<https://blog.csdn.net/skylj0209/article/details/3071507>

### 标准 (ANSI C, POSIX, SVID, XPG, ...)

* ANSI C：这一标准是 ANSI（美国国家标准局）于 1989 年制定的 C 语言标准。 后来被 ISO（国际标准化组织）接受为标准，因此也称为 ISO C。  
  ANSI C 的目标是为各种操作系统上的 C 程序提供可移植性保证，而不仅仅限于 UNIX。 该标准不仅定义了 C 编程语言的语发和语义，而且还定义了一个标准库。这个库可以根据 头文件划分为 15 个部分，其中包括：字符类型 (<ctype.h>)、错误码 (<errno.h>)、 浮点常数 (<float.h>)、数学常数 (<math.h>)、标准定义 (<stddef.h>)、 标准 I/O (<stdio.h>)、工具函数 (<stdlib.h>)、字符串操作 (<string.h>)、 时间和日期 (<time.h>)、可变参数表 (<stdarg.h>)、信号 (<signal.h>)、 非局部跳转 (<setjmp.h>)、本地信息 (<local.h>)、程序断言 (<assert.h>) 等等。
* POSIX：该标准最初由 IEEE 开发的标准族，部分已经被 ISO 接受为国际标准。该标准的具体内容 见 1.1.3。POSIX.1 和 POSIX.2 分别定义了 POSIX 兼容操作系统的 C 语言系统接口 以及 shell 和工具标准。这两个标准是通常提到的标准。POSIX 表示可移植操作系统接口（Portable Operating System Interface ，缩写为 POSIX 是为了读音更 像 UNIX）。电气和电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE） 最初开发 POSIX 标准，是为了提高 UNIX 环境下应用程序的可移植性。然而，POSIX 并不局限于 UNIX。 许多其它的操作系统，例如 DEC OpenVMS 和 Microsoft Windows NT，都支持 POSIX 标准，尤其是 IEEE Std. 1003.1-1990（1995 年修订）或 POSIX.1，POSIX.1 提供了源代码级别的 C 语言应用编程 接口（API）给操作系统的服务程序，例如读写文件。POSIX.1 已经被国际标准化组织（International Standards Organization，ISO）所接受，被命名为 ISO/IEC 9945-1:1990 标准。
* SVID：System V 的接口描述。System V 接口描述（SVID）是描述 AT&&;T Unix System V 操作 系统的文档，是对 POSIX 标准的扩展超集。
* XPG：X/Open 可移植性指南。X/Open 可移植性指南（由 X/Open Company, Ltd.出版）， 是比 POSIX 更为一般的标准。X/Open 拥有 Unix 的版权，而 XPG 则指定成为 Unix 操作系统必须满足的要求。

**Glibc**

众所周知，C 语言并没有为常见的操作，例如输入/输出、内存管理，字符串操作等提供内置的支持。 相反，这些功能一般由标准的“函数库”来提供。GNU 的 C 函数库，即 glibc，是 Linux 上最重要的 函数库，它定义了 ISO C 标准指定的所有的库函数，以及由 POSIX 或其他 UNIX 操作系统 统变种指定的附加特色，还包括有与 GNU 系统相关的扩展。目前，流行的 Linux 系统使用 glibc 2.0 以上的版本。glibc 基于如下标准：

* ISO C: C 编程语言的国际标准，即 ANSI C。
* POSIX：GNU C 函数库实现了 ISO/IEC 9945-1:1996 （POSIX 系统应用程序编程接口， 即 POSIX.1）指定的所有函数。该标准是对 ISO C 的扩展，包括文件系统接口原 语、设备相关的终端控制函数以及进程控制函数。同时，GUN C 函数库还支持部分由 ISO/IEC 9945-2:1993（POSIX Shell 和 工具标准，即 POSIX.2）指定的函数， 其中包括用于处理正则表达式和模式匹配的函数。
* Berkeley Unix：BSD 和 SunOS。GNU C 函数库定义了某些 UNIX 版本中尚未标准化的函数， 尤其是 4.2 BSD, 4.3 BSD, 4.4 BSD Unix 系统（即“Berkeley Unix”)以及“SunOS” （流行的 4.2 BSD 变种，其中包含有某些 Unix System V 的功能）。BSD 函数包括 符号链接、select 函数、BSD 信号处理函数以及套接字等等。
* SVID：System V 的接口描述。GNU C 函数库定义了大多数由 SVID 指定而未被 ISO C 和 POSIX 标准指定的函数。来自 System V 的支持函数包括进程间通信和共享内存、 hsearch 和 drand48 函数族、fmtmsg 以及一些数学函数。
* XPG：X/Open 可移植性指南。GNU C 函数库遵循 X/Open 可移植性指南（Issue 4.2） 以及所有的 XSI（X/Open 系统接口）兼容系统的扩展，同时也遵循所有的 X/Open Unix 扩展。

#### 系统调用

系统调用是操作系统提供给外部程序的接口。在 C 语言中，操作系统的系统调用通常通过 函数调用的形式完成，这是因为这些函数封装了系统调用的细节，将系统调用的入口、参数以及 返回值用 C 语言的函数调用过程实现。在 Linux 系统中，系统调用函数定义在 glibc 中。

谈到系统调用时，需要注意如下几点：

* 系统调用函数通常在成功时返回 0 值，不成功时返回非零值。如果要检查失败原因，则 要判断 errno 这个全局变量的值，errno 中包含有错误代码。
* 许多系统调用的返回数据通常通过引用参数传递。这时，需要在函数参数中传递一个 缓冲区地址，而返回的数据就保存在该缓冲区中。
* 不能认为系统调用函数就要比其他函数的执行效率高。要注意，系统调用是一个非常耗时 的过程。

有关系统调用我们将在以后详细讲述。

### 在线文档 (man, info, HOW-TO, ...)

#### 1. man

man，即 manunal，是 UNIX 系统手册的电子版本。根据习惯，UNIX 系统手册通常分为 不同的部分（或小节，即 section），每个小节阐述不同的系统内容。目前的小节划分如下：

1. 命令：普通用户命令
2. 系统调用：内核接口
3. 函数库调用：普通函数库中的函数
4. 特殊文件：/dev 目录中的特殊文件
5. 文件格式和约定：/etc/passwd 等文件的格式
6. 游戏。
7. 杂项和约定：标准文件系统布局、手册页结构等杂项内容
8. 系统管理命令。
9. 内核例程：非标准的手册小节。便于 Linux 内核的开发而包含

其他手册小节：

* l: PostgreSQL 数据库命令
* n: TCL/TK 命令

手册页一般保存在 /usr/man 目录下，其中每个子目录（如 man1, man2, ..., manl, mann） 包含不同的手册小节。使用 man 命令查看手册页。

man 命令行：  
man [-acdfFhkKtwW] [-m system] [-p string] [-C config\_file] [-M path] [-P pager] [-S section\_list] [section] name

常用命令行：  
$ man open  
$ man 7 man  
$ man ./myman.3

#### 2. info

Linux 中的大多数软件开发工具都是来自自由软件基金会的 GNU 项目，这些工具软件 件的在线文档都以 info 文件的形式存在。info 程序是 GNU 的超文本帮助系统。

info 文档一般保存在 /usr/info 目录下，使用 info 命令查看 info 文档。

要运行 info，可以在 shell 提示符后输入 info，也可以在 GNU 的 emacs 中键入 Esc-x 后跟 info。

info 帮助系统的初始屏幕显示了一个主题目录，你可以将光标移动到带有 \* 的主题菜单上面，然后按回车键  
进入该主题，也可以键入 m，后跟主题菜单的名称而进入该主题。例如，你可以键入 m，然后再键入 gcc 而进  
进入 gcc 主题中。  
如果你要在主题之间跳转，则必须记住如下的几个命令键：  
\* n：跳转到该节点的下一个节点；  
\* p：跳转到该节点的上一个节点；  
\* m: 指定菜单名而选择另外一个节点；  
\* f：进入交叉引用主题；  
\* l：进入该窗口中的最后一个节点；  
\* TAB：跳转到该窗口的下一个超文本链接；  
\* RET：进入光标处的超文本链接；  
\* u：转到上一级主题；  
\* d：回到 info 的初始节点目录；  
\* h：调出 info 教程；  
\* q：退出 info。

#DEMO#

#### 3. HOW-TO

可供用户参考的联机文档的另一种形式是 HOWTO 文件，这些文件位于系统的 /usr/doc/HOWTO 目录下。 HOWTO 文件的文件名都有一个 -HOWTO 后缀，并且都是文本文件。

每一个 HOWTO 文件包含 Linux 某一方面的信息，例如它支持的硬件或如何建立一个引导盘。

要想查看这些文件，进入 /usr/doc/HOWTO 目录，使用 more 命令，具体形式如下：  
$ cd /usr/doc/HOWTO; more *topic-name*-HOWTO

另外，HOWTO 文档还有其他格式的文件，例如 HTML 和 PS 等，保存在 /usr/doc/HOWTO/other-formats 下。

#### 4. 其他

Linux 的内核文档一般包含在内核源代码中，目录如下：/usr/src/*linux-2.x.x*/Documentation

/usr/doc 目录下包含有大量与特定软件或函数库相关的说明性文档。

### 库和头文件的保存位置

#### 1. 函数库

* /lib：系统必备共享库
* /usr/lib：标准共享库和静态库
* /usr/i486-linux-libc5/lib：libc5 兼容性函数库
* /usr/X11R6/lib：X11R6 的函数库
* /usr/local/lib：本地函数库

#### 2. 头文件

* /usr/include：系统头文件
* /usr/local/include：本地头文件

POSIX 现在已经发展成为一个非常庞大的标准族，某些部分正处在开发过程中。表 1-1 给出了 POSIX 标准的几个 重要组成部分。POSIX 与 IEEE 1003 和 2003 家族的标准是可互换的。除 1003.1 之外，1003 和 2003 家族也包括在表中。

1-1 POSIX 标准的重要组成部分

|  |  |
| --- | --- |
| 1003.0 | 管理 POSIX 开放式系统环境（OSE）。IEEE 在 1995 年通过了这项标准。 ISO 的版本是 ISO/IEC 14252:1996。 |
| 1003.1 | 被广泛接受、用于源代码级别的可移植性标准。1003.1 提供一个操作系统的 C 语言应 用编程接口（API）。IEEE 和 ISO 已经在 1990 年通过了这个标准，IEEE 在 1995 年 重新修订了该标准。 |
| 1003.1b | 一个用于实时编程的标准（以前的 P1003.4 或 POSIX.4）。这个标准在 1993 年 被 IEEE 通过，被合并进 ISO/IEC 9945-1。 |
| 1003.1c | 一个用于线程（在一个程序中当前被执行的代码段）的标准。以前是 P1993.4 或 POSIX.4 的一部分，这个标准已经在 1995 年被 IEEE 通过，归入 ISO/IEC 9945-1:1996。 |
| 1003.1g | 一个关于协议独立接口的标准，该接口可以使一个应用程序通过网络与另一个应用程序 通讯。 1996 年，IEEE 通过了这个标准。 |
| 1003.2 | 一个应用于 shell 和 工具软件的标准，它们分别是操作系统所必须提供的命令处理器 和工具程序。 1992 年 IEEE 通过了这个标准。ISO 也已经通过了这个标准（ISO/IEC 9945-2:1993）。 |
| 1003.2d | 改进的 1003.2 标准。 |
| 1003.5 | 一个相当于 1003.1 的 Ada 语言的 API。在 1992 年，IEEE 通过了这个标准。 并在 1997 年对其进行了修订。ISO 也通过了该标准。 |
| 1003.5b | 一个相当于 1003.1b（实时扩展）的 Ada 语言的 API。IEEE 和 ISO 都已经通过了 这个标准。ISO 的标准是 ISO/IEC 14519:1999。 |
| 1003.5c | 一个相当于 1003.1q（协议独立接口）的 Ada 语言的 API。在 1998 年， IEEE 通过了这个标准。ISO 也通过了这个标准。 |
| 1003.9 | 一个相当于 1003.1 的 FORTRAN 语言的 API。在 1992 年，IEEE 通过了这个标准， 并于 1997 年对其再次确认。ISO 也已经通过了这个标准。 |
| 1003.10 | 一个应用于超级计算应用环境框架（Application Environment Profile，AEP）的标准。 在 1995 年，IEEE 通过了这个标准。 |
| 1003.13 | 一个关于应用环境框架的标准，主要针对使用 POSIX 接口的实时应用程序。 在 1998 年，IEEE 通过了这个标准。 |
| 1003.22 | 一个针对 POSIX 的关于安全性框架的指南。 |
| 1003.23 | 一个针对用户组织的指南，主要是为了指导用户开发和使用支持操作需求的 开放式系统环境（OSE）框架 |
| 2003 | 针对指定和使用是否符合 POSIX 标准的测试方法，有关其定义、一般需求和指导方针的 一个标准。在 1997 年，IEEE 通过了这个标准。 |
| 2003.1 | 这个标准规定了针对 1003.1 的 POSIX 测试方法的提供商要提供的一些条件。 在 1992 年，IEEE 通过了这个标准。 |
| 2003.2 | 一个定义了被用来检查与 IEEE 1003.2（shell 和 工具 API）是否符合的测试方法的 标准。在 1996 年，IEEE 通过了这个标准。 |

除了 1003 和 2003 家族以外，还有几个其它的 IEEE 标准，例如 1224 和 1228，它们 也提供开发可移植应用程序的 API。要想得到关于 IEEE 标准的最新信息，可以访问 IEEE 标准的主页，网址是 [http://standard.ieee.org/](http://standards.ieee.org/)。 有关 POSIX 标准的概述信息，请访问 Web 站点 <http://standards.ieee.org/reading/ieee/stad_public/description/posix/>。