с префиксом и без префикса не определена

# Символы операций и скобки

- Скобки
  - []{}()
- Унарные
  - -- ++ ! ~ & \* + -
- Бинарные
  - && || << >> -> ., & ^ | \* + - / %
  - = == < > <= >= != += -= /= %= <<= >>= &= |= ^=
- Тернарные
  - ?:
- Другое
  - ... ;



# Перед делением на лексемы

Три-граф	ASCII	Три-граф	ASCII	Три-граф	ASCII
??=	#	??(	[	??<	{
??/	\	??)	]	??>	}
??'	^	??!	;	??-	~

- Строка текста программы, заканчивающаяся обратной наклонной чертой \, соединяется со следующей

# Форма Бекуса-Наура описания синтаксиса формальных

## ЯЗЫКОВ

- Терминальные символы
- Нетерминальные символы
- Правила вида
  - $\langle \text{нетерм.символ} \rangle ::= \langle \text{посл.симв.1} \rangle$ 
    - |  $\langle \text{посл.симв.2} \rangle$
    - | ...
    - |  $\langle \text{посл.симв.n} \rangle$

# Пример БНФ № 1

- $\langle \text{цифра} \rangle ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'$
- $\langle \text{знак} \rangle ::= '+' | '-' |$
- $\langle \text{число без знака} \rangle ::= \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \langle \text{число без знака} \rangle$
- $\langle \text{число} \rangle ::= \langle \text{знак} \rangle \langle \text{число без знака} \rangle$
- Множество строк, которые описывает  $\langle \text{число} \rangle$ :
  - 0, 1, ..., 9, +0, +1, ..., +9, -0, -1, ..., -9, 00, 01, ..., 09, +00, +01, ..., +09, -00, -01, ..., -09, ...

## Пример БНФ № 2

- Какое множество строк описывает  $\langle \text{ппс} \rangle$  ?
- $\langle \text{ппс} \rangle ::= | '(\langle \text{ппс} \rangle)'$  |  $\langle \text{ппс} \rangle \langle \text{ппс} \rangle$

# Расширенная БНФ

- [**<посл.симв.>**]
  - Необязательная  
последовательность символов
- {**<посл.симв.>**}
- Повторение последовательности  
символов

# Заключение

- Общие сведения о языке Си
  - Dennis Ritchie, 1973, Bell Laboratories, САСШ
- Лексика языка Си
  - Единица компиляции
  - Стадии работы компилятора
  - Лексемы
    - Символы-разделители
    - Идентификаторы
    - Ключевые слова
    - Константы, строковые литералы
    - Символы операций и скобки
- Форма Бекуса-Наура описания синтаксиса

