安卓应用程序永久获取Root权限的方法

申请号：201210436649.9
申请日：2012-11-02

申请(专利权)人　福州博远无线网络科技有限公司

地址　350000 福建省福州市福州开发区星发路8号火炬创新大厦四层403A室

发明(设计)人　张长安　刘德建　陈宏展　罗上清　潘运武

主分类号　G06F9/44(2006.01)I

分类号　G06F9/44(2006.01)I　G06F9/445(2006.01)I

公开(公告)号　102981835A

公开(公告)日　2013-03-20

专利代理机构　福州市鼓楼区京华专利事务所(普通合伙) 35212

代理人　宋连梅

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102981835 A
(43)申请公布日 2013.03.20

(21)申请号 201210436649.9

(22)申请日 2012.11.02

(71)申请人 福州博远无线网络科技有限公司
地址 350000 福建省福州市福州开发区星发路 8 号火炬创新大厦四层 403A 室

(72)发明人 张长安 刘德建 陈宏展 罗上清 潘运武

(74)专利代理机构 福州市鼓楼区京华专利事务所（普通合伙） 35212
代理人 宋连梅

(51)Int.Cl.
G06F 9/44(2006.01)
G06F 9/445(2006.01)

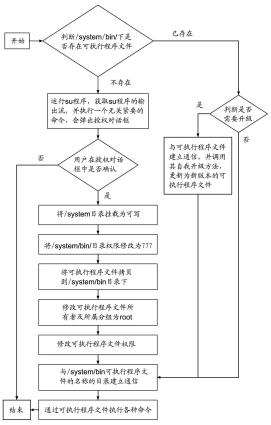
权利要求书 2 页 说明书 5 页 附图 1 页

(54)发明名称

安卓应用程序永久获取 Root 权限的方法

(57)摘要

本发明提供一种安卓应用程序永久获取 Root 权限的方法，将需要用到 Root 权限的操作进行编写，并编译成一可执行程序文件，将该文件包含在应用程序中，应用程序通过一次 su 程序授权，将文件拷贝到 /system/bin 的目录下，并修改其所有者和所属分组为 root，并设置其 suid 标识位；如此，就能够使该文件每次执行都拥有 Root 权限，从而避免再次弹出授权框。由于需求的变动，该文件程序可能会需要升级，因此文件需要实现自我升级的功能，以避免更新时需要再次弹出授权框。本发明通过该文件以 Root 的身份执行各种操作，就算应用程序被重装、升级，也无需再进行重复的 su 程序授权操作，给用户的使用带来极大方便。



CN 102981835 A

1. 一种安卓应用程序永久获取 Root 权限的方法，其特征在于：包括如下步骤：

步骤1、将安卓应用程序中需要用到Root权限的操作通过C/C++语言进行编写，并编译成一安卓系统可执行程序文件，且所述可执行程序文件能实现自我升级功能和与安卓应用程序的通信的功能；将可执行程序文件打包在安卓应用程序中；

步骤 2、在安卓系统下判断 /system/bin 的目录下是否存在所述可执行程序文件，否，则进入步骤 3；是，则判断可执行程序文件是否需要升级，要升级，则安卓应用程序与该可执行程序文件建立通信，并调用自我升级功能，对该可执行程序文件进行升级处理，并转入步骤 8；不升级，则直接进入步骤 8；

步骤 3、安卓应用程序运行安卓系统的 su 程序对操作用户进行第一次授权；

步骤 4、将安卓系统的 /system 目录进行重挂载为可写状态；

步骤 5、将安卓系统的 /system/bin 目录的权限进行设置，以保证安卓应用程序对 /system/bin 的目录拥有写的权限；

步骤 6、将所述可执行程序文件拷贝到所述 /system/bin 的目录下，并修改该可执行程序文件权限分类的所有者和所属分组为 Root；

步骤7、设置可执行程序文件的suid标识位，设置suid标识位后，可执行程序文件就能以所有者的身份被安卓应用程序所执行，即拥有 Root 权限的操作；

步骤 8、安卓应用程序执行可执行程序文件并与其建立通信，此时安卓应用程序需要用到 Root 权限的操作只要通过可执行程序文件进行执行即可，无需通过安卓系统的 su 程序进行重复授权。

2. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述 Root 权限的操作包括：获取文件目录、静默安装程序以及卸载程序。

3. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述步骤 3 具体为：安卓应用程序运行安卓系统的 su 程序，并获取 su 程序的输出流，之后的 su 命令可直接往输出流写入命令并执行，此时，往 su 程序的输出流写入一任意的字符命令，安卓系统的 SuperUser 会弹出授权对话框，操作用户点击允许，则继续执行所述步骤 4，操作用户点击拒绝，则结束流程。

4. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述步骤 4 具体为：往 su 程序的输出流写入第一命令，将安卓系统的 /system 目录进行重挂载为可写状态，该第一命令为：“mount-oremount-w - t[type][device]/system”，其中 [type] 为要挂载的文件系统类型，[device] 为要挂载的设备，此两项为 /system 目录挂载的相应参数，是通过执行安卓系统的 shell 命令来获取的。

5. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述步骤 5 具体为：往 su 程序的输出流写入第二命令，将安卓系统的 /system/bin 目录的权限改为 777，以保证安卓应用程序对 /system/bin 的目录拥有写的权限，该第二命令为：“chmod 777/system/bin”。

6. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述步骤 6 具体为：往 su 程序的输出流写入第三命令，将可执行程序文件拷贝到所述 /system/bin 的目录下，该第三命令为“dd if = [fromPath]of = [toPath]”，其中 [fromPath] 为可执行程序文件此时所在的路径，[toPath] 为要拷贝到的路径，即 /system/bin/ 可执行程

序文件的名称；并往 su 程序的输出流写入第四命令，修改该可执行程序文件权限分类的所有者和所属分组为 root，第四命令为“chown root :root/system/bin/ 可执行程序文件的名称”。

7. 根据权利要求 1 所述的安卓应用程序永久获取 Root 权限的方法，其特征在于：所述步骤 7 具体为：往 su 程序的输出流写入第五命令，设置可执行程序文件的 suid 标识位，并将可执行程序文件权限设置为可读可执行，即可执行程序文件权限设置为 4755，其中“4”是 suid 标识位，可执行程序文件就能以所有者的身份被安卓应用程序所执行，即拥有 Root 权限的操作；该第五命令为：“chmod 4755/system/bin/ 可执行程序文件的名称”。

安卓应用程序永久获取 Root 权限的方法

【技术领域】

[0001] 本发明涉及安卓系统上的应用程序权限管理技术领域，尤其涉及一种已经 Root 的安卓手机上，安卓应用程序永久获取 Root 权限的方法。

【背景技术】

[0002] 安卓 Android 系统中，第三方应用程序是以临时用户的身份进行运行的，这就使得该应用程序的权限受限制，如应用程序要访问或者修改安卓系统文件时，就会因为权限不足而失败。现在有的安卓手机已经进行了 Root 设置，该 Root 是 Linux 等类 UNIX 系统中的超级管理员用户帐户。对于 Android 手机，平常所说的 Root，其实就是通过各种方法，将系统的 su 程序文件拷贝到 /system/bin 目录下，并安装 SuperUser 授权管理，第三方应用程序可以通过 su 程序（su 程序是权限管理文件）执行需要 Root 权限的操作。但已经 Root 设置的手机，第三方应用程序就能直接拥有 Root 权限的操作吗？其实不然，Root 的手机上，会在 /system/bin 目录下存放一个名为 su 程序的文件，并且安装有 SuperUser（授权管理）应用程序。其他应用程序想要以 Root 权限执行，就必须通过 su 程序执行 shell 命令，而此时，SuperUser 就会弹出授权对话框，用户必须点击允许，命令才能成功被执行。点击允许时，如果没有选择“记住”选项，那么应用每次通过 su 程序执行 shell 命令时，都会弹出授权提醒的对话框。另外，如果应用程序升级了，那么就算之前已经记住允许操作了，再通过 su 程序执行 shell 命令时，也要重新通过 su 程序进行重复授权。

[0003] 另外弹出的授权提醒的对话框会带来如下问题：对于用户在手机的一个界面上操作手机上的第三方应用时，授权框的弹出并没有太大问题，因为此时用户肯定是看着手机在执行某些操作，用户能够及时的发现授权对话框弹出并确认，只是操作比较繁琐而已。但是，有些应用程序的操作界面并不再手机上，而是在 pc 机上，用户将手机连接到 pc 机，然后通过 pc 机来管理手机，此时，用户并不会去关注手机屏幕，如此，在授权对话框弹出时用户就不能及时发现并确认，而且会因为手机进入屏保，授权框无法显示，这样就导致授权失败。当然，用户在第一次授权时，可以勾选“记住”选项，下次就不用再授权，但是应用程序是会经常更新版本的，更新后，又得重新进行授权，给用户的使用带来极大不便。

[0004] 现有技术中提供了一种“Android 平台下的应用程序动态权限管理方法”，见公开号为：CN102289633A，公开日为：2011.12.21 的中国专利，包括应用程序安装时动态选择保存自身系统权限表；安装后，用户通过人机接口动态选择修改所述系统权限表；访问需要权限的 API 时，Android 平台根据该应用程序的系统权限表中的记录判断是否该 API 被禁用；当所述 API 被禁用时，该应用程序通过人机接口提示用户是否选择修改；若用户选择修改权限，Android 平台允许该应用程序访问所述 API，否则该应用程序退出访问。这种权限管理方法，用户能根据需要及个人经验随时修改应用程序的权限，从而能很大程度上防止应用程序的恶意访问资源，修改、窃取个人信息等危险行为。但该方法要是 Android 平台下的应用程序有更新时，该设置的权限管理将失效，还要重新进配置，操作麻烦。

[0005] 为了让审查人员更容易理解本发明这里需要指出的是：

[0006] 1、Linux 中，文件权限的问题 ：

[0007] 文件权限 ：每个 Linux 文件具有四种访问权限 ：可读 (r)、可写 (w)、可执行 (x)，无权限 (-)。

[0008] 权限分类 ：所有者，组成员，其他用户。每类的权限可以用字母表示，如 rwxrwxr-x 表示 ：即所有者及组成员拥有读、写、执行的权限，其他用户只拥有读和执行的权限（即 rwx ‘所有者’ rwx ‘组成员’ r-x ‘其他用户’）。数字表示法 ：所谓数字表示法，是指将 r、w 和 x 分别用 4、2、1 来代表，没有授予权限（-）的则为 0，然后把权限相加，如下 ：文件权限为 rwxr-xr-x，转换为数字为（421）（401）（401），相加后表示为 755。

[0009] 2、Linux 用户及分组 ：

[0010] Linux 是一个多任务多用户操作系统。

[0011] 多任务 ：多个前台应用程序和后台服务程序宏观上的同时运行。

[0012] 多用户 ：多个用户同时登录系统作业 ；为保证用户的私密性而建立的保护和限制机制。

[0013] 通常的多用户系统中都存在一个超级用户，例如 Linux 中的 root 用户，Windows 系统中的 administrator 等，系统中可以存在若干个具有不同的权限分类等级的其他用户，例如由超级用户建立的普通用户等，目前大多 Linux 发行版在安装时都要求建立一个普通用户，并使用该用户登陆。

[0014] 3、用户组 ：

[0015] 可以对用户进行分组，一个用户组可包含多个用户，一个用户也可以同时属于多个分组。

[0016] 例如，有以下分组及用户 ：

[0017] 分组 group1，包含用户 user1，user2

[0018] 分组 group2，包含用户 user2

[0019] 现在由 group1 中的 user1 创建一个文件 f，指定其权限为 775，那么 user1 及 user2 对文件 f 都有读、写、执行的权限，而 group2 中的 user2 则只有读和执行的权限

[0020] 4、系统中的标识位 suid ：

[0021] suid 是 Set User ID 的意思

[0022] 如果一个文件被设置了 suid 标识位，会表现在所有者的权限的可执行位上即可执行 (x) 的位置上，例如 ：

[0023] rwsr-xr-x 表示 suid 和权限分类中的所有者的权限中可执行位被设置。

[0024] 数字表示法 ：suid 对应的数字为 4，如 rwsr-xr-x 就表示为 4755

[0025] suid 的作用 ：让本来没有相应权限的用户运行这个程序时，可以访问他没有权限访问的资源（即让一些资源没有访问权限的有了访问权限）。

[0026] 例如，有用户 user1 和 user2 ；现在由用户 user1 创建一个普通文件 f，其权限为 rwxr-xr-x（755），并创建一个可执行文件 r，且其权限为 rwxr-xr-x（未设置 suid），此时由 user2 运行程序 r，r 的有效用户 id（euid）就是 user2，由于 user2 对文件 f 没有写的权限，所以 r 程序也就无法修改文件 f ；如果设置了文件 f 的 suid，即权限改为 rwsr-xr-x（4755），此时再由 user2 运行程序 r，r 的 euid 将被设置为其所有者的 id，即 user1，也就可以修改文件 f 了。

【发明内容】

[0027] 本发明要解决的技术问题，在于提供一种安卓应用程序永久获取 Root 权限的方法，其解决应用程序以 Root 身份执行命令时，SuperUser 会多次弹出授权提醒的问题。

[0028] 本发明是这样实现的：一种安卓应用程序永久获取 Root 权限的方法，包括如下步骤：

[0029] 步骤 1、将安卓应用程序中需要用到 Root 权限的操作通过 C/C++ 语言进行编写，并编译成一安卓系统可执行程序文件，且所述可执行程序文件能实现自我升级功能和与安卓应用程序的通信的功能；将可执行程序文件打包在安卓应用程序中；

[0030] 步骤 2、在安卓系统下判断 /system/bin 的目录下是否存在所述可执行程序文件，否，则进入步骤 3；是，则判断可执行程序文件是否需要升级，要升级，则安卓应用程序与该可执行程序文件建立通信，并调用自我升级功能，对该可执行程序文件进行升级处理，并转入步骤 8；不升级，则直接进入步骤 8；

[0031] 步骤 3、安卓应用程序运行安卓系统的 su 程序对操作用户进行第一次授权；

[0032] 步骤 4、将安卓系统的 /system 目录进行重挂载为可写状态；

[0033] 步骤 5、将安卓系统的 /system/bin 目录的权限进行设置，以保证安卓应用程序对 /system/bin 的目录拥有写的权限；

[0034] 步骤 6、将所述可执行程序文件拷贝到所述 /system/bin 的目录下，并修改该可执行程序文件权限分类的所有者和所属分组为 Root；

[0035] 步骤 7、设置可执行程序文件的 suid 标识位，设置 suid 标识位后，可执行程序文件就能以所有者的身份被安卓应用程序所执行，即拥有 Root 权限的操作；

[0036] 步骤 8、安卓应用程序执行可执行程序文件并与其建立通信，此时安卓应用程序需要用到 Root 权限的操作只要通过可执行程序文件进行执行即可，无需通过安卓系统的 su 程序进行重复授权。

[0037] 本发明具有如下优点：本发明将需要用到 Root 权限的操作进行编写，并编译成一可执行程序文件，将该可执行程序文件进行部署、设置，这样能够使该文件每次执行都拥有 Root 权限，从而避免再次弹出授权框。由于需求的变动，该文件程序可能会需要升级，因此文件需要实现自我升级的功能，以避免更新时需要再次弹出授权框。本发明通过该文件以 Root 的身份执行各种操作，就算应用程序被重装、升级，也无需再进行重复的 su 程序授权操作，给用户的使用带来极大方便。

【附图说明】

[0038] 图 1 为本发明方法流程示意图。

【具体实施方式】

[0039] 如图 1 所示，本发明的一种安卓应用程序永久获取 Root 权限的方法，包括如下步骤：

[0040] 步骤 1、将安卓应用程序中需要用到 Root 权限的操作通过 C/C++ 语言进行编写，并编译成一安卓系统可执行程序文件（如，该可执行程序文件的名称为 mySu），且所述可执行

程序文件能实现自我升级功能和与安卓应用程序的通信的功能；将可执行程序文件打包在安卓应用程序中；

[0041] 步骤 2、在安卓系统下判断 /system/bin 的目录下是否存在所述可执行程序文件，否，则进入步骤 3；是，则判断可执行程序文件是否需要升级，要升级，则安卓应用程序与该可执行程序文件建立通信（该通信时借于可执行程序文件与安卓应用程序有通信的功能），并调用自我升级功能，对该可执行程序文件进行升级处理，并转入步骤 8；不升级，则直接进入步骤 8；

[0042] 步骤 3、安卓应用程序运行安卓系统的 su 程序对操作用户进行第一次授权；其具体为：安卓应用程序运行安卓系统的 su 程序，并获取 su 程序的输出流，之后的 su 命令可直接往输出流写入命令并执行，而无需重新运行 su 程序，如此就只需要用户授权一次即可，此时，往 su 程序的输出流写入一任意的字符命令（该字符命令为无关紧要的命令，如“id”），该字符命令主要是为了让安卓系统的 SuperUser 会弹出授权对话框，此时，安卓系统的 SuperUser 会弹出授权对话框，操作用户点击允许，则继续执行所述步骤 4，操作用户点击拒绝，则结束流程。

[0043] 步骤 4、将安卓系统的 /system 目录进行重挂载为可写状态；其具体为：往 su 程序的输出流写入第一命令，将安卓系统的 /system 目录进行重挂载为可写状态，该第一命令为：“mount-o remount-w-t[type][device]/system”，其中 [type] 为要挂载的文件系统类型，[device] 为要挂载的设备，此两项为 /system 目录挂载的相应参数，是通过执行安卓系统的 shell 命令来获取的。

[0044] 步骤 5、将安卓系统的 /system/bin 目录的权限进行设置，以保证安卓应用程序对 /system/bin 的目录拥有写的权限；其具体为：往 su 程序的输出流写入第二命令，将安卓系统的 /system/bin 目录的权限改为 777，以保证安卓应用程序对 /system/bin 的目录拥有写的权限，该第二命令为：“chmod 777/system/bin”。

[0045] 步骤 6、将所述可执行程序文件拷贝到所述 /system/bin 的目录下，并修改该可执行程序文件权限分类的所有者和所属分组为 Root；其具体为：往 su 程序的输出流写入第三命令，将可执行程序文件拷贝到所述 /system/bin 的目录下，该第三命令为“dd if ＝ [fromPath]of ＝ [toPath]”，其中 [fromPath] 为可执行程序文件此时所在的路径，[toPath] 为要拷贝到的路径，即 /system/bin/ 可执行程序文件的名称；并往 su 程序的输出流写入第四命令，修改该可执行程序文件权限分类的所有者和所属分组为 root，第四命令为“chown root :root/system/bin/ 可执行程序文件的名称（即为“mySu”，该名称也可以用其他的命名来替换）”。

[0046] 步骤 7、设置可执行程序文件的 suid 标识位，设置 suid 标识位后，可执行程序文件就能以所有者的身份被安卓应用程序所执行，即拥有 Root 权限的操作；其具体为：往 su 程序的输出流写入第五命令，设置可执行程序文件的 suid 标识位，并将可执行程序文件权限设置为可读可执行，即可执行程序文件权限设置为 4755，其中“4”是 suid 标识位，可执行程序文件就能以所有者的身份被安卓应用程序所执行，即拥有 Root 权限的操作；这样应用程序即能永久获取 Root 权限的操作；该第五命令为：“chmod 4755/system/bin/ 可执行程序文件的名称（即为“mySu”）”。

[0047] 步骤 8、安卓应用程序执行可执行程序文件并与其建立通信，此时安卓应用程序需

要用到 Root 权限的操作只要通过可执行程序文件进行执行即可，无需通过安卓系统的 su 程序进行重复授权。

[0048] 其中，所述 Root 权限的操作包括：获取文件目录、静默安装程序以及卸载程序等等。

[0049] 这里值得一提的是：

[0050] 挂载和重挂载的区别，挂载：

[0051] Linux 系统中，需要将存储设备挂载到某个目录，我们才能在文件管理中正常使用；则一般通过 shell 中执行 mount 命令可以看到各个目录的挂载信息，例如：/dev/block/mmcblk0p9/system ext4 ro；该信息表示：将存储设备“/dev/block/mmcblk0p9”挂载到 /system 目录，其文件系统类型为“ext4”，其读写权限为“ro”（只读）

[0052] 重挂载：一般通过 shell 执行 remount 命令进行重新挂载存储设备：例如：“mount-o remount-w-t ext4/dev/block/mmcblk0p9/system”，表示将 /system 重挂载，其文件系统类型为“ext4”，其存储设备为“/dev/block/mmcblk0p9”，其读写权限为“-w”，即可读可写。步骤 4 中涉及的是进行重挂载的操作。

[0053] 总之，本发明通过该文件以 Root 的身份执行各种操作，就算应用程序被重装、升级，也无需再进行重复的 su 程序授权操作，给用户的使用带来极大方便。

[0054] 以上所述仅为本发明的较佳实施例，凡依本发明申请专利范围所做的均等变化与修饰，皆应属本发明的涵盖范围。

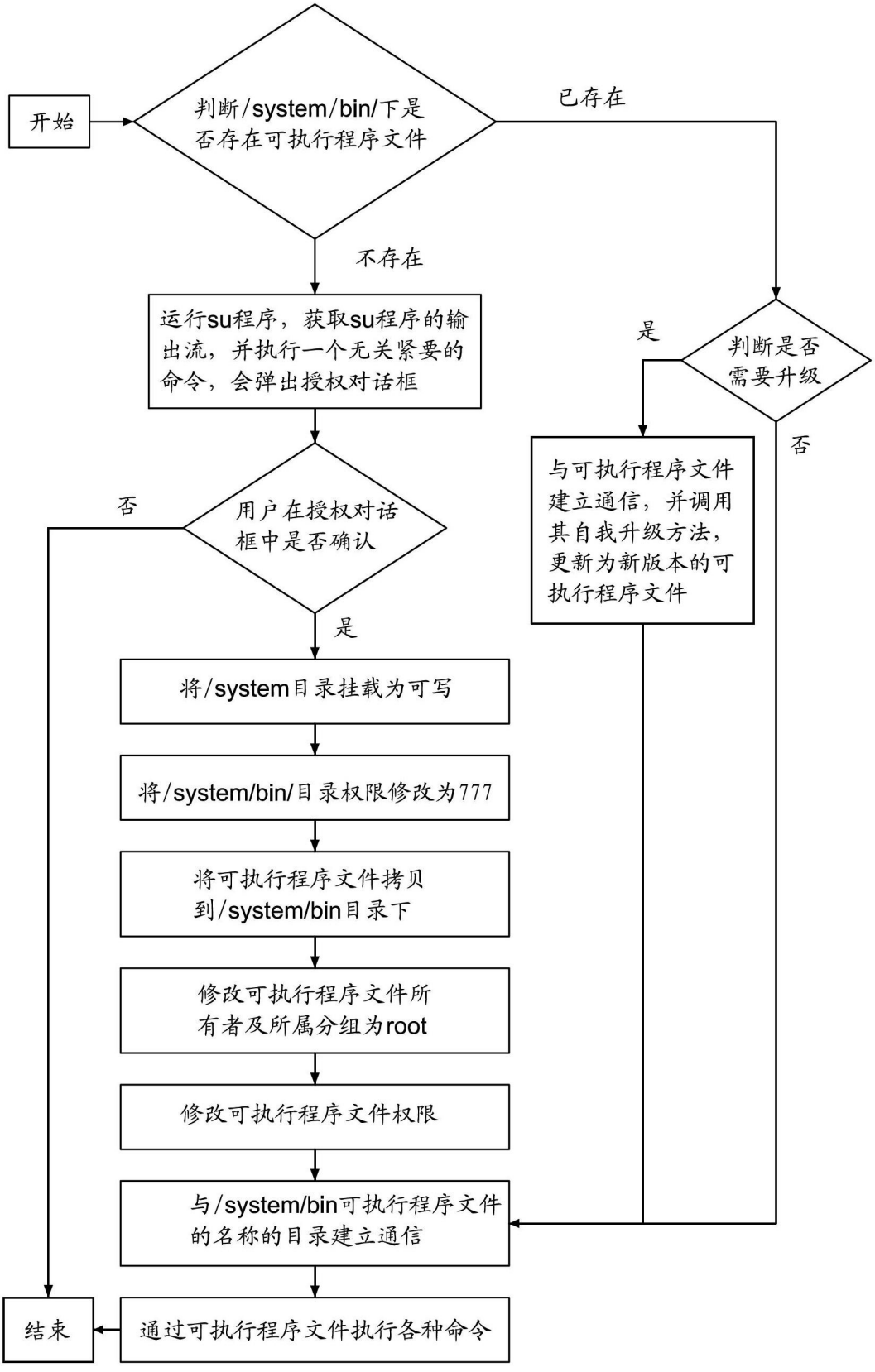

开始
判断/system/bin/下是否存在可执行程序文件
已存在
不存在
运行su程序，获取su程序的输出流，并执行一个无关紧要的命令，会弹出授权对话框
判断是否需要升级
是
否
与可执行程序文件建立通信，并调用其自我升级方法，更新为新版本的可执行程序文件
用户在授权对话框中是否确认
否
是
将/system目录挂载为可写
将/system/bin/目录权限修改为777
将可执行程序文件拷贝到/system/bin目录下
修改可执行程序文件所有者及所属分组为root
修改可执行程序文件权限
与/system/bin可执行程序文件的名称的目录建立通信
结束
通过可执行程序文件执行各种命令

图 1