

点击 IOT物联网小镇

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。

安装 WSL2 的前提条件

安装 WSL2

安装桌面环境

安装客户端

配置 DISPLAY 环境变量

安装服务端

启动客户端

中文支持

在上一篇文章中，我们主要聊了一下：在 Windows 系统中，安装 WSL 子系统。

文章在此：[WSL：在 Windows 系统中开发 Linux 程序的又一神器](#)

但是 WSL 只是长得像 Linux 而已，虽然在这个子系统中，我们可以执行 Linux 程序，但是到了底层，WSL 调用的还是 win32 API。

也就是说：[WSL 只是一个适配层](#)。

后来，微软终于放出了 WSL2 这个真正的 Linux 子系统，这是一个真正的 Linux 完整内核，与传统的虚拟机 VirtualBox、VMWare 一样，但是它与宿主机的互操作性更好。

为了完整性，我们就继续把这个流程给走完。

在这个过程中遇到的[问题](#)、需要的安装软件等[资源](#)，已经上传到[网盘](#)，在文末有下载地址。

安装 WSL2 的前提条件

在开始之前，我们先来看一下目前电脑中的 WSL 的版本。

在 Windows 命令行窗口，输入指令：wsl --list --verbose(或者输入简化版本：wsl -l -v)，得到结果：

	NAME	STATE	VERSION
*	Ubuntu-18.04	Stopped	1

可以看到我们上一篇文章中安装的 Ubuntu-18.04，而且 WSL 的版本是 1。

如果要升级到 WSL2，Windows 操作系统的**内核版本**必须满足一定的要求，如下：

若要更新到 WSL 2，需要运行 Windows 10。

- 对于 x64 系统：**版本 1903 或更高版本**，采用 **内部版本 18362 或更高版本**。
- 对于 ARM64 系统：**版本 2004 或更高版本**，采用 **内部版本 19041 或更高版本**。
- 低于 18362 的版本不支持 WSL 2。使用 [Windows Update 助手](#) 更新 Windows 版本。

若要检查 Windows 版本及内部版本号，选择 Windows 徽标键 + R，然后键入“winver”，选择“确定”。更新到“设置”菜单中的**最新 Windows 版本**。

① 备注

如果运行的是 Windows 10 版本1903 或 1909，请在 Windows 菜单中打开“设置”，导航到“更新和安全性”，然后选择“检查更新”。内部版本号必须是 18362.1049+ 或 18363.1049+，次要内部版本号需要高于 .1049。阅读详细信息：[WSL 2 即将支持 Windows 10 版本 1903 和 1909](#)。请参阅[疑难解答说明](#)。

我们就按照官方的要求，检查一下自己的电脑是否满足需求：选择 Windows 徽标键 **Win + R**，然后键入 winver，选择“确定”。

Microsoft Windows
版本 20H2 (OS 内部版本 19042.630)
© 2019 Microsoft Corporation。保留所有权利。

Windows 10 家庭中文版 操作系统及其用户界面受美国和其他国家/地区的商标法和其他待颁布或已颁布的知识产权法保护。

看来我的系统能满足要求，那就继续下一个步骤！

如果你的内核版本不满足这个需求，可以搜索关键字“[WIN10 获取内部预览版本](#)”，就可以找到解决方案。

安装 WSL2

继续以**管理员**身份启动 PowerShell，然后执行指令：

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart
```

```
管理员: Windows PowerShell
尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

部署映像服务和管理工具
版本: 10.0.19041.572

映像版本: 10.0.19042.630

启用一个或多个功能
[=====100.0%=====]
操作成功完成。
PS C:\WINDOWS\system32> █
```

此时，又到了重启电脑的时候！

重启之后，继续在 PowerShell 中，把 Ubuntu-18.04 设置为 WSL2 版本。

输入的指令是：

```
wsl --set-version Ubuntu-18.04 2
```

这个步骤可能会遇到这样的问题：

```
PS C:\WINDOWS\system32> wsl --set-version Ubuntu-18.04 2
正在进行转换，这可能需要几分钟时间...
有关与 WSL 2 的主要区别的信息，请访问 https://aka.ms/wsl2
WSL 2 需要更新其内核组件。有关信息，请访问 https://aka.ms/wsl2kernel
PS C:\WINDOWS\system32> █
```

解决方法：从微软的官网下载 [WSL2 Linux 内核升级包](#)，下载地址是：

[WSL2 Linux 内核升级包](#)

Step 4 - Download the Linux kernel update package


1. Download the latest package:

- [WSL2 Linux kernel update package for x64 machines](#) ↗

也可以从文末的网盘中下载

下载之后，一路 Next 安装即可。

此时，在 PowerShell 中重新执行 `wsl --set-version Ubuntu-18.04 2` 指令(需要等待一段时间，大概 1 分钟左右吧)：

 选择管理员: 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> wsl --set-version Ubuntu-18.04 2
正在进行转换，这可能需要几分钟时间...
有关与 WSL 2 的主要区别的信息，请访问 https://aka.ms/wsl2
转换完成。
PS C:\WINDOWS\system32>
```

同时，把 WSL 的默认版本改为 2： `wsl --set-default version 2`。

此时，我们再执行指令 `wsl -l -v` 来查看一下，就变成 WSL2 版本了：

```
PS C:\WINDOWS\system32> wsl -l -v
  NAME                STATE      VERSION
* Ubuntu-18.04        Stopped    2
PS C:\WINDOWS\system32>
```

现在，重新启动一下 Ubuntu-18.04 系统，虽然还是黑乎乎的窗口，但是它可是一个完整、纯正的 Linux 操作系统！

安装桌面环境

这部分的操作，参考：<https://autoize.com/xfce4-desktop-environment-and-x-server-for-ubuntu-on-wsl-2/>。

关于 Linux 的桌面环境，它与 Windows 桌面环境的理念有一些不一样。

在 Linux 系统中，桌面环境是一个[独立的应用程序](#)，不属于操作系统的一部分。

换句话说，即使没有桌面环境，也可以很愉快的玩 Linux 系统，比如那些服务器之类的设备。

Linux 中的桌面由[客户端](#)和[服务端](#)组成，它们之间通过一个协议来通信，俗称：X11。

服务端：负责图形的显示；

客户端：根据系统设置的 DISPLAY 环境变量，把显示图形的请求发送给服务端；

因此，为了让我们的 Ubuntu 显示图形界面，我们就需要安装一个服务端软件和客户端软件。

注意：这里的[服务端](#)软件是安装在 Windows 系统中的，[客户端](#)是安装在 Ubuntu-18.04 系统中的。

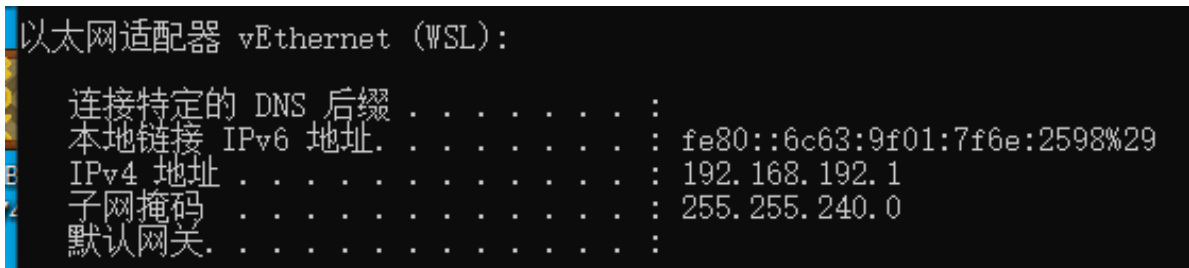
安装客户端

在 Ubuntu-18.04 的终端窗口中，执行如下指令安装客户端软件 xfce4：

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install xfce4
```

配置 DISPLAY 环境变量

因为 Ubuntu 中的客户端 xfce4 需要把显示请求，通过[网络](#)发送给 Windows 上的服务端，因此需要知道 Windows 系统的 IP 地址，在 Windows 系统中的命令行窗口中查看：ipconfig



```
以太网适配器 vEthernet (WSL):

   连接特定的 DNS 后缀 . . . . . : 
   本地链接 IPv6 地址. . . . . : fe80::6c63:9f01:7f6e:2598%29
   IPv4 地址 . . . . . : 192.168.192.1
   子网掩码 . . . . . : 255.255.240.0
   默认网关. . . . . :
```

然后在 Ubuntu-18.04 中把这个 IP 地址，配置到家目录下的 .bashrc 文件中：

```
$ cd
$ vi .bashrc
```

在文件末尾添加：

```
export DISPLAY=192.168.192.1:0
```

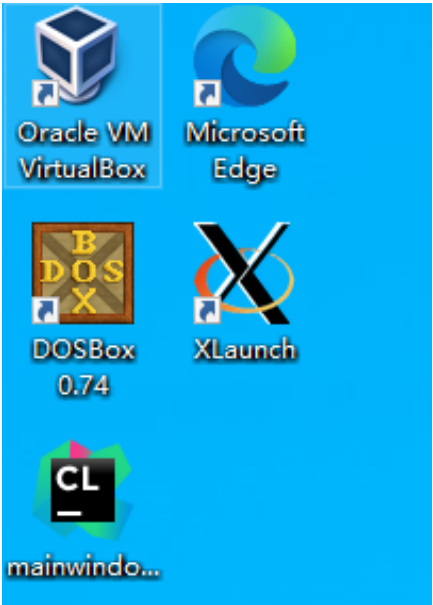
保存、关闭文件之后，执行：source ~/.bashrc 更新即可。

安装服务端

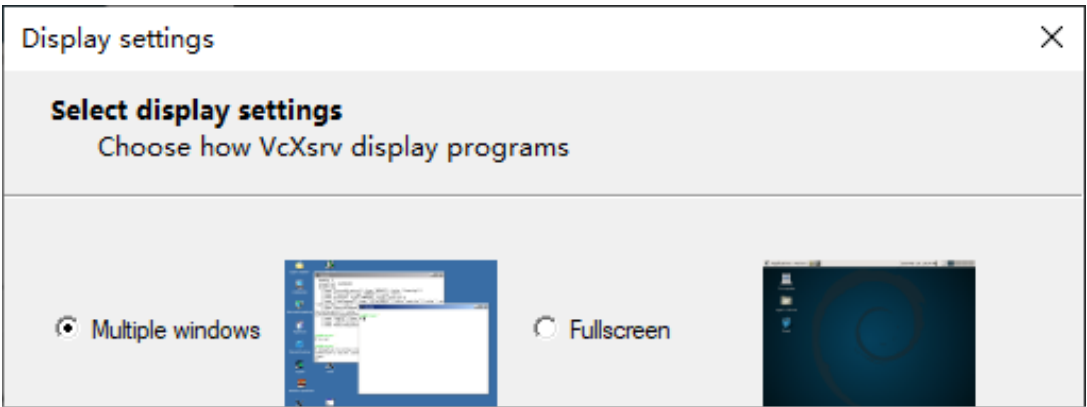
图形界面有 VcXsrv 和 Xming 两款软件，我们选择前面的 VcXsrv。

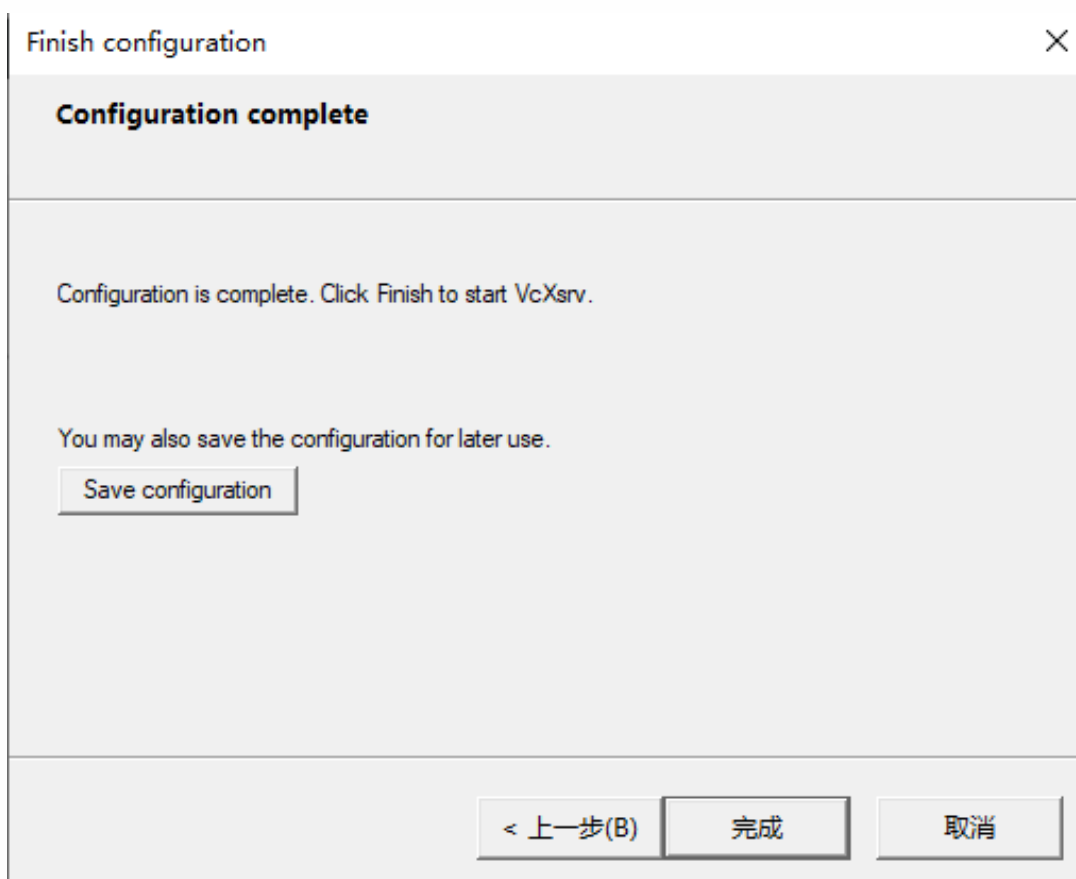
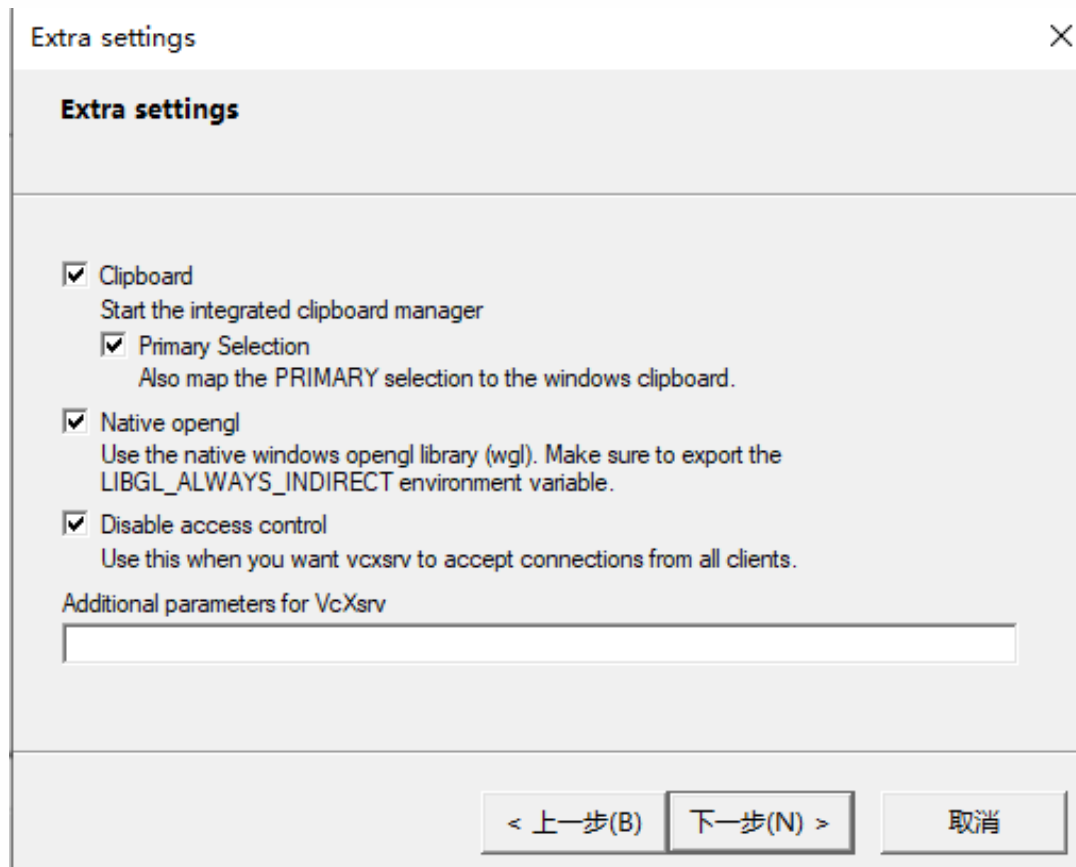
同样的，我已经把 [vcxsrv-64.1.20.9.0.installer.exe](#) 放在网盘里，下载后一路 Next 安装。

安装之后，启动 Windows 桌面上的 XLaunch 程序：



然后，按照参考文档中的配置，如下：





在最后一个步骤，可以把配置信息保存在桌面上。

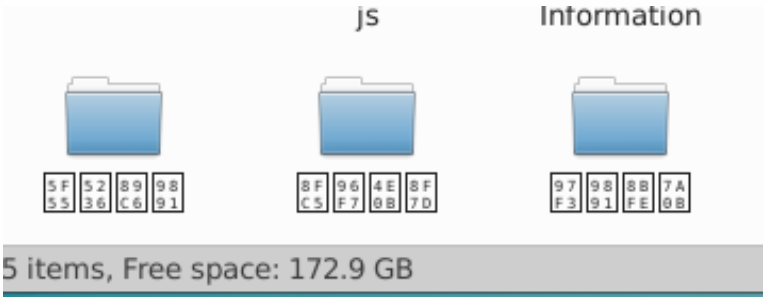
启动客户端

在 Ubuntu-18.04 系统中的命令行窗口，执行指令：`startxfce4`：

此刻，神器的事情就发生了：你的 Windows 系统中是不是就跳出了下面这个窗口：

可以打开桌面上 File System，是不是看到了很多熟悉的身影？

在打开 mnt 目录，是不是看到了 Windows 系统中的盘符？

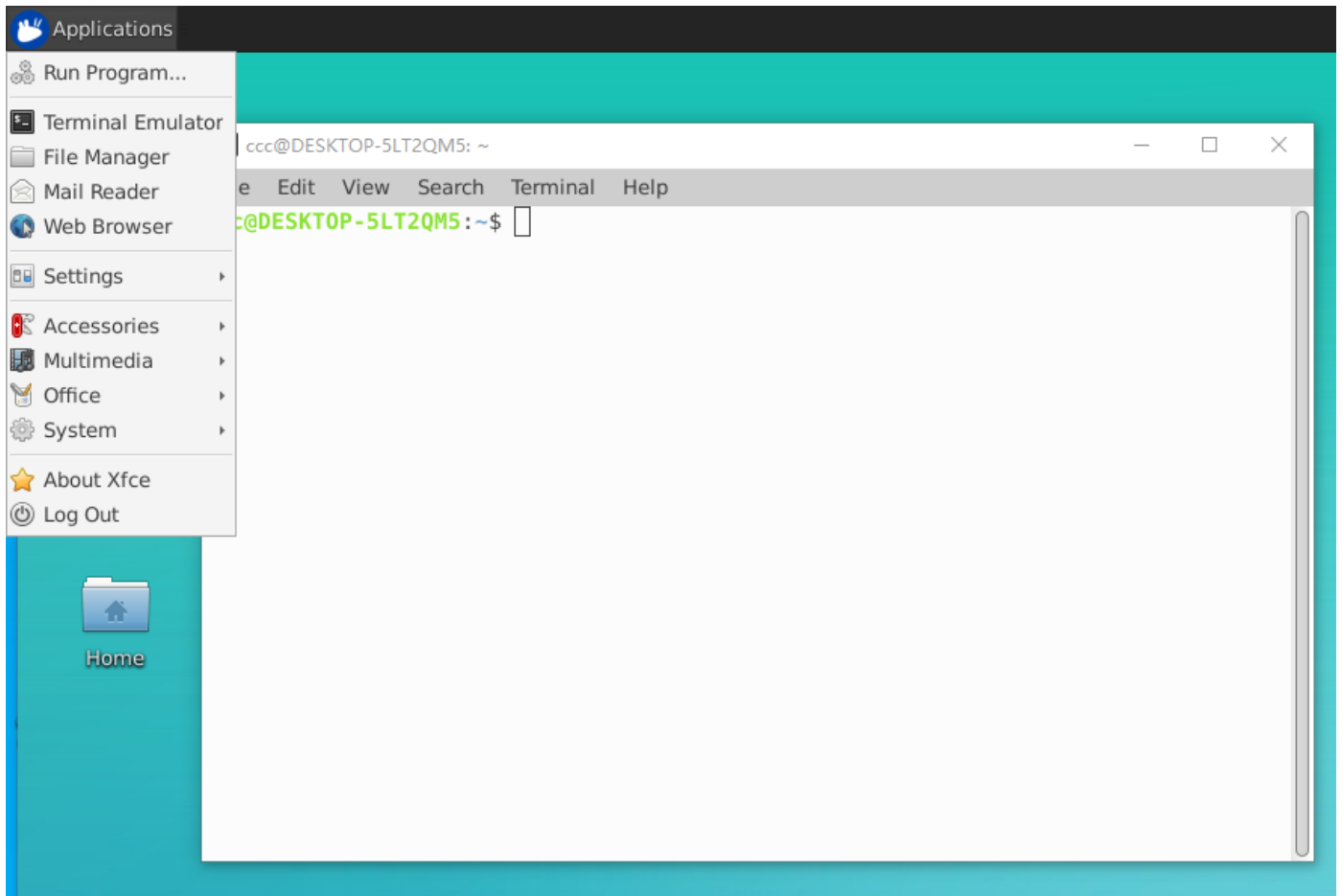


但是，中文信息显示的都是乱码！那我们就来安装中文语言支持包。

中文支持

在 Ubuntu-18.04 系统命令行窗口，设置 locale:

此时，我们可以使用桌面环境中的终端窗口，如下图所示：



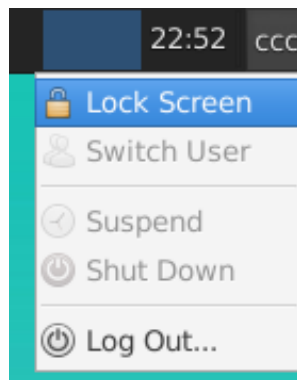
```
sudo vi /etc/locale.gen
```

把下图中的蓝色这一行复制，然后去掉前面的 # 和 空格，然后保存、退出。

然后在终端中执行指令： `sudo locale-gen`

```
ccc@DESKTOP-5LT2QM5: ~  
File Edit View Search Terminal Help  
ccc@DESKTOP-5LT2QM5:~$ sudo vi /etc/locale-gen  
[sudo] password for ccc:  
ccc@DESKTOP-5LT2QM5:~$ sudo vi /etc/locale.gen  
ccc@DESKTOP-5LT2QM5:~$ sudo locale-gen  
Generating locales (this might take a while)...  
  en_US.UTF-8... done  
  zh_CN.UTF-8... done  
Generation complete.  
ccc@DESKTOP-5LT2QM5:~$
```

单机右上角的用户名，选择 **【Log Out..】**



此时，Ubuntu 虚拟机的 WSL 窗口中的 xfce4 客户端会退出，再次执行 startxfce4 启动客户端：

再次，在桌面中的终端窗口中，把 locale 改为中文：

```
sudo vi /etc/default/locale
```

把 LANG=C.UTF-8 改为 LANG=zh_CN.UTF-8

然后，把下面这个几个中文语言包给它安装进去：

```
sudo apt install language-pack-zh-hans  
sudo apt install language-pack-gnome-zh-hans  
sudo apt install language-pack-kde-zh-hans
```

此刻，你再看一下刚才的那些乱码，是不是已经正确显示中文了？

----- End -----

接下来，你还可以安装各种 IDE，探索它们是如何共享文件的...

文中需要安装的软件，已经上传到网盘中了。

请在[公众号后台](#)回复721(不是留言区啊)，即可收到下载地址，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。