

Linux 开机自启动

- 1 `.desktop` 开机自启动
 - 1.1 XDG Autostart 规范
 - 1.2 `.desktop` 文件格式
 - 1.3 demo
 - 1.4 注意
- 2 修改 `/etc/rc.local` 文件
 - 2.1 说明
 - 2.2 demo
 - 2.3 注意
- 3 使用 `systemd` 服务
 - 3.1 说明
 - 3.2 使用步骤

我这里使用开机自启动的目的是：在 Jetson Nano 板上部署 YOLOv8 算法，用于可乐和保鲜盒的检测。希望在设备开机后，能够自动进行目标检测，并将检测结果发送给上位机。

下面介绍三种 Linux 开机自启动方式

- `.desktop` 开机自启动（尤其适用于有界面显示需求的程序）
- 修改 `/etc/rc.d/rc.local` 文件
- 使用 `systemd` 服务

1 `.desktop` 开机自启动

这里也是我选择的方式，我使用的硬件为 Jetson Nano B01 Ubuntu 18.04。

1.1 XDG Autostart 规范

XDG Autostart 规范是一套用于 Linux 桌面环境的标准，旨在定义和管理开机自启动的应用程序。它是由 X Desktop Group (XDG) 组织提出的，目的是为了提供一种统一的方式来处理开机自启动，使得在不同桌面环境中的用户体验更加一致和方便。

根据 XDG Autostart 规范，每个用户都可以在其 `~/.config/autostart/` 目录下创建 `.desktop` 文件，其中包含了关于开机自启动应用程序的相关信息。每个 `.desktop` 文件都包含了一些关键的键值对，用于定义应用程序的名称、命令、图标、是否启用自启动等。

1.2 `.desktop` 文件格式

`.desktop` 文件是文本文件，采用 INI 文件格式，其中包含了启动应用程序的相关信息。

```
1 [Desktop Entry]
2 Type=Application
3 Name=Application Name
4 Exec=/path/to/application
5 X-GNOME-Autostart-enabled=true
```

- `[Desktop Entry]`：表示这是 `.desktop` 文件的开始。
- `Type=Application`：表示这个 `.desktop` 文件定义了一个应用程序。
- `Name=Application Name`：设置应用程序的名称。
- `Exec=/path/to/application`：指定应用程序的可执行文件路径。
- `X-GNOME-Autostart-enabled=true`：这是一个可选项，表示允许应用程序在开机时自动启动。

1.3 demo

在 `~/.config/autostart/` 文件夹下创建 `demo.desktop` 文件

```
1 [Desktop Entry]
2 Type=Application
3 Exec=gnome-terminal -x bash -c
  "/home/lei/Desktop/infer/workspace/qzj.sh;exec bash"
4 Hidden=false
5 NoDisplay=false
6 X-GNOME-Autostart-enabled=true
7 Name[en_US]=qzj_infer
8 Name=qzj_infer
9 Comment[en_US]=start qzj_infer program
10 Comment=start qzj_infer program
```

1.4 注意

需要 `root` 权限

2 修改 `/etc/rc.local` 文件

2.1 说明

在 Ubuntu 20.04 及更新版本中，`rc.local` 已经被弃用并默认禁用。`rc.local` 脚本在旧版本的Ubuntu中用于在引导过程中执行命令或脚本，但它已被更现代的替代方案取代，如 `systemd` 服务

对于需要开机自启动的程序，只需要在 `/etc/rc.local` 文件中添加相应的路径。

下面来看一个具体的例子：

2.2 demo

打开 `/etc/rc.local` 文件，添加如下内容：

```
1 #!/bin/bash
2
3 # 假设我们要启动名为 my_script.sh 的脚本
4 # 并传递参数 123 和 "Hello"
5 /path/to/my_script.sh 123 "Hello"
6
7 exit 0
```

在文件末尾，确保添加了 `exit 0`，表示脚本执行完成

确保 `rc.local` 文件具有可执行权限：

```
1 | sudo chmod +x /etc/rc.local
```

最后，重启系统以使修改生效。

2.3 注意

该方法仅使用于无界面的程序！！！！

3 使用 `systemd` 服务

3.1 说明

使用 `systemd` 服务在 Ubuntu 中实现开机自启动是一种现代且推荐的方式

3.2 使用步骤

1. 创建一个新的 `.service` 服务单元文件，将服务单元文件保存在 `/etc/systemd/system/` 目录下。

2. 在该文件中添加以下内容

```
1 | [Unit]
2 | Description=Your Service Description
3 | After=network.target      # 如果你的服务需要在网络启动后运行，可以
   | 添加其他依赖项
4 |
5 | [Service]
6 | Type=simple               # 可选项：simple、forking、oneshot等
7 | ExecStart=/path/to/your_script # 指定你想要开机启动的脚本或
   | 程序的路径
8 | Restart=on-failure        # 可选项：on-failure、always、no等
9 |
10 | [Install]
11 | WantedBy=multi-user.target # 指定在哪个target（例如：
   | multi-user.target, graphical.target）下启动服务
```

接着来启动服务并设置开机自启动：

首先要重新加载systemd的配置

```
1 | sudo systemctl daemon-reload
```

然后设置自启动

```
1 | sudo systemctl enable your_service_name.service
```

运行完上述命令后，systemd 会在每次系统启动时自动启动该服务

下面是一些其他的常用命令

停止服务：停止当前服务，但下次重启机器会再次启动该服务

```
1 | sudo systemctl stop your_service_name.service
```

禁用服务：在下次系统启动时，该服务将不再自动运行

```
1 | sudo systemctl disable your_service_name.service
```

手动启动服务

```
1 | sudo systemctl start your_service_name.service
```

查看所有全局服务的状态

```
1 | sudo systemctl status
```

查看特定全局服务的状态

```
1 | sudo systemctl status your_service_name.service
```

如果服务正在运行，你会看到一些类似以下的输出：

```
1 | plaintextCopy code
2 | ● your_service_name.service - Your Service Description
3 |    Loaded: loaded
              (/etc/systemd/system/your_service_name.service; enabled;
              vendor preset: enabled)
4 |    Active: active (running) since Mon 2023-08-07 15:00:00
              UTC; 1h ago
5 |    Main PID: 1234 (your_program)
6 |       Tasks: 1 (limit: 4915)
7 |      Memory: 10.0M
8 |     CGroup: /system.slice/your_service_name.service
9 |             └─1234 /path/to/your_program
```

如果服务未运行，你会看到类似以下的输出：

```
1 | plaintextCopy code
2 | ● your_service_name.service - Your Service Description
3 |   Loaded: loaded
4 |   (/etc/systemd/system/your_service_name.service; enabled;
   vendor preset: enabled)
   Active: inactive (dead)
```

在这个输出中，`Active` 字段显示了服务的当前状态。`active (running)` 表示服务正在运行，`inactive (dead)` 表示服务未运行。

参考具体含义参考：<https://ruanyifeng.com/blog/2016/03/systemd-tutorial-part-two.html>