# Linux手机管家说明文档

目录

[Linux手机管家说明文档 1](#_Toc441677317)

[1.开发环境 2](#_Toc441677318)

[2.运行环境 2](#_Toc441677319)

[3.软件架构 2](#_Toc441677320)

[4.软件结构 2](#_Toc441677321)

[4.1多媒体管理 3](#_Toc441677322)

[4.2短信管理 3](#_Toc441677323)

[4.3通讯录管理 4](#_Toc441677324)

[4.4基本信息与应用管理 5](#_Toc441677325)

[5.核心函数说明 6](#_Toc441677326)

[6.软件安装 9](#_Toc441677327)

[6.1源码编译 9](#_Toc441677328)

[6.2 Deb包 11](#_Toc441677329)

[7.软件使用 11](#_Toc441677330)

[7.1运行软件 11](#_Toc441677331)

[7.2连接手机 12](#_Toc441677332)

[7.3导航栏 13](#_Toc441677333)

[7.4多媒体管理 14](#_Toc441677334)

[7.4.1我的音乐 14](#_Toc441677335)

[7.4.2我的图片 16](#_Toc441677336)

[7.4.3我的视频 18](#_Toc441677337)

[7.5我的短信 19](#_Toc441677338)

[7.6通讯录管理 20](#_Toc441677339)

[7.7应用管理 24](#_Toc441677340)

[7.7.1安装 24](#_Toc441677341)

[7.7.2卸载 25](#_Toc441677342)

[8.小组分工 26](#_Toc441677343)

# 1.开发环境

ADT,SHELL,GEDIT

# 2.运行环境

系统：Ubuntu14.04及以上版本

依赖：ADB，Java1.7及以上版本，Apache2，Python3，Netrw

# 3.软件架构

前台用HTML5\CSS3通过bootstrap框架实现显示，

后台主要使用CGI通过Apache2服务器进行与前台的交互。

结构图大致如下：



图1 软件架构图

# 4.软件结构

接下来将按模块来展示软件结构，所有的数据都是通过CGI发送请求给App，然后App通过Socket进行通信，按着不同的请求回复所需数据。而CGI与前台网页的通信主要是通过QUERY\_STRING这个环境变量来实现。

## 4.1多媒体管理

多媒体管理主要包括图片、音乐、视频管理三部分。结构如图2所示。

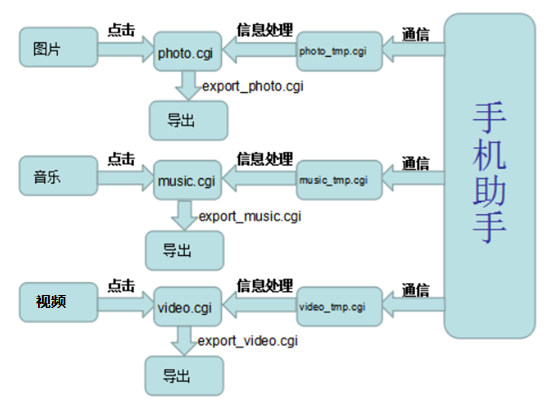


图2多媒体管理结构图

## 4.2短信管理

短信管理主要是以列表的方式显示短信的详细信息，结构如图3所示。

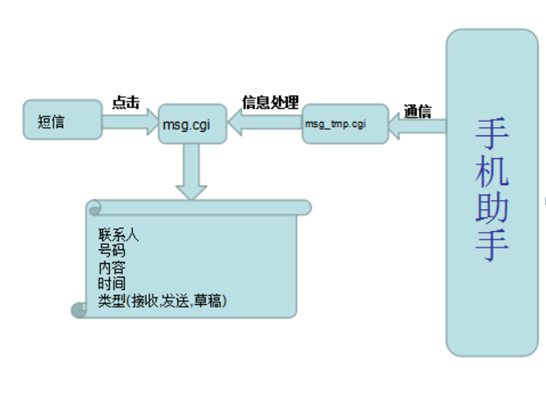


图3短信管理

## 4.3通讯录管理

通讯录管理包括对于联系人的增删改以及对整个通讯录的备份与还原，相当实用的功能。具体结构如图4所示。

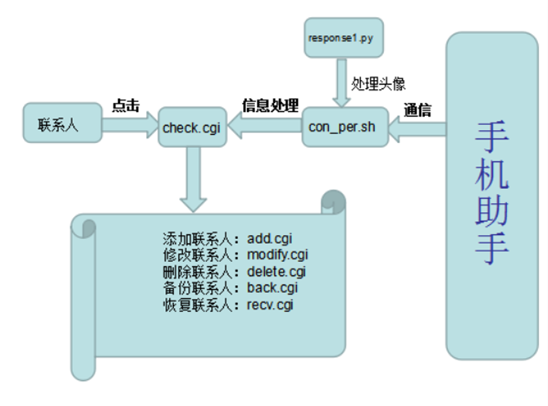


图4通讯录管理

## 4.4基本信息与应用管理

结构如图5所示，由于该模块的实际工作量巨大，涉及脚本众多，所以接下来将一一说明。

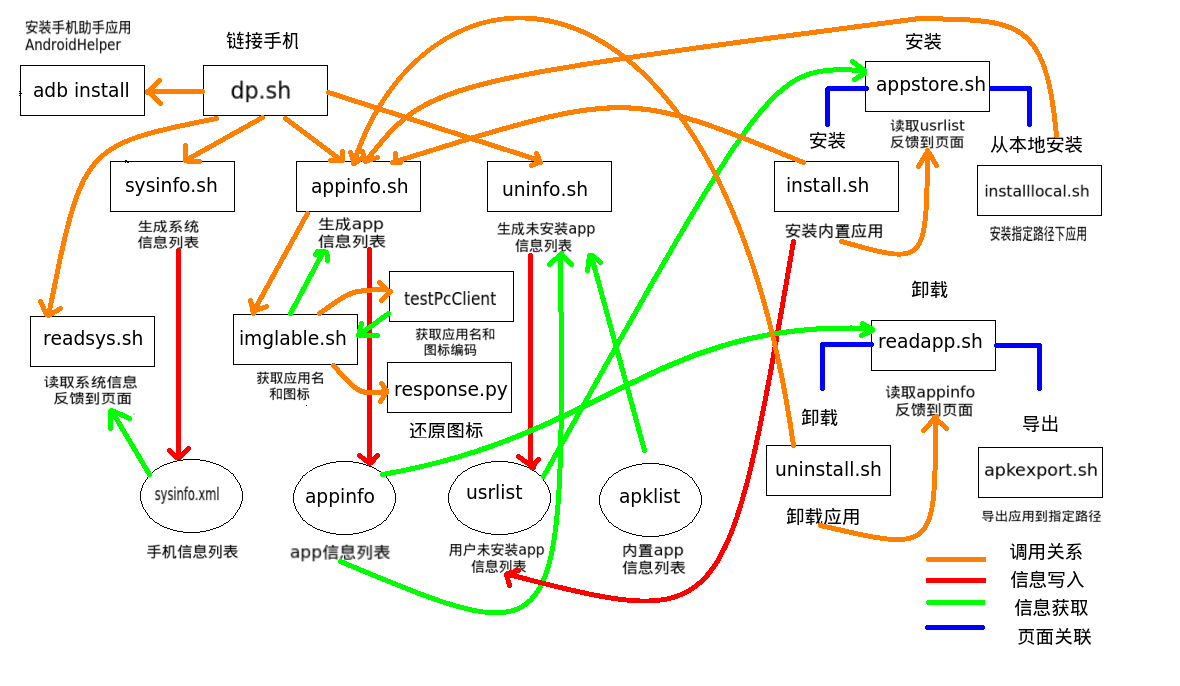


图5基本信息与应用管理结构图

该功能模块作为后台cgi，由页面上的点击事件，或者表单提交触发执行。主要涉及的页面按钮是左侧导航栏的三个按钮：“连接手机”，“安装”，“卸载”。三个cgi都会涉及到adb的使用，“安装”，“卸载”将会直接使用adb。

以下为三个按钮调用的cgi的运行结构的分析。

1.连接手机按钮

“连接手机”直接调用shell脚本dp.sh。

第一步：

在点击“连接手机”时向手机安装辅助APP——AndroidHelper。AndroidHelper回在手机上运行，根据PC端发送的信号返回指定信息。

第二步：

调用sysinfo.sh，用于生成手机系统信息列表sysinfo.xml文件。

第三步：

调用readsys.sh，将静态页面返回到界面，并且读取手机系统信息列表sysinfo.xml文件，将信息填充到html语句的合适位置，将页面的动态部分的返回到界面。

第四步：

调用appinfo.sh，用于生成已安装APP信息列表。该脚本将调用另外的脚本用于信息的生成。首先调用imglabel.sh获取应用图标的路径、应用名和包名，imglabel.sh使用java程序testPcClient从辅助APP——AndroidHelper获取应用图标的ASCII编码、应用名和包名。获取的应用图标ASCII编码后，imglabel.sh会调用Python脚本——response.py，根据图标的ASCII编码将突破还原并保存到指定路径下。最终imglabel.sh会将图标路径、应用名和包名返回给appinfo.sh脚本。appinfo.sh脚本最终将每个应用的信息以：“图标路径：应用名：安装位置：大小：包名”记录到appinfo文件中。

第五步：

调用uninfo.sh，根据内置应用的apklist列表和appinfo列表生成usrlist列表文件，保存未安装的内置应用信息。

2.安装按钮

直接调用appstore.sh，读取usrlist列表文件，填充页面信息反馈到界面上。此时界面上有连个按钮“安装”和“从本地安装”。点击内置应用的“安装”按钮将调用install.sh，将相应应用安装到手机上，然后更新usrlist列表文件，删除刚安装的应用信息，再次调用appstore.sh刷新页面，最后调用appinfo.sh刷新已安装应用列表；点击“从本地安装”按钮将会弹出文件选择对话框，找到本地的apk文件，点击“打开“然后文件路径就会传递给installlocal.sh,进行安装，最后调用appinfo.sh刷新已安装应用列表。

3.卸载按钮

“卸载“直接调用readapp.sh，读取appinfo列表文件，将信息填充到页面中反馈界面。此时界面上有”卸载“和”导出“两个按钮。通过复选框选中应用后，点击”卸载“，应用信息将会传递给uninstall.sh，执行卸载操作，然后执行appinfo.sh更新已安装应用列表appinfo，再次调用readapp.sh刷新页面；点击“导出”，会弹出路径选择对话框，选择路径点击“打开”应用信息将会传递给apkexport.sh，执行导出操作。

# 5.核心函数说明

#获取手机型号

model=`$adbpath/adb shell getprop ro.product.model|sed 's/\r$//'`

#获取厂商

brand=`$adbpath/adb shell getprop ro.product.brand|sed 's/\r$//'`

#获取系统版本号

android=`$adbpath/adb shell getprop ro.build.version.release|sed 's/\r$//'`

#获取屏幕截图

img="../image/1.png"

rm -rf $img

$adbpath/adb shell screencap -p|sed 's/\r$//'>$img

#获取总ROM

romtotal=`$adbpath/adb shell cat /proc/meminfo|grep "MemTotal"|awk '{print $2}'`

romtotal=`expr $romtotal / 1024`

#获取可使用ROM

romfree=`$adbpath/adb shell cat /proc/meminfo|grep "MemFree"|awk '{print $2}'`

romfree=`expr $romfree / 1024`

#获取总sdcard

sdtotal=`$adbpath/adb shell df sdcard|awk '{if(NR==2) print $2}'`

sdtotal=${sdtotal%G\*}

#获取可使用sdcard

sdfree=`$adbpath/adb shell df sdcard|awk '{if(NR==2) print $4}'`

sdfree=${sdfree%G\*}

#获取电量

battery=`$adbpath/adb shell dumpsys battery|grep "level"|grep -o [0-9]\*`

#readsys.sh 通过value函数在静态的html语句（$1）中寻找标签（$2）和相应的值（$3）,并返回到界面，flag置为1；如果没有匹配到将flag将保持0，在调用者中检测返回相应页面到界面

function value ()

{

ret=`echo $1|grep "$2"`

if [ -n "$ret" ]

then

if [ "$2" = "model" -o "$2" = "brand" -o "$2" = "android" ]

then

echo $ret|sed "s/></>$3</g"

fi

if [ "$2" = "shotscreen" ]

then

echo $ret|sed "s&src=\"\"&src=\"$3\"&"

fi

if [ "$2" = "battery" -o "$2" = "rom" -o "$2" = "sd" ]

then

per=$(echo "$4\*100/$3"|bc)

echo $ret|sed "s&value=\"1\"&value=\"$per\"&"|sed "s&>\([%kMG]\*\)</td>&>$4\/$3\1</td>&"

fi

flag=1

fi

}

……

cat ../html/home.html|while read line

do

flag=0

value "$line" "shotscreen" "${dict[0]}"

value "$line" "model" "${dict[1]}"

value "$line" "brand" "${dict[2]}"

value "$line" "android" "${dict[3]}"

value "$line" "battery" "100" "${dict[4]}"

value "$line" "rom" "${dict[5]}" $(echo "${dict[5]}-${dict[6]}"|bc)

value "$line" "sd" "${dict[7]}" $(echo "${dict[7]}-${dict[8]}"|bc|awk '{printf "%.1f",$0}')

if [ $flag -eq 0 ]

then

echo $line

fi

done

#appinfo.sh主要进行的是字符串的裁剪和拼接，比较重要的response.py脚本中将图标的ASCII编码还原成图片的过程

#获取!/usr/bin/python3

import base64

import sys

encode = sys.argv[2]

data = base64.b64decode(encode)

w = open("../image/"+sys.argv[1]+".png","wb")

w.write(data)

w.close()

应用名和图标的ASCII编码将作为命令行参数传递给该脚本，通过base64模块可以方便的将图标的ASCII编码转换为二进制，并写入文件，生成图片

#安装和卸载导出的具体实现脚本涉及到与界面的信息交互问题，我们通过环境变量QUERY\_STRING将页面信息传递给脚本。以最复杂的导出apk包到指定路径的脚本apkexport.sh为例：

printf -v QUERY\_STRING $(echo -n $QUERY\_STRING|sed 's/\\/\\\\/g;s/\(%\)\([0-9a-fA-F][0-9a-fA-F]\)/\\x\2/g')

exapk=${QUERY\_STRING#获取#获取\*;}

expak=${exapk#获取\*=}

apkselect=${QUERY\_STRING%;\*}

for i in `echo $apkselect|sed 's/%20/\t/g'`

do

raw=`cat ../data/appinfo|grep "$i"`

apkpath=`echo $raw|awk -F ":" '{print $3}'`

sudo $adbpath/adb pull $apkpath $exapk>/dev/null

done

由于语言环境的改变首先需要将QUERY\_STRING中的一些特殊的html字符转换成shell可以使用的字符，然后将apk安装路径和导出路径进行分离，将apk一一导出到指定路径

# 6.软件安装

提供了两种安装方式：源码编译、Deb包。

## 6.1源码编译

将源码包（图6）放在解压到任意目录，如图7所示，进入该目录，运行install.sh脚本，如图8所示，根据提示输入。

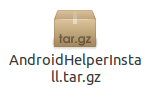


图6源码包图

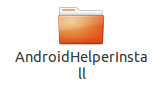


图7源码包解压图

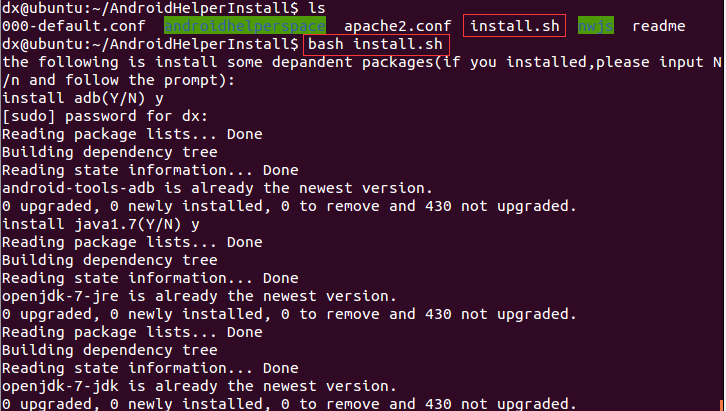


图8运行安装图

注意，Apache2最好进行安装，否则需要改很多配置（具体参照readme文件），如图9所示。完成后软件在家目录，cd进入家目录，如图10所示，运行androidhelper即可使用，如图11所示。

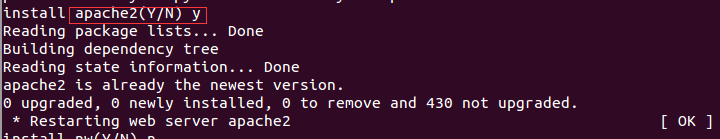


图9安装Apache2图

H:\dx\QQ截图20160127164050.png

图10运行图

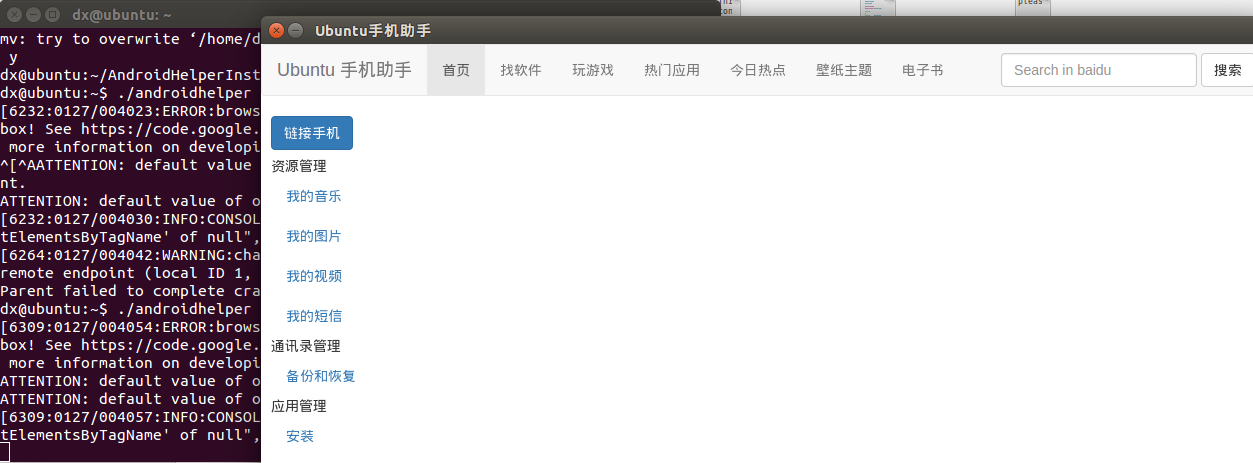


图11运行软件图

## 6.2 Deb包

Deb包（图12）的安装更为简单。

使用命令或双击运行Deb包（图13），注意图14的红色标注，若添加权限则无法正确运行。

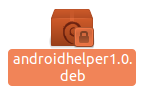


图12Deb包图

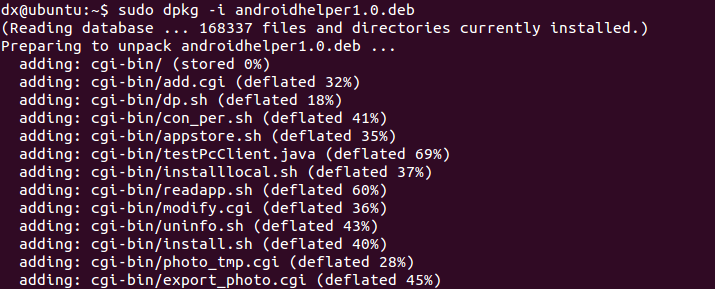


图13安装Deb包图

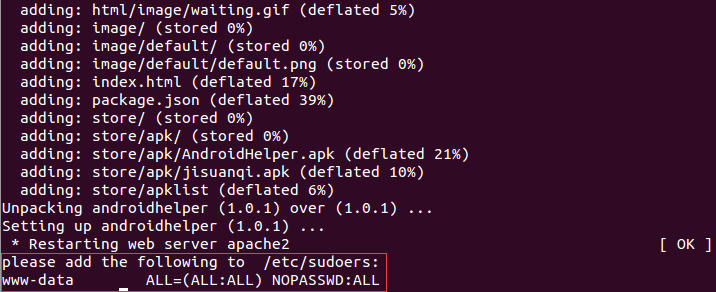


图14添加权限图

# 7.软件使用

## 7.1运行软件

双击Ubuntu\_phone即可运行软件，如图15所示。如果手机端出现授权提示，请选择确定选项。



图15软件运行首页图

## 7.2连接手机

点击主界面连接手机按钮即可连接手机，约1分钟后，进度条消失后，将显示手机基本信息，如图16、17所示。

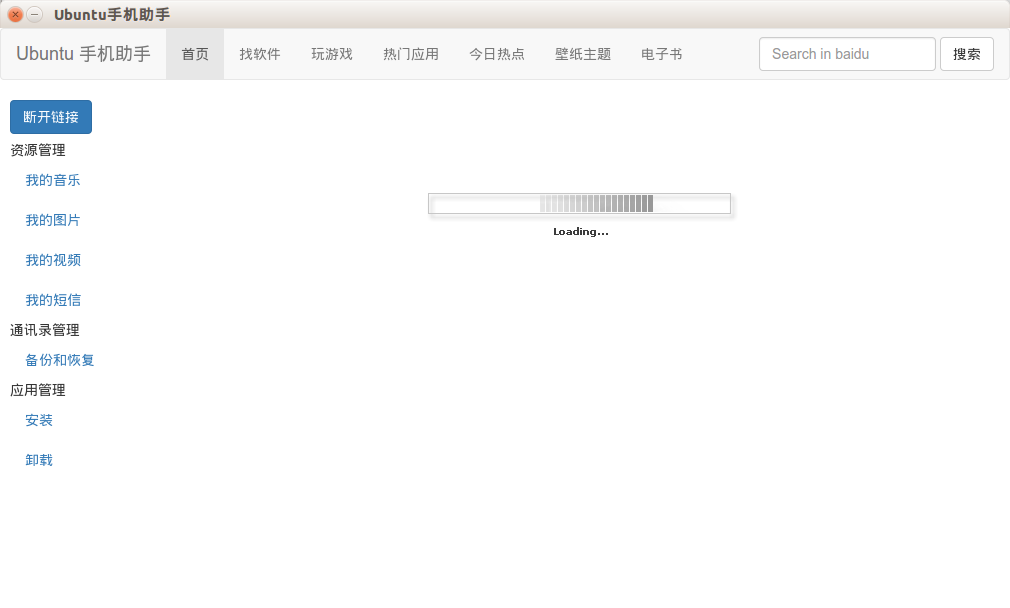


图16连接手机图



图17手机信息图

如果长时间没有连接成功，请打开终端输入：adb devices查看手机是否拥有权限

出现ccb78048 device 字样即为成功连接手机，其中ccb78048是手机识别码。

只出现List of devices attached字样，请检查手机是否插好，是否已经打开USB调试。

出现no permission字样，请打开USB调试，并运行link.sh脚本。

## 7.3导航栏

点击导航栏的‘首页’即可跳转到手机基本信息界面。

点击搜索，软件会跳转到百度的相应网页并显示搜索结果，如图18所示。

软件导航栏的其它功能尚未完善，敬请期待！



图18 搜索图

## 7.4多媒体管理

### 7.4.1我的音乐

点击左侧导航栏‘我的音乐’，即可跳转到我的音乐界面。如图19所示。

在我的音乐界面，可以使用查看和导出功能。你可以选择想要导出的音乐后，然后点击导出按钮，在弹出的路径选择框中选择导出路径，进行导出。也可以点击全选按钮后再点击导出按钮，导出全部音乐。如图20-22所示。

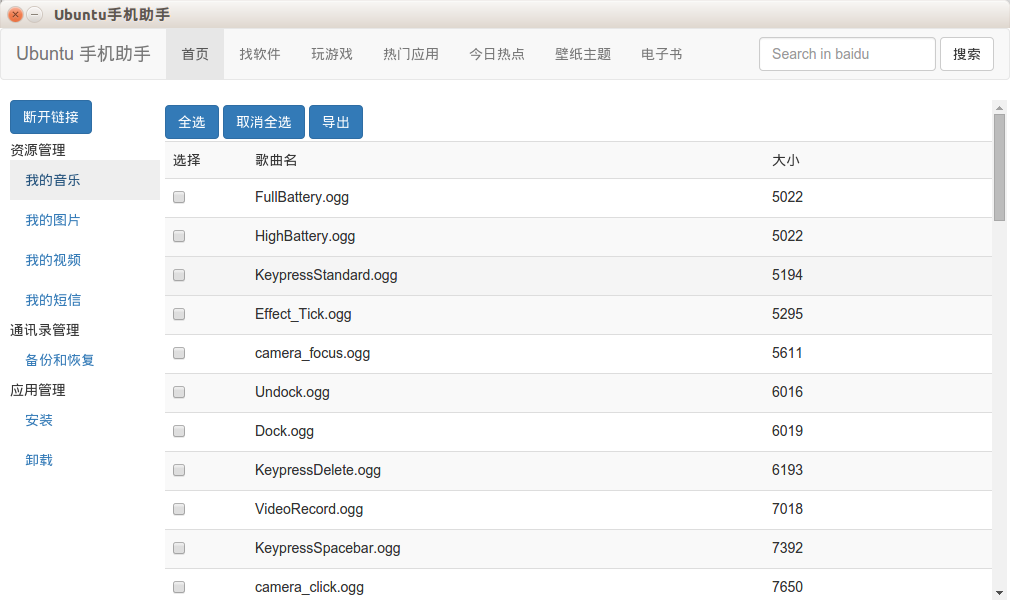


图19我的音乐图

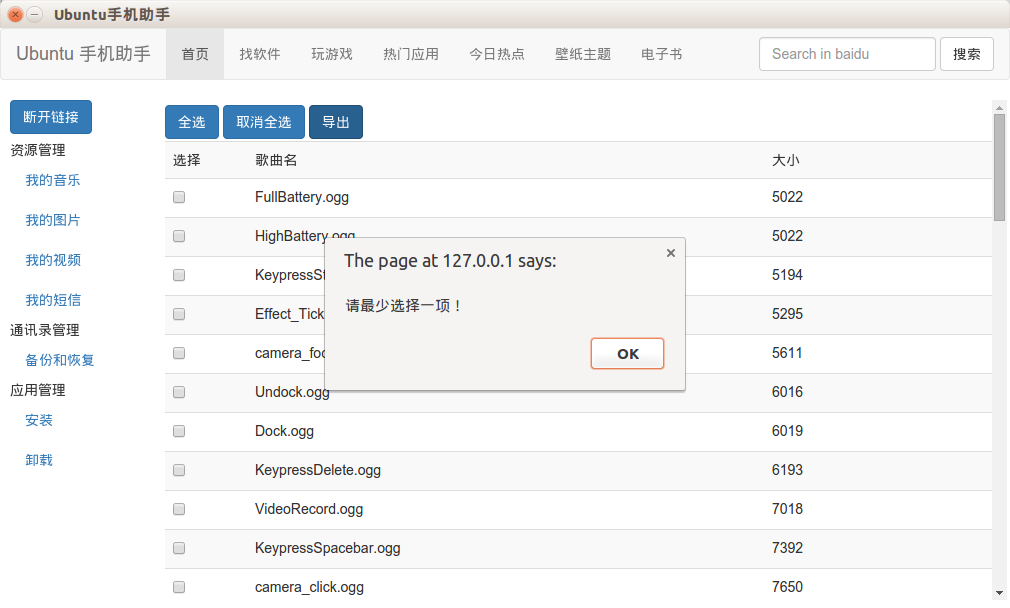


图20选择音乐图

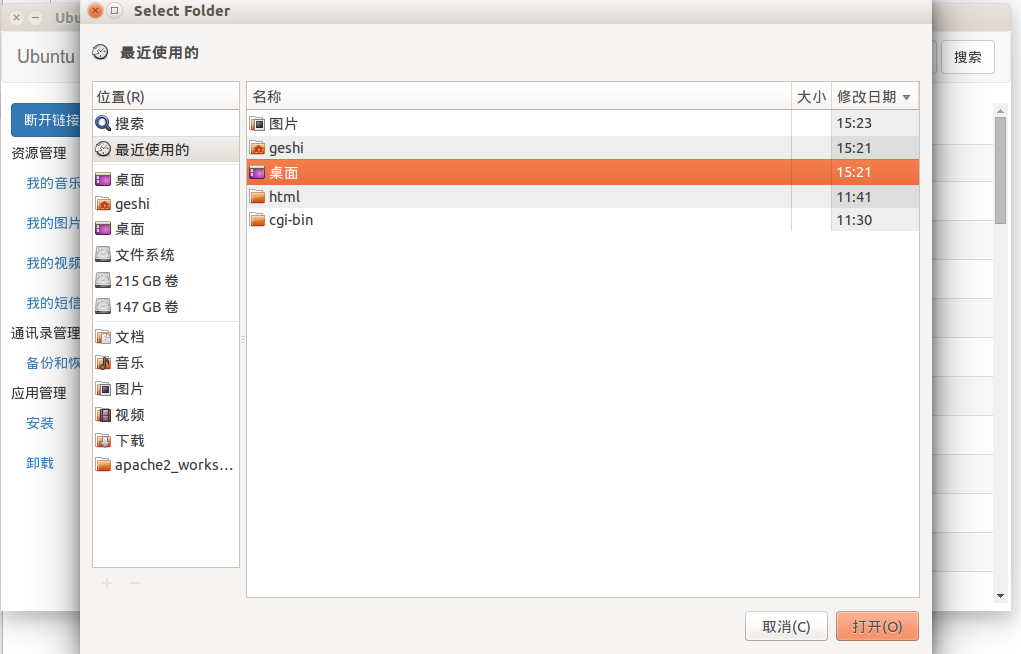


图21导出音乐图

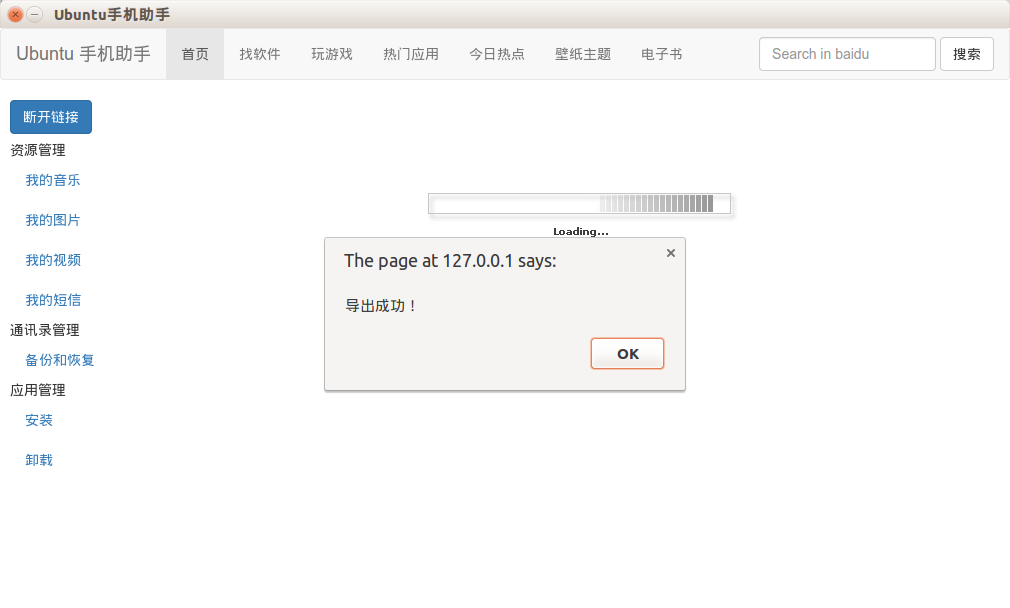


图22导出音乐成功图

### 7.4.2我的图片

点击左侧导航栏‘我的图片’，即可跳转到我的图片界面。如图23所示。

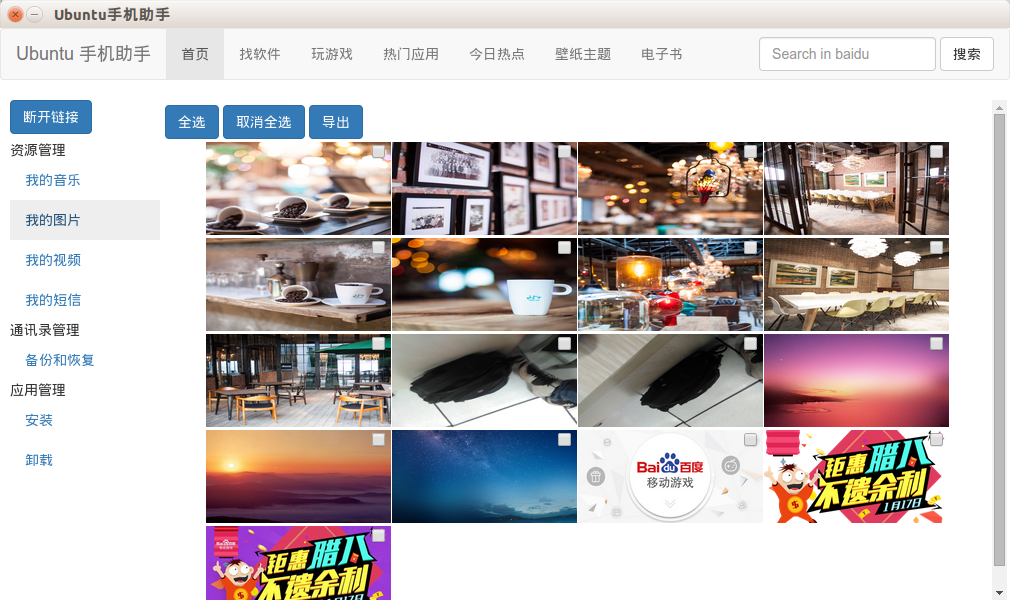


图23我的图片图

在我的图片界面，可以使用查看和导出功能。

使用导出功能。你可以选择想要导出的图片后，然后点击导出按钮，在弹出的路径选择框中选择导出路径，如图24所示，进行导出。也可以点击全选按钮后再点击导出按钮，导出全部图片。成功如图25。

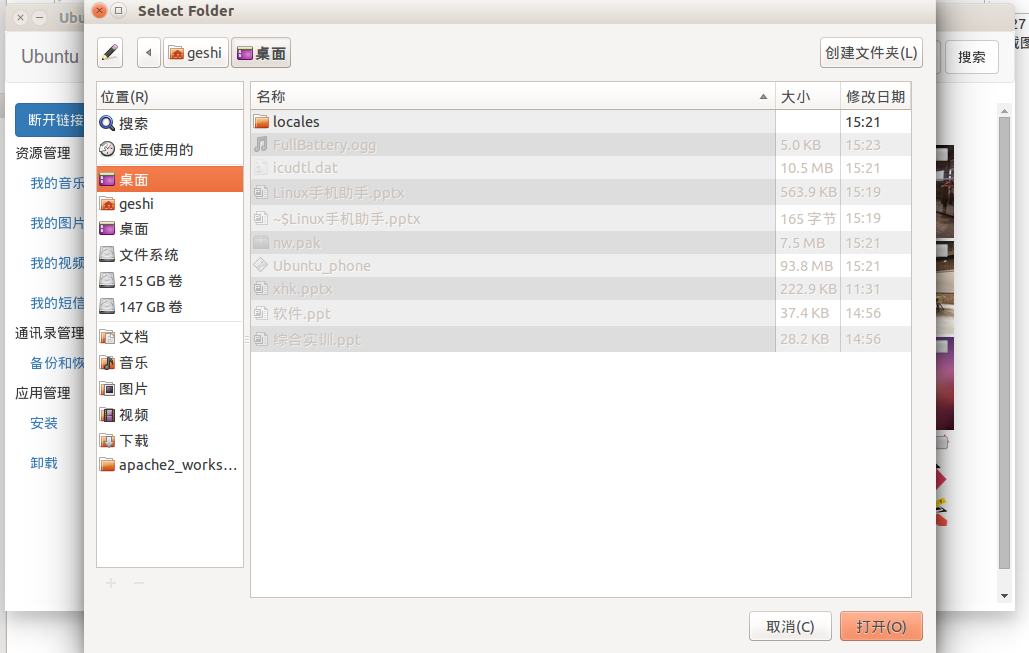


图24选择图片保存路径图

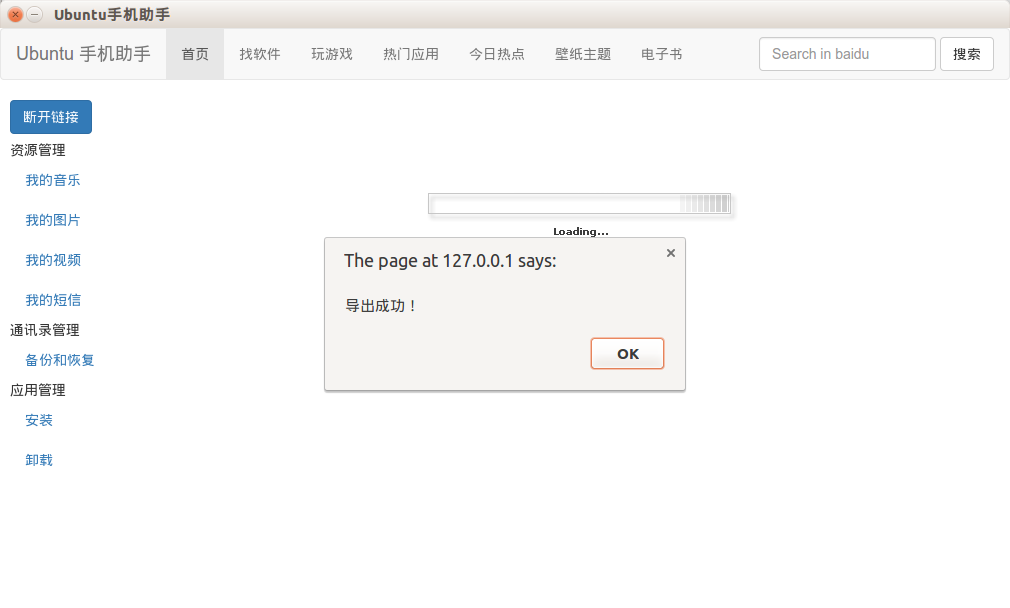


图25导出图片成功图

使用查看功能。你可以直接查看页面显示的缩略图，也可以点击缩略图，弹出原图进行查看。如图26所示。

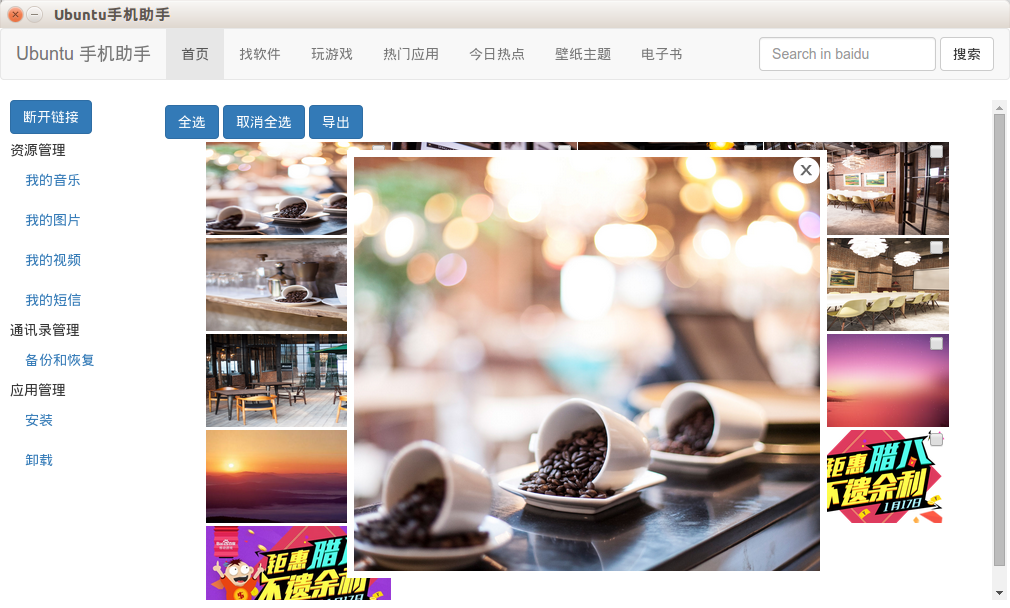


图26查看图片图

### 7.4.3我的视频

点击左侧导航栏‘我的视频’，即可跳转到我的视频界面，如图27所示。



图27我的视频

在我的视频界面，可以使用查看和导出功能。你可以选择想要导出的视频后，然后点击导出按钮，在弹出的路径选择框中选择导出路径，进行导出。也可以点击全选按钮后再点击导出按钮，导出全部视频。

## 7.5我的短信

点击左侧导航栏‘我的短信’，即可跳转到我的短信界面，如图28所示。



图28我的短息图

在我的视频界面，可以使用查看功能。在短信界面第一列为联系人，如果该联系人在你的通讯录中，该列将显示他的名字，否则显示他的手机号码。该页面的最后一列则是消息类型，你可以查看该消息是发送的，还是接收的。

## 7.6通讯录管理

点击左侧导航栏‘备份和恢复’，即可跳转到我的通讯录管理界面，如图29所示。



图29通讯录图

在通讯录管理界面你可以使用备份，恢复，删除，添加，修改联系人功能。

使用备份功能。你可以点击备份按钮，手机助手会帮你备份所有联系人。

使用恢复功能。你可以点击恢复按钮，手机助手会自动识别手机中已有的联系人，对于已有的联系人不会进行恢复，以免造成重复。不重复的联系人这回恢复到你的手机中。

使用修改联系人功能。你可以点击每一个联系人后面的修改按钮，按照提示输入联系人信息。点击确认后即可修改联系人。如图30-33所示。



图30 修改联系人图



图31 输入姓名图



图32修改电话图

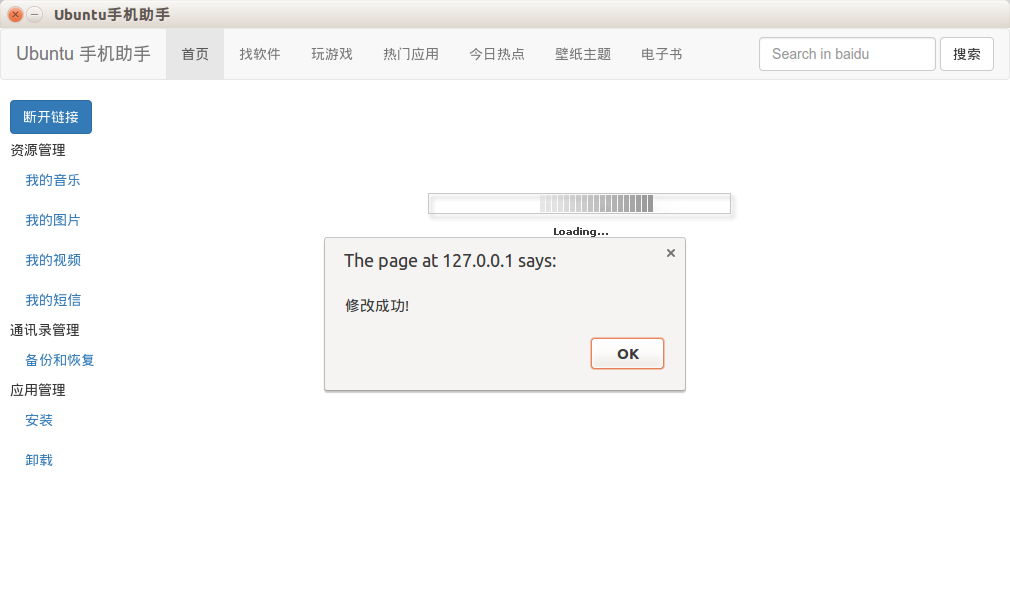


图33联系人修改成功图

使用删除联系人功能。你可以选择你想要删除的联系人，然后点击删除按钮，即可删除，如图34所示。也可以点击全选按钮，全部选择，这将删除所有联系人。成功如图35所示。



图34待选择联系人图



图35删除联系人成功图

使用添加联系人功能。你可以点击添加联系人按钮，如图36所示。按照提示输入联系人信息。点击确认后即可添加联系人，如图37所示。



图36添加联系人图

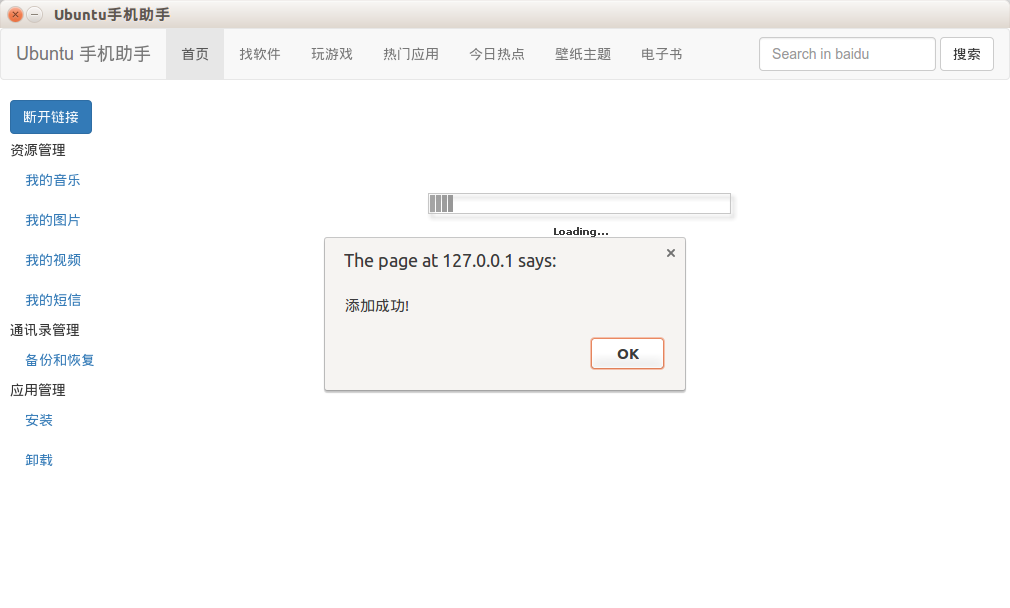


图37添加联系人成功图

## 7.7应用管理

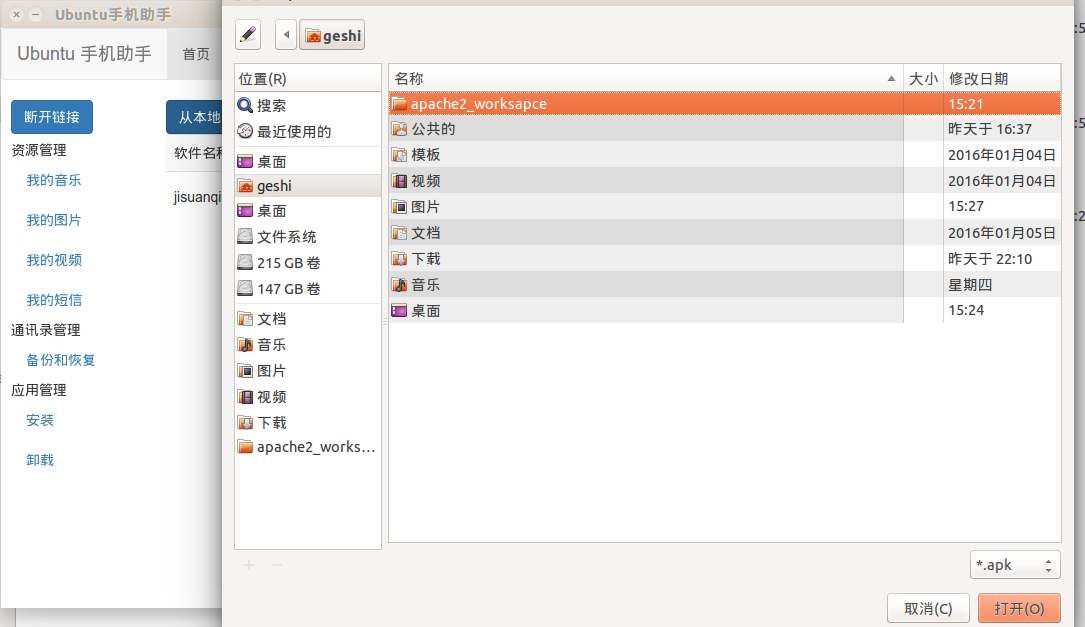
### 7.7.1安装

点击左侧导航栏‘安装’，即可跳转到安装apk界面。在这个界面中，你可以看见系统推荐安装的apk，也可以使用从本地安装功能。

使用推荐安装功能。只需要在需要安装的apk后点击安装按钮即可，如图38所示。

图38推荐安装图

使用从本地安装功能。点击从本地安装按钮，在弹出的对话框中选择需要安装的apk的路径，点击确定即可安装，如图39所示。

图39本地安装图

### 7.7.2卸载

点击左侧导航栏‘卸载’，即可跳转到卸载界面。在这个界面中，你可以看见手机上所有的apk。你可以使用卸载功能，也可以使用导出功能。卸载和导出按钮均实现了浮动，方便使用如图40、41所示。

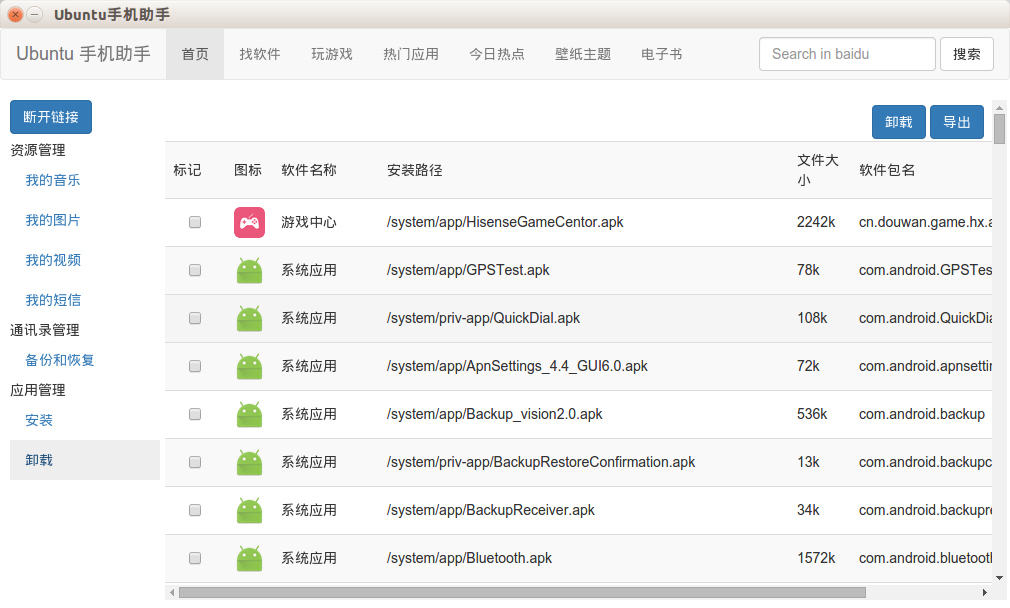


图40卸载图

使用卸载功能。你需要选择想要卸载的apk文件，点击卸载即可。

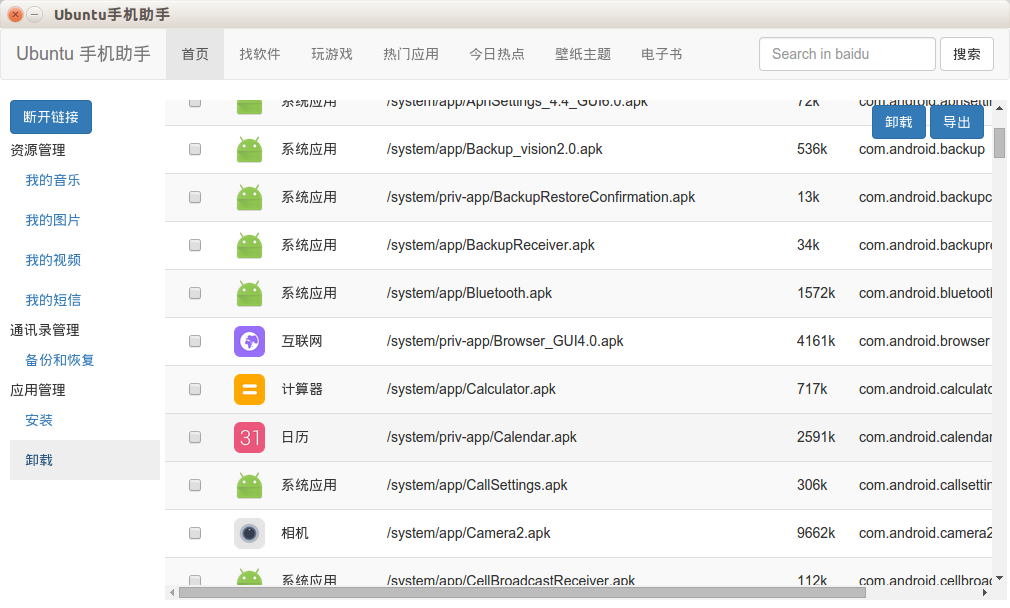


图41按钮浮动图

使用导出功能。导出功能将导出对于的apk的安装文件。在使用的时候只需要选择需要导出的apk后点击导出按钮，在弹出的对话框中选择导出路径，点击确定即可。

# 8.小组分工

|  |  |  |
| --- | --- | --- |
|  | 编码 | 文档 |
| 戴翔 | App端，软件打包 | 文档整合 |
| 葛石 | 软件UI | 对应的程序说明文档 |
| 邢浩堃 | 软件基本信息与应用管理模块 | 对应的程序说明文档 |
| 史东海 | 多媒体、短信、通讯录管理模块 | 对应的程序说明文档 |