**Linux系统环境搭建**

## 1 配置静态IP地址

**sudo vi /etc/network/interfaces**

当没有安装vim时，在vi命令行使用命令“：set nocompatible”，设置vi不使用兼容模式，就不会出现键盘上下键为AB了

添加以下内容指定固定IP地址:

|  |
| --- |
| # The loopback network interface（配置环回口）  auto lo # 开机自动激lo接口  iface lo inet loopback # 配置lo接口为环回口  # The primary network interface （配置主网络接口）  auto eth0 # 开机自动激活eth0接口  iface eth0 inet static # 配置eth0接口为静态设置IP地址  address 202.202.171.99  netmask 255.255.254.0  broadcast 202.202.171.255  gateway 202.202.171.10 |

或者添加以下内容由DHCP自动获取IP地址：

|  |
| --- |
| # The loopback network interface（配置环回口）  auto lo # 开机自动激lo接口  iface lo inet loopback # 配置lo接口为环回口  # The primary network interface （配置主网络接口）  auto eth0 # 开机自动激活eth0接口  iface eth0 inet dhcp # 配置eth0接口为DHCP自动获取 |

**sudo /etc/init.d/networking restart** # 重启网络

## 2 设置DNS server，使其ping通域名

**sudo vi /etc/resolv.conf**

添加以下内容：

|  |
| --- |
| nameserver 202.96.128.86  nameserver 202.96.128.166 |

系统重启后resolv.conf文件会复原，每次系统重启都要手动添加，下面介绍两种自动添加的方法。

1. 在脚本文件rc.local执行自启动执行命令

**sudo vi /etc/rc.local**

添加下面内容：

|  |
| --- |
| echo "nameserver 202.96.128.86" >> /etc/resolv.conf  echo "nameserver 202.96.128.166" >> /etc/resolv.conf |

1. 修改文件/etc/resolvconf/resolv.conf.d/base，系统启动后自动复制base内容到resolv.conf

**sudo vi /etc/resolvconf/resolv.conf.d/base**

添加下面内容：

|  |
| --- |
| nameserver 202.96.128.86  nameserver 202.96.128.166 |

## 3 更新软件源

**sudo apt-get update && sudo apt-get upgrade**

## 4 安装vim编辑器

**apt-get install vim**

修改配置：

**vim ~/.vimrc**

添加以下内容：

|  |
| --- |
| set nu  syntax on  set tabstop=4  set shiftwidth=4  set expandtab  set guioptions-=T  set nobackup  set incsearch  set encoding=utf-8  set fileencodings=utf-8,gb2312,gbk  set mouse=a  set ruler  set ff=unix |

## 5 安装新的终端软件

**sudo apt-get install terminator**

## 6 安装文本编辑器

**sudo apt-get install gedit**

## 7安装gcc编译器

**sudo apt-get install gcc**

## 8安装SSH服务

安装openssh-server：

**sudo apt-get install openssh-server**

也可以通过 Ubuntu Software Center 安装

由于 sshd 需要反查客户端的 dns 信息导致登录时间更长。可以通过禁用这个特性来提高登录的速度。

**sudo vim /etc/ssh/sshd\_config**

找到 GSSAPI options 这一节，将下面两行注释掉：

|  |
| --- |
| #GSSAPIAuthentication yes  #GSSAPIDelegateCredentials no |

修改其他配置

|  |
| --- |
| PasswordAuthentication yes  PermitRootLogin yes |

重启动SSH服务器：

**sudo /etc/init.d/ssh restart**

查看openssh-server是否启动：

**ps -e | grep ssh**

在window系统下打开SecureCRT的SSH2连接，输入ubuntu的IP、用户名和密码，可以代替telnet。

或者在其他系统终端上输入命令登录：ssh -l 用户 名远程主机

## 9安装Samba网络服务器,实现windows和ubuntu文件互访

**① 安装samba，smbclient，smbfs**

**sudo apt-get install samba**

**sudo apt-get install smbclient**

**② 编辑配置文件**

**sudo cp /etc/samba/smb.conf /etc/samba/smb.conf\_backup**

**sudo vim /etc/samba/smb.conf**

　　在字段**[global]**下面加入内容如下：

|  |
| --- |
| dos charset = cp936 # UTF-8 有可能需要改成 cp936  unix charset = UTF-8  workgroup = WORKGROUP  security = user # 需要身份验证  netbios name = smbserver # 这里填本机的计算机名 |

注销在字段#**[homes]**下所有内容如下：

|  |
| --- |
| #browseable = yes  #writable = yes |

分别找到**passwd program**和**security**行，将其改为

|  |
| --- |
| passwd program = /etc/samba/smbpasswd %u  security = user # 可能找不到，在附近添加 |

在文件末尾添加创建共享文件夹字段

|  |
| --- |
| [myShare] # 共享文件名  admin users = bananapi # 该共享的管理者  path = /home/bananapi # 共享目录路径  public = no # 用来指定该共享是否允许guest账户访问  writable = yes  browseable = yes  create mask = 0777  directory mask = 0777  guest ok = no # 是否允许guest账户访问 |

**③ 添加smb用户，并设置linux用户密码**

添加linux用户

**sudo adduser -a user** *# user为用户名*

　　把新添加的Linux用户添加到smb用户，并设置smb密码

**sudo smbpasswd -a user** *# 已存在的linux用户*

**④ 检验是否配置成功**

**sudo testparm**

重启下samba服务：

**sudo /etc/init.d/samba restart**

**⑤ 文件访问方式**

在ubuntu系统下，随便打开一个文件夹，然后输入路径**smb://202.202.170.113**(windows IP)

在windown系统下，随便打开一个文件夹，然后输入路径**\\202.202.171.99**(ubuntu IP)

## 10安装WiringPi(只在开发板banana Pro上)

wiringPi 库是由 Gordon Henderson 所编写幵维护的一个用 C 语言写成的类库，除了 GPIO 库，还包括了I2C 库、SPI 库、UART 库和软件 PWM 库等。

**① 先安装 git 工具。**

**sudo apt-get install git-core**

安装期间如果发生了某些错误，请更新apt库：sudo apt-get update

**② 安装 git 工具后，就可以使用如下的命令来获取 wiringPi。**

对于Banana Pro电路板:

**git clone https://github.com/LeMaker/WiringBP -b bananapro**

对于Banana Pi电路板:

**git clone https://github.com/LeMaker/WiringBP -b bananapi**

对于Banana Pi M2+或M3电路板:

**git clone https://github.com/BPI-SINOVOIP/BPI-WiringPi.git -b BPI\_M3**

**③ 下载完WiringBP后，进入WiringBP文件夹，修改build为可执行文件类型**。

**cd WiringBP/**

**sudo chmod +x ./build**

**sudo ./build**

**④ 检验WiringBP是否安装成功，列表显示具体管脚物理映射。**

**gpio readall**

**注意编译文件时需要加后缀 -lwiringPi -lpthread**

例如：sudo gcc test.c -o test -lwiringPi -lpthread

执行：sudo ./test

**WiringBP默认是没有启动SPI驱动的，设置SPI接口驱动：**

**把modprobe spi\_sun7i命令行添加到开机启动的脚本文件rc.local下。**

## 11 设置自动连接WIFI

**① 自动加载无线模块**

**sudo vim /etc/modules**

在文件末尾添加**ap6210**

**sudo modprobe ap6210**

**② 配置WiFi参数**

**sudo vim /etc/network/interfaces**

在文件尾添加下面内容(可能默认是注释掉的)：

|  |
| --- |
| auto wlan0  iface wlan0 inet dhcp # 自动分配ip地址  wpa-ssid "A900K" # “A900K”是连接wifi名称  wpa-psk "1234567890" # “1234567890”是登陆密码 |

**③ 重启系统**

**sudo reboot**

## 12 设置WIFI为共享热点

**这里提供2种设置方式，建议选择第二种设置。**

### 12.1设置方式一：不能打开网页连接，可以用微信和QQ

**① 安装hostapd，dnsmasq，iptables三个工具包**

**sudo apt-get install hostapd**

**sudo apt-get install dnsmasq**

**sudo apt-get install iptables**

**② 在目录/usr/share/doc/hostapd/examples下解压和编辑hostapd.conf.gz**

**cd /usr/share/doc/hostapd/examples**

**sudo gzip -d hostapd.conf.gz**

把修hostapd.conf文件复制到目录/etc下

**sudo cp hostapd.conf /etc**

修改hostapd.conf参数

**cd /etc**

**sudo vim hostapd.conf**

修改参数：**interface=实际使用wlan端口，ssid=自己定义的wifi名** 修改参数：

设置wifi密码：

|  |
| --- |
| wpa=2  wpa\_passphrase=1234567890  wpa\_key\_mgmt=WPA-PSK  rsn\_pairwise=TKIP CCMP |

**③ 修改/etc/dnsmasq.conf参数**

**cd /etc**

**sudo vim dnsmasq.conf**

修改接口，把第94行#interface=修改为interface=实际使用的wlan端口

修改分配地址范围，把第145行dhcp-range=192.168.123.10,192.168.123.254,12h

**④ 新建一个wifihotpoint.sh文件来启动、停止wifi热点**

**cd**

**vim wifihotpoint.sh**

添加下面内容（注意下面的wlan名称要和实际使用wlan端口一致）

|  |
| --- |
| #!/bin/sh  #Clean things up  init() {  #Stop NetworkManager, if already running (it will disturb you)  /usr/sbin/service network-manager stop  #Stop named, if already running. dnsmasq cannot run because it take up port 53  killall named  #Stop dnsmasq, if already running  /usr/sbin/service dnsmasq stop  #Stop hostapd, if already running  /usr/bin/pkill hostapd  #Bring down wlan2  /sbin/ip link set down dev wlan2  }  start() {  #First clean things up  init  #Start hostapd, and it will automatically be bringed up  /usr/sbin/hostapd -B /etc/hostapd.conf  #Set ip on wlan2  /sbin/ip addr add 192.168.123.1/24 dev wlan2  #Start dnsmasq  /usr/sbin/service dnsmasq start  #Start ip\_forward  echo 1 > /proc/sys/net/ipv4/ip\_forward  #add iptables rule for NAT  /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  #Recovery file /etc/resolv.conf  echo "nameserver 202.96.128.86" >> /etc/resolv.conf  echo "nameserver 202.96.128.166" >> /etc/resolv.conf  }  stop() {  #Remove iptables rule  /sbin/iptables -t nat -D POSTROUTING -o eth0 -j MASQUERADE  #Stop dnsmasq  /usr/sbin/service dnsmasq stop  #Stop hostapd  /usr/bin/pkill hostapd  #bring down wlan2, and its ip address will automatically be removed  /sbin/ip link set down dev wlan2  }  case "$1" in  'start')  start  ;;  'stop')  stop  ;;  \*)  echo "usage $0 start|stop"  esac |

接着把window文件转为unix文件，按ESC键，按shift+： 输入set ff=unix

**⑤ 启动wifi热点，成功启动提示可以看到ssid “Banana\_Pro”**

**sudo sh \usr\wifihotpoint.sh start**

停止wifi热点使用命令sudo sh \usr\wifihotpoint.sh stop

### 12.2设置方式二：可以连接外网

**① 安装hostapd，iptables，dhcp3-server三个工具包**

**sudo apt-get install hostapd**

**sudo apt-get install iptables**

**sudo apt-get install dhcp3-server**

如果dhcp3-server不能安装，执行下面步骤：

**sudo vim /etc/apt/sources.list**

在文件尾添加下面内容：

|  |
| --- |
| deb http://all.repository.backtrack-linux.org revolution main microverse non-free testing  deb http://64.repository.backtrack-linux.org revolution main microverse non-free testing  deb http://source.repository.backtrack-linux.org revolution main microverse non-free testing  deb http://updates.repository.backtrack-linux.org revolution main microverse non-free testing |

更新源列表：**sudo apt-get update**

**② 新建hostapd.conf配置文件**

**sudo vim /etc/hostapd/hostapd.conf**

添加下面内容：

|  |
| --- |
| interface=wlan2 # 表示无线端口  driver=nl80211  ssid=wifi\_test # 表示共享热点wifi名称  hw\_mode=g  channel=10  macaddr\_acl=0  auth\_algs=3  wpa=2  wpa\_passphrase=1234567890 # WiFi密码  wpa\_key\_mgmt=WPA-PSK  wpa\_pairwise=TKIP CCMP  rsn\_pairwise=TKIP CCMP |

**③ 修改自动分配IP的配置文件参数**

**sudo vim /etc/dhcp/dhcpd.conf**

在文件尾添加一下内容：

|  |
| --- |
| subnet 192.168.8.0 netmask 255.255.255.0 {  range 192.168.8.100 192.168.8.200; # 自动分配的IP范围  option domain-name-servers 202.96.128.86; # eth0的DNS解析IP  option routers 192.168.8.1; # 默认网关  } |

**④ 新建一个脚本文件wifihotpoint.sh**

**cd**

**vim wifihotpoint.sh**

添加下面内容：

|  |
| --- |
| hostapd -B /etc/hostapd/hostapd.conf #启动hostapd  ifconfig wlan2 192.168.0.1 netmask 255.255.255.0 #设置无线网卡  dhcpd wlan2 -pf /var/run/dhcp-server/dhcpd.pid #启动dhcp-server  bash -c "echo 1 >/proc/sys/net/ipv4/ip\_forward" #打开IP转发  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE #设置NAT，eth0为连接外网端口 |

**⑤ 启动wifi热点**

**sudo sh ./ wifihotpoint.sh**

停止wifi热点用sudo killall hostapd

## 13 设置WIFI路由

**① 设置静态IP**

**sudo vim /etc/network/interfaces**

设置为以下内容：

|  |
| --- |
| auto eth0  iface eth0 inet static  address 202.202.171.99  netmask 255.255.254.0  gateway 202.202.171.10  broadcast 202.202.171.255  auto wlan2  iface wlan2 inet static  address 192.168.8.1  network 192.168.8.0  netmask 255.255.255.0  broadcast 192.168.8.255 |

重启网络sudo /etc/init.d/networking restart

**② 安装hostapd、iptables和 DHCP服务器**

**sudo apt-get install hostapd**

**sudo apt-get install iptables**

**sudo apt-get install isc-dhcp-common isc-dhcp-serve**

**③ 配置hostapd.conf文件**

**sudo vim /etc/hostapd/hostapd.conf**

添加下面内容：

|  |
| --- |
| interface=wlan2 # 表示无线端口  driver=nl80211  ssid=wifi\_test # 表示共享热点wifi名称  hw\_mode=g  channel=10  macaddr\_acl=0  auth\_algs=3  wpa=2  wpa\_passphrase=1234567890 # WiFi密码  wpa\_key\_mgmt=WPA-PSK  wpa\_pairwise=TKIP CCMP  rsn\_pairwise=TKIP CCMP |

**④ 配置DHCP**

修改接口

**sudo vim /etc/default/isc-dhcp-server**

修改INTERFACES="wlan2"，其中wlan2为接口名称

分发IP

**sudo vim /etc/dhcp/dhcpd.conf**

在最后行添加下面内容：

|  |
| --- |
| subnet 192.168.8.0 netmask 255.255.255.0 {  range 192.168.8.100 192.168.8.200;  option routers 192.168.8.1;  option broadcast-address 192.168.8.255;  option domain-name-servers 192.168.8.1;  default-lease-time 600;  max-lease-time 7200;  } |

**⑤ 启动wifi**

新建一个脚本文件wifi\_route.sh，

**touch wifi\_route.sh**

**sudo chmod 777 wifi\_route.sh**

**vim wifi\_route.sh**

添加下面内容：

|  |
| --- |
| hostapd -B /etc/hostapd/hostapd.conf  dhcpd wlan2 -pf /var/run/dhcp-server/dhcpd.pid  bash -c "echo 1 >/proc/sys/net/ipv4/ip\_forward"  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE |

启动wifi

**sudo sh wifi\_route.sh**

**注意：/etc/network/interfaces文件下wlan2的address必须和/etc/dhcp/dhcpd.conf下的option routers的ip一样（即wifi的IP地址作为分发IP的网关）。如果不一样会导致PC可以ping通wifi连接的设备，但wifi连接的设备不能ping通PC的情况。**

## 14搭建web服务器

首先是升级更新源

**sudo apt-get update && sudo apt-get upgrade**

### 14.1 安装MySQL5

**① 安装**

**sudo apt-get install mysql-server mysql-client**

安装过程中要求提供一个MySQL的root用户的密码，输入两次密码:

New password for the MySQL "root" user: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Repeat password for the MySQL "root" user: \_\_\_\_\_\_\_\_\_\_\_\_

安装完后查看是否安装成功：**sudo netstat -tap | grep mysql**

**② 配置**

复制配置文件并改名为my.cnf：

**sudo cp /etc/mysql/mysql.conf.d/mysqld.cnf /etc/mysql/my.cnf**

修改配置：

**sudo gedit /etc/mysql/my.cnf**

注销行bind-address = 127.0.0.1

在[mysqld]下添加skip-grant-tables

在文件最后添加内容：

|  |
| --- |
| [client]  port = 3306  socket = /var/run/mysqld/mysqld.sock |

解决phpmyadmin登录问题：#2002 - The server is not responding (or the local MySQL server's socket is not correctly configured)

**③ 允许远程登录**

在Ubuntu终端登录mysql客户端后，输入命令。

允许所有远程登录（%号表示所有IP）：

**GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'%' IDENTIFIED BY '123456' WITH GRANT OPTION;**

或者只允许指定IP登录：

**GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'192.168.8.100' IDENTIFIED BY '123456' WITH GRANT OPTION;**

使修改生效：**FLUSH PRIVILEGES**

启动：sudo service mysql start

停止：sudo service mysql stop

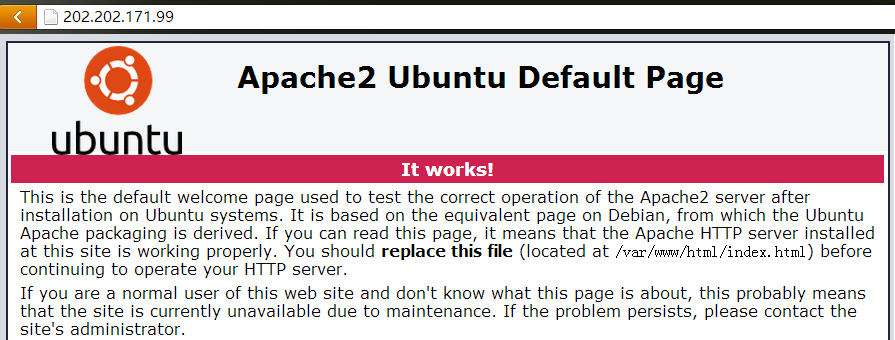
重启：sudo service mysql restart

### 14.2 安装Apache2

**sudo apt-get install apache2**

在你的浏览器上输入<http://localhost>或<http://主机ip地址>或<http://127.0.0.1>或者在连接主机的其他上网设备输入<http://主机ip地址>

能看到下面界面说明Apache2正常。



网站默认的文件根目录是/var/www/html（有些文章介绍默认根目录是/var/www/），具体是哪个，通过打开/etc/apache2/sites-available/000-default.conf文件，查看DocumentRoot后面跟着的路径。

启动：sudo /etc/init.d/apache2 start

重启：sudo /etc/init.d/apache2 restart

停止：sudo /etc/init.d/apache2 stop

### 14.3 安装PHP

① 安装PHP7和响应的扩展

**sudo apt-get install php7.0 php-pear libapache2-mod-php7.0 php7.0-mysql**

② 安装组件支持

**sudo apt-get install php7.0-curl php7.0-json php7.0-cgi**

可以根据需求配置文件/etc/php/7.0/apache2/php.ini

|  |
| --- |
| # 最大输入时间  max\_input\_time = 30  # 输出错误级别设置  error\_reporting = E\_COMPILE\_ERROR | E\_RECOVERABLE\_ERROR | E\_ERROR | E\_CORE\_ERROR  # 错误报告保存位置  error\_log = /var/log/php/error.log |

安装完需要重启Apache2

**sudo /etc/init.d/apache2 restart**

验证PHP5是否正常：在网站的文件根目录下（/var/www/html）新建info.php

**sudo vim /var/www/html/info.php**

输入下面内容：<?php phpinfo();?>

在服务器本机或正常连接服务器的设备上用浏览器访问http://主机ip/info.php，能看到下面界面说明PHP7.0正常。



### 14.4 安装phpMyAdmin

**sudo apt-get install phpmyadmin**

安装过程中遇到配置设置：

Web server to reconfigure automatically：选择apache2

Configure database for phpmyadmin with dbconfig-common? 选择NO

安装完后把phpmyadmin文件夹放入网站的文件根目录下

**sudo cp /usr/share/phpmyadmin/ /var/www/html/ -R**

或映射 **sudo ln -s /usr/share/phpmyadmin /var/www/html/**

在浏览器输入<http://主机IP/phpmyadmin/> 登陆服务器名：root 密码：数据库密码123456，登陆后界面如下图所示：



解决（phpMyAdmin 高级功能尚未完全设置，部分功能未激活。请点击这里查看原因）的方法

上传create\_tables.sql.zip文件

**cd /usr/share/doc/phpmyadmin/examples/**

**sudo gzip -d create\_tables.sql.gz**

**sudo zip create\_tables.sql.zip ./create\_tables.sql**



修改/etc/phpmyadmin/ config.inc.php配置文件，取消屏蔽下面的语句

|  |
| --- |
| // $cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';  // $cfg['Servers'][$i]['bookmarktable'] = 'pma\_\_bookmark';  // $cfg['Servers'][$i]['relation'] = 'pma\_\_relation';  // $cfg['Servers'][$i]['table\_info'] = 'pma\_\_table\_info';  // $cfg['Servers'][$i]['table\_coords'] = 'pma\_\_table\_coords';  // $cfg['Servers'][$i]['pdf\_pages'] = 'pma\_\_pdf\_pages';  // $cfg['Servers'][$i]['column\_info'] = 'pma\_\_column\_info';  // $cfg['Servers'][$i]['history'] = 'pma\_\_history';  // $cfg['Servers'][$i]['table\_uiprefs'] = 'pma\_\_table\_uiprefs';  // $cfg['Servers'][$i]['tracking'] = 'pma\_\_tracking';  // $cfg['Servers'][$i]['designer\_coords'] = 'pma\_\_designer\_coords';  // $cfg['Servers'][$i]['userconfig'] = 'pma\_\_userconfig';  // $cfg['Servers'][$i]['recent'] = 'pma\_\_recent'; |

**注：文件中的所有pma跟着的是双下划线，旧版本的是单下划线，如果是单下划线需要改为双下滑线。**

## 15 安装go语言

**① 安装Go语言依赖包**

Go语言部分功能是用C语言开发的，所以安装Go之前需要安装gcc，make等依赖

**sudo apt-get install bison ed gawk gcc libc6-dev make**

**② 获取Go语言源码**

从网站<http://www.golangtc.com/download>下载golang源码

**③ 配置Go环境变量**

在编译安装之前，还需要设置Go语言的环境变量，在Ubuntu下的配置go环境变量，打开HOME目录下的.bashrc文件

**vim ~/.bashrc**

在文件末尾添加以下内容：

|  |
| --- |
| export GOROOT=$HOME/go # Go安装包的根目录  export GOARCH=386 # 选项有amd64、386和arm  export GOOS=linux # 选项有linux、 window、freebsd、darwin和 nacl  export GOPATH=/home/pi/work/golang/path:/home/pi/work/golang/project # 默认存放工程目录  export GOBIN=$GOROOT/bin # 用于存放编译生成的执行文件  export PATH=$GOROOT/bin:$PATH # 设置环境变量 |

保存设置并运行命令使其生效：

**source ~/.bashrc**

**④ 编译Go源码包**

注意：需要安装过go1.4版本后才能通过编译

**cd $GOROOT/src**

**./all.bash**

编译会执行需要一点的时间（根据CPU速度而定）

**⑤ 验证是否安装成功**

在终端输入命令：**go version**

如果显示“go version go1.4.2 linux/arm”说明安装成功

新建一个hello.go文件，复制下面内容：

|  |
| --- |
| package main  import (  "fmt"  )  func main() {  fmt.Println(“hello golang!”)  } |

编译和执行命令：**go run hello.go**

如果输出结果：Hello golang!说明go语言已经安装成功了。

## 16 搭建ftp服务器

**① 安装ftp**

**sudo apt-get install vsftpd**

**② 配置vsftpd.conf**

**sudo vim /etc/vsftpd.conf**

按需求配置如下：

|  |
| --- |
| anonymous\_enable=NO #禁止匿名访问  local\_enable=YES #接受本地用户  write\_enable=YES #允许上传  chroot\_local\_user=YES #用户只能访问限制的目录  local\_root= /home/bananapi/work/golang/project/src/miniPC #在最后一行添加用户访问的目录（注意使用前把这个文件夹的权限设为0777，否则普通用户会有权限限制, 修改权限：sudo chmod a-w \_\_\_文件夹路径）  allow\_writeable\_chroot=YES #使用户锁定在自己的目录中  pam\_service\_name=vsftpd #配置成一个服务vsftpd的名称  pasv\_promiscuous=YES #在最后一行添加以pasv模式连接(可选) |

**③ 添加ftp用户**

**sudo useradd -d /home/bananapi/work/golang/project/src/miniPC -M \_\_\_**(ftp用户名)

**sudo passwd \_\_\_**( ftp用户名)

新增ftp用户登录ftp时会出现530 login incorrect，需要修pam.d/vsftpd

**sudo vim /etc/pam.d/vsftpd**

注释掉命令：#auth required pam\_shells.so

④ 重启ftp

**sudo service vsftpd restart**

## 17 安装ffmpeg

**① 下载FFmpeg和libx264的包 ffmpeg-2.4.1.tar.bz2**

**② 安装libx264包**

安装libx264前先安装依赖包yasm

**sudo apt-get install yasm**

**sudo apt-get install libx264-dev**

**③ 然后安装ffmpeg,ffmpeg有许多依赖包**

**sudo apt-get install libfaac-dev**

**sudo apt-get install libmp3lame-dev**

**sudo apt-get install libtheora-dev**

**sudo apt-get install libvorbis-dev**

**sudo apt-get install libxvidcore-dev**

**sudo apt-get install libxext-dev**

**sudo apt-get install libxfixes-dev**

**④ 安装ffmpeg**

解压tar -xjvf ffmpeg-2.4.1.tar.bz2

**./configure --prefix=/usr/local/ffmpeg --enable-gpl --enable-version3 --enable-nonfree --enable-postproc --enable-pthreads --enable-libfaac --enable-libmp3lame --enable-libtheora --enable-libx264 --enable-libxvid --enable-x11grab --enable-libvorbis**

**make**

**make install**

|  |
| --- |
| 简单安装  sudo add-apt-repository ppa:kirillshkrogalev/ffmpeg-next  sudo apt-get update  sudo apt-get install ffmpeg  卸载 ffmpeg 命令：  sudo apt-get remove ffmpeg |