**Android综合应用程序开发**

**应用名称**|二维码扫码登录功能实现

**开发日期**|2017-05-28 ~ 2017-06-03

**开发人员**|王婷 阮博男 侯玉玺

**一 基本说明**

**架构** 本次开发采用C/S架构。网页扫码模拟基本B/S架构

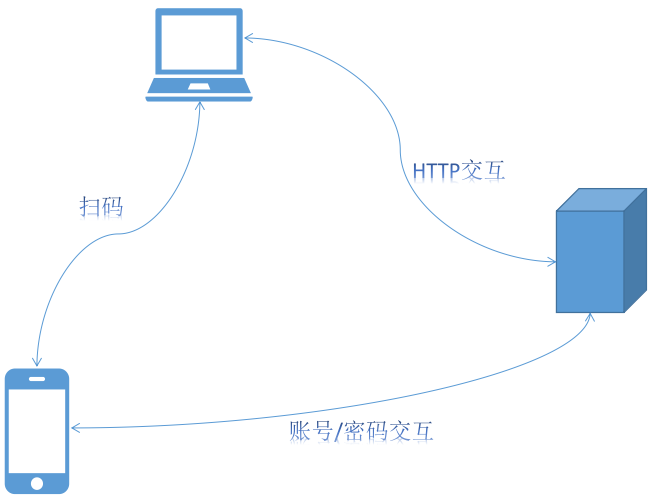
**C端** 运行于Android4.0以上设备

**C端依赖** ZXing

**S端** 运行于RedHat Enterprise Linux 7 64位系统

**S端依赖** Distrib-5.5.41-MariaDB Python-qrcode-5.3

**交互流程**



**二 协议说明**

**阅读说明**

设备：指Android设备

设备服务器：指与Android设备交互，完成登陆、注册、扫码、注销功能的服务程序

客户浏览器：指与网页服务器通过HTTP协议交互的程序

网页服务器：指与PC机浏览器进行HTTP交互，完成二维码生成、发送和扫码登录成功后的反馈、注销功能的服务程序

IP地址：指IPv4协议的地址字符串

进制：表格中所有数据均为十六进制

数据格式：报文传输时所有4字节均为网络序，报文解释时已转为主机序

报文头：每个报文的前8字节是报文头，最短的数据包只包含报文头

填充：对于报文中的定长字节数组，如果实际有意义长度不足该固定长度，则需用0x00补全至该固定长度

**设备连接**

功能：设备向设备服务器的指定端口发起TCP连接

方向：设备 => 设备服务器

**交互协议**

**1 设备向设备服务器发送各种认证请求**

**① 设备向设备服务器发送登录请求**

**·**功能：设备TCP连接成功后，设备向设备服务器发送登录请求

·方向：设备 => 设备服务器

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x91 | 0x01 | 总长度 | 数据长度 |
| 用户名（16字节） | | | |
|  | | | |
|  | | | |
|  | | | |
| 密码MD5（32字节） | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |

**② 设备向设备服务器发送注册请求**

**·**功能：设备TCP连接成功后，设备向设备服务器发送注册请求

·方向：设备 => 设备服务器

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x91 | 0x02 | 总长度 | 数据长度 |
| 用户名（16字节） | | | |
|  | | | |
|  | | | |
|  | | | |
| 密码MD5（32字节） | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |
|  | | | |

**2 设备服务器向设备发送各种认证应答**

**① 设备服务器向设备发送登录应答**

**·**功能：设备服务器收到设备的登录请求后，向设备发送的登录状态信息

·方向：设备服务器 => 设备

·注：如果用户已经在别处登录，则登录结果项为 0x00000002；如果用户名或密码错误，则登录结果项为 -0x00000001；如果登录成功，则返回设备服务器针对当前socket连接的文件描述符

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x11 | 0x01 | 总长度 | 数据长度 |
| 登录结果（4字节） | | | |

**② 设备服务器向设备发送注册应答**

**·**功能：设备服务器收到设备的注册请求后，向设备发送的注册状态信息

·方向：设备服务器 => 设备

·注：如果用户已经在别处登录，则注册结果项为 0x00000002；如果用户名已存在，则注册结果项为 0x00000001；如果注册成功，则返回设备服务器针对当前socket连接的文件描述符，并标记当前连接登录成功

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x11 | 0x02 | 总长度 | 数据长度 |
| 注册结果（4字节） | | | |

**3 设备向设备服务器发送扫码登录请求**

**·**功能：设备登录成功后，设备向设备服务器发送扫码登录请求

·方向：设备 => 设备服务器

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x91 | 0x04 | 总长度 | 数据长度 |
| 客户浏览器IP地址（16字节） | | | |
|  | | | |
|  | | | |
|  | | | |

**4 设备服务器向设备发送扫码登录应答**

**·**功能：设备服务器收到设备的扫码登录请求后，向设备发送的扫码登录状态信息

**·**方向：设备服务器 => 设备

·注：若当前设备尚未登录或当前设备已登录但扫码登录失败，则扫码登录结果项为 -0x00000001；若扫码登录成功，则扫码登录结果项为 0x00000000

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x11 | 0x04 | 总长度 | 数据长度 |
| 扫码登录结果（4字节） | | | |

**5 设备向设备服务器发送注销请求**

**·**功能：设备登录成功后，设备向设备服务器发送注销请求

·方向：设备 => 设备服务器

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x91 | 0x08 | 总长度 | 数据长度 |

**6 设备服务器向设备发送注销应答**

**·**功能：设备服务器收到设备的注销请求后，向设备发送的注销状态信息

**·**方向：设备服务器 => 设备

·注：注销结果总为0x00000000，且发送应答后设备服务器断开与当前设备的连接

0 8 16 24 32

|  |  |  |  |
| --- | --- | --- | --- |
| 0x11 | 0x08 | 总长度 | 数据长度 |
| 注销结果（4字节） | | | |

**Append 1 客户浏览器向网页服务器发送Web根路径的GET请求**

**Append 2 网页服务器向客户浏览器发送响应**

**① 若尚未成功进行扫码登录，则返回包含二维码的未登录界面**

**② 若已成功进行扫码登录，则返回包含用户信息的已登录界面**

**Append 3 客户浏览器向网页服务器发送”/logout”的GET请求**

**Append 4 网页服务器向客户浏览器发送响应**

**① 若尚未成功进行扫码登录，则返回包含二维码的未登录界面**

**② 若已成功进行扫码登录，则返回包含用户信息的注销界面**

**二维码内容规范**

二维码内容为加密过的客户浏览器IP地址字符串。

加密方式为：

ALPHA[] = "46CdeFGhIJK1mn0pQR5tuVwxYz"

IP[] = "xxx.xxx.xxx.xxx"

Status Encrypt(char \*dst, char \*src){

while(\*src){

\*dst = ALPHA[(\*src) % 26];

src++;

dst++;

}

return OK;

}

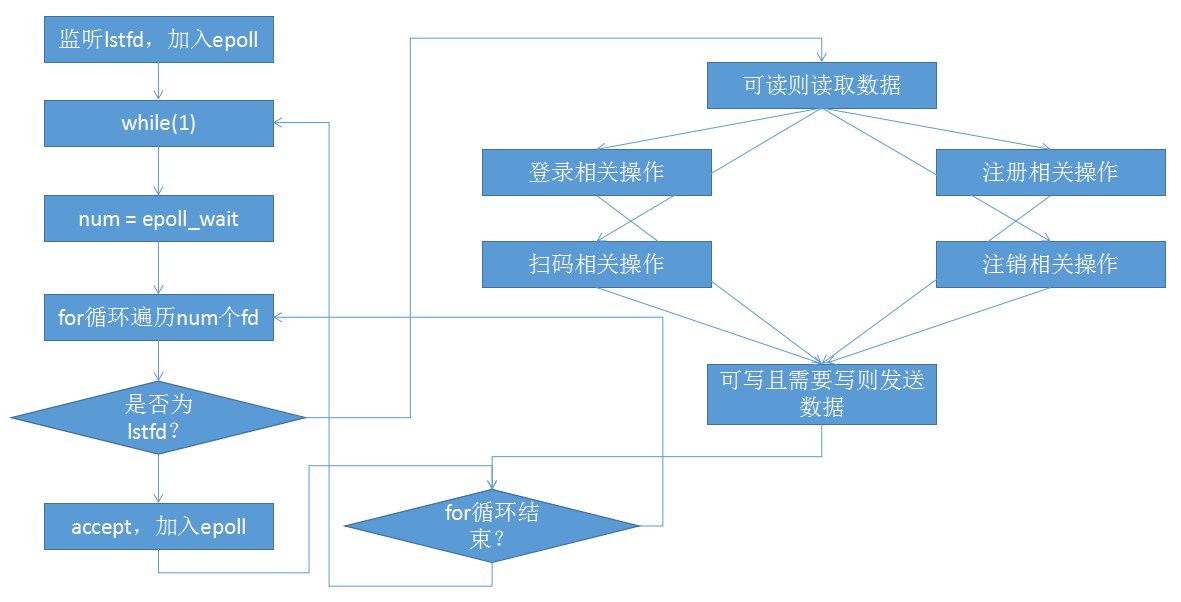
Encrypt(enIP, IP);

进行逆过程操作并结合ASCII码范围即可还原出IP。

**三 服务端开发**

**1 设备服务器开发**

采用“单进程-非阻塞Epoll”模型



**操作描述**

**① 登录相关操作**

根据用户名检查”./MobileLogin”目录下有无同名文件，如果有，说明用户已经在别处登录，生成登录结果为“已登录”的数据包等待发送；如果无，则在数据库中以收到的用户名+密码MD5作为条件查询结果，存在则生成登录结果为“当前socket文件描述符”的数据包等待发送，并在”./MobileLogin”目录下创建以该用户名为名的文件；否则，生成登录结果为“错误”的数据包等待发送。

**② 注册相关操作**

根据用户名检查”./MobileLogin”目录下有无同名文件，如果有，说明用户已经在别处登录，生成登录结果为“已登录”的数据包等待发送；如果无，则在数据库中以收到的用户名作为条件查询结果，存在则生成注册结果为“用户名重复”的数据包等待发送；否则，则在数据库中插入新的用户数据，生成注册结果为“当前socket文件描述符”的数据包等待发送，并在”./MobileLogin”目录下创建以该用户名为名的文件。

**③ 扫码相关操作**

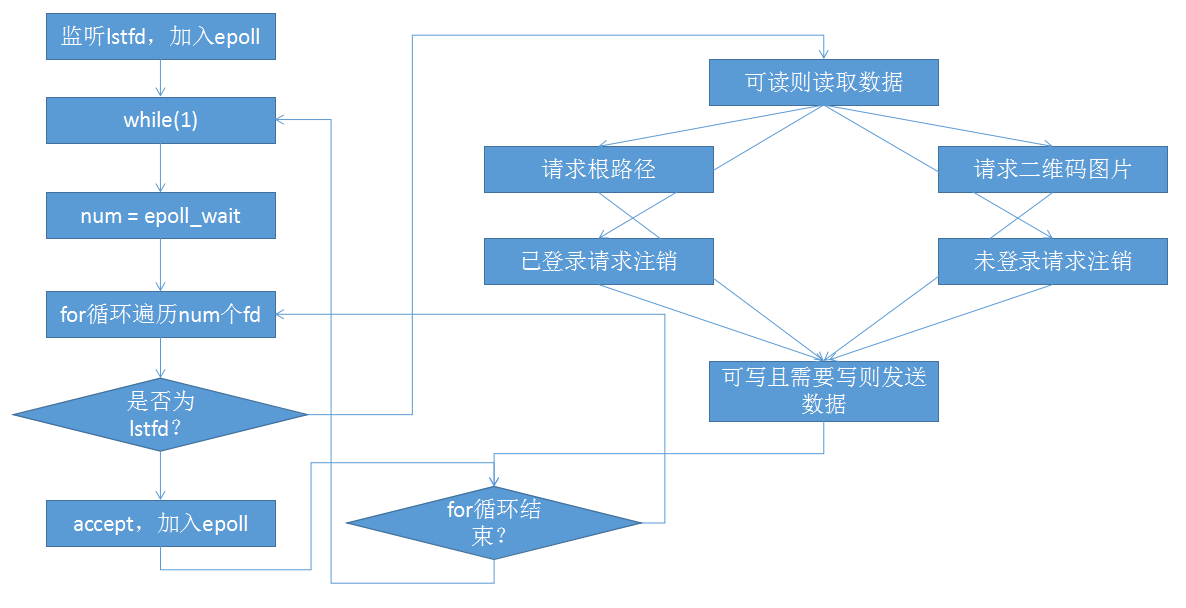
判断当前连接的设备是否为登录状态。是，则在”./WebLogin”文件夹下生成以收到IP为名，以当前设备登录用户名为内容的文件，并生成扫码结果为“无错误”的数据包等待发送；否则，生成扫码结果为“错误”的数据包等待发送。

**④ 注销相关操作**

生成注销结果为“无错误”的数据包等待发送，在发送完后断开连接，并删除”./MobileLogin”目录下相关文件。

**2 网页服务器开发**

采用“单进程-非阻塞Epoll”模型



**操作描述**

**① 请求根路径**

判断客户浏览器当前是否已登录（”./WebLogin”目录下是否有以当前客户浏览器IP为名的文件）。是，则读取该文件并获得用户名，生成特定用户的页面并等待发送；否，则生成包含二维码URL的文件并等待发送。

**② 请求二维码图片**

根据客户浏览器IP生成加密字符串，再针对加密字符串生成二维码，并等待发送。

**③ 已登录请求注销**

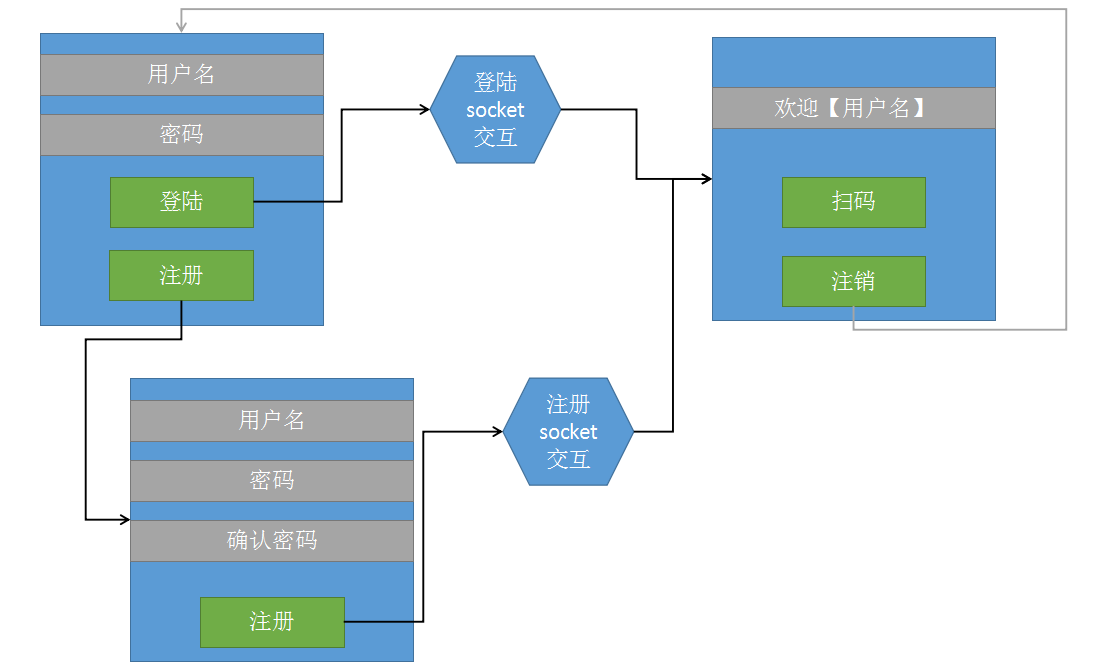
在已登录状态下，生成注销页面并等待发送，并删除”./WebLogin”目录下以客户浏览器IP为名的文件。

**④ 未登录请求注销**

生成包含二维码URL的文件并等待发送。

**四 客户端开发**

采用“多Activity-单例Application控制socket”模型



**五 总结与反思**

·用户名和密码可能导致SQL注入攻击，更好的做法是在服务端对收到的用户名和密码MD5的合法性进行检查

·Android开发中非主线程不可以进行`MainActivity.this`的Toast

·在Android 4.0后，所有网络通信操作要放入分支线程中进行

·可以使用一个单例`Application`实现在不同`Activity`之间维持同一个socket

·Java也有字节序问题，需要注意从socket收到的多字节数据类型要做字节序转换