# **Ubuntu Linux命令**

目录

[**Ubuntu Linux命令** 1](#_Toc173760218)

[录音功能 2](#_Toc173760219)

[打包RK3588开发板文件系统 3](#_Toc173760220)

[第一步骤 3](#_Toc173760221)

[第二步骤 3](#_Toc173760222)

[U盘/挂载 3](#_Toc173760223)

[NVME硬盘 3](#_Toc173760224)

[补丁包 4](#_Toc173760225)

[网络配置 4](#_Toc173760226)

[CPU配置查询 4](#_Toc173760227)

[内存信息查询 5](#_Toc173760228)

[wifi配置 5](#_Toc173760229)

[系统权限 5](#_Toc173760230)

[文件权限详细解释 6](#_Toc173760231)

[系统关机/重启 7](#_Toc173760232)

[系统日志 7](#_Toc173760233)

[系统进程管理 7](#_Toc173760234)

[修改系统时间及时区 8](#_Toc173760235)

[创建文件 8](#_Toc173760236)

[查看系统信息 8](#_Toc173760237)

[解压/压缩 文件 9](#_Toc173760238)

[GPIO相关 9](#_Toc173760239)

[动态链接库 10](#_Toc173760240)

[服务相关 10](#_Toc173760241)

[LINUX声音设置 10](#_Toc173760242)

[RK3588安装并编译OpenCV 11](#_Toc173760243)

[G++及编译文件 12](#_Toc173760244)

[CPU性能调整到性能模式 13](#_Toc173760245)

[查看NPU利用率 13](#_Toc173760246)

[Vim高级使用方法 13](#_Toc173760247)

[CONDA用法 14](#_Toc173760248)

[Linux22.04 路由表问题 14](#_Toc173760249)

[默认路由的作用 15](#_Toc173760250)

[路由表上下文 15](#_Toc173760251)

[网络工具nmtui 16](#_Toc173760252)

# 录音功能

arecord -D hw:1,0 -f dat test.wav -d 10：

使用 arecord 工具录制音频。具体而言：-D hw:1,0 指定了要使用的音频设备，hw:1,0 表示第 1 个硬件设备的第一个子设备。-f dat 指定了音频文件的格式为 dat 格式。test.wav 是要保存录制音频的文件名。-d 10 指定了录制音频的持续时间，这里是 10 秒。

# 打包RK3588开发板文件系统

https://wiki.t-firefly.com/zh\_CN/Firefly-Linux-Guide/first\_use.html#dao-chu-she-bei-xi-tong

## 第一步骤

在设备上导出Ubuntu跟文件系统rootfs

在开发板设备端操作

1.在设备的ubuntu环境下，安装fireflydev:

sudo apt update

sudo apt install fireflydev

2.安装fireflydev后，就能够使用ff\_export\_rootfs脚本导出根文件系统

## 第二步骤

二次打包完整的固件，将ubuntu rootfs 与发布固件的其他分区组合，完成二次打包，生成新的完整固件

# U盘/挂载

Ubuntu系统下默认的u盘挂载目录为 ：“/media/用户名/磁盘标识符”

# NVME硬盘

* lsblk：

显示系统中所有的块设备，包括硬盘和分区。挂载的硬盘将显示在列表中，并显示它们的挂载点

* sudo mkdir /mnt/myssd:

创建挂载点，其实就是创建一个新的文件夹

* sudo mount /dev/sdb1 /mnt/myssd/：

挂载固态硬盘，假设固态硬盘设备名称为 /dev/sdb1

* df -h：

验证固态硬盘是否成功挂载

* 永久挂载： 挂载信息添加到 /etc/fstab 文件中。编辑 /etc/fstab 文件，添加一行类似于以下内
* /dev/sdb1 /mnt/myssd ext4 defaults 0 0
* 请确保替换 **/dev/sdb1** 和 **/mnt/myssd** 为您的固态硬盘和挂载点的正确值，然后保存并退出编辑器。

# 补丁包

* sudo dpkg -i 文件名.deb：

打补丁包

# 网络配置

* ifconfig：

查看网络端口，一般（RK3588S）的网线端口为 eth0

* ip addr show：

查看当前网络接口的信息

* sudo ifconfig [interface] [new\_ip\_address] netmask [subnet\_mask]：修改ip地址和子网掩码，这个对于3588只能是临时修改，重启就会掉命令
* nmcli c modify Wired\ connection\ 1 ipv4.method manual ipv4.addresses 192.168.0.10/24 ipv4.gateway 192.168.0.1：

修改名为 "Wired connection 1" 的连接的 IPv4 地址配置，将连接设置为手动配置模式，并指定 IPv4 地址为 192.168.0.10，子网掩码为 24（即 255.255.255.0），网关为 192.168.0.1，这个配置是永久生效的

* nmcli c up Wired\ connection\ 1：

启用名为 "Wired connection 1" 的连接，使其生效

# CPU配置查询

* lscpu:

显示CPU架构，核心数，线程数，cpu频率信息

* cat /proc/cpuinfo

显示cpu的详细信息，包括制造商，型号，缓存大小

* nproc

显示cpu的逻辑核心数

* htop

类似于windows的任务管理器完全版

* top

任务管理器简化版

# 内存信息查询

* free -h

显示系统的内存使用情况，包括已用内存，空闲内存，缓冲区和缓存

* cat /proc/meminfo

显示关于系统内存的详细信息，包括内存总量，可用内存，缓存等

* htop

# wifi配置

* nmcli radio wifi on：

开启wifi

* nmcli device wifi list：

列出所有的wifi

* sudo nmcli device wifi connect MyWiFi password mypassword：

连接到名为 MyWiFi 的 WiFi 网络并输入密码

* nmcli connection down "WiFi-Name"

中断当前连接的名为WiFi-Name的wifi连接

# 系统权限

* chmod +x demo\_tty：

修改文件 **demo\_tty** 将被设置为可执行，可以直接运行该文件

* chmod 755 imx586\_default\_default.json：

修改文件权限为-rwxr-xr-x

* sudo su:

进入系统root权限

* exit：

退出root用户

* chown -R root:root /etc：

将/etc 目录以及其下的所有文件和子目录的所有者和组都设置为root:root

* chmod 777

设置文件或目录的权限为最大权限，允许所有用户对文件或目录进行，读，写和执行操作，包括所有者，所属组和其他用户

* chmod 775

相对于777较于宽松，对于777而言，它对其他用户的权限略有限制，允许所有者对文件或目录进行读，写和执行操作，允许所属组的用户进行读和执行操作，同时也允许其他用户进行读和执行操作，但不允许其他用户进行写操作

# 文件权限详细解释

第一列：文件或目录的权限标识符，它告诉您文件或目录的所有者，所属组和其他用户对该文件或目录具有的访问权限。每个权限标识符由10个字符组成，其中第一个字符表示文件类型，后面9个字符分三组，每组三个字符表示对应的权限

第一个字符表示文件类型：

-：表示普通文件

d：表示目录

l：表示符号链接（即linux中的快捷方式）

c：表示字符设备文件

b：表示块设备文件

p：表示命名管道

s：表示套接字

后面的九个字符分为三组，每组三个字符表示文件的访问权限

第一组（第2 到第4个字符）表示所有者的权限

第二组（第5到第7个字符）表示所属者的权限

第三组（第8到第10个字符）表示其他用户的权限

每个权限字符可以是以下之一

r：读取权限

w：写入权限

x：执行权限

字符设备文件

在类unix操作系统中表示字符设备的特殊文件。字符设备是一种设备，其数据以字符的形式流式传输，而不是以固定大小的块进行传输。典型的字符设备包括终端（tty），打印机，声卡，串口设备

其字符设备文件具有以下特点

1.数据流式传输：与块设备文件不同，字符设备文件的数据以字符的形式流式传输，一次一个字符

2.无法定位：字符设备通常不支持像磁盘这样的定位操作，例如，在字符设备中，您无法像磁盘文件那样直接访问文件的任意位置，而只是按顺序读取或写入数据

3.非缓冲：大多数字符设备不会缓冲数据，一旦数据被写入设备，它们就会立即传输到设备

4.一次一个字符读写：与块设备相比，字符设备文件通常只支持一次一个字符的读写操作，这意味着每个读取或写入操作都会涉及到单个字符

字符设备文件在类unix系统中的表示形式通常是在/dev 目录下。例如，典型的终端设备在linux系统中可能表示为/dev/ttyS0或/dev/ttyUSB0 ，其中tty表示终端设备，S0和USB0是设备的编号

块设备文件

块设备文件是一种在类unix操作系统中表示块设备的特殊文件，块设备是一种设备，其数据以固定大小的块进行传输，而不是以字符的形式流式传输，典型的块设备包括硬盘驱动器，固态硬盘，USB闪存驱动器和CD/DVD驱动器等。

块设备文件具有以下特点

1.数据以块为单位传输：与字符设备文件不同，块设备文件的数据以固定大小的块围为单位进行传输。这些块的大小通常是固定的，例如512字节或4KB

2.支持定位操作：块设备通常支持像磁盘这样的

# 系统关机/重启

* sudo shutdown now:

立即关机。

* sudo shutdown -h now:

立即关机，同时关闭电源。

* sudo shutdown -r now:

立即重新启动计算机。

* sudo shutdown -h +10:

10分钟后关机。

* sudo shutdown -r +5:

5分钟后重新启动计算机。

* sudo reboot:

重启设备

# 系统日志

* journalctl -r：

查看系统日志

* dmesg：

命令用于显示系统的消息日志，其中包含了开机期间的内核和驱动程序输出的信息。通过运行 dmesg 命令，您可以查看有关系统启动、硬件检测、设备驱动加载和其他系统级别事件的详细信息

* dmesg | grep es8323：

通过 dmesg 命令检索内核日志中包含 "es8323" 的行。 "es8323" 可能是某种硬件设备或驱动程序的名称

# 系统进程管理

一个作业（job）通常指的是一个正在运行的程序或者进程。命令用于列出当前 shell 中所有作业的状态和作业号。每个作业都有一个唯一的作业号，可以用来标识和管理作业。-l 选项显示更详细的信息，包括作业号、状态、命令和 PID

* jobs -l

[1]+ 1234 Running ./script.sh &

[2]- 5678 Stopped ./program

这表示当前有两个作业：作业号为 1 的作业正在运行 ./script.sh，作业号为 2 的作业已停止运行 ./program。

发送信号给进程号为5432的进程，信号编号为9，表示强制终止该进程

* kill -9 5432
* kill -9 3874
* ps -aux | grep demo：

使用 ps 命令来列出当前正在运行的进程，并使用 grep 过滤出包含 "demo" 字符串的进程信息

* ps -ef | grep plus：

命令用于列出系统上所有进程的详细信息。其中：-e 选项表示列出系统上所有的进程，包括其他用户的进程。-f 选项表示以完整格式显示进程信息，包括进程的详细信息，如 UID、PID、PPID、C、STIME、TTY、TIME、CMD 等。

# 修改系统时间及时区

* timedatectl list-timezones | grep Shang：

列出所有的时区，并通过 grep 过滤出包含Shang 的时区其中 | 被称之为管道符号，它将一个命令的输出传递给另一个命令作为输入，在这种情况下 timedatectl list-timezones 命令的输出被传递给grep Shang 命令 ，并且 greo Shang 命令将过滤出包含Shang的行

* timedatectl set-timezone Asia/Shanghai：

修改时区为亚洲/上海，注：在RK3588 Ubuntu20.04的系统中执行这句话会报错 Failed to set time zone : Access denied ，这是因为20.04固件Bug造成的使用chown -R root:root /etc：解决报错问题

* date：

查看当前时间

* sudo date 070412342023.56

MM 是月份

DD 是日期

hh 是小时

mm 是分钟

CC 是世纪

YY 是年份的后两位数字

ss 是秒钟（可选）

例如，要将日期设置为2023年7月4日，时间设置为12:34:56，可以运行以下命令

# 创建文件

* echo > gps.txt：

创建了一个名为 "gps.txt" 的空文件

* touch：

指令也可以做到创建文件

# 查看系统信息

* uname -m：

查看当前ubuntu系统架构

* htop：

使用htop工具查看内存状态

* sudo apt undate:

更新软件包列表：从您系统中已配置的软件源中下载最新的软件包信息，并将其保存到2.本地的软件包列表中。检查更新的软件包：检查软件源中是否有更新的软件包可用。如果有更新的软件包可用，它们会列在更新列表中。检查软件源的可用性：检查您系统中已配置的软件源是否可访问。如果某个软件源不可访问，apt update 命令会显示警告信息。

# 解压/压缩 文件

* tar -xzf your\_file.tar.gz：

请将 your\_file.tar.gz 替换为你要解压的 .tar.gz 文件的实际文件名，-x: 表示解压缩操作，-z: 表示使用 gzip 解压缩，-f: 指定要解压的文件名

* unzip your\_filr.zip：

解压.zip文件

* sudo apt-get install p7zip-full：

解压.7z文件

* 7z x filename.7z：

解压

* 7z x filename.7z -o /path/to/destination：

解压到指定目录

* unzip:

文件名.zip：解压zip文件

* unzip：

文件名.zip -d 目标目录：解压文件到目标目录中

zip -r my\_folder.zip my\_folder

这将在当前目录中创建一个名为 my\_folder.zip 的压缩文件，其中包含 my\_folder 文件夹的内容。

# GPIO相关

针对于RK3588GPIO控制文件所在目录/sys/class/gpio/

* echo 57 > export:

这一行命令将 GPIO 编号为 57 的引脚导出（export）给用户空间，这样用户空间就可以控制该引脚了。

* ls:

再次列出当前目录下的文件和文件夹，可以看到多了一个名为 gpio57 的文件夹，这个文件夹对应着 GPIO 编号为 57 的引脚。

* cd gpio57:

进入 GPIO 编号为 57 的引脚所在的目录。

* ls:

列出当前目录下的文件和文件夹，显示了 GPIO 引脚的相关属性文件

* cat direction:

读取并显示了 direction 文件的内容，这个文件用来控制 GPIO 引脚的输入/输出方向，in 表示该引脚为输入模式

* cat value:

读取并显示了 value 文件的内容，这个文件用来读取 GPIO 引脚的电平值，1 表示高电服务相关

# 动态链接库

在RK3588中，绝大部分动态链接库文件存放位置**/usr/lib/aarch64-linux-gun/ 这个目录存放了大多数针对64位ARM架构的Linux系统所需的动态链接库文件，这些库文件包括系统标准库，常用的第三方库及其他软件所依赖的库文件，因此，将动态链接库文件放在这个目录下可以确保系统中的各种程序都能够方便地找到并使用这些库文件**

# 服务相关

* sudo systemctl status ffmedia-capture.service：

其中ffmedia-capture.service是服务名称，查看名为 "ffmedia-capture.service" 的服务的状态。在运行该命令后，系统会返回该服务的当前状态信息，包括是否正在运行以及其他相关信息

* sudo systemctl restart ffmedia-capture.service：

使用 systemd 工具来重启名为 "ffmedia-capture.service" 的服务，systemctl restart ffmedia-capture.service: 这是 systemd 工具的命令，用于重启名为 "ffmedia-capture.service" 的服务。系统会尝试停止该服务（如果正在运行），然后再重新启动它。这个命令会尝试重新启动名为 "ffmedia-capture.service" 的服务，以使其重新加载配置或者应用最新的更改。

* sudo systemctl disable ffmedia-capture.service：

禁用了名为 "ffmedia-capture.service" 的 systemd 服务。系统已经移除了对应服务单元文件的链接，这意味着该服务将不会在系统启动时自动启动

* sudo systemctl enable ffmedia-capture.service：

服务在系统启动时自动启动

* sudo systemctl start ffmedia-capture.service：

启动名为 "ffmedia-capture.service" 的 systemd 服务

* sudo systemctl stop ffmedia-capture.service：

停止名为 "ffmedia-capture.service" 的 systemd 服务，及停止摄录程序

# LINUX声音设置

* alsamixer：

用于调整Linux系统中声音设置的命令行工具。它提供了一个简单的图形界面，用于控制声音的各种参数，例如音量、通道平衡、输入输出设置等。在运行alsamixer命令后，你将看到一个基于文本的界面，其中列出了各种声音控制参数，并且你可以使用键盘上的箭头键和其他键来调整这些参数。例如，你可以使用左右箭头键来选择不同的通道，上下箭头键来增加或减少音量。如果你有特定的声音设置需求，可以尝试使用alsamixer来进行调整

* systemctl restart pulseaudio：

重新启动 PulseAudio 音频服务器，使任何在其运行期间发生的更改都能生效。在 Linux 系统中，PulseAudio 负责处理音频输入和输出，以及应用程序之间的音频混音和路由。通过重新启动 PulseAudio，可以解决一些与音频相关的问题，例如声音失真、输入/输出设备无法识别等

# RK3588安装并编译OpenCV

软件列表更新

* sudo apt update

安装依赖

* sudo apt install build-essential cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
* sudo apt install libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-dev libdc1394-22-dev libv4l-dev libxvidcore-dev libx264-dev libatlas-base-dev gfortran

下载OpenCV源码

* 下载OpenCV (<https://github.com/opencv/opencv/releases/tag/4.8.1>) 下载源码格式

解压

* unzip opencv-4.8.1.zip （创建自己的文件夹，并放入及解压）

编译

* cd opencv-4.8.1
* mkdir build
* cd build
* cmake ..
* make -j8

安装

* sudo make install

配置环境，配置ld.so.conf文件

* sudo gedit /etc/ld.so.conf
* (2)添加内容
* include /etc/ld.so.conf.d/\*.conf
* include /usr/local/lib
* sudo ldconfig

查看版本，若显示正确，则表明安装完毕

* pkg-config --modversion opencv4

配置系统bash

* sudo gedit /etc/bash.bashrc
* 末尾添加
* PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig
* export PKG\_CONFIG\_PATH
* source /etc/bash.bashrc

查看版本

* pkg-config --modversion opencv4

验证并使用opencv

编写测试代码，文件名为opencv\_test.cpp

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace cv;

using namespace std;

int main() {

Mat image(300, 300, CV\_8UC3, Scalar(0, 255, 0));

imshow("OpenCV Test", image);

waitKey(0);

return 0;

}

方案一：指定 OpenCV 相关的头文件路径和库文件路径

g++ opencv\_test.cpp -o opencv\_test -I /usr/include/opencv4/ -L /usr/lib/aarch64-linux-gnu/ -lopencv\_core -lopencv\_highgui -lopencv\_imgproc -lopencv\_imgcodecs -pthread -lz -ldl

方案二：pkg-config 工具来自动获取 OpenCV 的编译选项和链接选项

/usr/lib/pkgconfig目录下添加opencv.pc

prefix=/usr

exec\_prefix=${prefix}

libdir=${prefix}/lib/aarch64-linux-gnu

includedir=${prefix}/include/opencv4

Name: opencv

Description: OpenCV - Open Source Computer Vision Library

Version: 4.5.2

Libs: -L${libdir} -lopencv\_core -lopencv\_imgproc -lopencv\_highgui

Cflags: -I${includedir}

执行编译命令

g++ opencv\_test.cpp -o opencv\_test `pkg-config --cflags --libs opencv`

**执行结果为一个小的绿色窗口，窗口名为OpenCV Test**

# G++及编译文件

* sudo apt update
* sudo apt install build-essential gdb
* sudo apt install g++
* sudo apt install gcc
* sudo apt install gdb
* sudo apt install camke

检测版本并查看是否安装成功

* cmake --version
* g++ --version
* gcc --version
* gdb –version

编译c++文件(.cpp)，比如文件名为serial\_write.cpp，执行g++ -o serial\_write serial\_write.cpp -std=c++11

运行程序sudo ./serial\_write

(base) firefly@firefly:~$

开头的(base) 使用conda deactivate 去掉

# CPU性能调整到性能模式

* echo performance | tee $(ls /sys/bus/cpu/devices/cpu\*/cpufreq/scaling\_governor)

# 查看NPU利用率

# Vim高级使用方法

i -- 切换到输入模式，在光标当前位置开始输入文本。

x -- 删除当前光标所在处的字符。

: -- 切换到底线命令模式，以在最底一行输入命令。

a -- 进入插入模式，在光标下一个位置开始输入文本。

o：在当前行的下方插入一个新行，并进入插入模式。

O -- 在当前行的上方插入一个新行，并进入插入模式。

dd -- 剪切当前行。

yy -- 复制当前行。

p（小写） -- 粘贴剪贴板内容到光标下方。

P（大写）-- 粘贴剪贴板内容到光标上方。

u -- 撤销上一次操作。

Ctrl + r -- 重做上一次撤销的操作。

:w -- 保存文件。

:q -- 退出 Vim 编辑器。

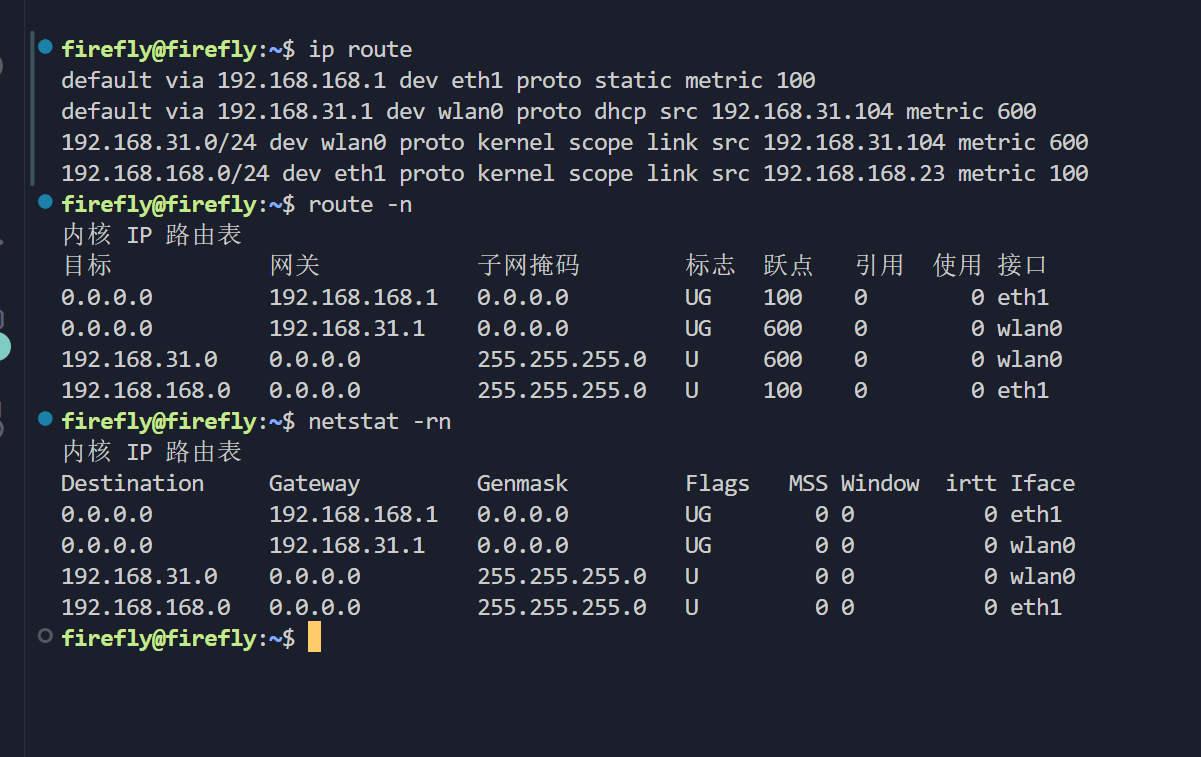
:q! -- 强制退出Vim 编辑器，不保存修改。

# CONDA用法

* conda activate rk3588 启动一定名为RK3588的虚拟环境

# Linux22.04 路由表问题

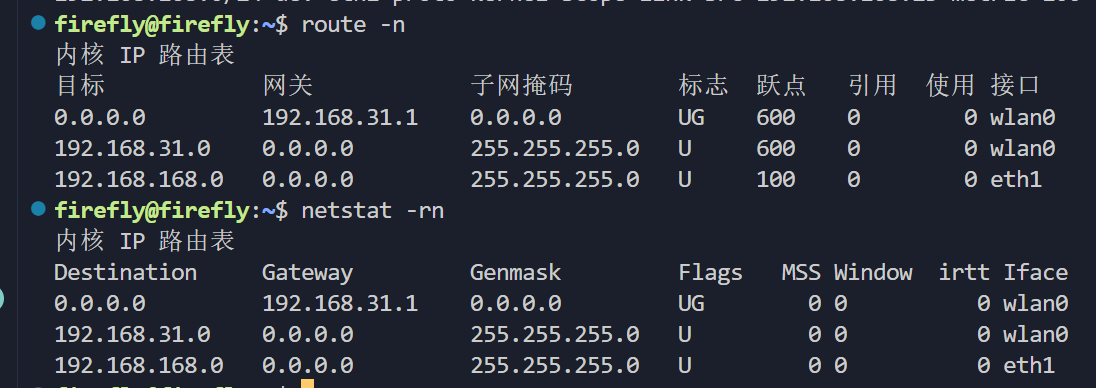
连接了网线ipv4为固定地址，同时连接wifi，这时wifi和网线SSH只能二选一，检查发现是路由表的问题，这是出问题状况的路由表



从这张图分析当前有2个默认路由（0.0.0.0 目标），分别通过 eth1 和 wlan0 接口。两个默认路由可能会导致路由选择不确定，进而引起网络连接问题

跃点数（Metric）不同: eth1 的默认路由跃点数（metric 100）低于 wlan0（metric 600），表示系统会优先使用 eth1 接口进行外部通信，而eth1就是网线网络，我现在用于网线SSH连接，没有互联网

执行删除指令

ip route delete default via 192.168.168.1 dev eth1

以下是详细解释

默认路由（default route）是指在路由表中用于转发所有无法匹配到其他特定路由条目的数据包的路由。默认路由的目标地址通常设为 0.0.0.0/0，表示所有IP地址都可以通过该路由传输。它是一个 "catch-all" 路由，通常用于转发到互联网或其他不在本地网络中的地址。

路由表项

目标 网关 子网掩码 标志 跃点 引用 使用 接口

0.0.0.0 192.168.31.1 0.0.0.0 UG 600 0 0 wlan0

**各字段的含义**

1. **目标（Destination）**: 0.0.0.0 表示这是一个默认路由，适用于所有目的地。
2. **网关（Gateway）**: 192.168.31.1 表示数据包要通过这个IP地址的网关进行转发。
3. **子网掩码（Genmask）**: 0.0.0.0，与目标地址 0.0.0.0 一起表示该路由适用于所有地址。
4. **标志（Flags）**: UG 表示该路由是一个网关路由（G）且是活动的（U）。
5. **跃点（Metric）**: 600 表示路由的优先级。数值越低，优先级越高。此项为600，表示优先级较低。
6. **接口（Iface）**: wlan0 表示数据包将通过这个网络接口（无线网卡）进行传输。

# 默认路由的作用

当你的计算机试图访问一个不在本地网络上的IP地址时（例如，访问一个互联网网站），它会查阅路由表。如果没有找到匹配的特定路由，它会使用默认路由。通过默认路由，计算机会将数据包发送到 192.168.31.1，这个IP地址通常是你的无线路由器的局域网地址。然后，无线路由器将负责将数据包转发到互联网。

**示例流程**

1. **请求目标**: 例如，你的计算机试图访问 8.8.8.8。
2. **路由表查找**: 计算机会检查路由表，寻找匹配的路由。
3. **使用默认路由**: 因为没有特定条目匹配 8.8.8.8，计算机会使用默认路由 0.0.0.0，通过 192.168.31.1（wlan0 接口）发送数据包。
4. **网关处理**: 网关 192.168.31.1（无线路由器）收到数据包，并将其转发到下一跳，通常是互联网服务提供商（ISP）的路由器，最终到达目标地址。

# 路由表上下文

除了默认路由，你的路由表还包括局域网的具体路由：

目标 网关 子网掩码 标志 跃点 引用 使用 接口

192.168.31.0 0.0.0.0 255.255.255.0 U 600 0 0 wlan0

192.168.168.0 0.0.0.0 255.255.255.0 U 100 0 0 eth1

这些路由表示：

* **192.168.31.0/24**: 通过 wlan0 接口连接的局域网，子网掩码 255.255.255.0 表示这个子网包括 IP 范围 192.168.31.1 到 192.168.31.254。
* **192.168.168.0/24**: 通过 eth1 接口连接的局域网，子网掩码 255.255.255.0 表示这个子网包括 IP 范围 192.168.168.1 到 192.168.168.254。

这些具体路由优先于默认路由，用于在局域网内直接传输数据。

通过理解默认路由及其配置，可以确保网络数据包正确传输，尤其是在多网络接口环境中。

# 网络工具nmtui

输入sudo nmtui进入此配置工具项目，此时用上下箭头按键进行上下选择，用tab反选确认选项，按 Enter 选择一个选项，按 Space 键选择复选框状态