## Android增量升级方案

1. **解决问题**

之前Android端升级方案一直是下载整个APK包，浪费时间和数据流量。此方案只需要下载新旧版本的差异包（一般比整个APK小80%），能节约大部分时间和数据流量，优化用户体验。

1. **原理**

  首先将应用的旧版本APK与新版本APK做二进制文件差异比较，得到差异部分（差异包）。Android端将获取的差异部分和当前安装的APK文件组合，在本地合成新版本的APK并进行安装。

本方案中的二进制差分、组合工具使用bsdiff /bspatch库（C/C++语言，提供差分和合并功能）。

1. **方案**

服务器端： 1、保存APK各版本及对应的MD5值（用于Android端的APK文件验证）

1. 根据新的APK生成各版本的差异包（使用bsdiff /bspatch库）  
    3、指定升级方式： 强制升级、非强制升级；

差异升级、整个APK下载升级。

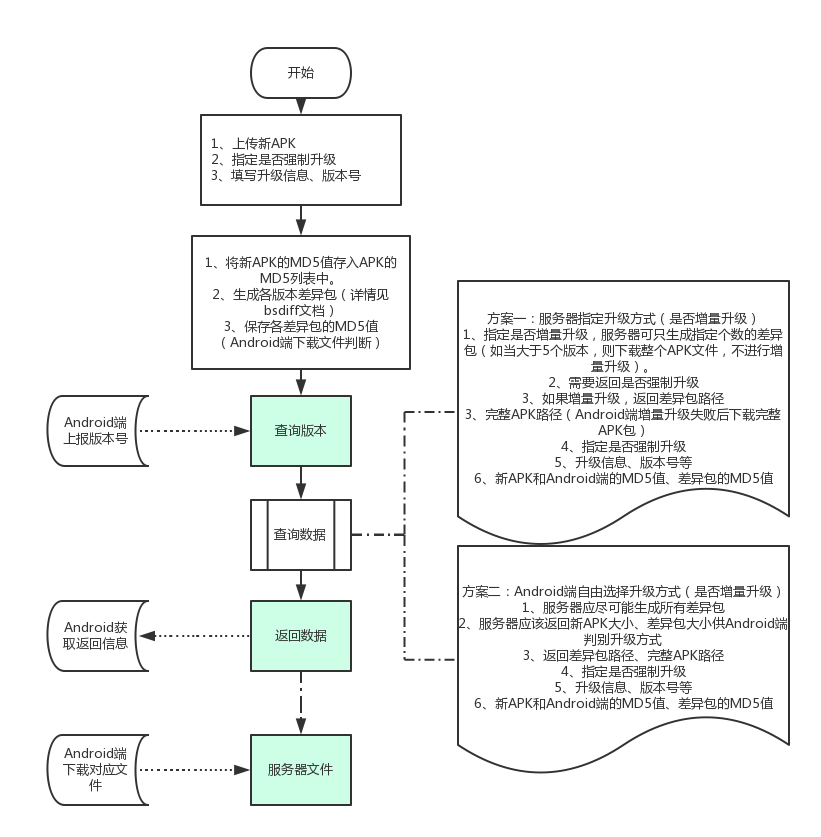
1. 提供下载整个APK方法（用于Android端增量升级失败后，下载整个APK）

Android端： 1、版本查询

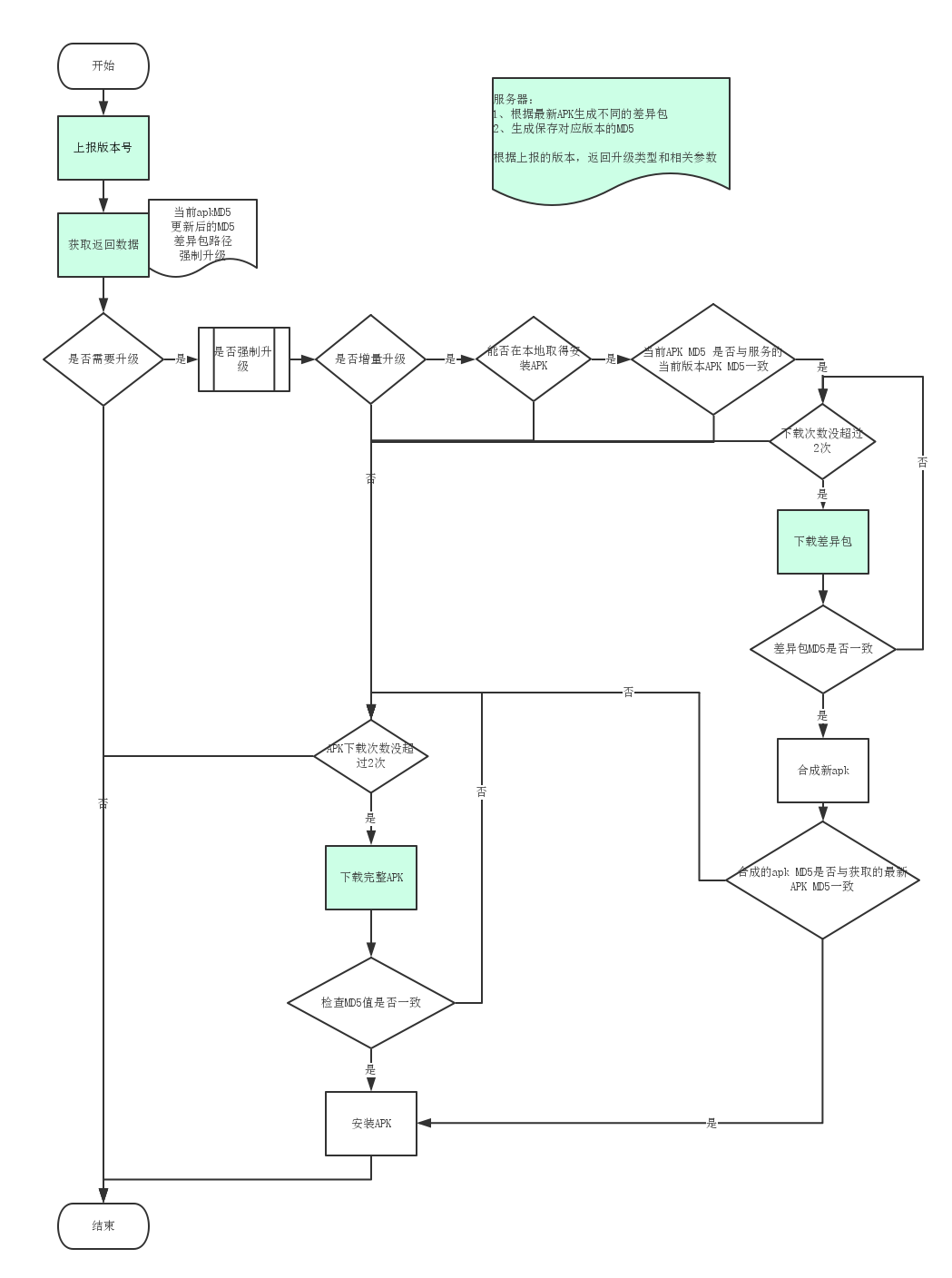
1. 差异升级时，根据差异包合成新的APK文件。（需要验证新旧APK的MD5值）

流程见下图：

服务器端：



Android端：

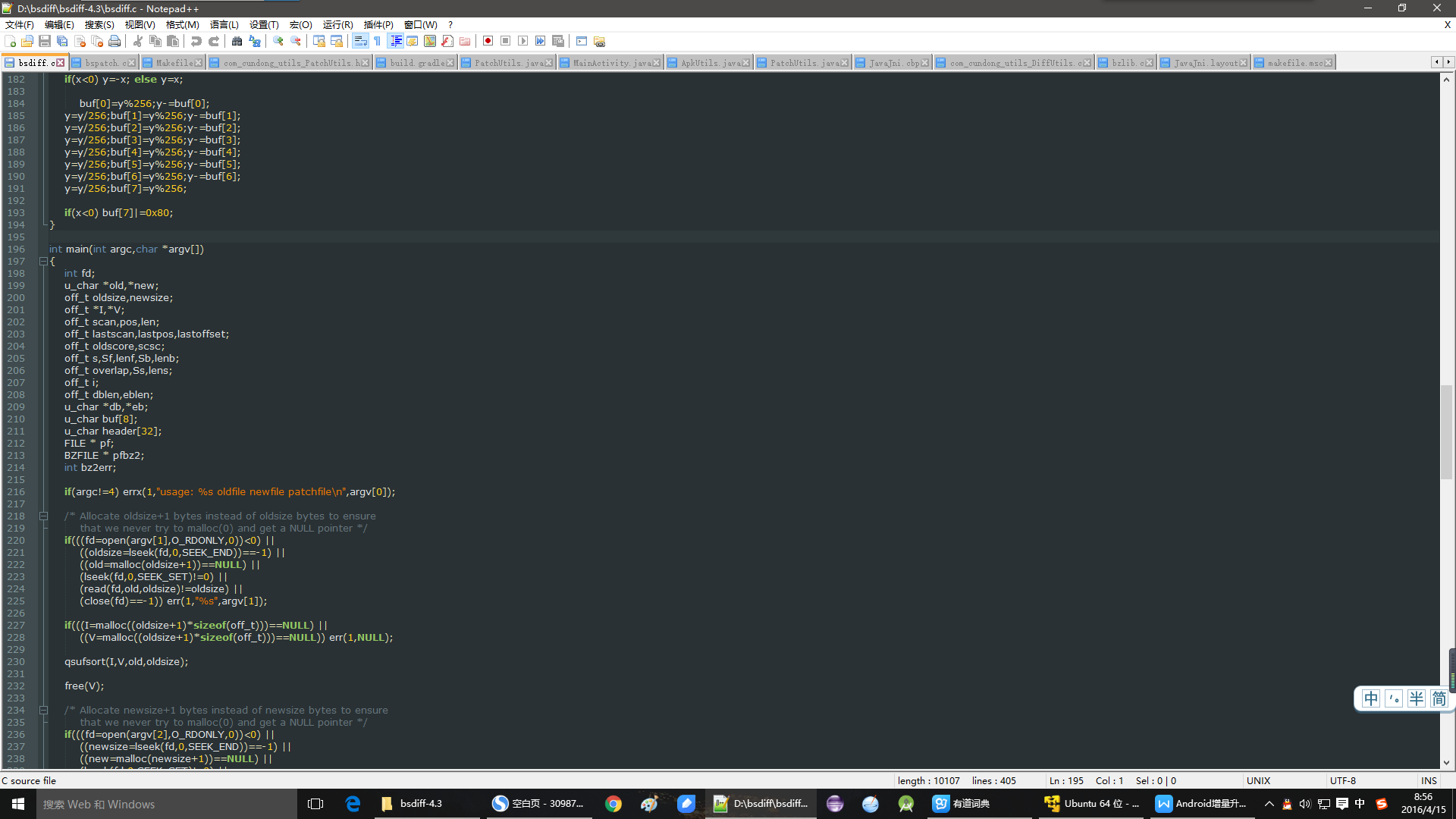


部分用法

生成差异包：bsdiff.c

参数： argc = 4；

argv[]: argv[0]=”bsdiff”；argv[1]=旧APK地址；agrv[2] = 新APK地址；argv[3]=生成差异包地址



合成差异包：bspatch.c

方法、参数同上。