



Cygwin系列（十一）：折腾终端2



silaoA

爱好编程的电气工程师, silaoa.github.io

已关注

15 人赞同了该文章

发布于 2020-01-14 08:30，编辑于 2020-11-27 13:22

本文共2500余字，预计阅读时间8分钟，本文知乎连接：[Cygwin系列（十一）：折腾终端2](#)，本文同步发布于silaoA的博客和微信公众号平台。

关注学习了解更多的Cygwin、Linux、Python技术。

目录

- 0x00 ansicon
- 0x01 winpty
- 0x02 ConPTY
- 0x03 Windows Terminal
- 0x04 总结
- 参考
- 更多阅读

本文承接前篇 [Cygwin系列（十）：折腾终端1](#)，继续深度扒终端。既然Windows Console与UNIX/Linux上的pty机制完全不同，要让基于不同机制的命令程序和终端配合起来，颇费周章。又双叒叕祭出David Wheeler大神的著名论断：

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）。

也就是说，解决办法无非是在已有条件下继续打造“中间层”，而Cygwin这个中间层试图直接补齐Windows API和UNIX/Linux API之间的差异，显得有点“重量”。

0x00 ansicon



Windows Console缺陷之一是无法处理ANSI转义序列，自然而然地有人想到做一款专门用于处理ANSI转义序列的工具， `ansicon` 便是，而且开源，项目地址 github.com/adoxa/ansicon...。从名字上看，应是“ansi”和“conversion”结合的缩写，望名知义。在MS-DOS还在流行的年代，有个 `ANSI.SYS` 的程序，也是干这个活。到Windows替代DOS时， `ansicon` 替代了 `ANSI.SYS` 的位置，它早在2010年就已诞生。

`ansicon` 十分轻量，软件包压缩后仅100多kB，程序文件就3个：`ANSI32.dll`、`ANSI64.dll` 和 `ansicon.exe`。 `--help` 选项输出 `ansicon` 的用法，如果不加任何选项、参数的话，`ansicon` 启动新的解释器实例，默认是 `cmd`。

```
ANSICON by Jason Hood <jadoxa@yahoo.com.au>.
Version 1.89 (29 April, 2019). Freeware.
http://ansicon.adoxa.vze.com/

Process ANSI escape sequences in Windows console programs.

ansicon [-lLEVEL] [-i] [-I] [-u] [-U] [-m[ATTR]] [-p[u]]
        [-e|E STRING] [-t|T [FILE...]] [PROGRAM [ARGS]]

-l          set the logging level (1=process, 2=module, 3=function,
            +4=output, +8=append, +16=imports, +32=files) for PROGRAM
-i          install - add ANSICON to CMD's AutoRun entry (also implies -p)
-u          uninstall - remove ANSICON from the AutoRun entry
-I -U       use local machine instead of current user
-m          use grey on black ("monochrome") or ATTR as default color
-p          hook into the parent process
-pu         unhook from the parent process
-e          echo STRING
-E          echo STRING, don't append newline
-t          display files ("- " for stdin), combined as a single stream
-T          display files, name first, blank line before and after
PROGRAM     run the specified program
nothing     run a new command processor, or display stdin if redirected

ATTR is one or two hexadecimal digits; please use "COLOR /?" for details.
It may start with '-' to reverse foreground and background (but not for -p).
```

ansicon用法

`ansicon` 用在什么场合呢？用在给基于Windows Console的终端增加ANSI转义序列处理能力，运行POSIX兼容的命令程序，比如在conhost窗口中运行git for Windows、mingw-gcc等，它被consolez等项目所采用。但受制于 conhost 的终端能力 (capability)，`ansicon` 处理ANSI转义序列也受到较大限制，比如只支持16色等。

0x01 winpty

鉴于基于Windows Console API的命令程序与基于pty的终端之间的鸿沟，特别是后者运行交互式程序（如 `cmd`、`python`）时，`winpty` 项目便设法在其中架起一座桥梁，项目开源，地址：github.com/rprichard/winpty...。它的作用实质是**提升基于pty的终端对基于Windows Console API的命令程序的兼容性**，也就是说和 `ansicon` 的作用方向恰恰相反。

`winpty` 在Windows Console基础之上造了一个类pty主设备的接口，用于和Windows命令程序通信。`winpty` 的程序文件可以分为两部分：

- `winpty-agent.exe`，运行在Windows命令行侧，是原生win32程序，创建一个隐藏的conhost窗口、执行用户指定的命令程序，同时监控键盘、鼠标、触屏事件，抓取命令程序输出，和其他基于Windows Console的第三方终端干着差不多的活；
- `winpty.exe`，运行在基于pty的终端侧，称作“Unix Adapter”（Unix适配器），是一个Cygwin程序，依赖于 `cygwin1.dll`，负责将输入输出传给基于pty的终端。

以上虽然是分别在两侧发挥作用，实际是打包在一起的，还有一个`winpty.dll`，都放在Cygwin或MSYS环境中运行，`winpty.exe`和`winpty-agent.exe`之间保持着进程通信。工作流程是这样的，在基于pty的终端（如 `Mintty`）中输入形如“`winpty <程序名> <参数>`”的命令，发起要执行的Windows命令程序，`winpty-agent.exe`创建出一个隐藏的conhost窗口、照常执行Windows命令程序，返回的输出交给`winpty.exe`处理，再传给终端渲染，两端相互配合。

```
$ ls # 当前目录是winpty-0.4.3-cygwin-2.8.0-x64
bin include lib share
$ ./bin/winpty ping www.cygwin.com
```



```
正在 Ping www.cygwin.com [209.132.180.131] 具有 32 字节的数据:
来自 209.132.180.131 的回复: 字节=32 时间=244ms TTL=43
来自 209.132.180.131 的回复: 字节=32 时间=238ms TTL=43
来自 209.132.180.131 的回复: 字节=32 时间=239ms TTL=43
来自 209.132.180.131 的回复: 字节=32 时间=238ms TTL=43
```

209.132.180.131 的 Ping 统计信息:

数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),

往返行程的估计时间(以毫秒为单位):

最短 = 238ms, 最长 = 244ms, 平均 = 239ms

\$ ping www.cygwin.com # 直接执行ping作为对比

```
Ping www.cygwin.com [209.132.180.131] 32 字节:
209.132.180.131 来自: 字节=32 时间=236ms TTL=43
209.132.180.131 来自: 字节=32 时间=238ms TTL=43
209.132.180.131 来自: 字节=32 时间=239ms TTL=43
209.132.180.131 来自: 字节=32 时间=242ms TTL=43
```

209.132.180.131 Ping 统计信息:

数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),

往返行程的估计时间(以毫秒为单位):

最短 = 236ms, 最长 = 242ms, 平均 = 238ms



如果还能想起 [Linux Cygwin知识库 \(一\)](#)：一文搞清控制台、终端、shell概念 中描述的终端上执行远程主机程序的过程，可以发现与上图有着惊人的相似性，**winpty像是客户端 (ssh)**，**winpty-agent像是代理程序 (ssh server)**，winpty、winpty-agent这一对名字都标识了他们的角色！Unix Adapter (winpty.exe) 可以在Cygwin/MSYS中编译、运行，当然也能在UNIX/Linux中编译、运行，winpty项目还提供了开发库和头文件，**若能做出基于winpty的ssh server，那么在UNIX/Linux上远程运行Windows命令程序就变为可能**！以前在UNIX/Linux主机上远程执行Windows命令程序是不可行的，个人认为这是winpty项目更大的意义所在。上述在Mintty中运行 winpty ping www.cygwin.com 过程，与 ssh ping www.cygwin.com 何其相似，只不过winpty和ping在同一个Windows主机上。

0x02 ConPTY

Windows Console组件已有多年的历史，支撑着数不清的命令行程序，纵使和pty管道机制不一样，也不能贸然抛弃。但如何支持pty管道，微软也一直没什么作为，第三方的 [ansicon](#) 增加ANSI转义序列处理能力毕竟只是临时的“补丁”办法。

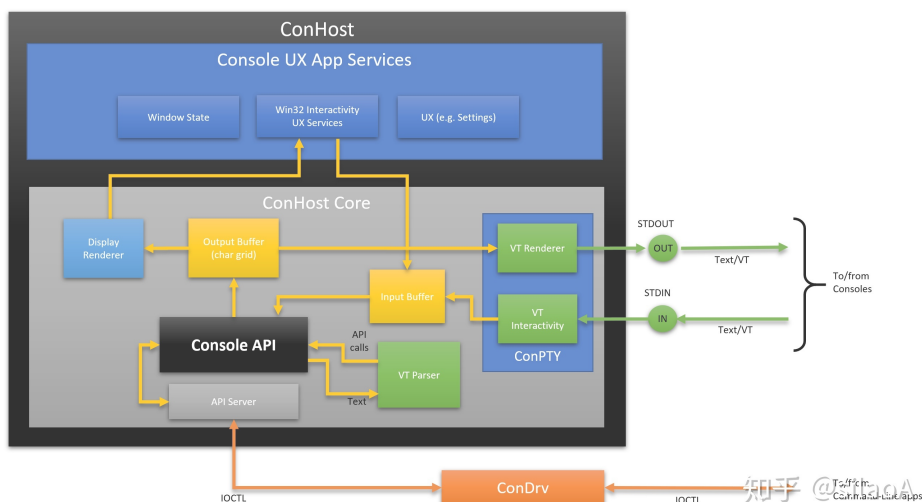
但，事情还是有所转机，自Satya Nadella执掌微软帝国以来，微软对开源社区、对Linux态度发生180度转变，以实际行动践行“Microsoft love Linux”的誓言：Azure支持Linux、开源.Net Core、开源VS Code、收购GitHub等，甚至主动在Windows 10中嵌入Linux这个毒瘤（WSL），一系列动作让人眼花缭乱。与此同时，微软还专门成立了命令行团队，下定决心完善命令行生态。终于在2018年，**以Windows Console核心为基础，官宣了Windows Pseudo Console (简称ConPTY) 组件和配套API，提供近似于pty的支持。**



ConPTY相当于微软自家打造的“中间层”。ConHost 还是那个 ConHost，ConDrv 也还是那个 ConDrv，但稍稍不同的是：

- ConHost 的核心之上，多了一层ConPTY；
- 更重要的是，ConPTY对外还留了标准输入/输出接口给终端，这个终端不再限死在微软自家的 conhost，可以是任何第三方软件，这就给了第三方终端开发更大的自由度。

Console Architecture with integrated ConPTY

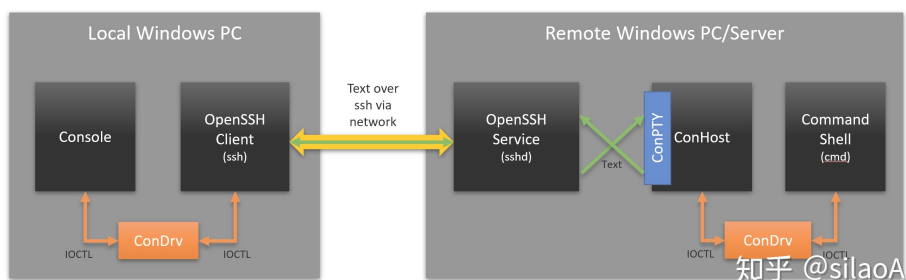


ConPTY架构 图片源于微软官方博客

那么现有环境下，**命令程序有Console API和ConPTY API两套可调用**，这和Cygwin环境下同时有Win32 API和POSIX API两套是类似的，旧有的程序继续用老的Console API，新的程序则建议直接上ConPTY API。在历史兼容性方面，旧有的程序不必做任何改变，输出照常传递给 conhost 核心，但经由ConPTY转成包含ANSI转义序列的文本，输出到新终端中，**也就是说基于ConPTY的终端，既能运行POSIX命令程序，也能运行基于Windows Console的命令程序！**

OpenSSH 也移植到了Windows 10中，ssh server可通过ConPTY与Windows命令程序连接，使得执行远程Windows主机的命令程序过程更为简洁。

Command-Line Remoting With ConPTY

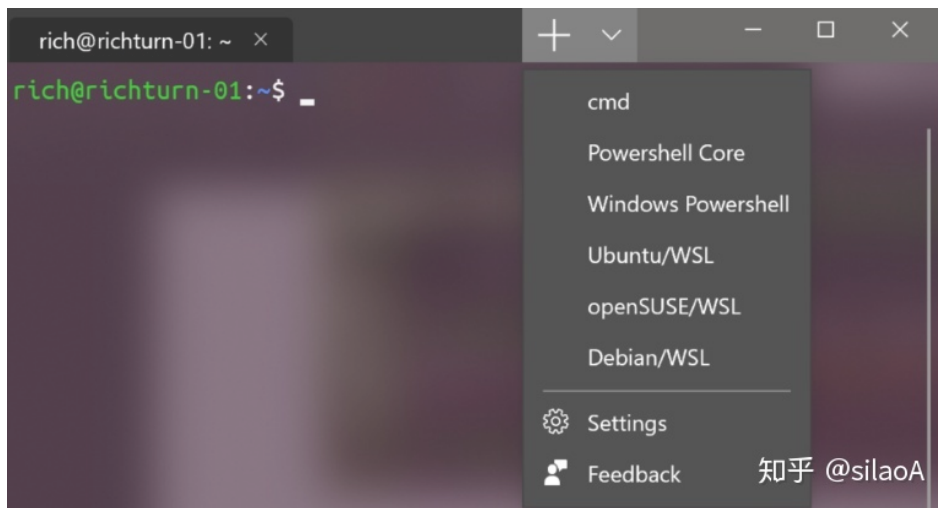


基于ConPTY的ssh server 图片源于微软官方博客

为了推广ConPTY，微软在WSL、VS、VS Code等产品中积极支持ConPTY，同时也在与第三方合作，如 ConEmu 开发团队。

0x03 Windows Terminal

Windows Terminal是微软出品的全新终端，是的，它用了ConPTY API！界面酷炫，支持连接到多个环境，一经官宣便迅速冲上人气榜首，目前仍是预览版状态。



Windows Terminal 图片源于微软官方博客

2019年build大会上，微软将Windows Console组件随同Windows Terminal开源，地址：[github.com/microsoft/te...](https://github.com/microsoft/terminal)

0x04 总结

Windows Console和POSIX的pty机制完全不同，为了补齐二者之间的差异性，ansicon、winpty项目打造的中间层，在不同方向上做出了很大努力，微软亲自打造的ConPTY基本填满两者间以往难以逾越的鸿沟，而且允许连接到任何终端窗口。基于ConPTY，微软还出品了酷炫的终端程序Windows Terminal。

可以预见，未来基于ConPTY和Windows Terminal加强版的第三方终端模拟器将大放异彩，Windows与Windows、Windows与UNIX/Linux主机之间远程执行命令程序也会十分常见。

参考

- [github.com/adoxa/ansicon...](https://github.com/adoxa/ansicon)
- [github.com/rprichard/wi...](https://github.com/rprichard/winpty)
- [Windows Command-Line: Introducing the Windows Pseudo Console \(ConPTY\).](#)

更多阅读

- [上一篇 Cygwin系列（十）：折腾终端1](#)
- [下一篇 Cygwin系列（十二）：了解X](#)
- [伪码人专栏目录导航 持续更新...](#)
- [Linux Cygwin知识库（一）：一文搞清控制台、终端、shell概念](#)
- [GNU Wget 爬虫？试一试](#)
- [Python操作Excel文件（0）：盘点](#)
- [silaoA的博客.https://silao.github.io](https://silao.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silaoA或联系公众号伪码人（We_Coder）。

最后，好书推荐。



Linux环境编程：从应用到内核

京东

¥68.30

去购买 >



发布于 2020-01-14 08:30，编辑于 2020-11-27 13:22

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

Cygwin

Linux

Windows 10

写评论... | 你赞过作者的回答



还没有评论，发表第一个评论吧

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



Linux 漫游之旅

还有什么能阻挡我学习 Linux?