



# (12)发明专利

(10)授权公告号 CN 103870310 B

(45)授权公告日 2017.07.14

(21)申请号 201410055020.9

(22)申请日 2014.02.18

(65)同一申请的已公布的文献号  
申请公布号 CN 103870310 A

(43)申请公布日 2014.06.18

(73)专利权人 小米科技有限责任公司  
地址 100085 北京市海淀区清河中街68号  
华润五彩城购物中心二期13层

(72)发明人 葛旭东 汪文俊 许瑞军

(74)专利代理机构 北京弘权知识产权代理事务  
所(普通合伙) 11363

代理人 逯长明 陈蕾

(51)Int.Cl.  
G06F 9/45(2006.01)

(56)对比文件

CN 101821711 A,2010.09.01,  
CN 102118500 A,2011.07.06,  
CN 102156643 A,2011.08.17,  
US 2009049434 A1,2009.02.19,

审查员 宫召英

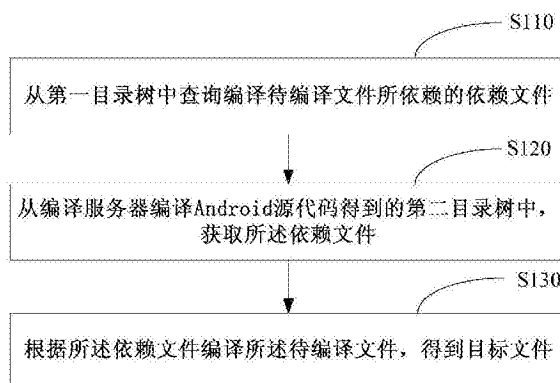
权利要求书3页 说明书14页 附图7页

(54)发明名称

程序编译方法及相关装置

(57)摘要

本公开是关于程序编译方法及装置,所述方法从存储在编译主机上的第一目录树中,查询编译选择的待编译文件所依赖的依赖文件;然后,从存储在编译服务器上的第二目录树中获取依赖文件;利用依赖文件编译待编译文件,得到目标文件。实现了在编译Android源代码时,能够依赖于除待编译文件所在的目录树之外的其它目录树(第二目录树),无需研发人员下载维护一套完整的Android源代码,只需下载本模块对应的源代码,减小了硬盘空间的占用。而且,能够直接从第二目录树中获得目标文件所需的依赖文件,无需通过编译得到依赖文件,从而节省了编译依赖文件占用的时间和资源。因此,大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。



1. 一种程序编译方法,其特征在于,包括:

从第一目录树中查询编译待编译文件所依赖的依赖文件;

从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件;

根据所述依赖文件编译所述待编译文件,得到目标文件;

其中,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息;

依据所述路径信息获取所述依赖文件。

2. 根据权利要求1所述的方法,其特征在于,从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息,采用如下方式:

接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;

根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。

3. 根据权利要求1所述的方法,其特征在于,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

4. 根据权利要求1所述的方法,其特征在于,所述从第一目录树中查询编译待编译文件所依赖的依赖文件,采用如下方式:

获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;

根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

5. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;

当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,执行所述从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖的步骤。

6. 一种程序编译方法,其特征在于,包括:

获取Android源代码;

编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

7. 根据权利要求6所述的方法,其特征在于,所述方法还包括:

检测编译主机中存储的Android源代码是否被修改;

当检测到编译主机中存储的所述Android源代码被修改时,依据修改后的Android源代码,更新编译服务器中存储的Android源代码;

编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

8. 根据权利要求7所述的方法,其特征在于,获取编译主机编译得到的各个目标文件对应的源代码,采用如下方式:

编译服务器从中心服务器中获取各个目标文件对应的Android源代码,所述Android源代码由编译主机将编译得到的目标文件对应的源代码上传至中心服务器;

或,

编译服务器从编译主机中获取各个目标文件对应的Android源代码。

9. 根据权利要求8所述的方法,其特征在于,所述方法还包括:

将所述映射关系发送给编译主机,所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件的路径信息,根据所述路径信息获取所述依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

10. 根据权利要求8所述的方法,其特征在于,所述方法还包括:

接收编译主机发送的所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

根据所述依赖文件的文件信息,查询所述映射关系获得所述依赖文件的路径信息。

11. 一种程序编译装置,其特征在于,包括:

第一查询单元,用于从第一目录树中查询编译待编译文件所依赖的依赖文件;

第一获取单元,用于从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件;

编译单元,用于根据所述依赖文件编译所述待编译文件,得到目标文件;

其中,所述第一获取单元,包括:

第一获取子单元,用于从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息;

第二获取子单元,用于根据所述路径信息获取所述依赖文件。

12. 根据权利要求11所述的装置,其特征在于,所述第一获取子单元,包括:

第一接收子单元,用于接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;

第一查询子单元,用于根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。

13. 根据权利要求11所述的装置,其特征在于,所述第一获取单元,包括:

发送单元,用于向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

第二接收子单元,用于接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

14. 根据权利要求11所述的装置,其特征在于,所述第一查询单元,包括:

第三获取子单元,用于获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;

第二查询子单元,用于根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

15.根据权利要求11所述的装置,其特征在于,所述装置还包括:

查找单元,用于在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,输出使所述第一获取单元执行相应动作的信号。

16.一种程序编译装置,其特征在于,包括:

第一获取单元,用于获取Android源代码;

第一编译单元,用于编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

17.根据权利要求16所述的装置,其特征在于,所述装置还包括:

检测单元,用于检测编译主机中存储的Android源代码是否被修改;

源代码更新单元,用于当检测到编译主机中的所述Android源代码被修改时,依据修改后的更新编译服务器中存储的Android源代码;

第二编译单元,用于编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

## 程序编译方法及相关装置

### 技术领域

[0001] 本公开涉及计算机技术领域,特别是涉及程序编译方法及相关装置。

### 背景技术

[0002] 程序编译是将高级程序设计语言转换成计算机硬件能识别的机器语言,以便计算机进行处理。

[0003] Android系统是以Linux为基础的半开源操作系统,主要应用于移动终端(例如,智能手机、平板电脑等),移动终端生产商可以根据自己的硬件个性定制属于自己的Android系统,此时,需要系统研发人员对系统的源代码进行开发及编译。

[0004] 编译Android的源代码时,可以将大型的开发项目分解为多个更易于管理的模块或子项目,而模块或子项目之间可能存在依赖关系,在编译某一目标文件时所依赖的文件具有不确定性,因此,系统研发人员需要在本地维护一套完整的Android源代码,而源代码非常庞大,在较好的网络环境下也需要几个小时才能下载完成,而且,随着Android版本的不断更新,为了增加软件的兼容性,通常需要在本地保存多个Android版本的源代码,占用硬盘和网络资源多;编译Android源代码需要30G左右的硬盘空间,占用大量的硬盘空间;而且编译时间通常需要数个小时,浪费了系统研发人员的大量时间。

### 发明内容

[0005] 本公开实施例中提供了一种程序编译方法及相关装置,以解决Android源代码编译占用资源多,且耗费时间长的问题。

[0006] 为了解决上述技术问题,本公开实施例公开了如下技术方案:

[0007] 根据本公开实施例的第一方面,提供一种程序编译方法,包括:从第一目录树中查询编译待编译文件所依赖的依赖文件;从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件;根据所述依赖文件编译所述待编译文件,得到目标文件。

[0008] 结合第一方面,在第一方面的第一种可能的实现方式中,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

[0009] 从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息;依据所述路径信息获取所述依赖文件。

[0010] 结合第一方面的第一种可能的实现方式,在第一方面的第二种可能的实现方式中,从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息,采用如下方式:

[0011] 接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。

[0012] 结合第一方面,在第一方面的第三种可能的实现方式中,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

[0013] 向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

[0014] 结合第一方面,在第一方面的第四种可能的实现方式中,所述从第一目录树中查询编译待编译文件所依赖的依赖文件,采用如下方式:

[0015] 获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;

[0016] 根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

[0017] 结合第一方面,在第一方面的第五种可能的实现方式中,所述方法还包括:

[0018] 在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;

[0019] 当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,执行所述从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖的步骤。

[0020] 根据本公开实施例的第二方面,提供一种程序编译方法,包括:

[0021] 获取Android源代码;编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0022] 结合第二方面,在第二方面的第一种可能的实现方式中,所述方法还包括:

[0023] 检测编译主机中存储的Android源代码是否被修改;

[0024] 当检测到编译主机中存储的所述Android源代码被修改时,依据修改后的Android源代码,更新编译服务器中存储的Android源代码;

[0025] 编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

[0026] 结合第二方面的第一种可能的实现方式,在第二方面的第二种可能的实现方式中,获取编译主机编译得到的各个目标文件对应的源代码,采用如下方式:

[0027] 编译服务器从中心服务器中获取各个目标文件对应的Android源代码,所述Android源代码由编译主机将编译得到的目标文件对应的源代码上传至中心服务器;

[0028] 或,

[0029] 编译服务器从编译主机中获取各个目标文件对应的Android源代码。

[0030] 结合第二方面的第二种可能的实现方式,在第二方面的第三种可能的实现方式中,所述方法还包括:

[0031] 将所述映射关系发送给编译主机,所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件的路径信息,根据所述路径信息获取所述依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0032] 结合第二方面的第二种可能的实现方式,在第二方面的第四种可能的实现方式中,所述方法还包括:

[0033] 接收编译主机发送的所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

- [0034] 根据所述依赖文件的文件信息,查询所述映射关系获得所述依赖文件的路径信息。
- [0035] 根据本公开实施例的第三方面,提供一种程序编译装置,包括:
- [0036] 第一查询单元,用于从第一目录树中查询编译待编译文件所依赖的依赖文件;
- [0037] 第一获取单元,用于从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件;
- [0038] 编译单元,用于根据所述依赖文件编译所述待编译文件,得到目标文件。
- [0039] 结合第三方面,在第三方面的第一种可能的实现方式中,所述第一获取单元,包括:
- [0040] 第一获取子单元,用于从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息;
- [0041] 第二获取子单元,用于根据所述路径信息获取所述依赖文件。
- [0042] 结合第三方面的第一种可能的实现方式,在第三方面的第二种可能的实现方式中,所述第一获取子单元,包括:
- [0043] 第一接收子单元,用于接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;
- [0044] 第一查询子单元,用于根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。
- [0045] 结合第三方面,在第三方面的第三种可能的实现方式中,所述第一获取单元,包括:
- [0046] 发送单元,用于向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;
- [0047] 第二接收子单元,用于接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。
- [0048] 结合第三方面,在第三方面的第四种可能的实现方式中,所述第一查询单元,包括:
- [0049] 第三获取子单元,用于获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;
- [0050] 第二查询子单元,用于根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。
- [0051] 结合第三方面,在第三方面的第五种可能的实现方式中,所述装置还包括:
- [0052] 查找单元,用于在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,输出使所述第一获取单元执行相应动作的信号。
- [0053] 根据本公开实施例的第四方面,提供一种程序编译装置,包括:
- [0054] 第一获取单元,用于获取Android源代码;

[0055] 第一编译单元,用于编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0056] 结合第四方面,在第四方面的第一种可能的实现方式中,所述装置还包括:

[0057] 检测单元,用于检测编译主机中存储的Android源代码是否被修改;

[0058] 源代码更新单元,用于当检测到编译主机中的所述Android源代码被修改时,依据修改后的更新编译服务器中存储的Android源代码;

[0059] 第二编译单元,用于编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

[0060] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

[0061] 本公开的实施例提供的技术方案可以包括以下有益效果:首先从存储在编译主机上的第一目录树中,查询编译选择的待编译文件依赖于哪些(或哪个)依赖文件,确定待编译文件所依赖的依赖文件后,从存储在编译服务器上的第二目录树中,获取所述依赖文件;然后,利用所述依赖文件编译所述待编译文件,得到最终的目标文件。实现了在编译Android源代码时,能够依赖于除待编译文件所在的目录树之外的其它的目录树(第二目录树),无需研发人员下载维护一套完整的Android源代码,只需下载本模块对应的源代码,减小了硬盘空间的占用。而且,能够直接从第二目录树中获得编译得到本模块的目标文件所需的依赖文件,无需通过编译得到所述依赖文件,从而节省了编译依赖文件占用的时间和资源。因此,本公开提供的程序编译方法大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。

## 附图说明

[0062] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本公开的实施例,并与说明书一起用于解释本公开的原理。

[0063] 图1是根据一示例性实施例示出的一种程序编译方法的流程图。

[0064] 图2是根据一示例性实施例示出的图1中步骤S120的一种流程图;

[0065] 图3是根据一示例性实施例示出的图1中步骤S120的另一种流程图;

[0066] 图4是根据一示例性实施例示出的另一种程序编译方法的流程图;

[0067] 图5是根据一示例性实施例示出的一种目标文件与依赖文件之间的依赖关系图;

[0068] 图6是根据一示例性实施例示出的一种编译方法的流程图;

[0069] 图7是根据一示例性实施例示出的另一种程序编译方法的流程图;

[0070] 图8是根据一示例性实施例示出的一种程序编译装置的框图;

[0071] 图9是根据一示例性实施例示出的另一种程序编译装置的框图;

[0072] 图10是根据一示例性实施例示出的一种用于实现程序编译的装置的框图;

[0073] 图11是根据一示例性实施例示出的一种用于实现程序编译的服务器的框图。

## 具体实施方式



[0074] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本公开相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本公开的一些方面相一致的装置和方法的例子。

[0075] 例如,某个模块对应目标文件A,而目标文件A依赖于文件B(即由文件B创建目标文件A),Make工具在编译目标文件A时,首先查找是否存在满足依赖条件的文件B,如果不存在,则编译得到文件B之后,才能利用文件B编译得到目标文件A。由此可知,系统研发人员可能只需要开发和测试某个模块或小项目,但是需要在本地维护一套完成的Android源代码。

[0076] 图1是根据一示例性实施例示出的一种程序编译方法的流程图,如图1所示,所述程序编译方法用于编译主机中,包括以下步骤:

[0077] 在步骤S110中,从第一目录树中查询编译待编译文件所依赖的依赖文件。

[0078] 本示例性实施例提供的程序编译方法主要用于编译Android源代码。所述待编译文件可以是整个Android源代码开发系统中划分的某个模块所对应的源代码。

[0079] 在编译之前,需要查找待编译文件所依赖的依赖文件,例如,某个模块编译完成后得到目标文件A,而目标文件A依赖于文件B,即由文件B编译得到目标文件A,在得到目标文件A之前,首先需要确定目标文件A依赖于哪个文件。

[0080] 所述待编译文件为Android源代码,例如,目标为A,则对应的待编译文件为目标A对应的源代码。所述第一目录树可以由编译主机编译得到,也可以由编译服务器编译或其它的编译主机编译得到。

[0081] 在本公开的一个示例性实施例中,步骤S110可以包括以下方法流程:

[0082] 1)获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于描述各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系。可以通过Make工具中的makefile文件描述Android源代码的各个模块之间的相互依赖关系,即makefile文件规定了最终得到的目标文件(可执行文件)与生成该目标文件的各个依赖文件之间的关系。

[0083] 2)根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

[0084] 在步骤S120中,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件。

[0085] 编译服务器对获得的与编译主机所要编译的Android源代码属于同一版本的Android源代码编译得到第二目录树,所述编译主机从所述第二目录树中获取所述待编译文件所依赖的符合条件的依赖文件。例如,符合条件可以是符合时间条件等。

[0086] 对于同一版本的Android源代码而言,某个目标与该目标所依赖的依赖文件之间的依赖关系是相同的,即编译主机对某一版本的Android源代码编译得到的第一目录树,与编译服务器(或另一编译主机)对所述某一版本的Android源代码编译得到的第二目录树,两个目录树中同一目标的依赖关系是相同的。

[0087] 在本公开的一个示例性实施例中,如图2所示,步骤S120可以包括以下子步骤:

[0088] S121,从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息。

[0089] 在本公开的一个示例性实施例中,获取所述依赖文件的路径信息可以通过以下流程实现:

[0090] 编译主机接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;然后,根据所述映射关系,查询得到所依赖的依赖文件对应的路径信息。即先将所述映射关系存储在编译主机上,在编译主机上查询所述映射关系获得所述依赖文件的路径信息。

[0091] 编译主机获得所述映射关系,可以首先由编译主机向编译服务器发送路径请求消息,所述路径请求消息用于请求编译服务器发送依赖文件的路径信息;然后,由编译服务器响应所述路径请求消息,向编译主机发送第二目录树中的各个依赖文件与路径信息之间的映射关系。还可以在编译服务器每次更新所述第二目录树后,自动向各个编译主机下发更新后的第二目录树中的各个依赖文件与路径信息之间的映射关系,此种方式能够降低各个编译主机对编译服务器的访问量,从而降低编译服务器的硬件配置要求。

[0092] S122,依据所述路径信息获取所述依赖文件。编译主机将获得的所述路径信息发送给编译服务器,然后,由编译服务器获取所述路径信息对应的存储空间中存储的依赖文件,并下发给所述编译主机。

[0093] 在本公开的另一个示例性实施例中,如图3所示,步骤S120可以包括以下子步骤:

[0094] S123,向编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称。

[0095] 当所述依赖文件为编译得到的某个目标文件时,所述文件名称即所述目标文件的文件名称。

[0096] S124,接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

[0097] 编译服务器根据所述编译主机发送的依赖文件的文件信息,在编译服务器中查找所述依赖文件的路径信息,并获取所述路径信息对应的存储空间中存储的文件,并下发给所述编译主机。其中,所述依赖文件的路径信息查询第二目录树中各个文件与路径信息之间的映射关系得到。

[0098] 在步骤S130中,根据所述依赖文件编译所述待编译文件,得到目标文件。

[0099] 本示例性实施例提供的程序编译方法,首先从编译主机中的第一目录树中,查询待编译文件依赖于哪些依赖文件,然后,从存储在编译服务器上的第二目录树中,获取所述依赖文件;再利用所述依赖文件编译所述待编译文件,得到最终的目标文件,从而实现了在编译Android源代码时,能够依赖于除待编译文件所在的目录树之外的其它的目录树(第二目录树)。研发人员只需在本地保存所述待编译文件(本模块对应的源代码),无需下载维护一套完整的Android源代码,减小了硬盘空间的占用。而且,能够直接从第二目录树中获得符合条件的依赖文件,无需通过编译得到所述依赖文件,从而节省了编译依赖文件占用的时间和资源。因此,本公开提供的程序编译方法大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。

[0100] 图4是根据一示例性实施例示出的另一种程序编译方法的流程图,如图4所示,所述方法应用于编译主机中,可以用于编译Android源代码,包括以下步骤:

[0101] 在步骤S110中,从第一目录树中查询编译待编译文件所依赖的依赖文件。

[0102] 在步骤S210中,在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件。

[0103] 当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,在步骤S120,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件。

[0104] 当所述第一目录树中存在所述待编译文件所依赖的依赖文件时,在步骤S220中,从所述第一目录树中获取所述依赖文件。

[0105] 在步骤S130中,根据所述依赖文件编译所述待编译文件,得到目标文件。

[0106] 本示例性实施例示出的程序编译方法,在编译Android源代码的过程中,首先确定待编译文件依赖于哪些依赖文件,在编译主机本地存储的第一目录树中查找是否存在符合条件的所述依赖文件,如果存在,优先从第一目录树中获取所述依赖文件,降低了编译服务器的访问量,从而能够降低编译服务器的硬件要求。

[0107] 下面结合图5所示的目标文件与依赖文件之间的依赖关系对上述的程序编译流程进行说明。图5示出了目标文件G与所有的依赖文件之间的关系,具体的,目标文件G依赖于E和F;目标文件E依赖于C;目标文件F依赖于D;目标文件C依赖于A;目标文件D依赖于A和B。

[0108] 假设,编译主机中待编译文件对应的目标文件为G,则根据图5所示的依赖关系图,可查找到目标文件G所依赖的依赖文件为E和F。然后,从编译服务器中存储的第二目录树中获取符合条件的依赖文件E和F。根据依赖文件编译待编译文件得到目标文件G。

[0109] 可选地,当查找到目标文件G的依赖文件为E和F时,可以先查找编译主机中的第一目录树中是否存在依赖文件E和F,当第一目录树中存在依赖文件E和F时,优先从第一目录树中获取所述依赖文件E和F,从而降低编译主机对编译服务器的访问量。

[0110] 图6是根据一示例性实施例示出的一种程序编译方法的流程图,如图6所示,所述方法应用于编译服务器,包括以下步骤:

[0111] 在步骤S610中,获取Android源代码。

[0112] 编译服务器需要获取与编译主机上的源代码版本相同的Android源代码。

[0113] 在步骤S620中,编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0114] 在本公开的一个示例性实施例中,编译主机根据编译服务器中的所述映射关系获取待编译文件所依赖的依赖文件,可以是编译服务器将所述映射关系发送给编译主机,编译主机根据所述映射关系获取待编译文件所依赖的依赖文件的路径信息,然后,编译主机将所述路径信息发送给编译服务器,由编译服务器获取所述路径信息对应的存储空间中存储的依赖文件,并下发给所述编译主机。

[0115] 在本公开的另一个示例性实施例中,编译主机向编译服务器发送所述依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称。

[0116] 编译服务器根据所述依赖文件的文件信息,从所述映射关系中查询所述依赖文件的路径信息,并根据所述路径信息获取所述依赖文件下发给编译主机。

[0117] 本示例性实施例提供的程序编译方法,编译服务器编译获得的Android源代码得到第二目录树,并获得第二目录树中各个文件与路径信息之间的映射关系。编译主机根据所述映射关系,从第二目录树中获得存储的待编译文件所依赖的依赖文件,从而避免编译

主机在本地编译才能获得所述依赖文件,节省了编译依赖文件占用的时间和资源。而且,所述编译主机无需在本地下载维护一套完整的Android源代码,只需下载本模块对应的源代码,减小了硬盘空间的占用。大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。

[0118] 由于Android源代码非常庞大,在Android源代码的系统研发时,通常将Android源代码系统项目分解为多个容易管理的模块,不同研发组负责不同的模块,因此,编译服务器需要获取各个模块的最新修改的Android源代码。图7是根据一示例性实施例示出的另一种程序编译方法的流程图,如图7所示,所述方法包括以下步骤:

[0119] 在步骤S610中,获取Android源代码。

[0120] 在步骤S620中,编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系。所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0121] 在步骤S710中,检测编译主机中的所述Android源代码是否被修改。

[0122] 在本公开的一个示例性实施例中,各个编译主机在编译过程中修改Android源代码后,向中心服务器上传修改后的Android源代码,编译服务器从中心服务器中存储的各个模块(目标文件)对应的Android源代码。

[0123] 可选地,编译主机可以在空闲状态时,向中心服务器上传修改后的Android源代码。

[0124] 编译服务器比较在本地存储的Android源代码与中心服务器中相对应的Android源代码是否相同,如果不同,确定Android源代码有更新。需要说明的是,比较同一版本同一模块在本地存储的Android源代码及中心服务器中存储的Android源代码是否相同。

[0125] 可选地,编译服务器可以直接从各个编译主机上获取各个模块(目标文件)对应的Android源代码。

[0126] 在步骤S720中,当检测到编译主机中的所述Android源代码被修改时,依据修改后的源代码,更新所述Android源代码。

[0127] 系统研发组对本模块的Android源代码编译研发的过程中,必然会修改本模块的Android源代码,因此,需要对应更新编译服务器中存储的Android源代码,以及更新第二目录树。

[0128] 在步骤S730中,编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

[0129] 然后,编译主机获得待编译文件所依赖的最新的依赖文件,并根据所述最新的依赖文件编译所述待编译文件得到最终的目标文件。

[0130] 本示例性实施例提供的程序编译方法,能够根据各个模块修改后的Android源代码更新第二目录树中相应的依赖文件,保证第二目录树中的依赖文件总是最新的依赖文件,进而将最新的依赖文件提供给对应的编译主机,使编译主机总是能从第二目录树中获得符合条件的依赖文件。

[0131] 相应于上述的程序编译方法,本公开还提供了程序编译装置实施例。

[0132] 图8是根据一示例性实施例示出的一种程序编译装置的框图,该装置应用于编译主机中,主要用于编译Android源代码,如图8所示,该装置包括第一查询单元110,第一获取

单元120和编译单元130。

[0133] 第一查询单元110被配置为从第一目录树中查询编译待编译文件所依赖的依赖文件。

[0134] 在本公开的一个示例性实施例中,所述第一查询单元110可以包括:第三获取子单元和第二查询子单元;

[0135] 第三获取子单元被配置为获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;

[0136] 第二查询子单元被配置为根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

[0137] 所述待编译文件为Android源代码,例如,目标为A,则对应的待编译文件为目标A对应的源代码。所述第一目录树可以由编译主机编译得到,也可以由编译服务器编译或其它的编译主机编译得到。

[0138] 第一获取单元120被配置为从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件。

[0139] 在本公开的一个示例性实施例中,所述第一获取单元可以包括:第一获取子单元和第二获取子单元;

[0140] 所述第一获取子单元被配置为从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息。

[0141] 所述第一获取子单元可以包括第一接收子单元和第一查询子单元;

[0142] 所述第一接收子单元被配置为接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系。

[0143] 所述第一查询子单元被配置为根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。

[0144] 所述第二获取子单元被配置为根据所述路径信息获取所述依赖文件。

[0145] 在本公开的另一个示例性实施例中,所述第一获取单元可以包括:发送单元和第二接收子单元;

[0146] 所述发送单元被配置为向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称。

[0147] 所述第二接收子单元被配置为接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

[0148] 编译单元130被配置为根据所述依赖文件编译所述待编译文件,得到目标文件。

[0149] 可选地,如图8所示,所述程序编译装置还包括:查找单元140。所述查找单元140被配置为在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,输出使所述第一获取单元执行相应动作的信号。

[0150] 本实施例提供的所述程序编译装置,利用第一查询单元从编译主机中的第一目录

树中,查询待编译文件依赖于哪些依赖文件,然后,通过第一获取单元从存储在编译服务器中的第二目录树中,获取所述依赖文件;再由编译单元利用所述依赖文件编译所述待编译文件,得到最终的目标文件,从而实现了在编译Android源代码时,能够依赖于除待编译文件所在的目录树之外的其它的目录树(第二目录树)。研发人员只需在本地保存所述待编译文件(本模块对应的源代码),无需下载维护一套完整的Android源代码,减小了硬盘空间的占用。而且,能够直接从第二目录树中获得符合条件的依赖文件,无需通过编译得到所述依赖文件,从而节省了编译依赖文件占用的时间和资源。因此,本公开提供的程序编译方法大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。

[0151] 关于上述实施例中的装置,其中各个单元执行操作的具体方式已经在有关该方法的实施例中进行了详细描述,此处将不做详细阐述说明。

[0152] 图9是根据一示例性实施例示出的另一种程序编译装置的框图,该程序编译装置应用于编译服务器,如图9所示,所述装置包括第一获取单元210和第一编译单元220;

[0153] 第一获取单元210被配置为获取Android源代码。

[0154] 第一编译单元220被配置为编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0155] 可选地,如图9所示,所述装置还可以包括检测单元230、源代码更新单元240和第二编译单元250。

[0156] 检测单元230被配置为检测编译主机中存储的Android源代码是否被修改。

[0157] 源代码更新单元240被配置为当检测到编译主机中的所述Android源代码被修改时,依据修改后的更新编译服务器中存储的Android源代码。

[0158] 第二编译单元250被配置为编译更新后的Android源代码,更新所述第二目录树中的各个依赖文件。

[0159] 可选地,图9所示的程序编译装置还可以包括第一发送单元和第一查找单元。

[0160] 所述第一发送单元被配置为将编译服务器中的所述映射关系发送给各个编译主机,由所述编译主机根据所述映射关系获取所述待编译文件所依赖的依赖文件的路径信息,并提供给所述编译服务器。

[0161] 所述第一查找单元被配置为根据编译主机提供的路径信息获取所述路径信息的存储空间中存储的依赖文件。

[0162] 可选地,图9所示的程序编译装置还可以包括接收单元,第二查找单元和第二发送单元。

[0163] 所述接收单元被配置为接收编译主机上传的所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

[0164] 所述第二查找单元被配置为根据编译服务器中存储的各个依赖文件与路径信息之间的映射关系,查找所述依赖文件对应的路径信息,并根据所述路径信息获取所述依赖文件。

[0165] 所述发送单元被配置为将获得的所述依赖文件下发给所述编译主机;所述依赖文件用于使编译主机编译得到所述待编译文件对应的目标文件。

[0166] 本实施例提供的程序编译装置,由第一获取单元获取Android源代码,利用第一编译单元编译所述Android源代码得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;然后,编译主机根据所述映射关系获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。实现了在编译Android源代码时,能够依赖于除待编译文件所在的目录树之外的其它的目录树(第二目录树)。避免编译主机在本地编译才能获得所述依赖文件,节省了编译依赖文件占用的时间和资源。而且,所述编译主机无需在本地下载维护一套完整的Android源代码,只需下载本模块对应的源代码,减小了硬盘空间的占用。大大减少了所需的硬盘空间及编译时间,提高了系统开发效率。

[0167] 图10是根据一示例性实施例示出的一种用于实现程序编译的装置800的框图。例如,装置800可以是移动电话,计算机,数字广播终端,消息收发设备,游戏控制台,平板设备,医疗设备,健身设备,个人数字助理等。

[0168] 参照图10,装置800可以包括以下一个或多个组件:处理组件802,存储器804,电源组件806,多媒体组件808,音频组件810,输入/输出(I/O)的接口812,传感器组件814,以及通信组件816。

[0169] 处理组件802通常控制装置800的整体操作,诸如与显示,电话呼叫,数据通信,相机操作和记录操作相关联的操作。处理元件802可以包括一个或多个处理器820来执行指令,以完成上述的方法的全部或部分步骤。此外,处理组件802可以包括一个或多个模块,便于处理组件802和其他组件之间的交互。例如,处理部件802可以包括多媒体模块,以方便多媒体组件808和处理组件802之间的交互。

[0170] 存储器804被配置为存储各种类型的数据以支持在设备800的操作。这些数据的示例包括用于在装置800上操作的任何应用程序或方法的指令,联系人数据,电话簿数据,消息,图片,视频等。存储器804可以由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器(SRAM),电可擦除可编程只读存储器(EEPROM),可擦除可编程只读存储器(EPROM),可编程只读存储器(PROM),只读存储器(ROM),磁存储器,快闪存储器,磁盘或光盘。

[0171] 电力组件806为装置800的各种组件提供电力。电力组件806可以包括电源管理系统,一个或多个电源,及其他与为装置800生成、管理和分配电力相关联的组件。

[0172] 多媒体组件808包括在所述装置800和用户之间的提供一个输出接口的屏幕。在一些实施例中,屏幕可以包括液晶显示器(LCD)和触摸面板(TP)。如果屏幕包括触摸面板,屏幕可以被实现为触摸屏,以接收来自用户的输入信号。触摸面板包括一个或多个触摸传感器以感测触摸、滑动和触摸面板上的手势。所述触摸传感器可以不仅感测触摸或滑动动作的边界,而且还检测与所述触摸或滑动操作相关的持续时间和压力。在一些实施例中,多媒体组件808包括一个前置摄像头和/或后置摄像头。当设备800处于操作模式,如拍摄模式或视频模式时,前置摄像头和/或后置摄像头可以接收外部的多媒体数据。每个前置摄像头和后置摄像头可以是一个固定的光学透镜系统或具有焦距和光学变焦能力。

[0173] 音频组件810被配置为输出和/或输入音频信号。例如,音频组件810包括一个麦克风(MIC),当装置800处于操作模式,如呼叫模式、记录模式和语音识别模式时,麦克风被配置为接收外部音频信号。所接收的音频信号可以被进一步存储在存储器804或经由通信组

件816发送。在一些实施例中,音频组件810还包括一个扬声器,用于输出音频信号。

[0174] I/O接口812为处理组件802和外围接口模块之间提供接口,上述外围接口模块可以是键盘,点击轮,按钮等。这些按钮可包括但不限于:主页按钮、音量按钮、启动按钮和锁定按钮。

[0175] 传感器组件814包括一个或多个传感器,用于为装置800提供各个方面的状态评估。例如,传感器组件814可以检测到设备800的打开/关闭状态,组件的相对定位,例如所述组件为装置800的显示器和小键盘,传感器组件814还可以检测装置800或装置800一个组件的位置改变,用户与装置800接触的存在或不存在,装置800方位或加速/减速和装置800的温度变化。传感器组件814可以包括接近传感器,被配置用来在没有任何的物理接触时检测附近物体的存在。传感器组件814还可以包括光传感器,如CMOS或CCD图像传感器,用于在成像应用中使用。在一些实施例中,该传感器组件814还可以包括加速度传感器,陀螺仪传感器,磁传感器,压力传感器或温度传感器。

[0176] 通信组件816被配置为便于装置800和其他设备之间有线或无线方式的通信。装置800可以接入基于通信标准的无线网络,如WiFi,2G或3G,或它们的组合。在一个示例性实施例中,通信部件816经由广播信道接收来自外部广播管理系统的广播信号或广播相关信息。在一个示例性实施例中,所述通信部件816还包括近场通信(NFC)模块,以促进短程通信。例如,在NFC模块可基于射频识别(RFID)技术,红外数据协会(IrDA)技术,超宽带(UWB)技术,蓝牙(BT)技术和其他技术来实现。

[0177] 在示例性实施例中,装置800可以被一个或多个应用专用集成电路(ASIC)、数字信号处理器(DSP)、数字信号处理设备(DSPD)、可编程逻辑器件(PLD)、现场可编程门阵列(FPGA)、控制器、微控制器、微处理器或其他电子元件实现,用于执行上述方法。

[0178] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储器804,上述指令可由装置800的处理器820执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0179] 一种非临时性计算机可读存储介质,当所述存储介质中的指令由移动终端的处理器执行时,使得移动终端能够执行一种程序编译方法,所述方法包括:

[0180] 从第一目录树中查询编译待编译文件所依赖的依赖文件;

[0181] 从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件;

[0182] 根据所述依赖文件编译所述待编译文件,得到目标文件。

[0183] 可选地,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

[0184] 从所述第二目录树中获取所述依赖文件在所述编译服务器中的路径信息;

[0185] 依据所述路径信息获取所述依赖文件。

[0186] 可选地,从所述第二目录树中,获取所述依赖文件的路径信息,采用如下方式:

[0187] 接收所述编译服务器发送的所述第二目录树中的各个依赖文件与路径信息之间的映射关系;

[0188] 根据所述映射关系,查询得到所述待编译文件所依赖的依赖文件对应的路径信息。



[0189] 可选地,从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖文件,采用如下方式:

[0190] 向所述编译服务器发送所述待编译文件所依赖的依赖文件的文件信息,所述文件信息至少包括所述依赖文件的文件名称;

[0191] 接收所述编译服务器根据所述依赖文件的文件信息获得的所述依赖文件;由编译服务器查询所述第二目录树中的各个依赖文件与路径信息之间的映射关系,得到的所述依赖文件的路径信息,并依据所述路径信息获得所述依赖文件;所述映射关系保存在所述编译服务器中。

[0192] 可选地,所述从第一目录树中查询编译待编译文件所依赖的依赖文件,采用如下方式:

[0193] 获取所述第一目录树中各个待编译文件的依赖关系,所述依赖关系用于存储各个待编译文件和各个所述待编译文件所依赖的依赖文件之间的对应关系;

[0194] 根据所述依赖关系,查询得到所述待编译文件所依赖的依赖文件。

[0195] 可选地,所述方法还包括:在所述第一目录树中查找是否存在所述待编译文件所依赖的依赖文件;当所述第一目录树中不存在所述待编译文件所依赖的依赖文件时,执行所述从编译服务器编译Android源代码得到的第二目录树中,获取所述依赖的步骤。

[0196] 图11是本公开实施例中的结构示意图。该服务器1900可因配置或性能不同而产生比较大的差异,可以包括一个或一个以上中央处理器(central processing units,CPU)1922(例如,一个或一个以上处理器)和存储器1932,一个或一个以上存储应用程序1942或数据1944的存储介质1930(例如一个或一个以上海量存储设备)。其中,存储器1932和存储介质1930可以是短暂存储或持久存储。存储在存储介质1930的程序可以包括一个或一个以上模块(图示没标出),每个模块可以包括对服务器中的一系列指令操作。更进一步地,中央处理器1922可以设置为与存储介质1930通信,在服务器1900上执行存储介质1930中的一系列指令操作。

[0197] 服务器1900还可以包括一个或一个以上电源1926,一个或一个以上有线或无线网络接口1950,一个或一个以上输入输出接口1958,一个或一个以上键盘1956,和/或,一个或一个以上操作系统1941,例如Windows Server™,Mac OS X™,Unix™,Linux™,FreeBSD™等等。

[0198] 在示例性实施例中,还提供了一种包括指令的非临时性计算机可读存储介质,例如包括指令的存储介质1930,上述指令可由服务器1900的处理器1922执行以完成上述方法。例如,所述非临时性计算机可读存储介质可以是ROM、随机存取存储器(RAM)、CD-ROM、磁带、软盘和光数据存储设备等。

[0199] 一种非临时性计算机可读存储介质,当所述存储介质中的指令由服务器的处理器执行时,使得服务器能够执行一种程序编译方法,所述方法包括:

[0200] 获取Android源代码;

[0201] 得到第二目录树,并保存所述第二目录树中的各个依赖文件与路径信息的映射关系;依据所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件,并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。

[0202] 可选地,所述方法还包括:

- [0203] 检测编译主机中存储的Android源代码是否被修改；
- [0204] 当检测到编译主机中存储的所述Android源代码被修改时，依据修改后的Android源代码，更新编译服务器中存储的Android源代码；
- [0205] 编译更新后的Android源代码，更新所述第二目录树中的各个依赖文件。
- [0206] 可选地，获取编译主机编译得到的各个目标文件对应的源代码，采用如下方式：
- [0207] 编译服务器从中心服务器中获取各个目标文件对应的Android源代码，所述Android源代码由编译主机将编译得到的目标文件对应的源代码上传至中心服务器；
- [0208] 或，
- [0209] 编译服务器从编译主机中获取各个目标文件对应的Android源代码。
- [0210] 可选地，所述方法还包括：
- [0211] 将所述映射关系发送给编译主机，所述映射关系用于使编译主机获取待编译文件所依赖的依赖文件的路径信息，根据所述路径信息获取所述依赖文件，并根据所述依赖文件编译第一目录树中的待编译文件得到目标文件。
- [0212] 可选地，所述方法还包括：接收编译主机发送的所述待编译文件所依赖的依赖文件的文件信息，所述文件信息至少包括所述依赖文件的文件名称；根据所述依赖文件的文件信息，查询所述映射关系获得所述依赖文件的路径信息。
- [0213] 本领域技术人员在考虑说明书及实践这里公开的发明后，将容易想到本公开的其它实施方案。本申请旨在涵盖本公开的任何变型、用途或者适应性变化，这些变型、用途或者适应性变化遵循本公开的一般性原理并包括本公开未公开的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的，本公开的真正范围和精神由下面的权利要求指出。
- [0214] 应当理解的是，本公开并不局限于上面已经描述并在附图中示出的精确结构，并且可以在不脱离其范围进行各种修改和改变。本公开的范围仅由所附的权利要求来限制。
- [0215] 需要说明的是，在本文中，诸如“第一”和“第二”等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。而且，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、物品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、物品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、物品或者设备中还存在另外的相同要素。

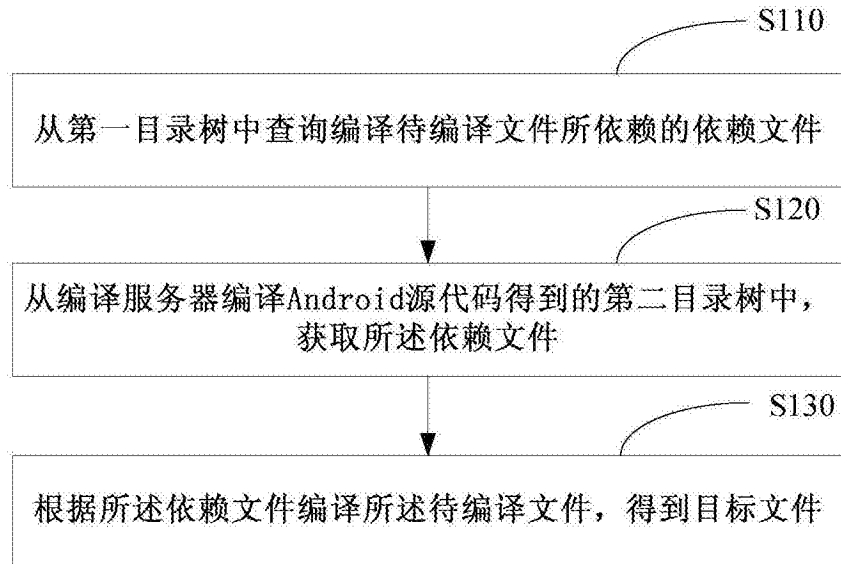


图1

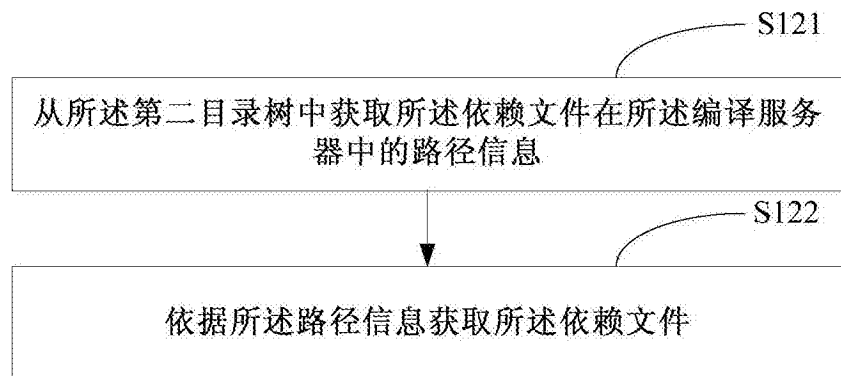


图2

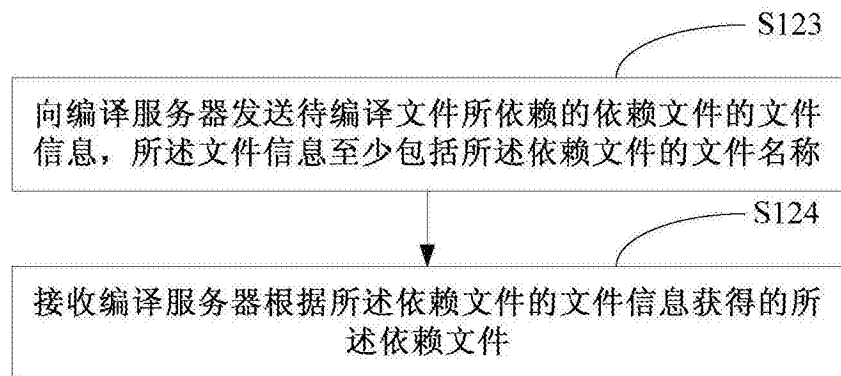


图3

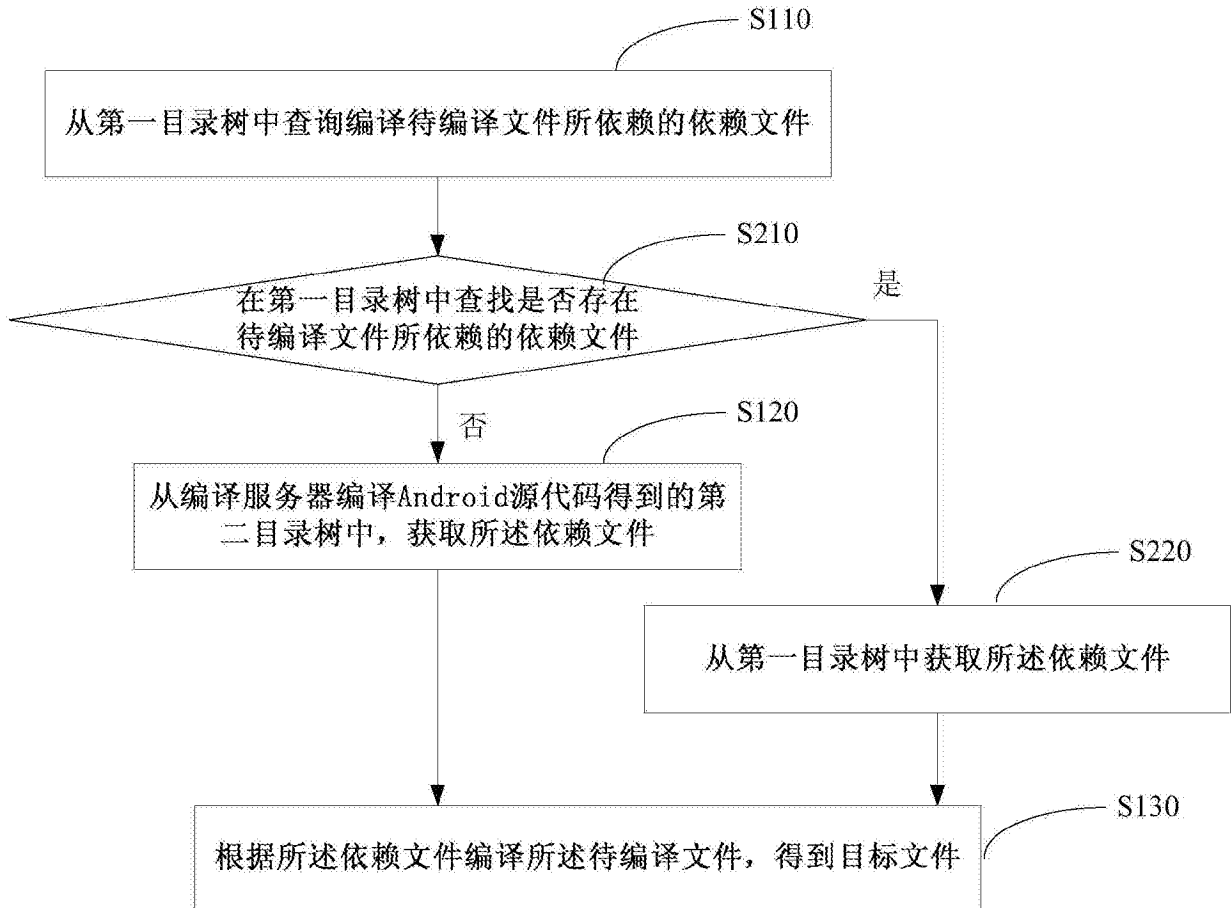


图4

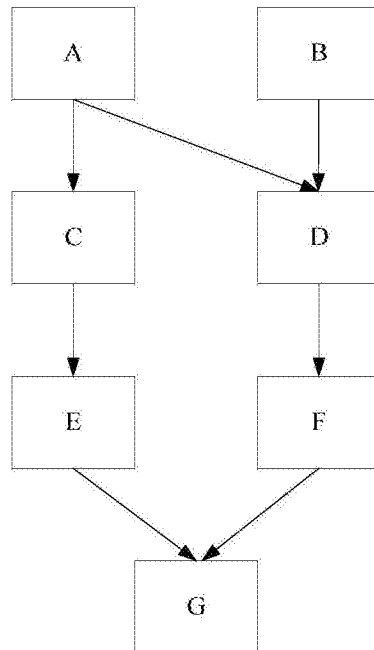


图5

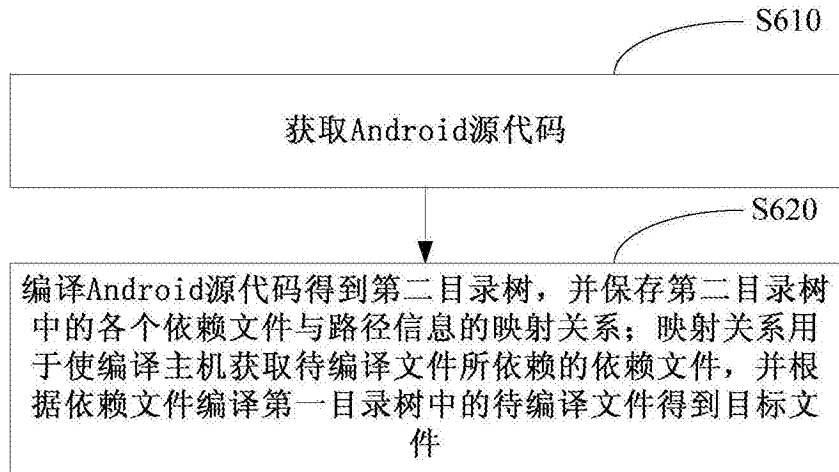


图6

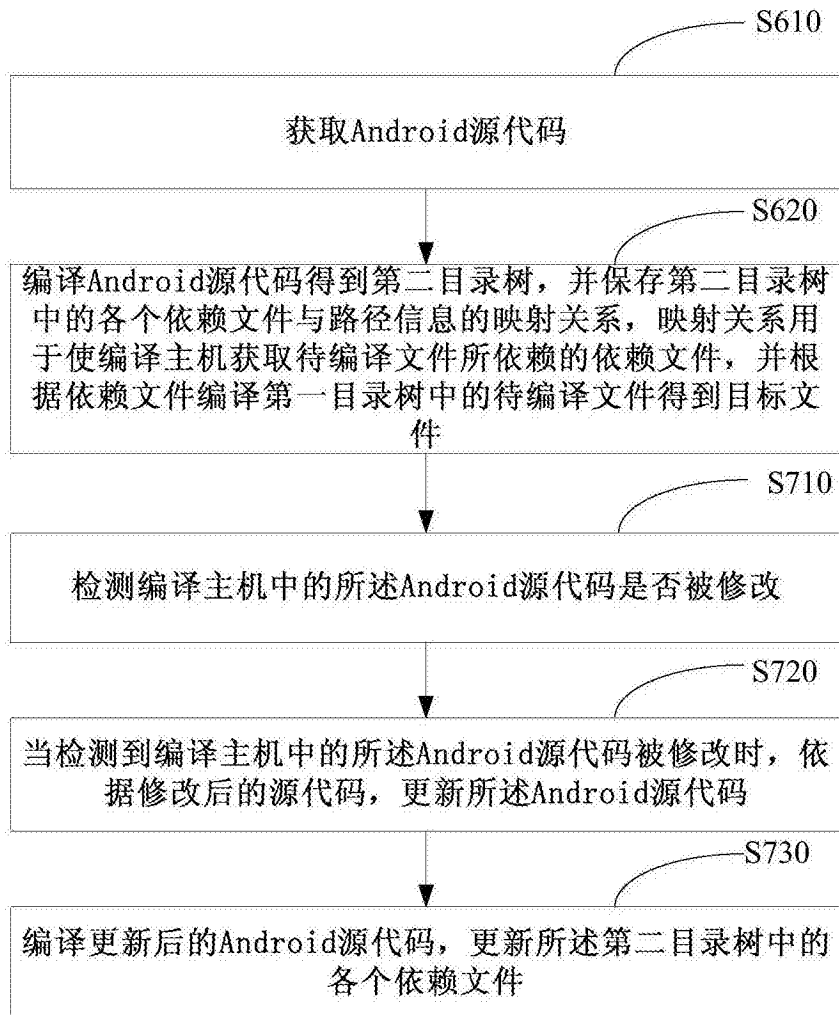


图7

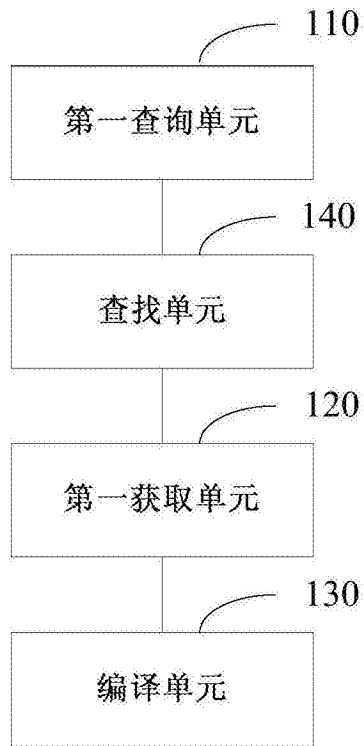


图8

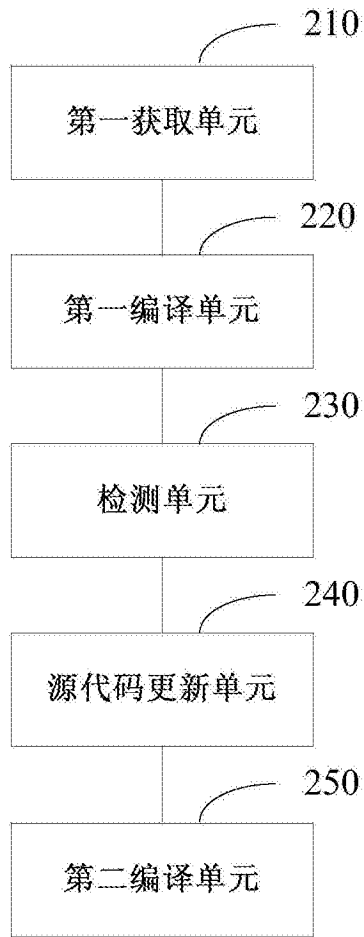


图9

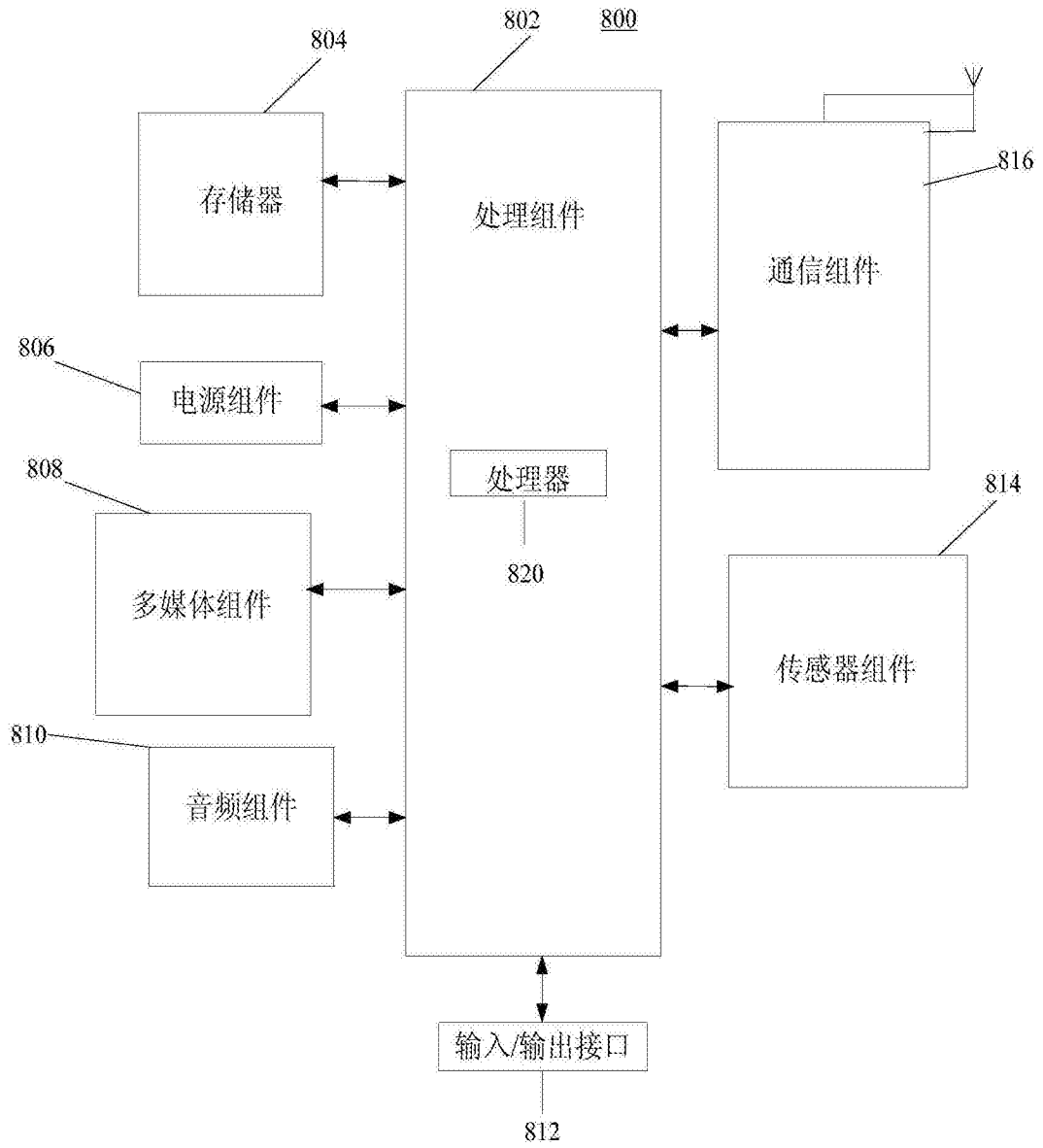


图10



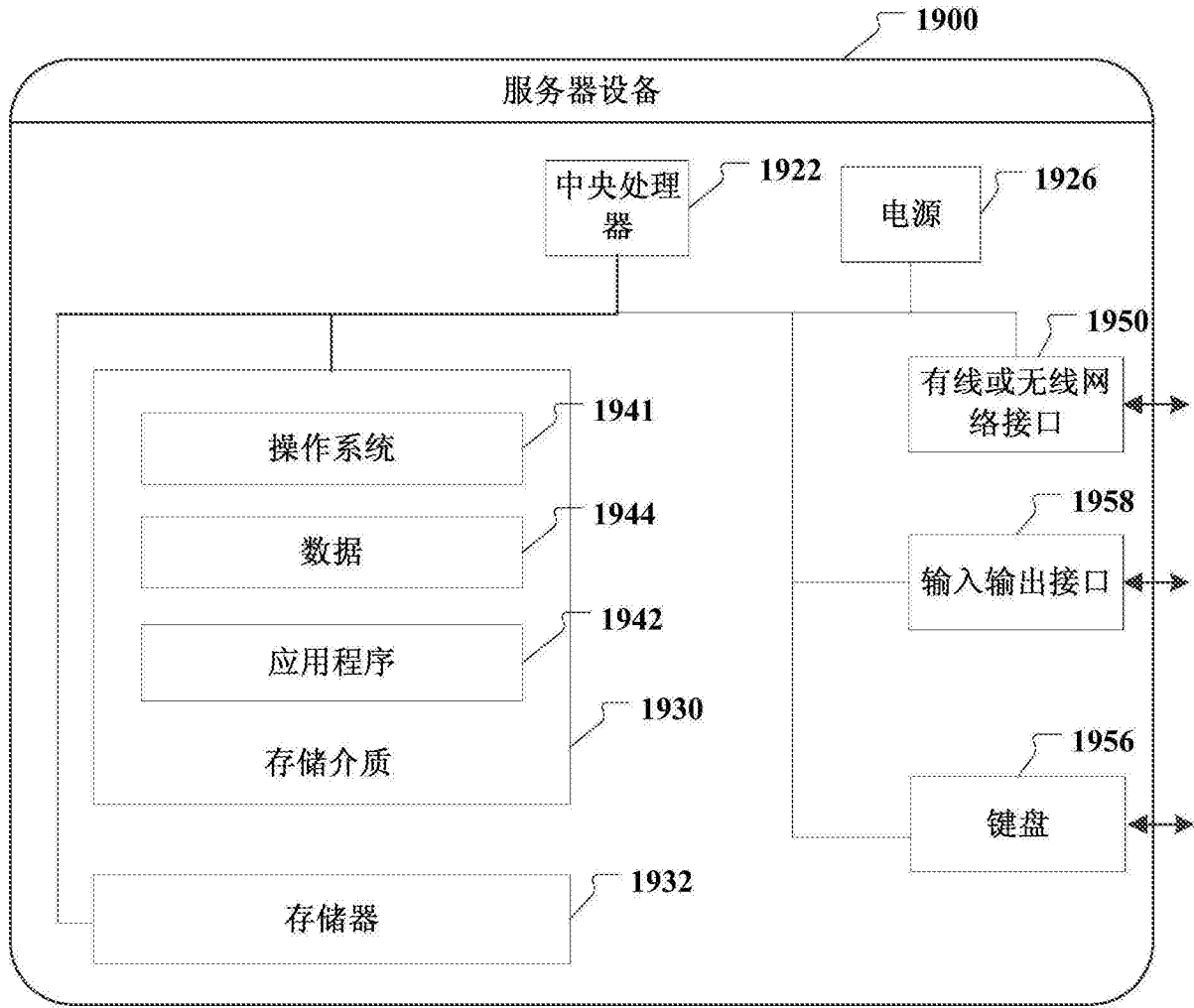


图11