

作者：道哥，10+年的嵌入式开发老兵。

公众号：**【IOT物联网小镇】**，专注于：C/C++、Linux操作系统、应用程序设计、物联网、单片机和嵌入式开发等领域。 公众号回复**【书籍】**，获取 Linux、嵌入式领域经典书籍。

转载：欢迎转载文章，转载需注明出处。

[什么是 WSL?](#)

[什么是 WSL2?](#)

[激活 WSL 服务](#)

[安装 Ubuntu-18.04 操作系统](#)

[通过 ssh 远程登录](#)

[编译 Hello,World!](#)

作为一名嵌入式软件开发工程师，在 Linux 系统下写代码、编译、调试是避免不了的事情。

但是[很少](#)有同学会直接拿一台 Linux 真机来作为工作平台，除非是那种利用远程服务器来编译的场景。

更普遍的使用场景是：在一台 Windows/Mac 系统中，安装[虚拟机](#)，然后在虚拟机中安装 Linux 操作系统。

Windows 系统中常用的虚拟机有：Virtualbox、VMWare，Mac 系统中一般都是用 Parallels Desktop，速度确实比较快！

在很久以前，我参与的项目主要是[物联网网关](#)，只需要编译 Linux 系统下的可执行程序，因此在日常工作中，都是直接在 Ubuntu 虚拟机中敲代码(编辑器：VSCode、SubLime)、编译、调试。

后来需要开发[跨平台](#)的应用，开发模式就变成了：

1. 敲代码：indows 系统中中的 Visual Studio;
2. 编译 Windows 应用程序: 直接用 Visual Studio 中的 VC 编译器来编译;
3. 编译 Linux 应用程序：通过远程部署的方式，把代码同步到 Ubuntu 虚拟机中，然后远程编译、调试;

这样的开发模式是属于比较常见的，只不过每次 Ubuntu 虚拟机比较耗费资源，启动比较慢而已！

编写跨平台的代码，使用 CMake 工具来管理构建过程，也是非常方便的。

之前写过 2 篇相关的文章：

[《使用 cmake 来搭建跨平台的应用程序框架：C语言版本》](#)

[《使用 cmake 来搭建跨平台的应用程序框架：C++版本》](#)

这 2 篇总结，算是一个简单的工程模板吧。

平时如果我需要写一些简单的 Demo 发给别人，我就会直接复制一下，然后添加上演示代码就可以了。

说了这么多，都是在描述跨平台开发的工作场景，或者说是开发模式。

其实，在 Windows 系统中，还有一种安装 Linux 系统的方式，这就是 WSL/WSL2！

下面，我就把自己的安装、配置过程与大家分享一下！

什么是 WSL?

这部分没有什么好掰扯的，直接从微软官网摘录如下：

WSL(Windows Subsystem for Linux): Windows 系统中的一个子系统，在这个子系统上可以运行 Linux 操作系统。

可以让开发人员直接在 Windows 上按原样运行 GNU/Linux 环境（包括大多数命令行工具、实用工具和应用程序），且不会产生传统虚拟机或双启动设置开销。

您可以：

- 在 [Microsoft Store](#) 中选择你偏好的 GNU/Linux 分发版。
- 运行常用的命令行软件工具（例如 `grep`、`sed`、`awk`）或其他 ELF-64 二进制文件。
- 运行 Bash shell 脚本和 GNU/Linux 命令行应用程序，包括：
 - 工具：vim、emacs、tmux
 - 语言：NodeJS、Javascript、Python、Ruby、C/C++、C# 与 F#、Rust、Go 等。
 - 服务：SSHD、MySQL、Apache、lighttpd、MongoDB、PostgreSQL。
- 使用自己的 GNU/Linux 分发包管理器安装其他软件。
- 使用类似于 Unix 的命令行 shell 调用 Windows 应用程序。
- 在 Windows 上调用 GNU/Linux 应用程序。

什么是 WSL2?

WSL2 是适用于 Linux 的 Windows 子系统体系结构的一个新版本，它是对基础体系结构的一次重大改造。

它使用虚拟化技术和 Linux 内核来实现其新功能，主要目标是提高文件系统性能和添加完全的系统调用兼容性。

个人理解：

WSL：并不是一个真正的 Linux 操作系统，仅仅是 Linux 应用程序与 Windows 操作系统之间的一个适配层。

在这个适配层之上，可以运行 Linux 应用程序，有点类似于以前的 cygwin 的方式。

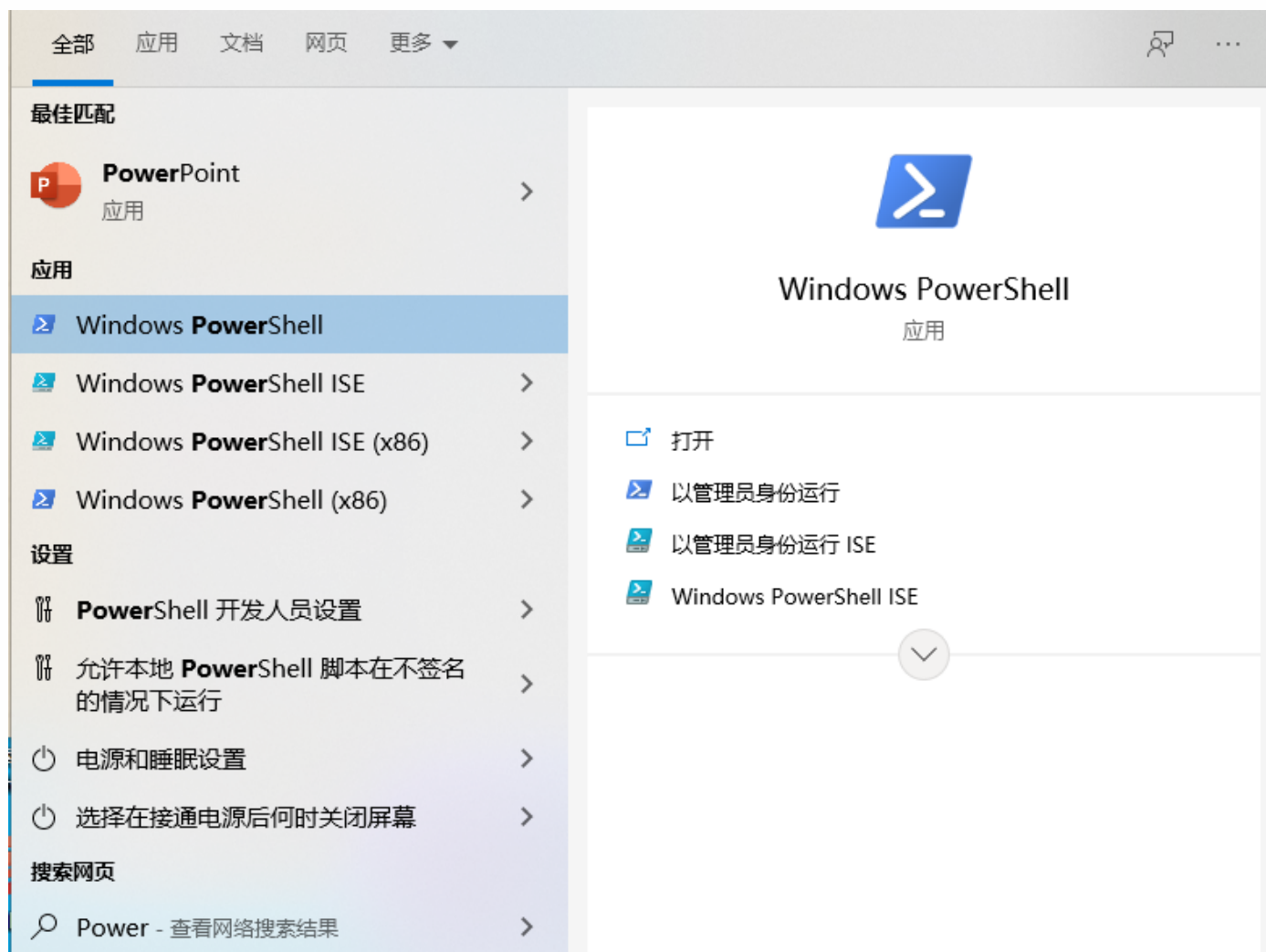
WSL2：它就是一个虚拟机，类似于 Virtual Box，在这个虚拟机之上，运行一个完整的 Linux 操作系统。

相对于 Virtual Box、VMWare 来说，WSL2 提供更全面的兼容性、与 Windows 系统的互操作性更好、运行速度更快、占用系统资源更少。

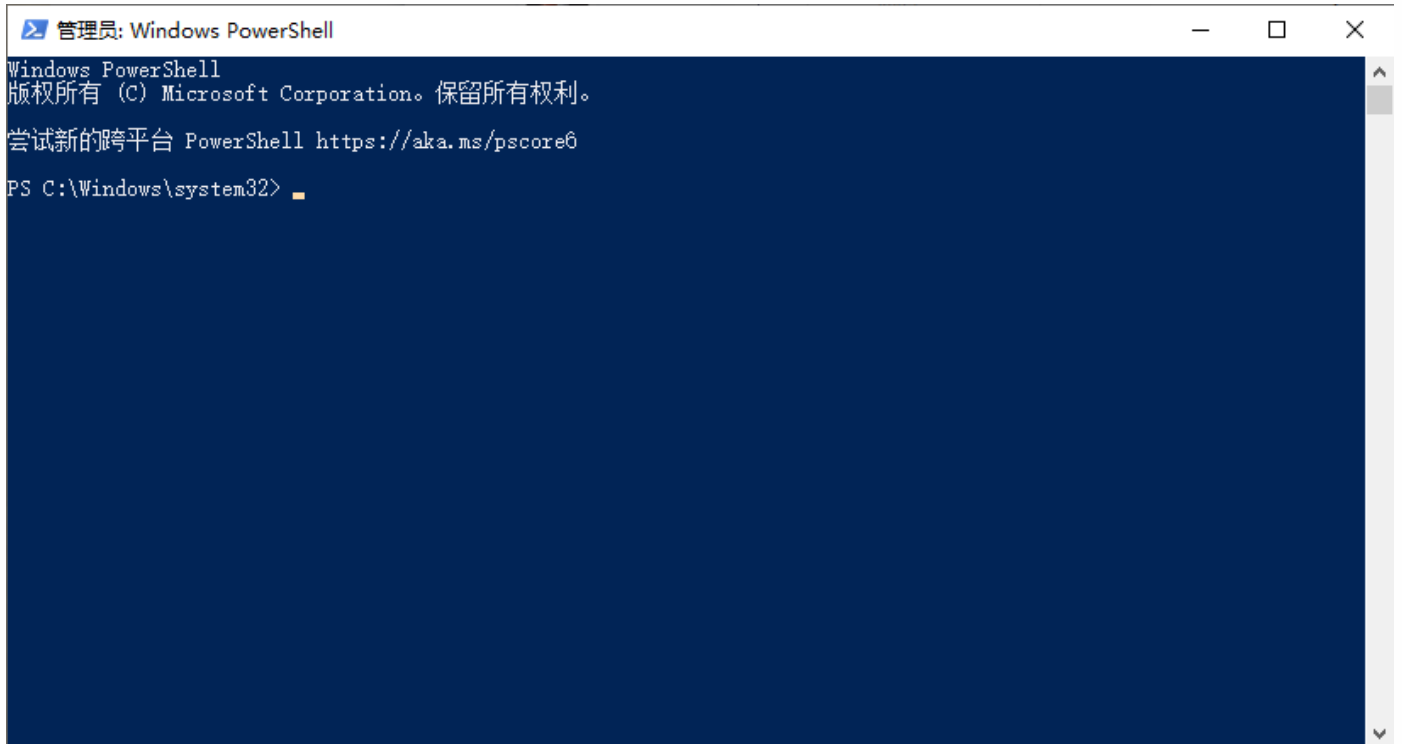
激活 WSL 服务

按 Win+X, 启动 [Windows PowerShell \(管理员\)](#), 注意: 是带有[管理员](#)的这个啊:

或者直接在搜索窗口中输入 [Power](#), 然后选择以管理员方式运行:



PowerShell 的窗口是:



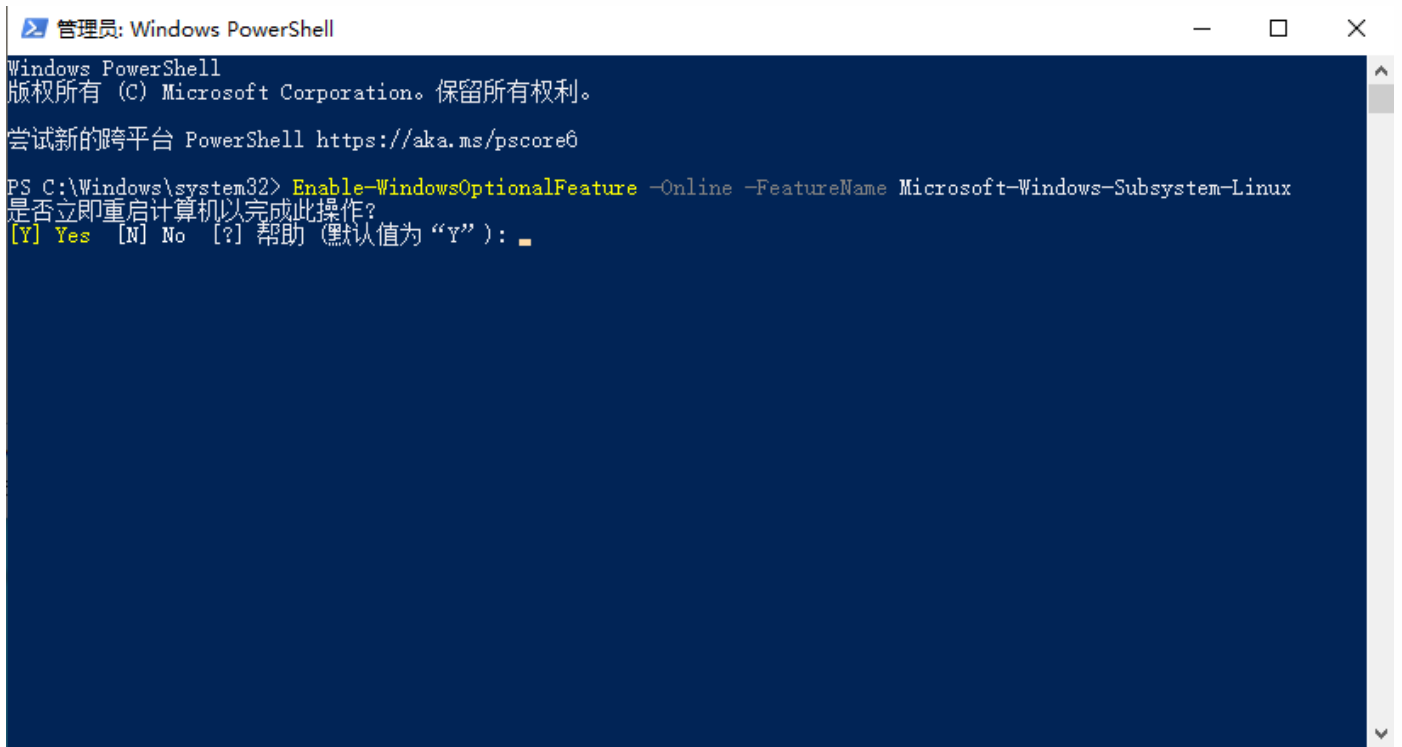
```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
尝试新的跨平台 PowerShell https://aka.ms/pscore6
PS C:\Windows\system32>
```

可以把 PowerShell 理解成升级版的、功能更强劲的 cmd。

在 PowerShell 窗口中，输入如下指令来激活 WSL 服务：

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

此时，提示需要重启系统(不用说，肯定选择 Y)：



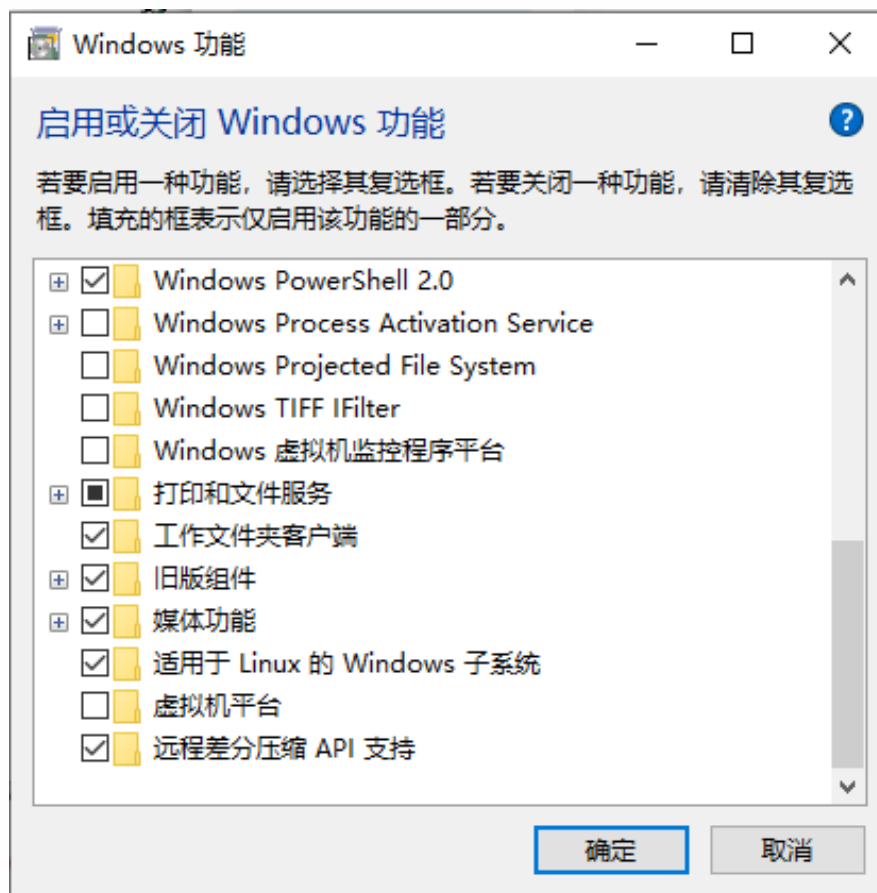
```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
尝试新的跨平台 PowerShell https://aka.ms/pscore6
PS C:\Windows\system32> Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
是否立即重启计算机以完成此操作?
[Y] Yes [N] No [?] 帮助 (默认值为“Y”):
```

电脑重启之后，按下 Win + R，调出命令输入窗口。输入指令 `appwiz.cpl`。

点击左侧的 **【启动或关闭 Windows 功能】**：



弹出下面这个窗口：

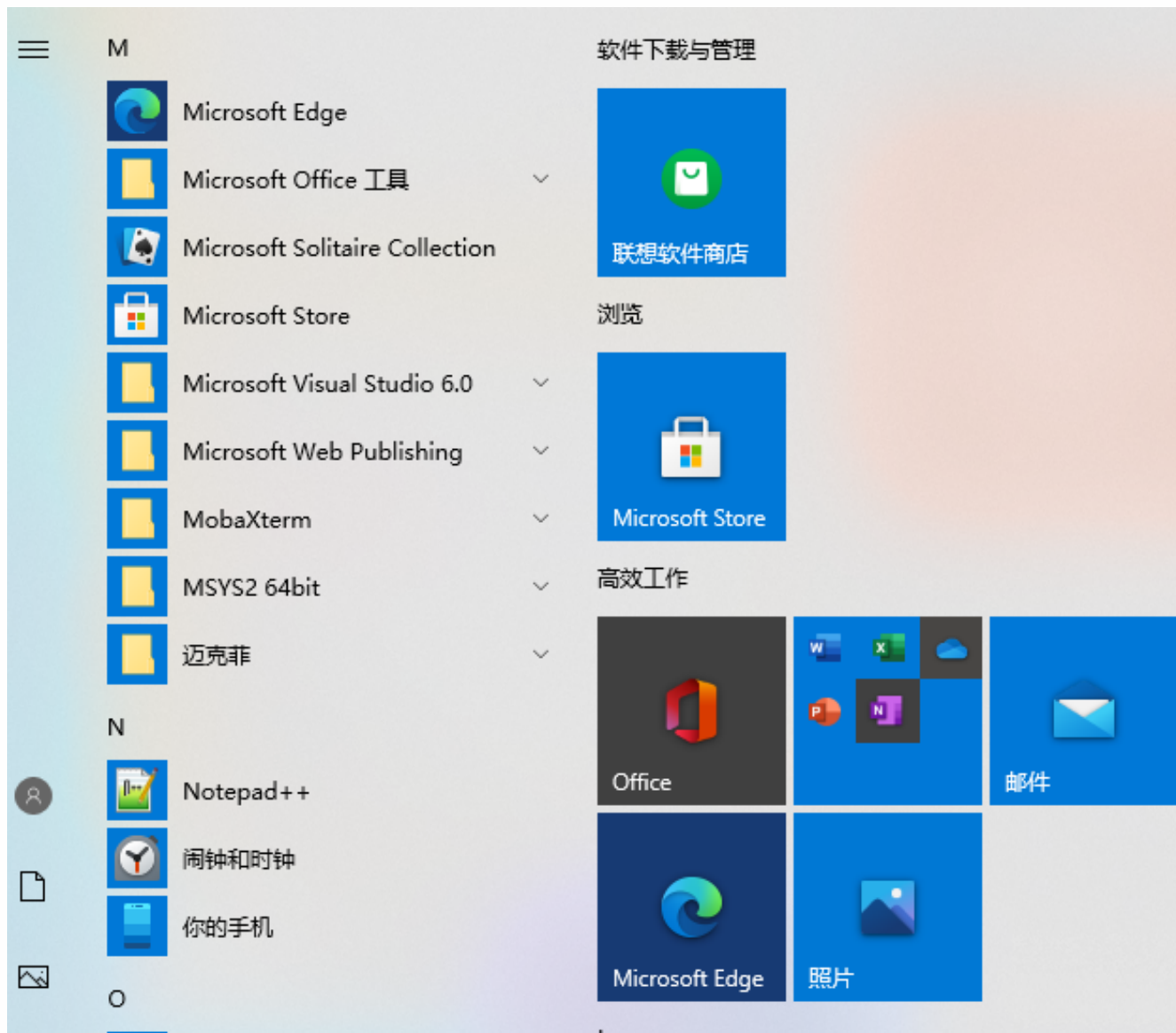


可以看到：【适用于 Linux 的 Windows 子系统】这一栏已经被勾选了，说明 WSL 服务已经启动。

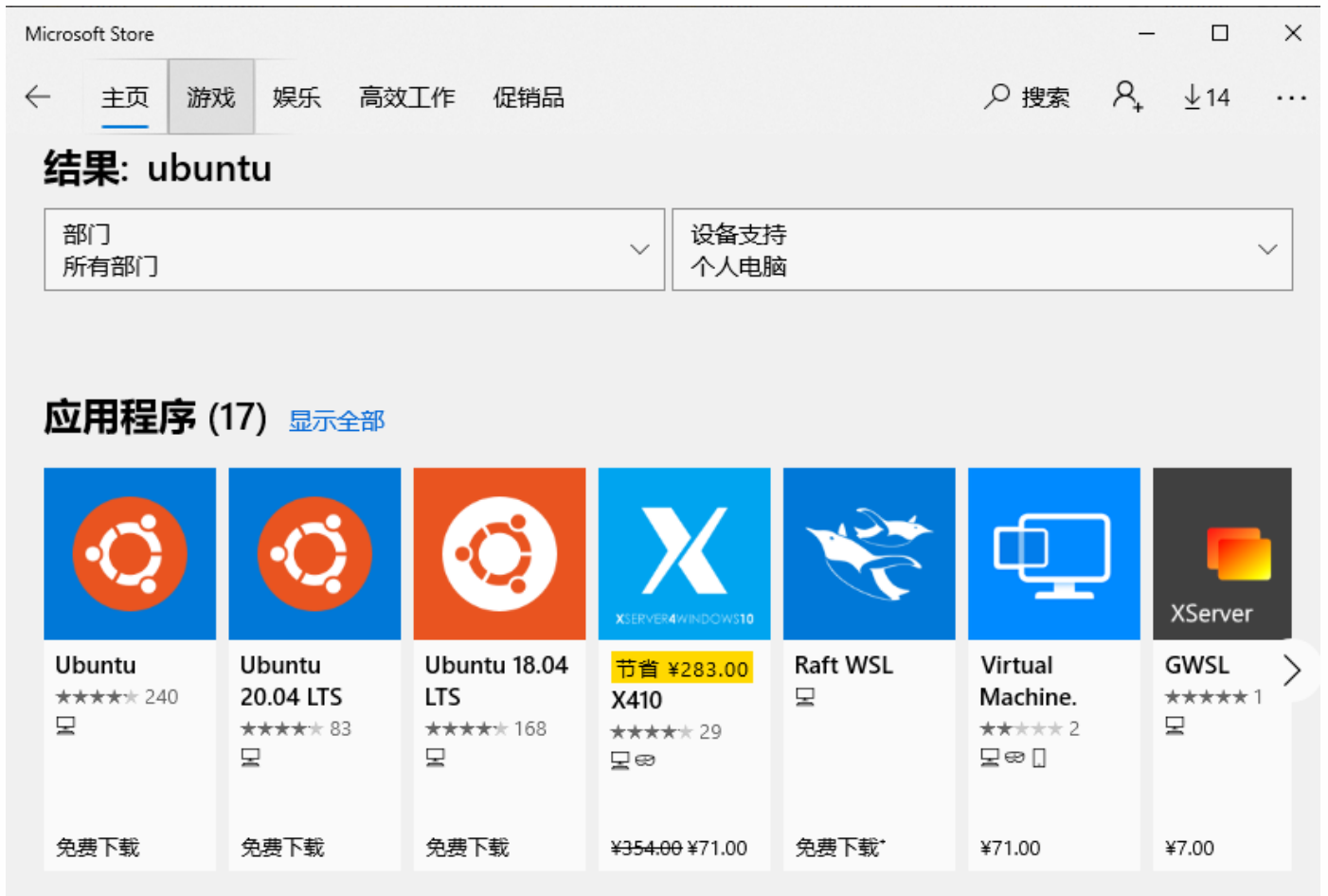
接下来要做的事情，就是安装 Ubuntu 操作系统。

安装 Ubuntu-18.04 操作系统

启动 【Windows Store】：



在右上角的搜索栏中，输入：Ubuntu



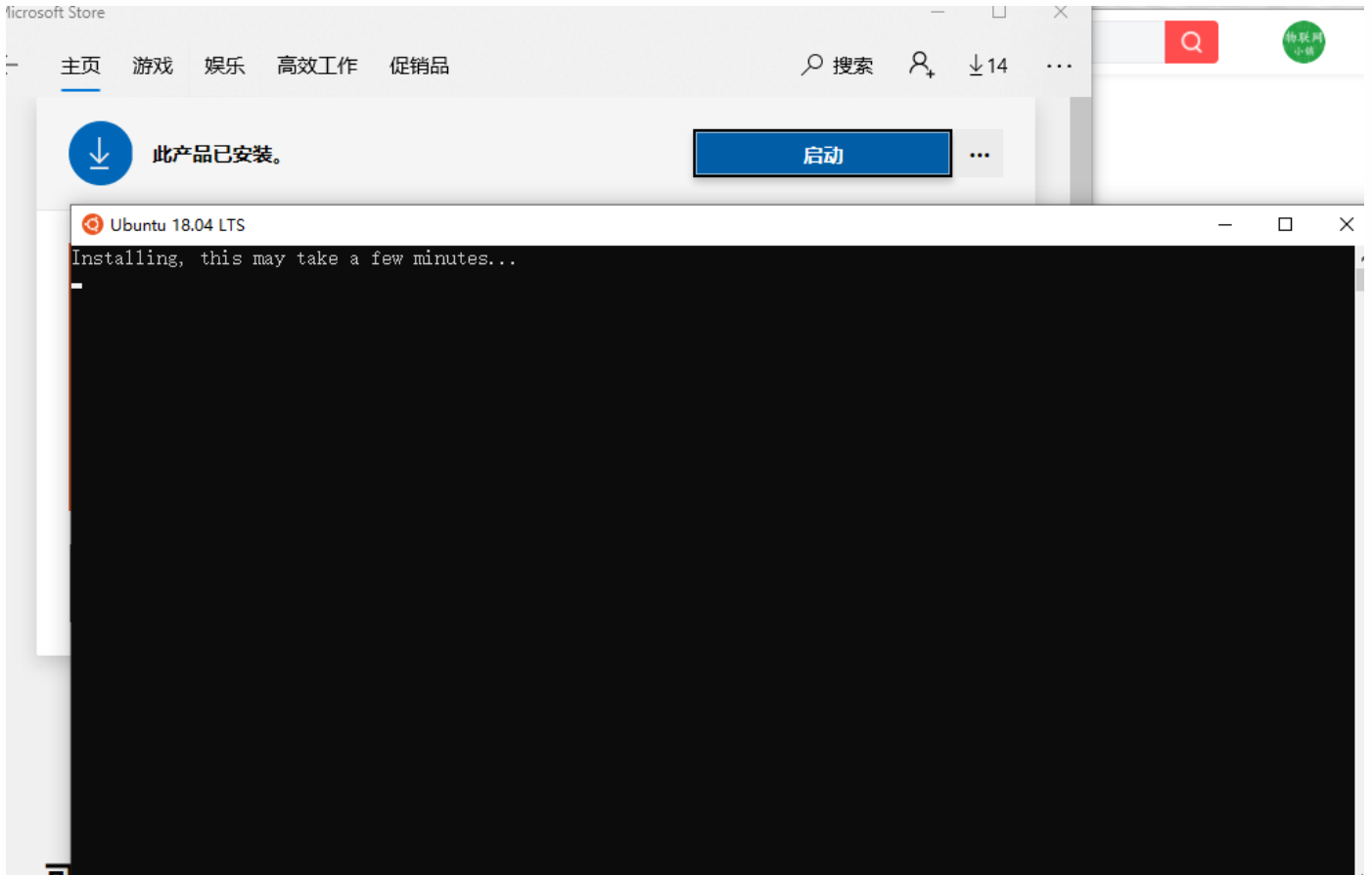
我们这里就选择 Ubuntu-18.04 吧，这也是我工作中使用的版本。



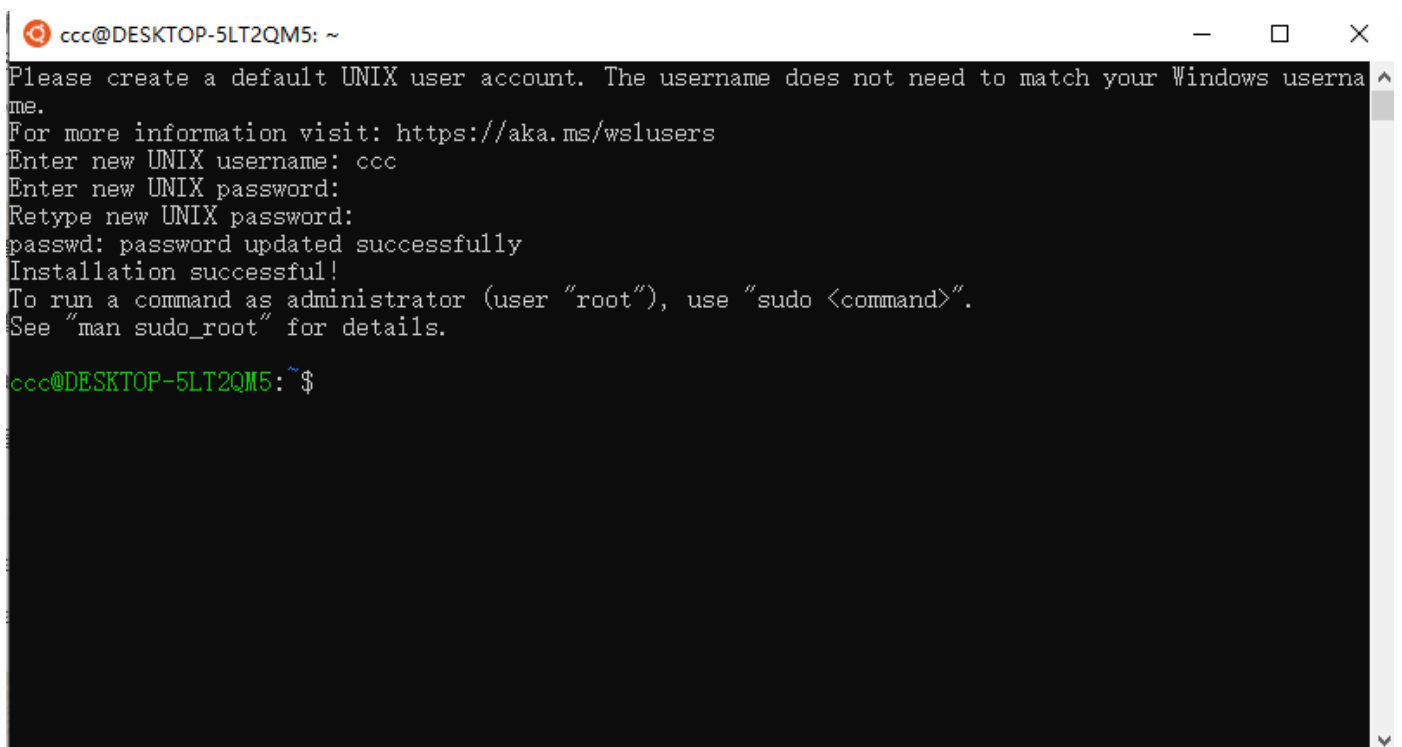
下载速度挺快的：



安装完成之后，点击【启动】按钮：



第一次打开速度稍微慢一些，大约 1 分钟左右吧，提示设置用户名、密码，然后就进入我们熟悉的窗口了：



这样，Ubuntu-18.04 系统就安装好了！

我一般在系统安装之后，会把 root 用户的密码也设置一下，以后总会需要使用 root 身份来登录系统的。

```
$ sudo passwd
```

```
ccc@DESKTOP-5LT2QM5:/etc/apt$ sudo passwd
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
ccc@DESKTOP-5LT2QM5:/etc/apt$
```

系统安装之后，第一件事情一般都是[更换软件源](#)。

可以选择阿里、清华或者中科大的源，这里就直接使用清华的软件源：

```
$ cd /etc/apt
$ sudo cp sources.list source.list.bak
$ sudo vim sources.list
```

把下面软件源复制、粘贴进去：

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe
multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted
universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted
universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted
universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted
universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main
restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted
universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main
restricted universe multiverse
```

保存之后，执行更新指令：

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

这个步骤，请同学们耐心等待，稍安勿躁。出现提示输入，选择 Y 就行了。

通过 ssh 远程登录

如果您觉得这个黑乎乎的窗口有点丑，那么可以通过 Windows 系统中的 Xshell 工具来远程登录(ssh)。

为了这个目的，我们需要在 Ubuntu-18.04 中重新安装 sshd 服务：

```
$ sudo apt purge openssh-server  
$ sudo apt install openssh-server
```

安装完毕之后，需要修改配置文件，以运行远程登录：

```
$ sudo vim /etc/ssh/sshd_config
```

修改1：把 13 行的 #Port 22 改为 Port 22。

修改2：把 32 行的 #PermitRootLogin prohibit-password 改为 PermitRootLogin yes。

修改3：把 56 行的 #PasswordAuthentication yes 改为 PasswordAuthentication yes。

修改之后，启动 sshd 服务：

```
$ sudo service ssh restart
```

```
ccc@DESKTOP-5LT2QM5:/etc/apt$ sudo service ssh restart  
* Restarting OpenBSD Secure Shell server sshd
```

[OK]

此时，启动 Windows 系统中的 Xshell 来远程登录一下，主机名只需要填写：localhost 即可：

新建会话属性

?

×

类别(C):

连接

- 用户身份验证
 - 登录提示符
 - 登录脚本
- SSH
 - 安全性
 - 隧道
 - SFTP
- TELNET
- RLOGIN
- 串口
- 代理
- 保持活动状态

终端

- 键盘
- VT 模式
- 高级

外观

- 窗口
- 突出

高级

- 跟踪
- 响铃
- 日志记录

文件传输

- X/YMODEM
- ZMODEM

连接

常规

名称(N): wsl

🖨️

协议(P): SSH

主机(H): localhost

端口号(O): 22

说明(D):

重新连接

☐ 连接异常关闭时自动重新连接(A)

间隔(V): 0 秒 限制(L): 0 分钟

TCP选项

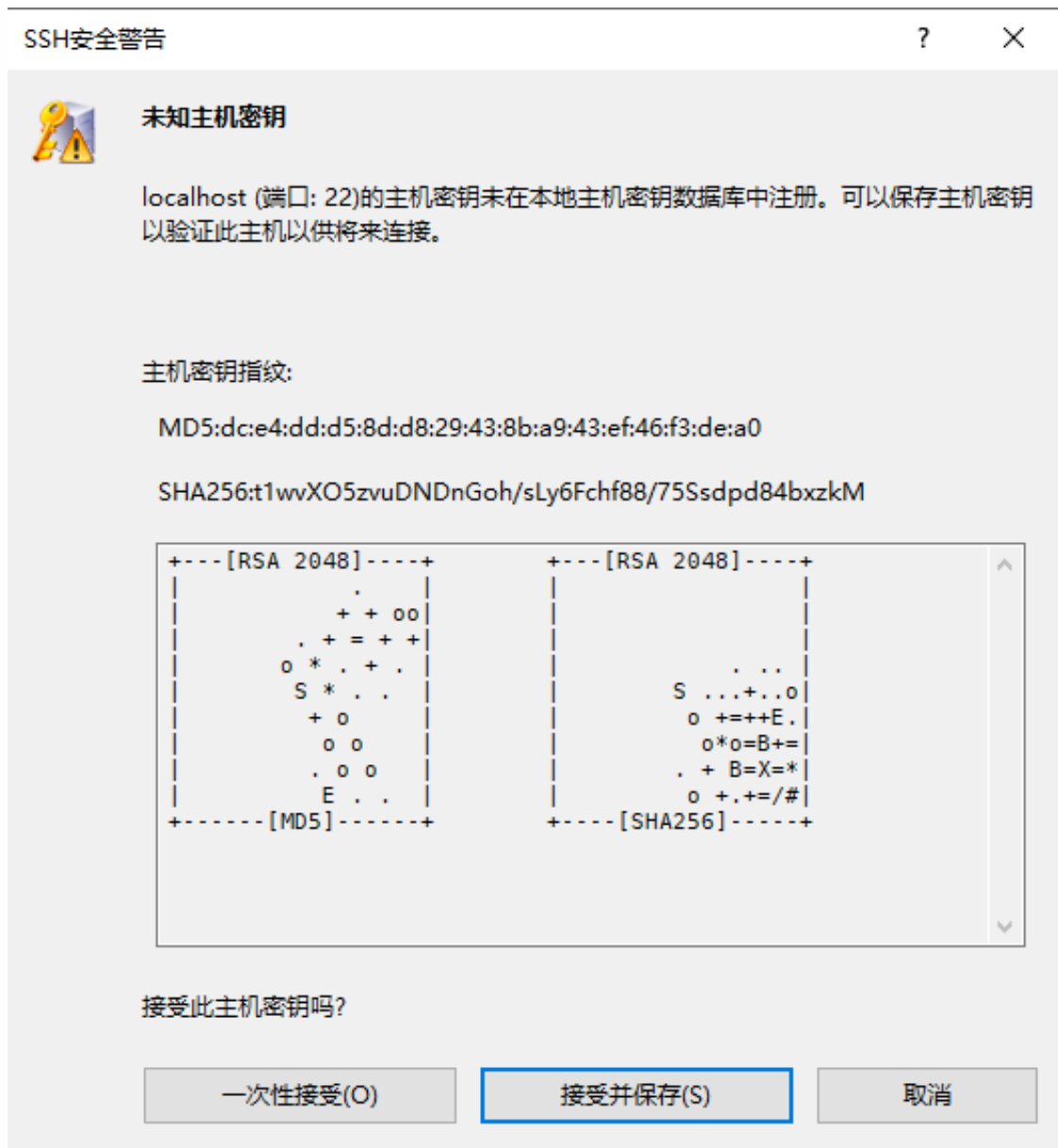
☐ 使用Nagle算法(U)

连接

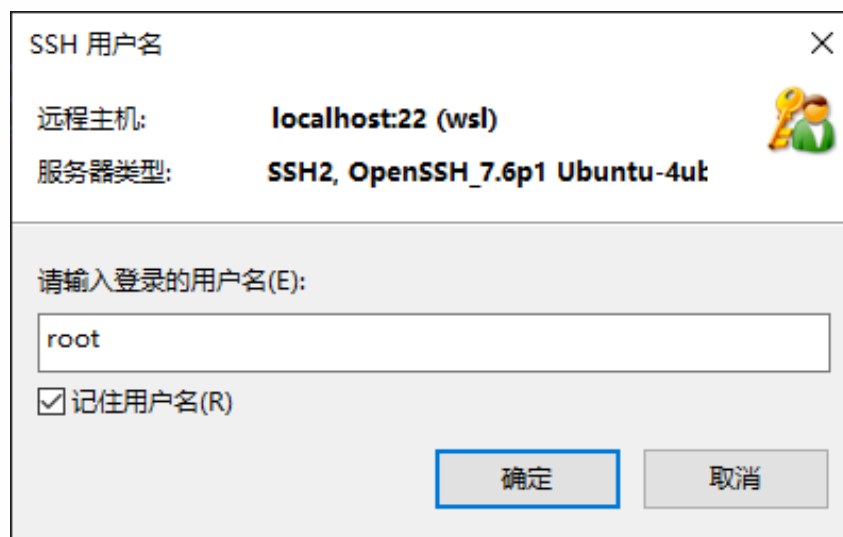
确定

取消

建立链接的时候，需要保存秘钥：



然后输入用户名和密码:




SSH用户身份验证?×

远程主机:localhost:22 (wsl)

登录名:root

服务器类型:SSH2, OpenSSH_7.6p1 Ubuntu-4ubuntu0.3



请在下面选择恰当的身份验证方法并提供登录所需的信息。

☒ Password(P)

密码(W):

☐ Public Key(U)

用户密钥(K):

密码(H):


☐ Keyboard Interactive(I)

使用键盘输入用户身份验证。

☒ 记住密码(R)

如果出现下面这个提示，先不用理会它，直接选择【否】，这是与 Linux 的图形窗口相关的，后面会解释。

Xshell×



需要Xmanager软件来处理X11转发请求。
当你安装Xmanager时，你可以直接在Windows中使用从Xshell运行的X11程序，例如xterm和gnome-terminal。

你想立即下载Xmanager试用版吗？

(要关闭此消息，只需关闭会话属性 - > 连接 - > SSH->隧道页面中的X11转发选项。)

此时，就可以在 Xshell 中愉快的操作 Ubuntu 的命令行窗口了。

```
ccc@DESKTOP-5LT2QM5: ~ $ ll
total 24
drwxr-xr-x 1 ccc ccc 4096 Jul 19 22:20 ./
drwxr-xr-x 1 root root 4096 Jul 19 21:50 ../
-rw-r--r-- 1 ccc ccc 220 Jul 19 21:50 .bash_logout
-rw-r--r-- 1 ccc ccc 3771 Jul 19 21:50 .bashrc
drwxr-xr-x 1 ccc ccc 4096 Jul 19 21:50 .cache/
drwx----- 1 ccc ccc 4096 Jul 19 21:50 .config/
-rw-r--r-- 1 ccc ccc 807 Jul 19 21:50 .profile
-rw-r--r-- 1 ccc ccc 0 Jul 19 21:54 .sudo_as_admin_successful
-rw----- 1 root root 12155 Jul 19 22:20 .viminfo
ccc@DESKTOP-5LT2QM5: ~ $
```

是不是感觉少了点什么东西？那些 Desktop、Documents、Picture 等默认文件夹怎么都不见了？

这个问题，后面在演示 WSL2 的时候就见分晓了！

编译 Hello,World!

最后，我们来编译一个应用程序。首先安装 gcc 编译器：

```
$ sudo apt-get install gcc
```

再写一个 hello.c 文件：

```
#include <stdio.h>

int main()
{
    printf("Hello,World! \n");
    return 0;
}
```

然后编译、执行：

```
ccc@DESKTOP-5LT2QM5:~/tmp$ gcc hello.c -o hello
ccc@DESKTOP-5LT2QM5:~/tmp$ ./hello
Hello,World!
```

完美！

----- End -----

接下来要介绍的就是 WSL2 以及 图形窗口了，过程我已经验证完毕，等整理好之后再分享给大家，谢谢！

推荐阅读

- 【1】C语言指针-从底层原理到花式技巧，用图文和代码帮你讲解透彻
- 【2】一步步分析-如何用C实现面向对象编程
- 【3】原来gdb的底层调试原理这么简单
- 【4】内联汇编很可怕吗？看完这篇文章，终结它！

其他系列专辑：[精选文章](#)、[C语言](#)、[Linux操作系统](#)、[应用程序设计](#)、[物联网](#)



微信搜一搜

Q IOT物联网小镇

星标公众号，能更快找到我！

C/C++、物联网、嵌入式、Lua语言 Linux 操作系统、应用程序开发设计



扫码关注公众号



道哥 个人微信

喜欢请**分享**，满意点个**赞**，最后点**在看**。