

目录

Q1.	如何下载 Vmware 及 Linux 系统软件.....	2
Q2.	虚拟机中 Linux 系统如何选择?	2
Q3.	虚拟机如何配置可以流畅运行 Debian 12.7.0 标准版?	3
Q4.	桥接的网络模式是什么意思	3
Q5.	Debian 安装时, 在 Boot menu 界面如何选择?	4
Q6.	Debian 12.7.0 标准版安装后, 需要安装哪些软件包.....	4
Q7.	系统安装 Debian 标准版后, 没有 SSH?	5
Q8.	Debian 和 Ubuntu 配置网络环境所使用的命令和配置文件有哪些?	5
Q9.	TShark 和 Wireshark 什么关系? Debian 上如何安装?	8
Q10.	如何使用 vs code 通过 SSH 远程登录到 Linux 系统进行 C 程序开发	8
Q11.	VS code 调试程序遇到以下错误提示, 我该怎么办?	10
Q12.	防火墙如何开放接收 UDP 流量	11
Q13.	UFW 防火墙有什么特点?	12
Q14.	可以在系统中同时安装并使用 UFW 和 iptables?	13
Q15.	如何在 Debian Ubuntu Linux 主机中访问虚拟机的共享文件夹?	13
Q16.	ubuntu (普通用户) 登录后, 设置 root 账户密码, 及如何切换账户	14
Q17.	在 Ubuntu Server 查看网络接口和网络配置的命令	14
Q18.	Debian 和 Ubuntu 如何配置双网口主机?	14
Q19.	在 Ubuntu 主机上如何配置双网口	16
Q20.	如何启用 IP 转发和配置防火墙 (使用 iptables 或 nftables)	17

Q1. 如何下载 Vmware 及 Linux 系统软件

从 2024 年 5 月开始，知名虚拟机软件 VMware Workstation Pro 宣布免费供个人使用。这意味着我们可以在无需许可证密钥或任何持续费用的前提下，在 Windows 或 Linux 上下载并使用这款强大的虚拟机软件的全部功能。官网：

```
https://www.vmware.com/products/desktop-hypervisor/workstation-and-fusion
```

linux 系统的 ISO 文件下载，可使用清华大学开源软件镜像站 | Tsinghua Open Source Mirror，网址：

```
https://mirrors.tuna.tsinghua.edu.cn/#
```

Q2. 虚拟机中 Linux 系统如何选择？

考虑资源消耗，通常选择不带桌面环境的系统，比较推荐 Ubuntu Server 或者 Debian Standard 。

以下是 Ubuntu Server 和 Debian Standard 之间的一些主要比较：

特性	Ubuntu Server	Debian Standard
基础	基于 Debian Unstable	自己的独立发行
支持周期	每两年 LTS（长期支持）版本，支持 5 年（可延长到 10 年）	每两年一个稳定版本，支持 3 年
安装和使用	安装和使用更简单，适合新手	安装和使用稍微复杂，适合有经验的用户
社区和支持	社区支持强大，有官方支持	社区支持强大，但没有官方支持
软件包管理	使用 APT，与 Debian 类似	使用 APT，与 Ubuntu 类似
性能	性能较好，但资源占用较多	性能优良，适合资源有限的环境
更新频率	LTS 版本每两年更新一次，标准版本每 6 个月更新一次	稳定版每两年更新一次，测试版和不稳定版更新频率更高

总结

- **Ubuntu Server**：适合新手和需要频繁更新的环境，有官方支持，社区活跃。

- **Debian Standard**: 适合有经验的用户，注重稳定性和安全性，适合生产环境。

考虑到虚拟机对物理主机的资源消耗，**建议采用 Debian Standard**。

Q3. 虚拟机如何配置可以流畅运行 Debian 12.7.0 标准版？

以下是推荐的配置：

虚拟机硬件配置

1. **处理器**:
 - 配置 2 个虚拟处理器核（更高的性能可以选择 4 核）。
2. **内存**:
 - 至少分配 2GB 内存（推荐 4GB）。
3. **硬盘空间**:
 - 分配 20GB 硬盘空间（根据需要可以调整）。
4. **网络适配器**:
 - 使用“桥接模式”或“NAT 模式”。
5. **显示设置**:
 - 一般不用调整默认值，因为标准版没有图形界面。

Q4. 桥接的网络模式是什么意思

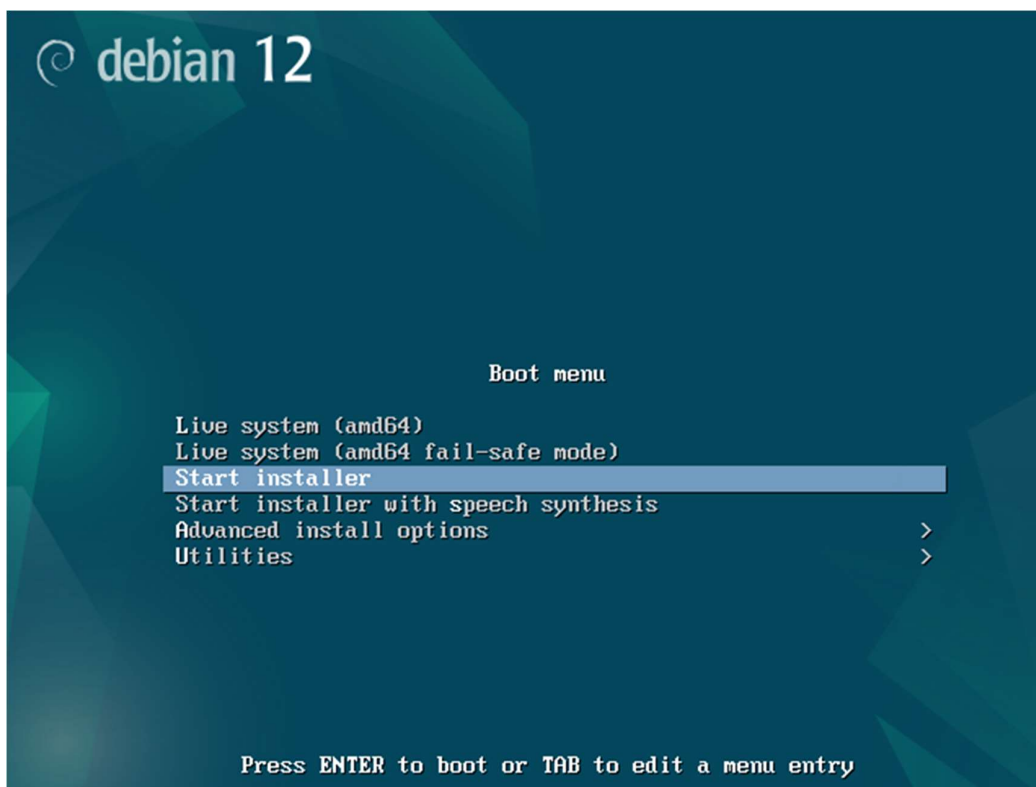
桥接模式（Bridged Networking）是一种将虚拟机的网络适配器直接连接到主机的物理网络的模式。这使得虚拟机可以像一台物理机一样，获得与宿主机相同的网络访问权限和 IP 地址范围。简单来说，虚拟机就像直接插入到物理网络中的一台独立设备。

主要特性

1. **独立 IP 地址**: 虚拟机会获得一个与宿主机不同的 IP 地址，通常通过 DHCP 服务器分配。
2. **与其他设备通信**: 虚拟机可以与网络中其他设备（包括宿主机）直接通信，就像它们是同一网络中的设备一样。
3. **适用于多种网络环境**: 适用于需要虚拟机作为独立设备参与网络操作的场景，例如测试网络应用、文件共享等。

在使用桥接模式时，确保网络配置正确，以避免 IP 冲突和其他网络问题。

Q5. Debian 安装时，在 Boot menu 界面如何选择？



如图所示，选择“Start installer”

注意，选择语言时，要选择英文，以避免终端窗口出现乱码。

Q6. Debian 12.7.0 标准版安装后，需要安装哪些软件包

安装基本工具

安装完成后，可以配置一些基本开发工具：

1. 更新包列表：

```
sudo apt update
```

2. 安装 GCC、GDB 和其他开发工具：

```
sudo apt install build-essential gdb
```

3. 安装常用编辑器：

```
sudo apt install vim nano
```

这样配置可以确保你顺畅地运行 Debian 并进行 C 编程调试。

Q7. 系统安装 Debian 标准版后，没有 SSH?

要在 Debian 上打开 SSH 服务器，您可以按照以下步骤操作：

1. **安装 OpenSSH 服务器：** 打开终端，运行以下命令来安装 OpenSSH 服务器：

```
sudo apt update
sudo apt install openssh-server
```

2. **启动 SSH 服务：** 安装完成后，启动 SSH 服务：

```
sudo systemctl start ssh
```

3. **设置开机启动：** 确保 SSH 服务在系统启动时自动启动：

```
sudo systemctl enable ssh
```

4. **检查 SSH 服务状态：** 可以通过以下命令检查 SSH 服务是否正常运行：

```
sudo systemctl status ssh
```

5. **配置防火墙：** 如果您使用的是 UFW (Uncomplicated Firewall)，可以允许 SSH 流量：

```
sudo ufw allow ssh
sudo ufw status
```

6. **连接到 SSH 服务器：** 使用 SSH 客户端连接到服务器，例如：

```
ssh username@server_ip
```

Q8. Debian 和 Ubuntu 配置网络环境所使用的命令和配置文件有哪些？

Ubuntu Server 和 Debian Standard 在配置系统环境时，虽然很多命令和工具是类似的，但也有一些差别：

包管理系统

Ubuntu Server：

- 使用 apt 进行包管理。
- 配置文件 /etc/apt/sources.list 中，官方源通常以 archive.ubuntu.com 开头。

Debian Standard：

- 也使用 apt 进行包管理。
- 配置文件 /etc/apt/sources.list 中，官方源通常以 deb.debian.org 开头。

网络配置

Ubuntu Server:

- 使用 netplan 进行网络配置。
- 配置文件位于 /etc/netplan/ 目录下，文件通常名为 00-installer-config.yaml。

```
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: true
```

Debian Standard:

- 传统上使用 /etc/network/interfaces 文件进行网络配置，不过也可以使用 netplan。

```
auto ens33
iface ens33 inet dhcp
```

激活新的网络接口和重启网络时使用的命令

Ubuntu Server

激活新的网络接口

```
sudo ip link set ens33 up
```

重启网络

1. 使用 netplan:

```
sudo netplan apply
```

2. 使用 systemd-networkd:

```
sudo systemctl restart systemd-networkd
```

3. 使用 NetworkManager:

```
sudo systemctl restart NetworkManager
```

Debian Standard

激活新的网络接口

```
sudo ip link set ens33 up
```

重启网络

1. 使用 ifupdown (传统方法):

```
sudo ifdown ens33 && sudo ifup ens33
```

2. 使用 systemd-networkd:

```
sudo systemctl restart systemd-networkd
```

3. 使用 NetworkManager:

```
sudo systemctl restart NetworkManager
```

服务管理

Ubuntu Server:

- 使用 systemd 进行服务管理。

```
sudo systemctl start <service>
sudo systemctl stop <service>
sudo systemctl enable <service>
```

Debian Standard:

- 也使用 systemd 进行服务管理，命令与 Ubuntu 相同。

用户和权限管理

Ubuntu Server:

- 默认启用 sudo，用户可以通过 sudo 获得超级用户权限。

```
sudo adduser <username>
sudo usermod -aG sudo <username>
```

Debian Standard:

- 也使用 sudo，但初始安装时可能需要手动安装并配置。

```
sudo apt install sudo
sudo adduser <username>
sudo usermod -aG sudo <username>
```

默认软件包

Ubuntu Server:

- 通常会预装一些常用的服务和工具，例如 curl、wget 和 ssh。

Debian Standard:

- 通常只包含最基本的系统工具，用户需要手动安装额外的软件包。

Q9. TShark 和 Wireshark 什么关系？ Debian 上如何安装？

Tshark 是一个命令行工具，用于网络数据包捕获和分析，它是 Wireshark 的命令行版本。Tshark 可以在多个操作系统上使用，包括 Linux、Windows 和 macOS。

在 Debian 上单独安装 Tshark：

1. 更新软件包列表：

```
sudo apt-get update
```

2. 安装 Tshark：

```
sudo apt-get install tshark
```

这样，Tshark 就会被安装到你的系统中，而无需安装 Wireshark。

运行以下命令确认安装成功：

```
tshark -v
```

注意，TShark 运行需要 Root 权限。

Q10. 如何使用 vs code 通过 SSH 远程登录到 Linux 系统进行 C 程序开发

让我们详细介绍如何使用 VS Code 通过 SSH 远程登录 Linux 系统进行 C 程序的编写、编译、运行及调试。以下是具体步骤：

1. 准备工作

- **安装 VS Code**：确保你已经安装了 Visual Studio Code。
- **安装 Remote - SSH 扩展**：打开 VS Code，点击左侧的扩展图标（或按 Ctrl+Shift+X），搜索并安装 **Remote - SSH** 扩展。

2. 配置 SSH 连接

1. **打开命令面板**：按 Ctrl+Shift+P，输入并选择“Remote-SSH: Connect to Host...”
2. **添加新的 SSH 主机**：输入 SSH 连接字符串，例如：ssh user@hostname
3. **输入密码或使用 SSH 密钥认证**：根据提示完成连接。

3. 创建和编辑 C 程序

1. **创建文件**：在远程连接的 Linux 文件系统中，创建一个新的文件，命名为 hello.c

2. **编写代码**：编写一个简单的 Hello World 程序：

```
#include <stdio.h>

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

4. 编译 C 程序

1. **打开终端**：在 VS Code 中，点击终端按钮（或按`Ctrl+```）。
2. **编译代码**：使用 gcc 编译器编译程序：

```
gcc -g hello.c -o hello
```

5. 运行 C 程序

1. **运行程序**：在终端中运行编译后的可执行文件：

```
./hello
```

你应该会看到输出：Hello, World!

6. 配置调试

1. **添加调试配置**：点击左侧的调试图标，点击齿轮图标选择“C++ (GDB/LLDB)”，生成 launch.json 文件。
2. **编辑 launch.json**：

```
{
    "version": "0.2.0",
    "configurations": [
        {
            "name": "GDB",
            "type": "cppdbg",
            "request": "launch",
            "program": "${workspaceFolder}/hello",
            "args": [],
            "stopAtEntry": false,
            "cwd": "${workspaceFolder}",
            "environment": [],
            "externalConsole": true,
            "MIMode": "gdb",
            "miDebuggerPath": "/usr/bin/gdb",
            "setupCommands": [
                {
```

```

        "description": "Enable pretty-printing for gdb",
        "text": "-enable-pretty-printing",
        "ignoreFailures": true
    },
    ],
    "preLaunchTask": "build",
    "internalConsoleOptions": "openOnSessionStart",
    "linux": {
        "MIMode": "gdb",
        "miDebuggerPath": "/usr/bin/gdb"
    }
}
]
}

```

7. 调试 C 程序

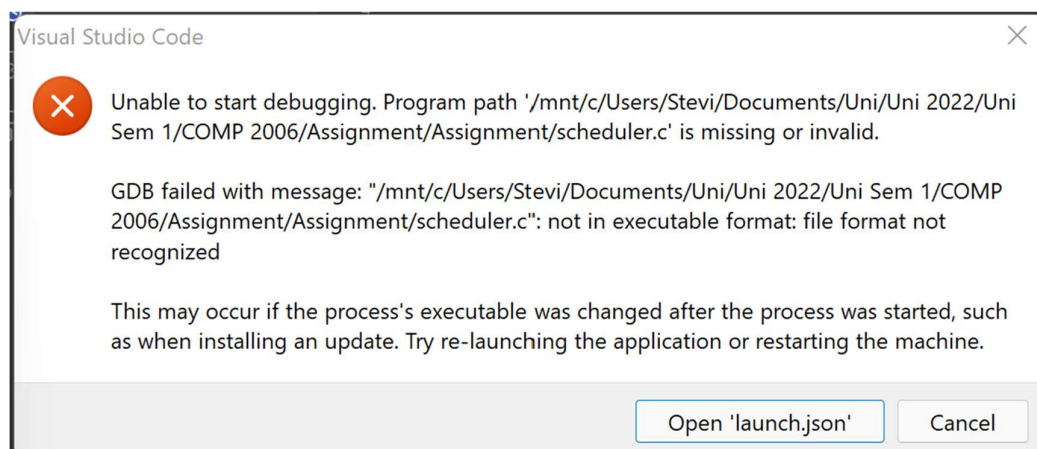
1. **编译程序**：确保你的程序已经编译：

```
gcc -g hello.c -o hello
```

2. **启动调试**：在 VS Code 调试面板中选择“GDB”，然后点击“启动调试”。

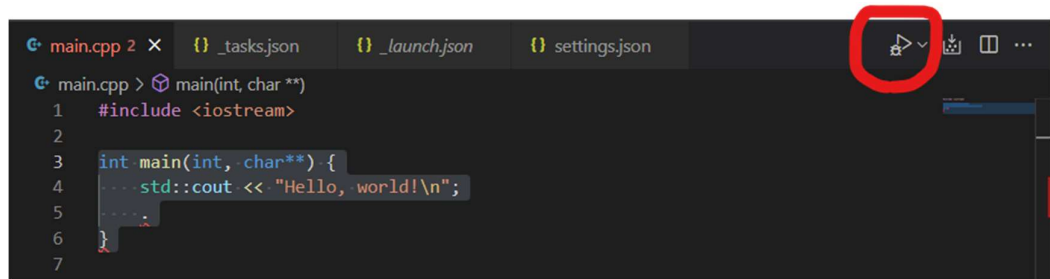
通过这些步骤，你就可以成功在 VS Code 中通过 SSH 远程登录 Linux 系统，编写、编译、运行及调试 C 程序了。

Q11. VS code 调试程序遇到以下错误提示，我该怎么办？

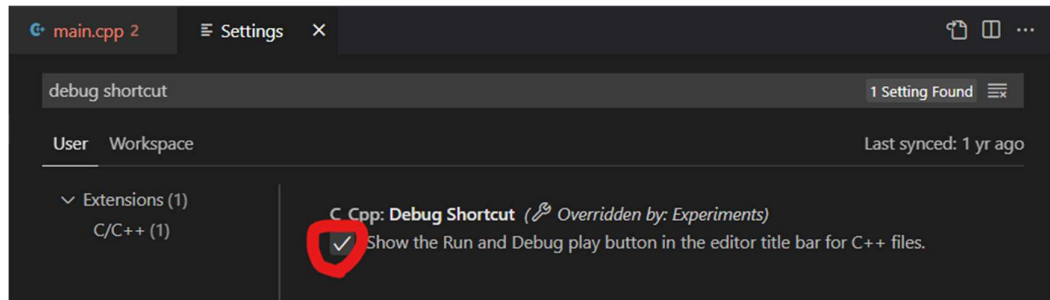


请按照以下步骤操作

1. 打开一个文件夹，然后打开要调试的文件（scheduler.c）
2. 同时删除当前的 tasks.json 和 launch.json。
3. 你会看到右上方有一个调试按钮；



如果没有看到该按钮，可能需要进入设置并激活它：



4. 单击调试按钮后，您将看到一个调试配置列表，供您选择。选择有 "PreLaunchTask: C/C++: gcc build active file" 的配置。
5. 调试应该能启动
6. 你会发现任务.json 已创建。

Q12. 防火墙如何开放接收 UDP 流量

防火墙阻止 UDP 流量是一种常见的网络安全措施。UDP（用户数据报协议）流量通常用于快速传输数据，但也可能被用于恶意软件和网络攻击。防火墙可以通过以下几种方式来阻止 UDP 流量：

1. **过滤规则**：设置防火墙规则，拒绝特定 UDP 端口的流量。
2. **应用屏蔽**：使用应用屏蔽技术，识别并阻止恶意的 UDP 流量。
3. **状态检查**：启用状态检查，确保只允许已建立的连接的 UDP 流量。

防火墙开放接收 UDP 流量

要开放防火墙以接收 UDP 流量，可以按照以下步骤操作，具体取决于你所使用的防火墙管理工具。

使用 UFW (Uncomplicated Firewall)

1. **安装并启用 UFW**（如果未安装）：

```
sudo apt update
sudo apt install ufw
```

```
sudo ufw enable
```

2. 允许特定端口的 UDP 流量（例如端口 1234）：

```
sudo ufw allow 1234/udp
```

3. 检查 UFW 状态：

```
sudo ufw status
```

使用 iptables

1. 允许特定端口的 UDP 流量（例如端口 1234）：

```
sudo iptables -A INPUT -p udp --dport 1234 -j ACCEPT
```

2. 保存 iptables 规则（确保规则在重启后仍然有效）：

```
sudo apt install iptables-persistent
sudo netfilter-persistent save
sudo netfilter-persistent reload
```

这两种方法都可以帮助你开放防火墙以接收特定端口的 UDP 流量。

Q13. UFW 防火墙有什么特点？

UFW（Uncomplicated Firewall）和 iptables 都是用于管理 Linux 系统防火墙的工具，但它们有一些关键区别：

UFW

- **简单易用**：UFW 是 iptables 的一个简化接口，旨在使防火墙配置更加简单和直观。
- **高级用户友好**：适合那些不太熟悉 iptables 命令的用户。
- **基于 iptables**：UFW 实际上是在 iptables 之上构建的，所以它的功能是基于 iptables 的。

iptables

- **灵活性高**：iptables 提供了更多的灵活性和控制，适合需要更复杂配置的高级用户 1。
- **命令繁琐**：iptables 的命令语法相对复杂，需要更多的学习和维护 1。
- **功能强大**：iptables 可以实现更复杂的网络策略和规则，包括 NAT、网络地址转换（NAT）、数据包修改等 2。

总的来说，如果你是一个初学者或需要简单的防火墙配置，UFW 是一个不错的选择。如果你需要更高级的控制和灵活性，iptables 会更适合你。

Q14. 可以在系统中同时安装并使用 UFW 和 iptables?

其实，UFW 就是基于 iptables 的一个更简化的界面，所以两者是兼容的。

安装和使用步骤

1. 安装 UFW 和 iptables:

```
sudo apt update
sudo apt install ufw iptables
```

2. 启用 UFW:

```
sudo ufw enable
```

3. 使用 UFW 管理简单规则:

- 例如，允许 SSH 连接:

```
sudo ufw allow ssh
```

4. 使用 iptables 管理复杂规则:

- 例如，添加一条规则以阻止特定 IP 地址:

```
sudo iptables -A INPUT -s 192.168.0.1 -j DROP
```

这使得你可以利用 UFW 的简便性，同时也能使用 iptables 的强大功能。

Q15. 如何在 Debian Ubuntu Linux 主机中访问虚拟机的共享文件夹?

可以按照以下步骤进行:

1. **确保 VMware Tools 已安装:** 在虚拟机中安装 VMware Tools，这是访问共享文件夹的基础。可以通过虚拟机的“帮助”菜单找到安装选项。
2. **设置共享文件夹:** 在虚拟机中，进入“设置”->“共享文件夹”，启用共享文件夹功能，并添加你想要共享的文件夹路径。
3. **在 Ubuntu 中挂载共享文件夹:**

- 打开终端，输入以下命令来挂载共享文件夹:

```
sudo vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other -o uid=0 -o gid=0 -o umask=022
```

- 如果挂载失败，可以尝试以下命令:

```
sudo vmhgfs-fuse .host:/ /mnt/hgfs -o allow_other -o uid=0 -o gid=0 -o nonempty
```

- 确保 /mnt/hgfs 目录存在，如果不存在，可以先创建：

```
sudo mkdir /mnt/hgfs
```

4. **检查挂载状态**：输入以下命令查看共享文件夹是否成功挂载：

```
ls /mnt/hgfs
```

5. **永久挂载**（可选）：如果希望共享文件夹在每次开机时自动挂载，可以编辑 /etc/fstab 文件，添加以下行：

```
.host:/ /mnt/hgfs fuse.vmhgfs-fuse allow_other,uid=0,gid=0,umask=022 0 0
```

完成以上步骤后，你就可以在 Ubuntu 主机中访问虚拟机的共享文件夹了。

Q16. ubuntu（普通用户）登录后，设置 root 账户密码，及如何切换账户

```
sudo passwd root
```

切换到 root 账户

```
su -
```

从 root 账户返回到普通用户

```
exit
```

Q17. 在 Ubuntu Server 查看网络接口和网络配置的命令

```
ip a
```

重启网络

```
sudo netplan apply
```

Q18. Debian 和 Ubuntu 如何配置双网口主机？

Ubuntu

Ubuntu 使用 Netplan 来配置网络，Netplan 通过 YAML 文件定义网络设置 1。

1. **配置文件路径**：/etc/netplan/
2. **示例配置文件**：

```
network:
  version: 2
  ethernets:
```

```
eth0:
  dhcp4: no
  addresses: [192.168.1.100/24]
  gateway4: 192.168.1.1
  nameservers:
    addresses: [8.8.8.8, 8.8.4.4]
eth1:
  dhcp4: no
  addresses: [192.168.2.100/24]
  gateway4: 192.168.2.1
  nameservers:
    addresses: [8.8.8.8, 8.8.4.4]
routes:
  - to: 192.168.2.0/24
    via: 192.168.1.1
    metric: 100
  - to: 192.168.1.0/24
    via: 192.168.2.1
    metric: 100
```

3. **应用配置**：保存文件后，运行以下命令应用配置：

```
sudo netplan apply
```

Debian

Debian 使用/etc/network/interfaces 文件来配置网络 2。

1. **配置文件路径**：/etc/network/interfaces
2. **示例配置文件**：

```
auto eth0
iface eth0 inet static
  address 192.168.1.100
  netmask 255.255.255.0
  gateway 192.168.1.1
  dns-nameservers 8.8.8.8 8.8.4.4

auto eth1
iface eth1 inet static
  address 192.168.2.100
  netmask 255.255.255.0
  gateway 192.168.2.1
  dns-nameservers 8.8.8.8 8.8.4.4

up ip route add 192.168.2.0/24 via 192.168.1.1
```

```
up ip route add 192.168.1.0/24 via 192.168.2.1
```

3. **应用配置**：保存文件后，运行以下命令应用配置：

```
sudo ifup eth0  
sudo ifup eth1
```

这两个系统都提供了灵活的网络配置选项，但 Ubuntu 的 Netplan 配置更为简洁和直观，而 Debian 的配置方式更为传统。

Q19. 在 Ubuntu 主机上如何配置双网口

您可以按照以下步骤进行操作：

1. **检查网络接口**：首先，使用以下命令列出所有网络接口：

```
ip a
```

2. **配置静态 IP**：如果需要为某个接口配置静态 IP，可以编辑 `/etc/netplan/*.yaml` 文件。例如：

```
network:  
  version: 2  
  ethernet:  
    eth0:  
      dhcp4: no  
      addresses: [192.168.1.100/24]  
      gateway4: 192.168.1.1  
      nameservers:  
        addresses: [8.8.8.8, 8.8.4.4]  
    eth1:  
      dhcp4: no  
      addresses: [192.168.2.100/24]  
      gateway4: 192.168.2.1  
      nameservers:  
        addresses: [8.8.8.8, 8.8.4.4]
```

3. **应用配置**：保存文件后，运行以下命令应用配置：

```
sudo netplan apply
```

4. **检查配置**：最后，再次运行 `ip a` 命令检查网络接口配置是否正确。

Q20. 如何启用 IP 转发和配置防火墙（使用 iptables 或 nftables）

在 Ubuntu 和 Debian 系统上的操作步骤是类似的，但有些小差异。我们来看看具体的步骤：

Ubuntu

启用 IP 转发

1. 编辑 sysctl 配置文件：

```
sudo nano /etc/sysctl.conf
```

2. 找到并取消注释以下行：

```
net.ipv4.ip_forward=1
```

3. 重新加载 sysctl 配置：

```
sudo sysctl -p
```

配置防火墙（使用 iptables）

1. 允许转发流量：

```
sudo iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

2. 配置 NAT：

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

3. 保存 iptables 规则：

```
sudo iptables-save | sudo tee /etc/iptables/rules.v4
```

Debian

启用 IP 转发

1. 编辑 sysctl 配置文件：

```
sudo nano /etc/sysctl.conf
```

2. 找到并取消注释以下行：

```
net.ipv4.ip_forward=1
```

3. 重新加载 sysctl 配置：

```
sudo sysctl -p
```

配置防火墙（使用 iptables）

1. 允许转发流量:

```
sudo iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

2. 配置 NAT:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
sudo iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

3. 保存 iptables 规则:

```
sudo iptables-save > /etc/iptables/rules.v4
```

使用 nftables (Ubuntu 和 Debian)

nftables 是 iptables 的替代工具，可以在两者上使用。

1. 安装 nftables (如果未安装):

```
sudo apt-get install nftables
```

2. 配置文件:

```
sudo nano /etc/nftables.conf
```

3. 示例配置:

```
table inet filter {
    chain input {
        type filter hook input priority 0; policy accept;
    }

    chain forward {
        type filter hook forward priority 0; policy accept;
        iif "eth0" oif "eth1" accept
        iif "eth1" oif "eth0" accept
    }

    chain output {
        type filter hook output priority 0; policy accept;
    }
}

table ip nat {
    chain postrouting {
        type nat hook postrouting priority 100; policy accept;
        oif "eth0" masquerade
    }
}
```

```
oif "eth1" masquerade  
}  
}
```

4. 启动 nftables:

```
sudo systemctl enable nftables  
sudo systemctl start nftables
```