Logging in From a Linux System or Localhost
3 days 23 hours ago
Auto response
4 days 5 hours ago
Re: final issue.
4 days 21 hours ago
Re: Rocky, thank you for this
4 days 22 hours ago

Learning C/C++ Step-By-Step

Submitted by ganesh35 (Contact Author) (Forums) on Wed, 2009-01-07 18:03. :: C/C++

Learning C/C++ Step-By-Step

01. Step-by-Step C/C++ --- Introduction

Many people are really interested in learning and implementing C/C++ programs on their favorite platforms like DOS/Windows or Linux. If you are the one looking for a step-by-step guide to get started, this tutorial is for you. Let me know your comments on my tiny attempt to serve the community.

Contents

Introduction to Language & Expressions

What is C?

C is a compiler based programming language supports both high level and low level statements to interact directly with the hardware.

Development of C Language

The C programming language evolved from a succession of programming languages developed at Bell Laboratories in early 1970s. It was not until the late 1970s that this programming language began to gain widespread popularity and support. This was because until that time C compilers were not readily available for commercial use outside of Bell Laboratories.

The C language was the outcome of Dennis Ritchie's work on a project in Bell Laboratories, to invent a suitable high level language for writing an Operating

System which manages the input and output devices of a computer, allocates its storage and schedules the running of other programs.

UNIX operating system is written in the C language. Hence the Unix Operating system has C as its standard programming language. In fact over 90% of the operating system itself is written in the C language. So originally C language was designed and implemented on the Unix Operating System.

C as a general purpose Language

C is a high level, procedural/structured, and general purpose programming language and resembles few other high level languages such as Fortran, Pascal, and PL/1. However, we cannot call the C language as a "Purely High Level Language".

C stands somewhere between the high-level languages meant for carrying on special activities and the low level languages such as assembly language of a machine because of some features like "System Independence", "Limited Data Type", "High Flexibility", it is considered as a powerful language C has also become popular because of its portability across systems.

History of C

Year	Language	Developed by	Remarks
1960	ALGOL	International Committee	Too general, Too abstract
1963	CPL	Cambridge University	Hard to learn, Difficult to implement
1967	BCPL	Martin Richards	Could deal with only specific problems
1970	B	Ken Thompson AT & T Bell Labs	Could deal with only specific problems
1972	C	Dennis Ritchie AT & T Bell Labs	Lost generality of BCPL and B restored
Early 80's	C++	Bjarne Stroustrup AT & T	Introduces OOPs to C.

Features of C

- Simple, versatile, general purpose language
- Programs are fast and efficient
- Has got rich set of operators
- more general and has no restrictions
- can easily manipulates with bits, bytes and addresses
- Varieties of data types are available
- separate compilation of functions is possible and such functions can be called by any C program
- block-structured language
- Can be applied in System programming areas like operating systems, compilers & Interpreters, Assemblers etc.,

II. Programming Basics

Components of a program

1. Constants
2. Variables
3. Operators
4. Statements

So, before writing serious programming we must be clear with all the above components of programs. According to above example every program is a set of statements, and statement is an instruction to the computer, which is a collection of constants, variables, operators and statements.

Constants

A constant is a fixed value, which never altered during the execution of a program.
Constants can be divided into two major categories:

1. Primary Constants
2. Secondary Constants

Data Types

The kind of data that the used variables can hold in a programming language is known as the data type.

Basic data types are as follows:

1. Numeric Data Type
2. Non-Numeric Data Type
3. Integer Data Type
4. Real Data Type
5. Logical Data Type
6. Enumerated Data Type

1. Numeric Data Type: Totally deals with the numbers. These numbers can be of integer (int) data type or real (float) data type.

2. Non-Numeric Data Type : Totally deals with characters. Any character or group of characters enclosed within quotes will be considered as non-numeric or character data type.

3. Integer Data Type : Deals with integers or whole numbers. All arithmetic operations can be achieved through this data type and the results are again integers.

4. Real Data Type : deals with real numbers or the numeric data, which includes fractions. All arithmetic operations can be achieved through this data type and the results can be real data type.

5. Logical or Boolean Data Type : can hold only either of the two values TRUE or FALSE at a time. In computer, a 1 (one) is stored for TRUE and a 0 (zero) is stored for FALSE.

6. Enumerated Data Type : Includes the unstructured data grouped together to lead to a new type. This data type is not standard and us usually defined by user.
Ex.

```
Week_days = { "sun", "mon", "tue", "wed", "thu", "fri", "sat" };
Directions = {"North", "East", "West", "South" };
```

The following table shows the standard data types with their properties.

Keyword	Range: low	Range: high	Digits of precision	Bytes of memory	Format-ID
	-128	127	n/a	1	%c
vhar					
int	**-32, 768**	**32, 767**	**N/a**	**2 (on 16 bit processor)**	**%d**
long	-2,147, 483, 648	2, 147, 483, 647	N/a	4	%ld
float	3.4 x 10-38	3.4 x 1038	7	4	%f
double	1.7 x 10-308	1.7 x 10308	15	8	%lf
long double	3.4 x 10-4932	3.4 x 10-4932	19	10	%Lf

NOTE: *The required ranges for signed and unsigned **int** are identical to those for signed and unsigned short. On compilers for 8 and 16 bit processors (including Intel x86 processors executing in 16 bit mode, such as under MS-DOS), an int is usually 16 bits and has exactly the same representation as a short. On compilers for 32 bit and larger processors (including Intel x86 processors executing in 32 bit mode, such as Win32 or Linux) an int is usually 32 bits long and has exactly the same representation as a long.*

I want you to refer this page for more information on int type for different processors:

Ref: http://www.jk-technology.com/c/inttypes.html

next
Learning C/C++ Step-By-Step - Page 02

add comment | view as pdf | print: this | all page(s)

Related Tutorials

- Beginner's Guide To c++
- An Explanation of Pointers (C++)

Please do not use the comment function to ask for help! If you need help, please use our forum.
Comments will be published after administrator approval.

Why use Microsoft libraries ?
Submitted by Jason (not registered) on Tue, 2013-04-16 15:58.

Why using conio.h ? It is not part of C standard, C++ standard, nor is it part of any Linux or Unix libraries.

The inclusions of Microsoft specific libraries in chapters 4 & 5 have me doubting the validity the rest of the tutorial (learning c/cpp on Linux).

The tutorial is well written, however, but I think it would be more appropriate on a .Net or Msdn blog.