# 一．C标准

C标准主要由两部分组成，一部分描述C的语法，另一部分描述C标准库。C标准库定义了一组标准头文件，每个头文件中包含一些相关的函数、变量、类型声明和宏定义。要在一个平台上支持C语言，不仅要实现C编译器，还要实现C标准库，这样的实现才算符合C标准。不符合C标准的实现也是存在的，例如很多单片机的C语言开发工具中只有C编译器而没有完整的C标准库。

# 二．glibc

　　在Linux平台上最广泛使用的C函数库是glibc，其中包括C标准库的实现，所有系统函数的实现。几乎所有C程序都要调用glibc的库函数，所以glibc是Linux平台C程序运行的基础。glibc提供一组头文件和一组库文件，最基本、最常用的C标准库函数和系统函数在libc.so库文件中，几乎所有C程序的运行都依赖于libc.so，有些做数学计算的C程序依赖于libm.so，以后我们还会看到多线程的C程序依赖于libpthread.so。以后我说libc时专指libc.so这个库文件，而说glibc时指的是glibc提供的所有库文件。glibc并不是Linux平台唯一的基础C函数库，也有人在开发别的C函数库，比如适用于嵌入式系统的uClibc，klibc。

　　在计算术语中，uClibc 是一个面向嵌入式Linux系统的小型的C标准库。最初uClibc是为了支持uClinux而开发，这是一个不需要内存管理单元的Linux版本，因此适合于微控制器系统（uCs；此处"u"是代表"micro"的μ的罗马化）. 　项目领导人是Erik Andersen.其他主要贡献者是Manuel Novoa III.许可证遵从GNU LGPL。uClibc是自由软件。 　uClibc比一般用于Linux发行版的C库GNU C Library (glibc)要小得多，glibc目标是要支持最大范围的硬件和内核平台的所有C标准，而uClibc专注于嵌入式Linux.很多功能可以根据空间需求进行取舍。uClibc运行于标准的以及无MMU的Linux系统上，支持i386，x86 64，ARM (big/little endian), AVR32，Blackfin，h8300，m68k，MIPS (big/little endian), PowerPC，SuperH (big/little endian), SPARC，和v850等处理器。

　　 klibc，一种C标准函式库，开发者为汉·彼得·艾文（Hans Peter Anvin）。它是自由软件，采用GNU 通用公众授权条款或BSD授权条款。它主要应用于Linux开机流程中，而且它也是早期用户空间（Early user space）与initramfs的一部份。在此时，这些应用程式无法使用glibc。它也适用于嵌入式系统的开发。

# 三．C标准库

### 0．概述

      C语言函数手册也称为C标准库。C标准库由在15个头文件中声明的函数、类型定义和宏组成，每个头文件都代表了一定范围的编程功能。有人说，C标准库可以分为 3 组，如何正确并熟练的使用它们，可以相应的可区分出 3 个层次的程序员：

合格程序员：<stdio.h>、<ctype.h>、<stdlib.h>、<string.h>

熟练程序员：<assert.h>、<limits.h>、<stddef.h>、<time.h>

优秀程序员：<float.h>、<math.h>、<error.h>、<locale.h>、<setjmp.h>、<signal.h>、<stdarg.h>

### 1．<math.h>里的数学计算公式介绍

    math.h 头文件定义了各种数学函数和一个宏。在这个库中所有可用的功能都带有一个 double 类型的参数，且都返回 double 类型的结果。

### 2．<assert.h>函数

    assert.h C标准库的头文件中提供了一个宏称为断言可以用于验证程序的假设，如果这个假设是错误的，并打印诊断消息。

### 3．<ctype.h>函数

ctype.h是C标准函数库中的头文件，定义了一批C语言字符分类函数（C character classification functions），用于测试字符是否属于特定的字符类别，如字母字符、控制字符等等。既支持单字节（Byte)字符，也支持宽字符。

### 4．<errno.h>函数

errno.h中的C标准库的头文件中定义的整数变量errno，这是由系统调用和一些库函数的错误事件指明什么地方出了错。该宏展开为int类型的修改的左值，因此它可以同时读取和修改程序。

### 5．float.h函数

float.h中的C标准库的头文件包含了一组浮点值相关的各种平台相关的常数。这些常量是由ANSI C允许更多的可移植程序提出。

### 6.limits.h函数

limits.h头中确定各种变量类型的各种属性。在这个头中定义的宏限制的char，int和long类型，如各种变量的值。

### 7.locale.h函数

locale.h头文件定义了特定的位置设置，如日期格式和货币符号。

### 8.setjmp.h函数

### 9.signal.h函数

signal.h头文件中定义变量类型sig\_atomic\_t，两个函数调用和几个宏处理程序的执行过程中不同的信号报告。

### 10.<stdarg.h>函数

stdarg.h头文件定义了一个变量va\_list类型和三个宏，可以用来获取一个函数的参数的个数，即不知道可变数目的参数。可变参数函数定义的参数列表的末尾的省略号（...）。

### 11.<stddef.h>函数

stddef.h 头文件定义了各种变量的类型和宏。许多这些定义也出现在其他头文件中。

### 12.<stdio.h>函数

stdio.h头定义了三个变量的类型，几个宏及各种功能进行输入和输出。

### 13.<stdlib.h>函数

stdlib .h 头文件定义了四个变量类型、一些宏和各种通用工具函数。\*库变量

### 14.<string.h>函数

string.h 头定义了一个变量的类型，操作字符数组的一个宏观和各种功能。

### 15.<time.h>函数

time.h 头定义了四个变量类型，两个宏和用于操作的日期和时间的各种功能。

# 四.使用math.h的注意事项

1.首先在使用gcc时，默认并不连接数学库/usr/lib/libm.a，只连接/usr/lib/libc.a，因此在使用math.h时，需要加参数选项-lm。

2.在ansi标准中，M\_PI并未定义，因此为满足ansi标准，使用PI时，需要自定义

const double PI = 4.0 \* atan(1.0);

3.另外，math.h中，三角函数，反三角函数，双曲三角函数的输入为弧度制，不是角度制。因此输入时需要进行调整：