

## 一、监考终端部署

### 1、安装树莓派系统

去[官网](#)下载树莓派系统，然后使用[balenaEtcher](#)等工具将系统刷入SD卡，在boot分区创建文件名为ssh的文件,用于SSH连接。默认账号为pi,密码为raspberrypi

### 2、安装树莓派3.5英寸显示屏驱动

使用SSH工具连接树莓派或使用HDMI线将树莓派连接至显示屏，在终端中输入以下代码，安装驱动：

```
git clone https://github.com/waveshare/LCD-show.git
cd LCD-show/
sudo ./LCD35B-show
```

重启之后就可以点亮屏幕

### 3、其他配置

在终端中输入命令：`sudo raspi-config`，进入树莓派设置，开启摄像头、SSH、VNC等工具，也可以调整分辨率等设置。

### 4、给树莓派换源

此处使用的系统是debian10(buster)，其他系统换源可以借鉴[清华大学镜像站](#)

编辑 `/etc/apt/sources.list` 文件，删除原文件所有内容，用以下内容取代：

```
deb http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main non-free contrib rpi
deb-src http://mirrors.tuna.tsinghua.edu.cn/raspbian/raspbian/ buster main non-free contrib rpi
```

编辑 `/etc/apt/sources.list.d/raspi.list` 文件，删除原文件所有内容，用以下内容取代：

```
deb http://mirrors.tuna.tsinghua.edu.cn/raspberrypi/ buster main ui
```

安装后使用以下命令更新源：

```
sudo apt update
```

### 5、安装pip并更换pip源

安装pip3

```
sudo apt install python3-pip
```

更新pip并换源

```
pip3 install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U
pip3 config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
```

## 6、安装opencv依赖

```
sudo apt-get install libhdf5-dev -y
sudo apt-get install libatlas-base-dev -y
sudo apt-get install libjasper-dev -y
sudo apt-get install libqt4-test -y
sudo apt-get install libqtgui4 -y
sudo apt-get install libhdf5-serial-dev -y
sudo apt install libqtgui4 -y
sudo apt install libqt4-test -y
```

## 7、安装opencv和opencv-contrib

此处安装opencv使用的是清华源，但是安装opencv-contrib时使用的是豆瓣源，豆瓣源有编译好的opencv-contrib的轮子，其他源需要编译，花费的时间较短

```
pip3 install opencv-python
pip3 install -i http://pypi.douban.com/simple/ opencv-contrib-python
```

## 8、安装pyside2

```
sudo apt-get install python3-pyside2.qt3dcore python3-pyside2.qt3dinput python3-
pyside2.qt3dlogic python3-pyside2.qt3drender python3-pyside2.qtcharts python3-
pyside2.qtconcurrent python3-pyside2.qtc core python3-pyside2.qtgui python3-
pyside2.qthelp python3-pyside2.qtl ocation python3-pyside2.qtmultimedia python3-
pyside2.qtmultimediawidgets python3-pyside2.qtnetwork python3-pyside2.qtopengl
python3-pyside2.qtpositioning python3-pyside2.qtprintsupport python3-
pyside2.qtml python3-pyside2.qtquick python3-pyside2.qtquickwidgets python3-
pyside2.qtscrip t python3-pyside2.qtscrip ttools python3-pyside2.qtsensors python3-
pyside2.qtsql python3-pyside2.qtsvg python3-pyside2.qtt est python3-
pyside2.qtt exttospeech python3-pyside2.qtu itools python3-pyside2.qtw ebchannel
python3-pyside2.qtw ebsockets python3-pyside2.qtw idgets python3-
pyside2.qtx11extras python3-pyside2.qtxml python3-pyside2.qtxmlpatterns
sudo apt-get install shiboken2
```

## 9、安装可能用到的python3第三方库

```
pip3 install pymysql
pip3 install paramiko
pip3 install dlib
pip3 install face_recognition
pip3 install requests
```

## 10、项目部署

### (1) 放置文件

将项目文件放在 `/home/pi/Desktop` 下，项目文件夹名称为 `qt`

### (2) 设置开机启动

在 `/home/pi/.config/` 新建一个名为 `autostart` 的文件夹

```
mkdir /home/pi/.config/autostart
```

在 `autostart` 目录下新建 `boot.desktop`

```
nano /home/pi/.config/autostart/qt.desktop
```

文件内容如下

```
[Desktop Entry]
Categories=Application;Programme;
Comment=Demo
Encoding=UTF-8
Exec=python /home/pi/Desktop/qt/main.py
Name=Demo Desktop
Type=Application
```

## 二、巡考终端部署

巡考终端可以部署在 windows、Linux 和 Mac 系统上，我们暂时部署在电脑上，之后会部署在树莓派上，做成相应的便携终端。巡考端同监考端一样，使用了 pyside2、pymysql、paramiko 等 python 库，详细安装方法同监考终端部署。将巡考端代码拷贝到设备任意位置后可以直接运行

**PS:**在Windows下安装python第三方库，使用的是 pip，而不是 pip3，当然，也可以使用conda 等方式安装python第三方库

## 三、管理后台部署

我们使用的是阿里云服务器作为我们后台的服务器，服务器安装系统为CentOS8，此管理后台依然可以部署在内网中

### 1、安装MySQL

```
sudo yum install mysql-server
```

安装时设置账号和密码均为root，安装后使用

```
service mysqld start
```

来开启MySQL服务

### 2、安装python3.8和pip3

```
sudo yum install python38
sudo yum install python38-pip
```

### 3、安装一些可能用到的第三方库

```
pip3 install django==3.1.7
pip3 install pymysql==1.0.2
pip3 install django-simpleui==2021.4.11
```

### 4、安装uwsgi

```
sudo yum install python38-devel #安装依赖
pip3 install uwsgi #安装uwsgi
```

## 5、安装nginx

```
sudo yum install nginx
```

PS:运行yum安装命令时最好先运行 `sudo yum update` 来更新源

## 6、运行

### (1)、文件上传与运行

#### ①、上传位置

上传至 `/home/admin`，项目文件夹名为 `MySite`

#### ②、通过uwsgi启动

进入目录 `/home/admin/MySite/Recognition/script`，运行

```
sudo uwsgi --ini uwsgi.ini
```

#### ③、启动nginx

```
sudo nginx
```

### (2)、可能用到的命令

#### ①、调试

- `python ./manage.py makemigrations` #查找改变
  - `python ./manage.py migrate` #将改变提交到数据库
  - `python ./manage.py collectstatic` #收集静态文件
  - `python ./manage.py runserver` #运行项目
- 在Linux上可能需要将python命令替换为python3命令

#### ②、uwsgi和nginx部署

- 启动uwsgi: `uwsgi --ini uwsgi.ini`
- 启动nginx: `nginx`
- 停止uwsgi: `uwsgi --stop uwsgi.pid`
- 停止uwsgi或nginx(使用Linux命令): `ps -ef | grep uwsgi/nginx`

然后使用: `kill -9 进程id` 结束进程

## 7、相关配置文件

### (1)、阿里云服务器中uwsgi配置文件位置:

```
/home/admin/MySite/Recognition/script
```

```
[root@iZuf65i2eouaa5z1ofoxe0Z script]# ls -la
total 156
drwxr-xr-x 2 root root    57 Apr  2 20:42 .
drwxr-xr-x 6 root root    85 Apr  1 18:19 ..
-rw-r--r-- 1 root root  705 Apr  2 20:42 uwsgi.ini
-rw-r----- 1 root root 148148 Apr  2 20:56 uwsgi.log
-rw-r--r-- 1 root root    5 Apr  2 20:47 uwsgi.pid
```

如上图，含有三个文件，从上到下分别为uwsgi的配置文件、日志和进程文件

uwsgi.ini 文件内容如下：

```
[uwsgi]
#项目目录
chdir=/home/admin/MySite/Recognition
#启动uwsgi的用户名和用户组
uid=root
pid=root
#配置wsgi接口模块文件路径,也就是wsgi.py这个文件所在的目录名
wsgi-file=Recognition/wsgi.py
#静态文件路径
static-map=/static=/home/admin/MySite/Recognition/
#配置和nginx连接的socket连接
socket=127.0.0.1:8000
#启动主进程
master=true
#进程个数
worker=4
pidfile=/home/admin/MySite/Recognition/script/uwsgi.pid
#自动移除unix socket和pid文件当服务停止时
vacuum=true
#序列化接受内容，如果可能的话
thunder-lock=true
#启用线程
enable-threads=true
#设置自中断时间
harakiri=30
#设置缓冲
post-buffering=4096
#设置日志目录
daemonize=/home/admin/MySite/Recognition/script/uwsgi.log
```

## (2)、项目静态文件位置:

```
/home/admin/MySite/Recognition/static
```

```
[admin@iZuf65i2eouaa5z1ofoxe0Z static]$ ls -la
total 0
drwxrwx--- 5 admin admin 45 Apr  3 16:01 .
drwxrwx--- 7 admin admin 98 Apr  3 16:01 ..
drwxrwx--- 7 admin admin 69 Apr  3 16:01 admin
drwxrwx--- 4 admin admin 29 Apr  3 16:01 index
drwxrwx--- 3 admin admin 20 Apr  3 16:01 media
```

[https://blog.csdn.net/qq\\_29183811](https://blog.csdn.net/qq_29183811)

如上图，从上到下分别为管理后台、前端的静态文件，以及用户上传的文件（即上传的人脸和证件照片）

### (3)、nginx配置文件位置:

```
/etc/nginx/nginx.conf
```

nginx文件内容如下:

```
# For more information on configuration, see:
# * Official English Documentation: http://nginx.org/en/docs/
# * Official Russian Documentation: http://nginx.org/ru/docs/

user root;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections 1024;
}

http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    client_max_body_size 100m;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
```

```

# See http://nginx.org/en/docs/nginx_core_module.html#include
# for more information.
include /etc/nginx/conf.d/*.conf;

server {
    server_name xjfyf.top; # managed by Certbot
    # Load configuration files for the default server block.
    include /etc/nginx/default.d/*.conf;

    location / {
        include uwsgi_params;
        uwsgi_connect_timeout 30;
        uwsgi_pass 127.0.0.1:8000;
        uwsgi_param UWSGI_SCRIPT Recognition.wsgi; #项目wsgi.py目录
        uwsgi_param UWSGI_CHDIR /home/admin/MySite/Recognition; #项目目录
    }

    location /static {
        alias /home/admin/MySite/Recognition/static;
    }

    error_page 404 /404.html;
        location = /40x.html {
    }

    error_page 500 502 503 504 /50x.html;
        location = /50x.html {
    }

    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/xjfyf.top/fullchain.pem; # managed by
Certbot
    ssl_certificate_key /etc/letsencrypt/live/xjfyf.top/privkey.pem; # managed
by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}

server {
    if ($host = xjfyf.top) {
        return 301 https://$host$request_uri;
    } # managed by Certbot
    listen 80 ;
    listen [::]:80 ;
    server_name xjfyf.top;

    return 404; # managed by Certbot
}
}

```

(4)、nginx日志文件位置: `/var/log/nginx`

通过nginx相关日志文件可以查看nginx运行情况