<https://segmentfault.com/a/1190000012461553>

## C语言标准

1. 早期的计算机汇编语言是与机器平台紧密耦合的，为了屏蔽这种耦合，增加代码的可移植性，C语言随机出现。
2. 二十世纪八十年代，为了避免各开发厂商用的C语言语法产生差异，由美国国家标准局为C语言制定了一套完整的国际标准语法，称为ANSI C，作为C语言最初的标准。
3. 与ANSI C标准同时出现的就是ISO组织，将ANSI C加入了ISO的大家庭，定义了ISO C。除了在格式和排版等方面存在一些差别外，其他都与ANSI C相同。
4. **所以一句话ANSI C与ISO C 对于我们开发者来说完全相同。**
5. 目前出现的C标准如下：
   * C89
   * C99
   * C11
6. 对于编程来说，不同的机器平台的编译器可能不同，但是他必须按照ISO C的标准来实现，即他必须支持对于C标准的语法规则的编译。当然随着C标准的演进，编译器也是需要随之演进的，比如当前的C11标准，不是所有的编译器均支持。
7. 交叉编译，就是在一个平台上通过交叉编译器，编译另个平台上运行的程序。编译器起到重要作用哦~

## C标准库

* C标准库是严格按照C标准规范实现的一个C库，在Linux下有个叫libc的库就是标准库了。有些Linux下可能已经和glibc打包到一起了。
* 起初的C标准库存在15个头文件(后续介绍)，我认为这15个头文件是我们作为C开发者必须知道的。每个语言的诞生都会有一些语言对应的库提供给开发者，这是一个语言能够发展起来的基础。（个人理解哈~）

## POSIX C

1. 说起POSIX C有人会鄙视我，说根本就没有POSIX C，对是没有POSIX C标准，但是有POSIX C库。
2. 首先要理解什么事POSIX，全称：可移植操作系统接口。故名思议，由于当时Unix诞生之后，由于各个厂商都实现自己的Unix（都敢使用Unix，由于版权的问题），接口的不统一，导致在各个厂商下基于不同的操作系统开发变得很乱，为了解决这一问题，便有了POSIX标准。
3. 一句话：**POSIX标准的诞生是为了统一个操作系统的接口，方便开发者开发程序，写出可移植的代码程序。基于POSIX标准的库函数都是可以在持之此标准的操作系统平台上移植的。**
4. 我们说的posix c库就是基于POSIX标准定义的一套头文件实现的lib库。
5. 所以说我们man手册中经常会看到POSIX Program信息，表示是遵循POSIX标准而实现的。

展示一下ANSI C和POSIX C库的头文件哪些。如下展示的C标准库头文件是被扩展之后的标准，而不是起初的标准库了。增加了一些功能。  
[http://www.cnitblog.com/guopi...](http://www.cnitblog.com/guopingleee/archive/2007/08/13/31701.html)

## GNU C

1. GNU 是一个自由软件基金会，GNU's Not Unix.
2. GNU 是为了实现自由开源目的一个基金会，它提供了很多基于POSIX标准的软件和库，比如glibc、gcc、emacs等等。
3. 聊到这里就要聊聊Linux了，Linux诞生相信大家都听说过了，此处不再提。由于Linux开发者拥护GNU理念，将Linux也就遵循了GNU标准，并且Linux的开发也使用了GNU提供的大量的库函数。
4. 所以每个Linux版本都存在glibc的库。
5. Linux为了兼容性考虑在设计之初就按照POSIX标准，支持了POSIX标准。
6. **GNU C叫做glibc是Linux上的一个基础库，glibc C实现了POSIX C标准的库函数功能，有些POSIX标准是单独的库函数存在的**
7. 小插曲：由于GNU组织没有开发出很完善的操作系统内核，使用了Linux内核，所以将Linux纳入进来，又由于GNU组织存在很多开源的很好的工具组件促使Linux快速成长，所以说GNU和Linux是相互成全。（个人感觉仅供参考~）

## C语言学习个人总结

1. 想要使用C语言开发软件系统，必须了解C标准包括语法规则，其次了解C标准库清楚哪些基础库可以使用，使用时再查也是可以的。
2. 对于C标准库应该各个操作系统平台都支持，放心使用即可。
3. 在类Unix、Linux操作系统上编程，当然要学习POSIX标准，了解POSIX提供了哪些编程接口，方便快速开发。
4. 在Windows上当然也有一套编程接口，不了解不瞎哔哔了。但是：不是说windows完全不支持POSIX标准。Windows NT-based 系统不能直接支持新版POSIX接口，仅支持第一版POSIX v1:ISO/IEC9945-1:1990[1]版标准。由于仅支持第一版POSIX的Windows NT-based系统不能创建符合POSIX接口标准的线程和窗体、套接字，所以微软公司提供POSIX兼容层Microsoft POSIX subsystem软件包（Windows Services for UNIX）以支持新版POSIX接口，Windows 系统还可以运行其他POSIX兼容层例如Cygwin.
5. C标准不断的演进有些新增语法也是值得学习的，毕竟新增的语法和头文件更加方便我们开发程序。
6. GNU C中存在一些对于C标准的优化，如果是在Linux上开发程序，那我们应该大胆的使用它，方便我们程序的开发。例如关于\_\_attribute\_\_的机制就是GNU C的一大特色。所以关于GNU C我们也是要好好的学习一下的。