

## Cygwin系列（一）：Cygwin是什么



silaoA

爱好编程的电气工程师, silaoa.github.io

已关注

217 人赞同了该文章

发布于 2019-02-14 14:49，编辑于 2021-04-09 08:31

本文4300余字，阅读约11分钟，本文知乎链接：[Cygwin系列（一）：Cygwin是什么](#)。

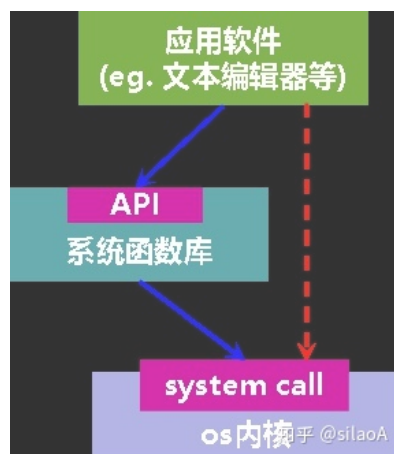
本文接上篇Cygwin前传：从割据到互补。先一句话回答标题：**Cygwin是一个可原生运行于Windows系统上的POSIX兼容环境。**

### 前言：从软件角度理解系统

计算机世界里存在各种各样的操作系统，目前通用操作系统有主流的三大类：

- UNIX，通用操作系统鼻祖，发展出特别多衍生系统；
- Windows，微软家根基，桌面市场霸主；
- GNU/Linux，UNIX近亲，有各种发行版如Ubuntu、CentOS等。

这些系统有各自的内核，出于系统稳定性考虑是不允许用户程序直接操作内核，同时也将内核开发和应用软件开发隔离开来，系统将必要的函数封装成库供应用软件调用，约定的规范即为应用软件接口(Application Program Interface, API)。



软件系统层级关系简要示意图

API函数库是连接用户软件和系统内核桥梁，或者是“协议”，操作系统厂商写好函数库说明书，应用软件开发人员不必关心其内部是如何实现的，用的时候对照着API手册查询就够了；应用软件也可以越过系统函数库通过system call（系统调用）直接调用os内核函数，如图中红色虚线所示，这种方式是进入到内核空间操作，并不被推荐，一般是调用系统函数API在用户空间操作。

如果各系统平台都能提供相同的系统函数库，那么开发者在这个系统函数库基础之上编写软件代码，那么就很容易将软件移植到各个系统平台。然而，这只是个美好的愿望，IEEE就是为了达成这样的愿望才牵头制定POSIX标准。POSIX标准主要就是针对UNIX API而制订，不管函数如何包装、功能如何实现，但API按照标准约定来（比如函数变量等符号名称、数据结构、参数类型与个数、基本工具命令名称等），Linux完全兼容POSIX标准（部分函数符合POSIX，部分函数是Linux专有，即是POSIX的超集），微软声称Windows部分兼容POSIX标准。

主流os内核通常是C语言写成，系统函数库通常以一个或多个链接库文件的形式提供，其中最重要的是C标准库，其他的链接库往往调用C标准库而实现，当然也可能直接调用系统内核函数，甚至混合。不同系统平台有多种主流的C标准库共存：

- BSD libc，由BSD系统发布；
- GNU C Library (glibc)，由GNU项目发布，可在Linux、多种UNIX系统上运行；



- Microsoft C run-time library(MSVCRT.DLL)，由微软随Windows发布，给Visual C++编辑器链接使用的；
- **Newlib**，由Cygnus Solution公司开发，Cygwin环境中的libc.a正是此版本，目前广泛用在嵌入式系统中；
- dietlibc、μClibc等，功能经过适度裁剪的C标准库，主要用在嵌入式系统。

## Cygwin的组成

先辈David Wheeler大神曾经曰过：

“All problems in computer science can be solved by another level of indirection（计算机科学领域的任何问题都可以通过增加一个间接的中间层来解决）”。

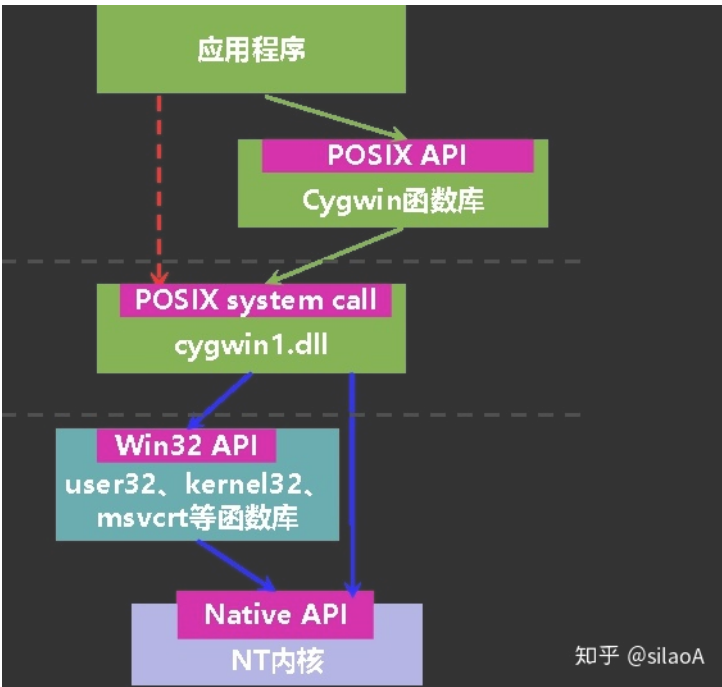
David Wheeler （图片来源：[https://en.wikipedia.org/wiki/David\\_Wheeler\\_\(computer\\_scientist\)](https://en.wikipedia.org/wiki/David_Wheeler_(computer_scientist))）

就如同上图，在硬件基础之上逐步堆叠了系统内核、系统函数库等中间层，在应用程序内部还可以继续细分多个层次，这样把最终用户与硬件隔离开来，增强了抽象能力、屏蔽底层细节，也让开发者分工，专注于各自层次的开发，同时降低了软件迁移的难度。

Cygwin就是在Windows中增加了一个中间层——兼容POSIX的模拟层，并在此基础上构建了大量Linux-like的软件工具。再来解释本文开头的回答，如下图，POSIX兼容环境包括以下两部分：

- cygwin1.dll，作为实现POSIX系统调用的模拟层，可原生运行在Windows中；
- 在cygwin1.dll之上构建的大量函数库、应用程序，如libc、zlib、bash、gcc、vim、awk、sed、git等等，与UNIX/Linux几乎等同\*。

\*注：Cygwin的libc是Newlib，Linux的libc是GNU libc，UNIX有的是BSD libc。



Cygwin环境层次简要示意图

Cygwin API首先尽可能地遵从Single Unix Specification V3（2018版），这个标准内容同时也是POSIX.1和IEEE Std 1003.1的标准内容，由Open Group和IEEE共同制定，最新已更新到V4



(2018版)，其次再尽可能地遵从Linux最佳实践。Cygwin API中还有些是Cygwin独有的，在POSIX中并未涉及。

Cygwin将cygwin1.dll、函数库、应用程序等文件按照UNIX/Linux的目录树架构进行组织存放，如/bin、/usr、/lib、/etc、/var、/home等等都存在于Cygwin安装路径下，用户从终端登陆进Cygwin的shell后，就可以像在UNIX/Linux系统那样使用相同的命令、工具，随着开发工作推进，越来越多的GNU、UNIX、Linux软件都移植到了Cygwin中。不仅如此，甚至像X Server、Gnome/KDE桌面环境等都移植到了Cygwin中，UNIX/Linux系统中的图形界面软件也能使用。

### Cygwin的优点

- 首先自然是近乎一致的UNIX/Linux体验；
- 完备且相对轻量，普通用户不必安装整个Linux系统或虚拟机，就可以获得近乎一致的体验，Cygwin的程序运行与Windows互不干扰，高效的命令行工具与Windows图形界面各有所长、形成互补；
- 开源免费，cygwin1.dll本身是按照GPLv3协议发布的，其他的应用程序有GPL、LGPL、X11等多种协议；
- 安装卸载方便，Cygwin提供了包管理工具，可按需安装/卸载软件包，一个能运行起来的最小Cygwin系统只需要几十上百MB，而完全的Cygwin系统需要几十GB；
- 源码级兼容性，GNU、UNIX、Linux软件的源代码几乎不用修改就可以在Cygwin环境中编译构建成功；
- 与Windows互操作，Cygwin把Windows的磁盘挂载到/cygdrive下，如c盘就是/cygdrive/c、d盘就是/cygdrive/d，Cygwin中的应用程序可以读写Windows磁盘中的文件，Windows应用程序也可以读写Cygwin目录中的文件（但要注意不要把文件搞乱了）；Cygwin的shell中可以启动Windows程序，Windows的cmd中也可以启动Cygwin的程序，但由于字符编码不同可能造成乱码；
- 多一套可用的API，对于Windows开发者，程序代码既可以调用Win32 API，又可以调用Cygwin API，甚至混合。

### Cygwin的不足

- 效率相对低，由于是在Win32系统之上模拟实现POSIX兼容层，应用程序在底层就多了一个层级的函数调用，效率比UNIX/Linux系统上原生的应用程序肯定低，不过这也是在效率和兼容性之间选择的一个平衡；
- 未实现二进制文件级别的兼容，Cygwin系统上的应用程序编译后仍然是Windows PE格式的可执行文件，**UNIX/Linux系统上的二进制可执行文件在Cygwin上不能运行。**
- 与Windows互操作不足，Windows原生程序并不能利用cygwin1.dll提供的与UNIX/Linux兼容的信号、pty设备等，除非改写程序代码重新编译，但这样新的程序就依赖于cygwin1.dll，就不是“Windows原生程序”了。

### Cygwin常见应用场景

Cygwin可资利用的是已经移植的大量GNU、UNIX、Linux软件和兼容POSIX的模拟层，其使用场合也就是针对这两点。



高效命令行工具（图片来源于网络）

常见的应用场合包括但不限于：

### （1）Shell命令行使用

Shell是UNIX/Linux的精华所在，骨灰级玩家可以做到不用鼠标只敲命令完成所有工作，用户最常用的大量命令在Cygwin下均可照常使用，在UNIX/Linux编写的脚本也可以几乎不加修改地在Cygwin下运行。例如安卓厨房本是在Linux-like环境下运行的脚本集合，用于修改安卓系统固件包，有了Cygwin，Windows用户也可以拿来修改安卓系统固件包。高效的命令行工具与Windows图形界面各有所长、形成互补。

### （2）交叉编译构建环境搭建

Cygwin环境中已移植好了gcc等开发工具，大量的交叉工具链（如arm-none-gnu-eabi-gcc、arm-none-gnu-eabi-binutils）也可以在Cygwin中制作，就算只有Windows原生版本的，Cygwin shell中也能调用，那么利用Cygwin就能搭建起交叉编译构建环境；另外，使用Cygwin API，编写代码以及后续编译构建过程，与在UNIX/Linux中差异也很小了。

### （3）程序移植

把符合POSIX标准的程序移植到Windows下，还有更多正在由个人、社区、商业公司、研究机构不断贡献的开源自由软件，造福广大Windows用户，利用已有的GNU、UNIX、Linux软件会使程序移植越来越容易。这一点不多说。

### （4）兼用POSIX API和Win32 API开发

有的开发者可能对UNIX/Linux和Win32的API都熟悉，两套API也各有其优点，在Cygwin下开发者自己可以任意选取、混合使用。

## 参考

- [zh.wikipedia.org/zh-cn/...](http://zh.wikipedia.org/zh-cn/...)
- [zh.wikipedia.org/wiki/P...](http://zh.wikipedia.org/wiki/P...)
- [en.wikipedia.org/wiki/C...](http://en.wikipedia.org/wiki/C...)
- <http://www.cygwin.com>
- [en.wikipedia.org/wiki/L...](http://en.wikipedia.org/wiki/L...)
- [en.wikipedia.org/wiki/C...](http://en.wikipedia.org/wiki/C...)
- [unix.org/apis.html](http://unix.org/apis.html)

## 更多阅读

- [上一篇 Cygwin前传：从割据到互补](#)
- [下一篇 Cygwin系列（二）：初窥Cygwin背后](#)

- [伪码人专栏目录导航 \(持续更新...\)](#)
- [微软WSL——Linux桌面版未来之光](#)
- [GNU Wget 爬虫? 试一试](#)
- [silaoA的博客.https://silaoa.github.io](https://silaoa.github.io)



如本文对你有帮助，或内容引起极度舒适，欢迎分享转发与留言交流

► 本文为原创文章，如需转载请私信知乎账号或联系公众号：伪码人（We\_Coder）。

发布于 2019-02-14 14:49，编辑于 2021-04-09 08:31

「真诚赞赏，手留余香，感谢！」

赞赏

1 人已赞赏



[Cygwin](#) [Linux](#) [计算机](#)

写评论... | 你赞过作者的回答

12 条评论

默认 时间



yujian

自从有了wsl之后，感觉cygwin就有点鸡肋了😂

2021-12-06 · 作者回复了

● 回复 3



silaoA 作者

是的吧。cygwin历史早，在老机器上还能一战。

2021-12-06

回复 赞



奇诺城堡

强烈建议在这系列每篇文章下面放一个指向下一篇文章的连接，这样便于寻址👍

2019-10-10 · 作者回复了

回复 1 赞



奇诺城堡 > silaoA



2019-10-11

回复 赞



silaoA 作者 > 奇诺城堡

🔥 我再改改，欢迎留言交流。

2019-10-11

回复 赞

展开其他 2 条回复 >



血刃飘香

好文，读着轻松

2021-09-28 · 作者回复了

回复 赞



silaoA 作者

希望点赞转发关注三连😁

2021-09-28

回复 赞



ksadfhlg

兄弟你第一个posix写错了

2021-04-09 · 作者回复了

回复 赞



silaoA 作者

我修改一下，感谢。

2021-04-09

回复 赞



至强处理器

感谢作者的精彩文章！

2020-11-20

回复 赞

写评论... | 你赞过作者的回答

## 文章被以下专栏收录



伪码人 We\_Coder

专注Cygwin、Linux、电气等技术交流分享



技术部落联盟

涉及容器、大数据、人工智能、物联网等领域