ubuntu开机脚本自启动(含ROS程序启动)方法

原创 北方南方2020 2020-05-31 15:59:59 3225 收藏 62 版权

分类专栏: ubuntu常用操作 文章标签: ubuntu

ubuntu常用操作 专栏收录该内容 0 订阅 1 篇文章 订阅专栏

ubuntu 开机脚本自启动(含ROS程序启动)方法

```
    平时有用到一些机器人或者带有ubuntu系统的移动设备,比如NVIDIA TX2,NVIDIA XAVIA,
Jetson NANO,秒算,工控机等等,写好了程序后,如果想要运行,开机之后还得连接键盘鼠标,
显示器,等程序运行之后还需要重新拔掉键盘鼠标显示器等等,步骤繁琐是一部分,还有可能
因为反应不及时发生危险,尤其对于无人机,强扭矩的机器人等设备.之前在网上也找了很多,
各种方法都有,这里总结一下自己常用的,方便日后查找.
        如果哪里写得不正确的,还请各位大佬指正,谢谢!!!!
```

操作步骤

A.普通执行文件

- **1.首先创建脚本文件**
  如下面的start.sh,这里以简单的hellow CSDN!为例

```
#! /bin/bash

cd /home/xxx/Desktop/CSDN/test_demo/build; #填写自己的可执行文件所在目录

./hellow_world    #运行可执行文件
wait
exit 0
```

- **2.给脚本添加可执行权限**
  因为刚创建的.sh文件是不可执行的,需要添加权限才可执行,添加方法很见单,使用chmod命令即可
  `chmod +x start.sh`

- **3.在终端中运行脚本进行测试**
  打开终端,进入脚本所在路径,运行./start.sh

```
$ ./start.sh
Hellow CSDN!
```

终端输出正常的运行结果,测试有效.

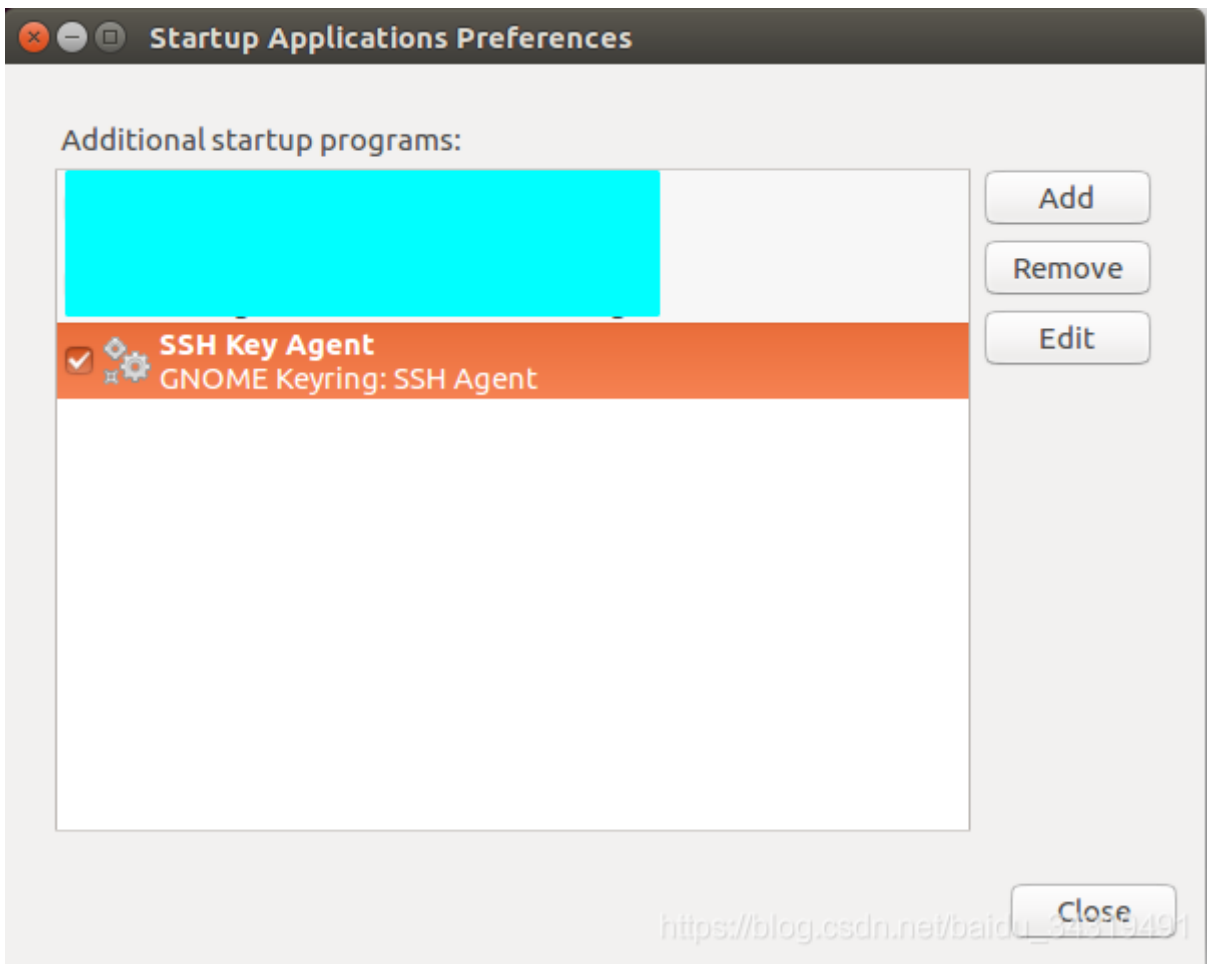
- **4.在新的终端运行**
  在之前的步骤之后,即使添加到了开机自启动,就算运行,也不能看出来是否真的在运行,甚至可能就没有在运行,也可能运行之后瞬间结束,为此,可以**采用打开一个新终端加延时**的方式运行,修改 `start.sh` 脚本:

```
#! /bin/bash
gnome-terminal -x bash -c "cd /home/xxx/Desktop/CSDN/test_demo/build;./hellow_csdn; sleep 5"
wait
exit 0
```

其中 `gnome-terminal -x bash -c` 会打开一个新的bash终端并执行c后面的语句,语句用';',隔开,最后的sleep 5表示延迟5秒退出,这样就可以在新的终端看到运行效果,添加这个延时是为了方便验证是否启动程序或者节点
!注意,如果是ubuntu 18.04的系统,
gnome-terminal -x bash -c需要改写成gnome-terminal – bash -c

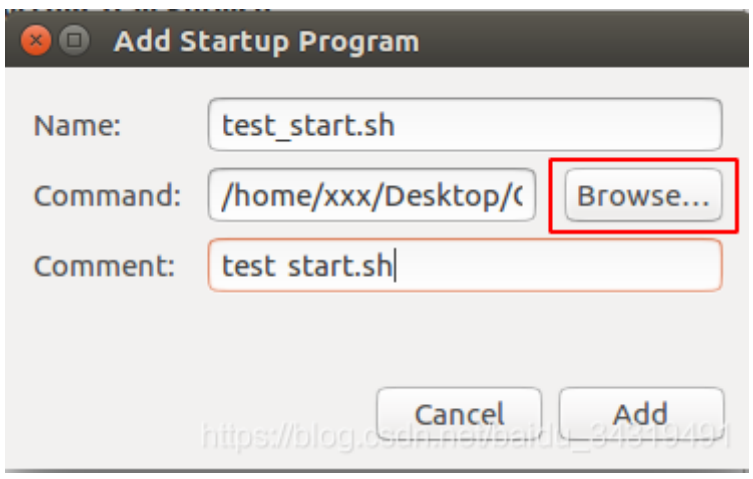
在完成了上述步骤之后,可以将脚本添加到ubuntu开机启动管理器里头,具体操作如下:

- **1.打开ubuntu启动管理器**
  在终端输入 `gnome-session-properties` 运行,会打开一个界面,这里记录着系统上层应用的启动软件:

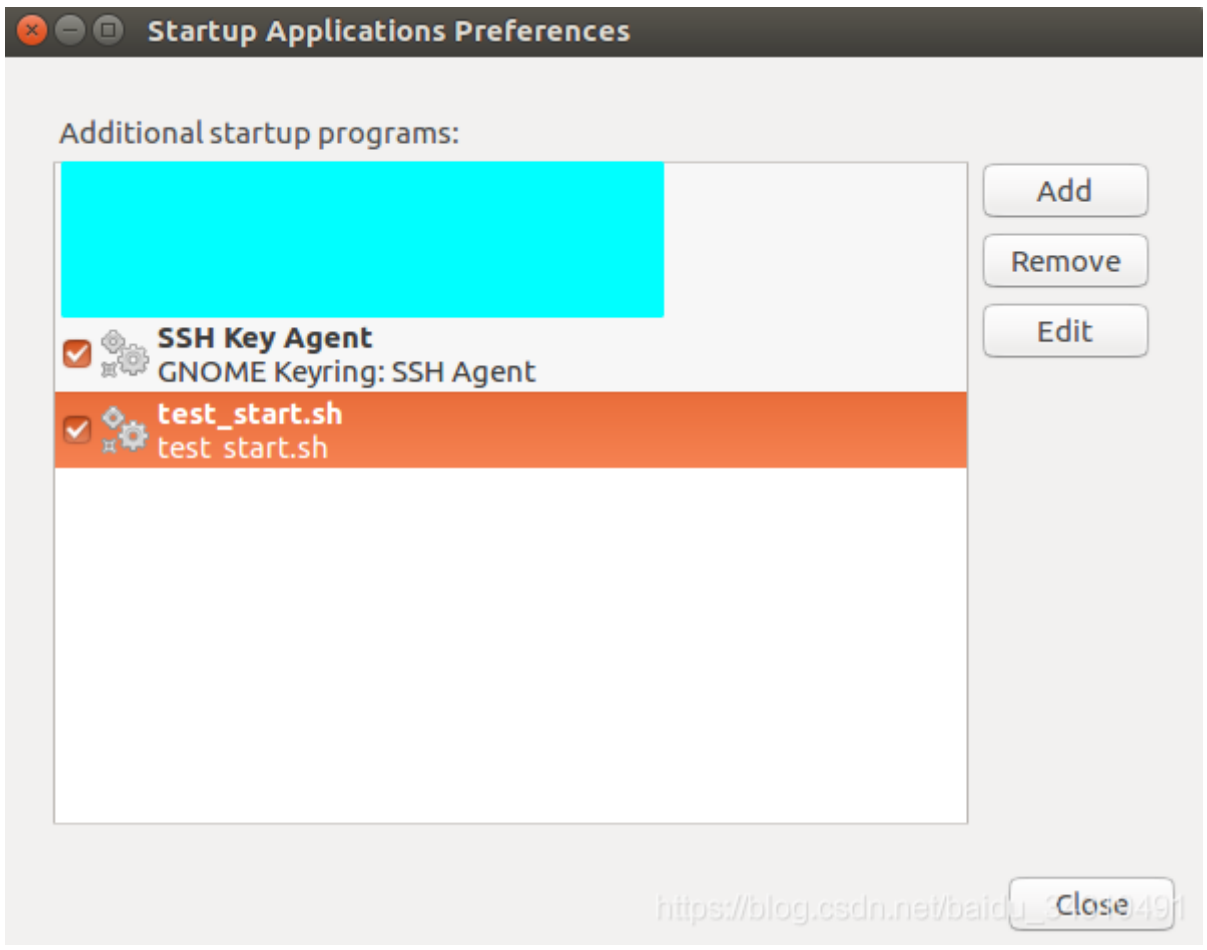


可以根据需要点击Add将对应的启动项目加进来,启动项可以是脚本,个人猜想也可以是桌面图标(未测试,大家有兴趣可以测试),这里以脚本为例,把上面的 `start.sh` 脚本添加到启动项里来

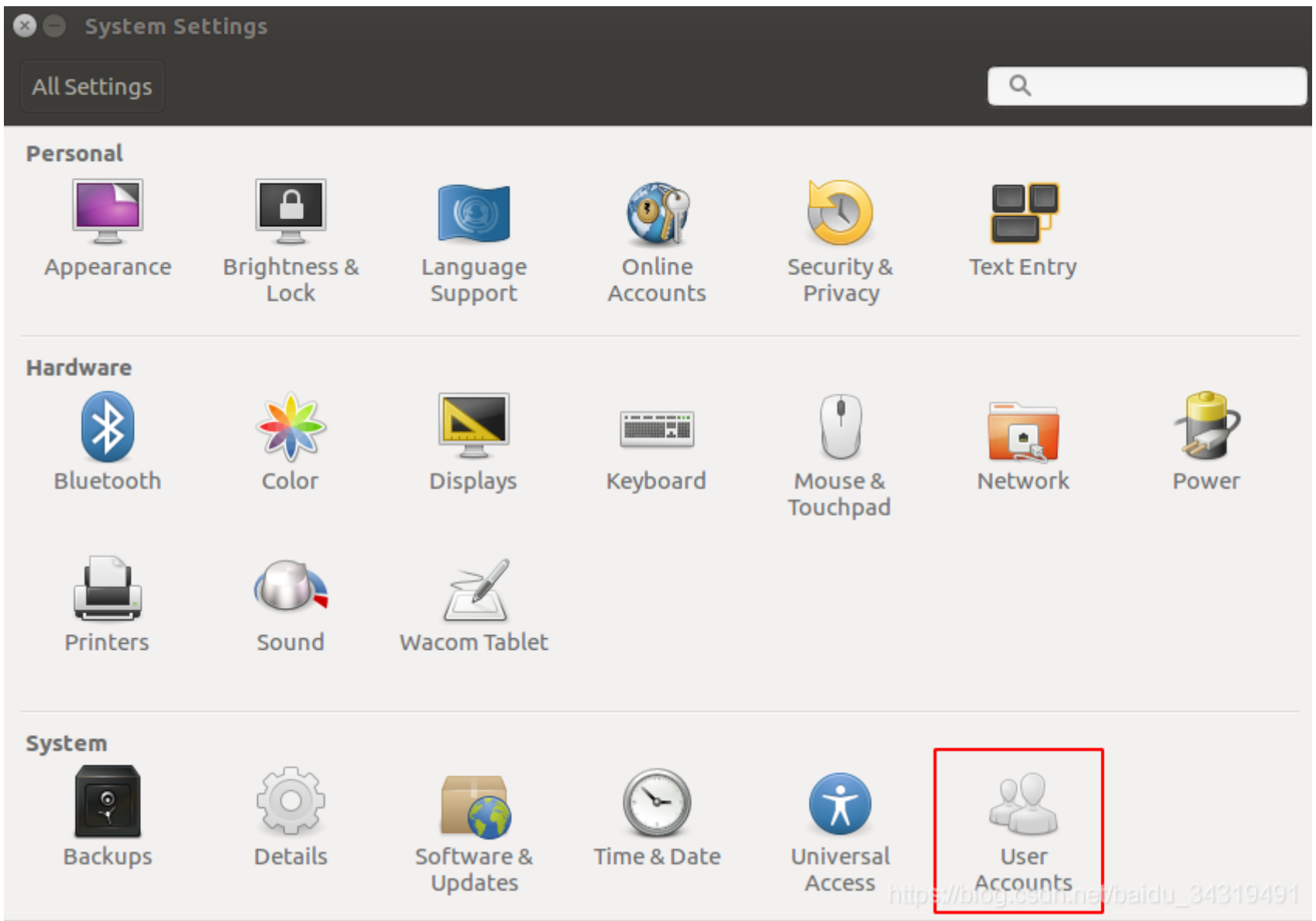
- **2.添加执行项目**
  点击 `Add` 按键
  输入相应的信息,这里 `Name` 和 `Comment` 都可以自己随意设置,只是对启动项的一个说明,中间的 `Command` 才是要运行的内容,可以点击 `Browse...` 来进行选择,这里就选择刚刚的那个脚本文件(注意脚本文件一定要有可执行的权限).



最后点击 Add 就可以了,界面中就会多了一项刚刚设置的内容

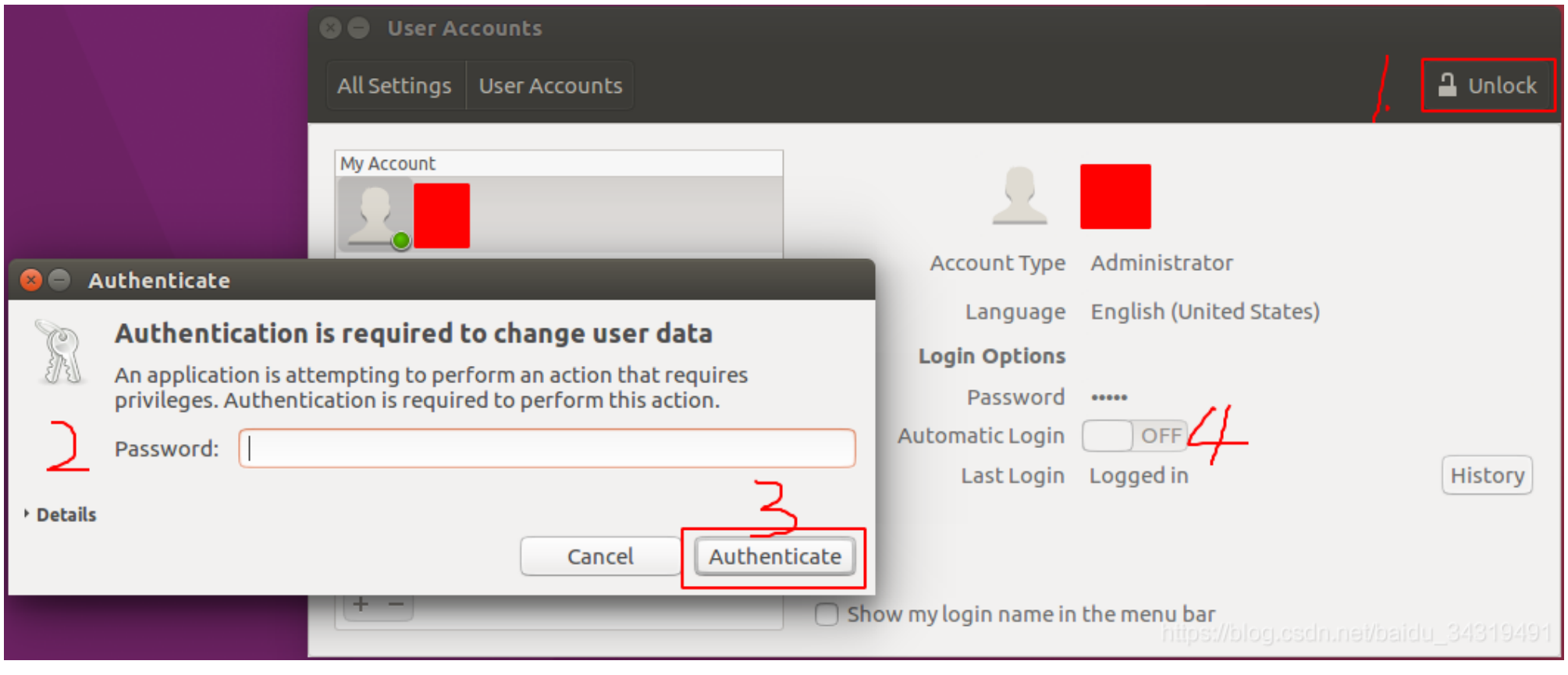


- **3.重启测试**
  点击重启就可以测试了,会发现开机之后确实打开了一个新的终端,并运行了脚本里的程序.
  由于一般设备都设置了密码,而管理器里的这些程序,是需要登录之后才可以进行的,所以,可以将设备设置为自动登录,具体的方法如下:
  A. 进入系统设置==>用户设置



B.设置自动登录

按照如下的顺序进行设置

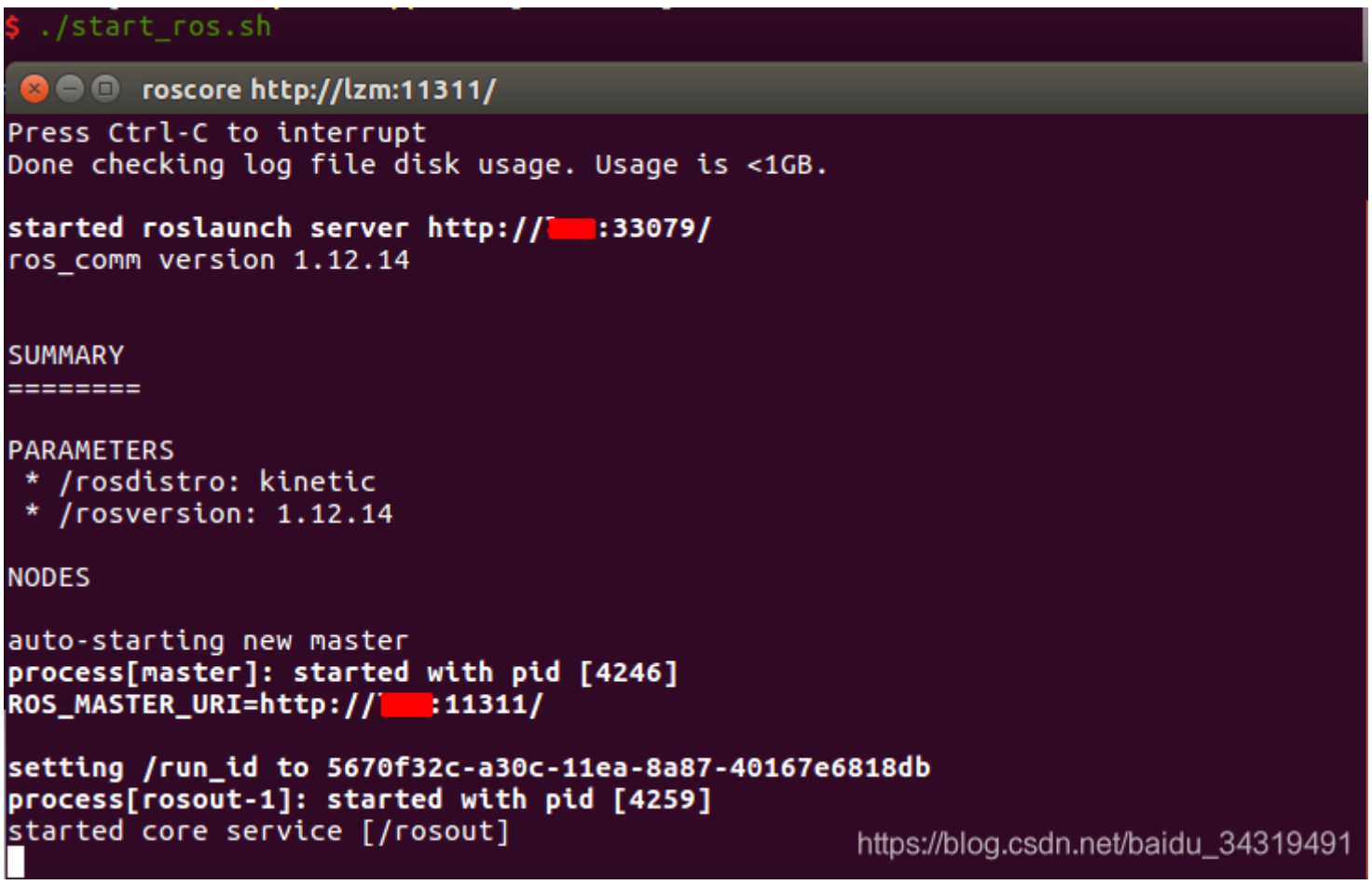


B.ROS程序开机启动

ROS的开机自启动也是类似,只不过脚本里头要进行相应的设置,由于ROS是第三方库,而终端要运行ros必须 `source /opt/ros/${ros-distro}/setup.bash` 才可以运行ros,所以在脚本中,新建一个终端运行时候,需要添加对应的source语句,类似 `start.sh` 文件新建 `start_ros.sh` 文件

```
#! /bin/bash
#根据系统启动相应版本的roscore
gnome-terminal -x bash -c "source /opt/ros/kinetic/setup.bash;roscore" &  #启动roscore
sleep 2 #等待roscore启动(可以不写)
gnome-terminal -x bash -c "cd <path>/ros_ws;source devel/setup.bash;rosrun package node" #启动节点
wait
exit 0
```

运行 `start_ros.sh` 文件



可以发现ros正常运行.

运行launch文件

launch可以避免启动roscore,所以比较方便,启动launch的方式和启动ros_node的方式是一样的

```
#! /bin/bash
gnome-terminal -x bash -c "cd <path>/ros_ws;source devel/setup.bash;rosrun package1 node1" &
gnome-terminal -x bash -c "cd <path>/ros_ws;source devel/setup.bash;rosrun package1 node2" &
gnome-terminal -x bash -c "cd <path>/ros_ws;source devel/setup.bash;rosrun package2 node1" &
gnome-terminal -x bash -c "cd <path>/ros_ws;source devel/setup.bash;rosrun package3 node1" &
wait
exit 0
```

有几个就可以添几个,根据需要自行处理,最后将 `start_ros.h` 脚本添加到 `gnome-session-properties` 管理器中即可.

C.需要sudo权限的脚本开机自启动

```
有的时候,我们的设备连接外部设备,需要打开串口,甚至是gpio口,这需要使用到权限,串口可以通过
添加rules来设置为普通用户,但是之前在DJI Manifold产品上需要程序对GPIO口进行设置,这个时候
就需要用到sudo权限了,因为在进入sudo权限需要输入密码,这就不好完成开机自启动.
```

所以需要设置脚本能够sudo运行程序,好在还是有办法的,打开终端,进入脚本所在路进,运行

`echo 'password' | sudo -S "./start.sh"`



发现可以直接运行,所以只需要新开一个脚本,使用sudo权限启动程序脚本就可以实现开机sudo启动了.

具体方法

- 在 `start.sh` 所在文件夹下添加 `sudo_start.sh` 并添加执行权限

  ```
  #! /bin/bash
  echo '电脑密码' | sudo -S "./start.sh"
  wait
  exit 0
  ```

- 运行./sudo_start.sh即可
- ROS节点类似,只需要将./start.sh替换成./start_ros.sh即可

结束语

本博文记录了作者日常使用的ubuntu系统开机自启动的方法,真实亲测有效,有需要的可以用作参考,如有错误,请指正.

另外文中使用flameshot截图软件,可以添加批注,画线等,比较方便.