



(12) 发明专利申请

(10) 申请公布号 CN 105183580 A

(43) 申请公布日 2015. 12. 23

(21) 申请号 201410250492. X

(22) 申请日 2014. 06. 06

(71) 申请人 西安中兴新软件有限责任公司
地址 710114 陕西省西安市高新区长安通讯
产业园东西四号路 1 号

(72) 发明人 段瑞楠

(74) 专利代理机构 北京派特恩知识产权代理有
限公司 11270
代理人 张颖玲 王黎延

(51) Int. Cl.

G06F 11/14(2006. 01)

G06F 9/445(2006. 01)

G06F 3/06(2006. 01)

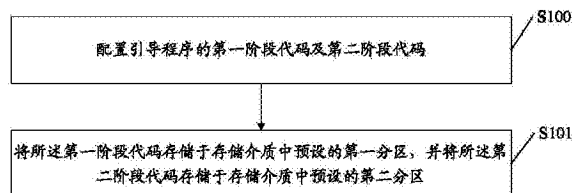
权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种引导程序的存储方法、故障恢复方法及
设备

(57) 摘要

本发明公开了一种引导程序的存储方法, 该方法包括: 配置引导程序的第一阶段代码及第二阶段代码; 其中, 所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作; 所述第二阶段代码用于加载内核到内存并运行; 将所述第一阶段代码存储于存储介质中预设的第一分区, 并将所述第二阶段代码存储于存储介质中预设的第二分区。本发明还同时公开了一种引导程序的故障恢复方法、存储设备及嵌入式设备。



1. 一种引导程序的存储方法,其特征在于,所述方法包括:

配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行;

将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

配置所述第二阶段代码的备份代码,得到第二阶段备份代码;

将所述第二阶段备份代码存储于存储介质中预设的第三分区。

3. 根据权利要求2所述的方法,其特征在于,所述方法还包括:

在所述第二阶段代码升级后,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

4. 一种引导程序的故障恢复方法,其特征在于,将第一阶段代码和第二阶段代码分别存储于存储介质的第一分区和第二分区中;所述方法还包括:

所述第一阶段代码在完成基本参数设置后,检测所述第二阶段代码是否故障;

检测到所述第二阶段代码故障时,确定是否存在第二阶段备份代码;

确定不存在所述第二阶段备份代码时,进入引导程序的下载模式。

5. 根据权利要求4所述的方法,其特征在于,所述方法还包括:

确定存在所述第二阶段备份代码时,检测所述第二阶段备份代码是否故障;

检测到所述第二阶段备份代码无故障时,利用所述第二阶段备份代码恢复故障的存储于第二分区中的第二阶段代码;

检测到所述第二阶段备份代码故障时,进入引导程序的下载模式。

6. 根据权利要求4或5所述的方法,其特征在于,所述方法还包括:

检测到所述第二阶段代码无故障时,确定是否存在所述第二阶段备份代码;

确定不存在所述第二阶段备份代码时,跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;

确定存在所述第二阶段备份代码时,检测所述第二阶段代码与所述第二阶段备份代码是否一致;

若检测到所述第二阶段代码与所述第二阶段备份代码一致,则跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;

若检测到所述第二阶段代码与所述第二阶段备份代码不一致,则确定存储于第二分区中的第二段代码已被升级,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

7. 一种存储设备,其特征在于,所述存储设备包括:配置单元、存储单元;其中,

所述配置单元,用于配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行;

所述存储单元,用于将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区。

8. 根据权利要求 7 所述的存储设备,其特征在于,所述配置单元,还用于配置所述第二阶段代码的备份代码,得到第二阶段备份代码;

所述存储单元,还用于将所述第二阶段备份代码存储于存储介质中预设的第三分区。

9. 根据权利要求 8 所述的存储设备,其特征在于,所述存储设备还包括:

更新单元,用于在所述第二阶段代码升级后,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

10. 一种嵌入式设备,其特征在于,所述嵌入式设备包括:配置单元、存储单元、检测单元、确定单元、故障恢复单元;其中,

所述配置单元,用于配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行;

所述存储单元,用于将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区;

所述检测单元,用于所述第一阶段代码在完成基本参数设置后,检测第二阶段代码是否故障;

所述确定单元,用于在检测单元检测到所述第二阶段代码故障时,确定是否存在第二阶段备份代码;

所述故障恢复单元,用于在确定单元确定不存在所述第二阶段备份代码时,进入引导程序的下载模式。

11. 根据权利要求 10 所述的嵌入式设备,其特征在于,所述配置单元,还用于配置所述第二阶段代码的备份代码,得到第二阶段备份代码;相应的,所述存储单元,还用于将所述第二阶段备份代码存储于存储介质中预设的第三分区;

所述检测单元,还用于检测所述第二阶段备份代码是否故障;

所述故障恢复单元,还用于在确定单元确定存在所述第二阶段备份代码,并在检测单元检测到所述第二阶段备份代码无故障时,利用所述第二阶段备份代码恢复故障的存储于第二分区中的第二阶段代码;在确定单元确定存在所述第二阶段备份代码,并在检测单元检测到所述第二阶段备份代码故障时,进入引导程序的下载模式。

12. 根据权利要求 10 或 11 所述的嵌入式设备,其特征在于,所述检测单元,还用于检测所述第二阶段代码与所述第二阶段备份代码是否一致;

所述嵌入式设备还包括:

执行单元,用于在检测单元检测到所述第二阶段代码无故障,并在确定单元确定不存在所述第二阶段备份代码时,跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;在检测单元检测到所述第二阶段代码无故障,并在确定单元确定存在所述第二阶段备份代码时,若所述检测单元检测到所述第二阶段代码与所述第二阶段备份代码一致,则跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;

更新单元,用于在检测单元检测到所述第二阶段代码无故障,并在确定单元确定存在所述第二阶段备份代码时,若所述检测单元检测到所述第二阶段代码与所述第二阶段备份代码不一致,则确定所述第二阶段代码已被升级,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

一种引导程序的存储方法、故障恢复方法及设备

技术领域

[0001] 本发明涉及嵌入式领域的引导程序管理技术,尤其涉及一种引导程序的存储方法、故障恢复方法及设备。

背景技术

[0002] UBoot 及 BootLoader 是用于多种嵌入式设备的中央处理器 (Central Processing Unit, CPU) 的引导程序,UBoot 是 BootLoader 升级后的版本;UBoot 不仅支持 Linux 系统,而且支持 VxWorks、NetBSD、LyxOS 等嵌入式操作系统。UBoot 可以支持的处理器包括:80×86、ARM、MIPS、XScale、PowerPC, UBoot 的主要功能是初始化硬件设备,待将软硬件运行环境设置好后,将操作系统内核从存储介质中载入内存,最终跳转到操作系统入口,将控制权移交给操作系统。

[0003] 空中接口固件更新 (Firmware Over The Air, FOTA) 是终端设备的固件升级空中下载技术,指终端设备通过云端设备管理 (Device Management, DM) 及下载协议,将固件升级包通过互联网下载到终端设备内部;然后,由终端设备内部的安装程序进行自升级,从而实现软件 Bug 的修复、新功能的扩展。

[0004] FOTA 技术升级通常采用差分升级的方式,差分升级的思想是对比两个软件版本,根据差分算法计算出两个软件版本的差异部分,通过对存储介质 Flash 块中差异页进行搬运、修改、擦除来完成固件升级,通过对文件进行随机读写完成文件系统的升级。

[0005] 发明人在实现本发明的过程中,发现现有引导程序升级的技术方案至少存在以下缺陷:

[0006] 1) 现有技术中,若升级引导程序时出现掉电,下次启动系统时会出现系统自举问题。

[0007] 2) 现有技术中,引导程序中的第一阶段代码和第二阶段代码是合并在一起存储于存储介质中的,这里,通常将用于完成基本参数设置和加载第二阶段代码到内存的代码称为第一阶段代码,将用于加载内核到内存并运行的代码称为第二阶段代码;若引导程序在运行或升级过程中出现了故障,系统将难以恢复,这种情况下,就需要进行存储介质拆卸,用烧片器进行软件重写或者用联合测试工作组 (Joint Test Action Group, JTAG) 接口进行写入,这样不但不安全,而且破坏了问题现场,为软件研发过程带来了许多不便。

发明内容

[0008] 有鉴于此,本发明实施例期望提供一种引导程序的存储方法、故障恢复方法及设备,能够使嵌入式系统的引导程序保持稳定运行,从而提高嵌入式系统的安全性。

[0009] 为达到上述目的,本发明的技术方案是这样实现的:

[0010] 本发明实施例提供了一种引导程序的存储方法,该方法包括:

[0011] 配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作;所述

第二阶段代码用于加载内核到内存并运行；

[0012] 将所述第一阶段代码存储于存储介质中预设的第一分区，并将所述第二阶段代码存储于存储介质中预设的第二分区。

[0013] 上述方案中，所述方法还包括：

[0014] 配置所述第二阶段代码的备份代码，得到第二阶段备份代码；

[0015] 将所述第二阶段备份代码存储于存储介质中预设的第三分区。

[0016] 上述方案中，所述方法还包括：

[0017] 在所述第二阶段代码升级后，利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0018] 本发明实施例提供了一种引导程序的故障恢复方法，将第一阶段代码和第二阶段代码分别存储于存储介质的第一分区和第二分区中；该方法还包括：

[0019] 所述第一阶段代码在完成基本参数设置后，检测所述第二阶段代码是否故障；

[0020] 检测到所述第二阶段代码故障时，确定是否存在第二阶段备份代码；

[0021] 确定不存在所述第二阶段备份代码时，进入引导程序的下载模式。

[0022] 上述方案中，所述方法还包括：

[0023] 确定存在所述第二阶段备份代码时，检测所述第二阶段备份代码是否故障；

[0024] 检测到所述第二阶段备份代码无故障时，利用所述第二阶段备份代码恢复故障的存储于第二分区中的第二阶段代码；

[0025] 检测到所述第二阶段备份代码故障时，进入引导程序的下载模式。

[0026] 上述方案中，所述方法还包括：

[0027] 检测到所述第二阶段代码无故障时，确定是否存在所述第二阶段备份代码；

[0028] 确定不存在所述第二阶段备份代码时，跳转到所述第二阶段代码的入口处，完成剩余的初始化操作；

[0029] 确定存在所述第二阶段备份代码时，检测所述第二阶段代码与所述第二阶段备份代码是否一致；

[0030] 若检测到所述第二阶段代码与所述第二阶段备份代码一致，则跳转到所述第二阶段代码的入口处，完成剩余的初始化操作；

[0031] 若检测到所述第二阶段代码与所述第二阶段备份代码不一致，则确定存储于第二分区中的第二段代码已被升级，利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0032] 基于上述方法，本发明实施例提供了一种存储设备，该存储设备包括：配置单元、存储单元；其中，

[0033] 所述配置单元，用于配置引导程序的第一阶段代码及第二阶段代码；其中，所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作；所述第二阶段代码用于加载内核到内存并运行；

[0034] 所述存储单元，用于将所述第一阶段代码存储于存储介质中预设的第一分区，并将所述第二阶段代码存储于存储介质中预设的第二分区。

[0035] 上述方案中，所述配置单元，还用于配置所述第二阶段代码的备份代码，得到第二阶段备份代码；

[0036] 所述存储单元,还用于将所述第二阶段备份代码存储于存储介质中预设的第三分区。

[0037] 上述方案中,所述存储设备还包括:

[0038] 更新单元,用于在所述第二阶段代码升级后,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0039] 基于上述方法,本发明实施例还提供了一种嵌入式设备,该嵌入式设备包括:配置单元、存储单元、检测单元、确定单元、故障恢复单元;其中,

[0040] 所述配置单元,用于配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对所述第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行;

[0041] 所述存储单元,用于将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区;

[0042] 所述检测单元,用于所述第一阶段代码在完成基本参数设置后,检测第二阶段代码是否故障;

[0043] 所述确定单元,用于在检测单元检测到所述第二阶段代码故障时,确定是否存在第二阶段备份代码;

[0044] 所述故障恢复单元,用于在确定单元确定不存在所述第二阶段备份代码时,进入引导程序的下载模式。

[0045] 上述方案中,所述配置单元,还用于配置所述第二阶段代码的备份代码,得到第二阶段备份代码;相应的,所述存储单元,还用于将所述第二阶段备份代码存储于存储介质中预设的第三分区;

[0046] 所述检测单元,还用于检测所述第二阶段备份代码是否故障;

[0047] 所述故障恢复单元,还用于在确定单元确定存在所述第二阶段备份代码,并在检测单元检测到所述第二阶段备份代码无故障时,利用所述第二阶段备份代码恢复故障的存储于第二分区中的第二阶段代码;在确定单元确定存在所述第二阶段备份代码,并在检测单元检测到所述第二阶段备份代码故障时,进入引导程序的下载模式。

[0048] 上述方案中,所述检测单元,还用于检测所述第二阶段代码与所述第二阶段备份代码是否一致;

[0049] 所述嵌入式设备还包括:

[0050] 执行单元,用于在检测单元检测到所述第二阶段代码无故障,并在确定单元确定不存在所述第二阶段备份代码时,跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;在检测单元检测到所述第二阶段代码无故障,并在确定单元确定存在所述第二阶段备份代码时,若所述检测单元检测到所述第二阶段代码与所述第二阶段备份代码一致,则跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;

[0051] 更新单元,用于在检测单元检测到所述第二阶段代码无故障,并在确定单元确定存在所述第二阶段备份代码时,若所述检测单元检测到所述第二阶段代码与所述第二阶段备份代码不一致,则确定所述第二阶段代码已被升级,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0052] 本发明实施例所提供的引导程序的存储方法、故障恢复方法及设备,配置引导程

序的第一阶段代码及第二阶段代码；其中，所述第一阶段代码用于完成基本参数设置、对第二阶段代码进行故障检测及根据检测结果执行相应操作；所述第二阶段代码用于加载内核到内存并运行；将所述第一阶段代码存储于存储介质中预设的第一分区，并将所述第二阶段代码存储于存储介质中预设的第二分区；如此，在引导程序运行或升级过程中，不会对存储介质中预设的第一分区进行频繁擦写，仅是对存储介质中预设的第二分区进行擦写，这样，能够缩小存储介质中出现问题的存储区域，从而延长存储介质的使用寿命。

[0053] 本发明实施例在第二阶段代码故障恢复时，若不存在所述第二阶段备份代码，可以进入引导程序的下载操作模式；若存在所述第二阶段备份代码且检测所述第二阶段备份代码无故障，可以利用存储于第三分区中的第二阶段备份代码恢复故障的存储于第二分区第二阶段代码；若存在所述第二阶段备份代码但检测所述第二阶段备份代码故障，可以进入引导程序的下载操作模式；如此，能够及时地对引导程序运行或升级过程中出现的故障进行恢复，从而提高嵌入式系统的安全性。

附图说明

- [0054] 图 1 为本发明实施例引导程序的存储方法实现流程示意图；
[0055] 图 2 为本发明实施例引导程序的故障恢复方法实现流程示意图；
[0056] 图 3 为本发明实施例存储设备的组成结构示意图；
[0057] 图 4 为本发明实施例嵌入式设备的组成结构示意图。

具体实施方式

[0058] 本发明实施例中，配置引导程序的第一阶段代码及第二阶段代码；其中，所述第一阶段代码用于完成基本参数设置、对第二阶段代码进行故障检测及根据检测结果执行相应操作；所述第二阶段代码用于加载内核到内存并运行；将所述第一阶段代码存储于存储介质中预设的第一分区，并将所述第二阶段代码存储于存储介质中预设的第二分区。

[0059] 下面结合附图和具体实施方式对本发明所述方法和装置作进一步说明。

[0060] 本发明实施例提出了一种引导程序的存储方法，如图 1 所示，该方法包括：

[0061] 步骤 S100：配置引导程序的第一阶段 stage1 代码及第二阶段 stage2 代码。

[0062] 这里，引导程序可以为 BootLoader、Uboot，也可以为更高级版本的引导程序。

[0063] 这里，引导程序通常包括启动加载模式和下载模式，而且允许用户在这两种工作模式之间进行切换；其中，启动加载模式，也称为“自主引导方式”，也即引到程序从目标机的存储介质上将操作系统加载到内存随机存取存储器 (Random Access Memory, RAM) 中并引导运行，整个过程并没有用户的介入，该引导方式是引导程序的正常工作模式下普遍采用的一种引导方式，因此，在嵌入式设备发布的时候，引到程序一般以该引导方式对内核代码进行启动引导；下载模式，该引导方式下目标机上的引导程序将通过串口连接或网络连接等通信手段从主机下载文件，如下载应用程序、数据文件、内核映像等，从主机下载的文件通常首先被引导程序保存到目标机的内存 RAM 中，然后再被引导程序写到目标机上的存储介质中，最后完成内核的引导运行。

[0064] 一般，引导程序在启动加载模式下分为：stage1 和 stage2，stage1 代码通常用汇编语言实现，包含以下几个基本步骤：

- [0065] 1) 基本参数设置；
- [0066] 2) 为 stage2 准备内存 RAM 空间；
- [0067] 3) 将 stage2 从存储介质中读取到内存 RAM 空间。
- [0068] 4) 设置好堆栈；
- [0069] 5) 跳转到 stage2 的入口处。

[0070] Stage2 代码通常用 C 语言实现,包含以下几个基本步骤：

- [0071] 1) 初始化本阶段需要用到硬件；
- [0072] 2) 检测系统的内存映象；
- [0073] 3) 将内核和根文件系统从存储介质上加载到内存 RAM 空间；
- [0074] 4) 为内核设置启动参数；
- [0075] 5) 调用内核。

[0076] 本步骤中,配置的所述 stage1 代码用于完成基本参数设置、对所述 stage2 代码进行故障检测及根据检测结果执行相应的操作；其中,基本参数,例如,中央处理器 (Central Processing Unit, CPU) 参数、内存参数以及一些基本的输入 / 输出 (Input/Output, I/O) 控制参数；配置的所述 stage2 代码用于完成剩余的初始化操作,即：加载内核到内存并运行。

[0077] 步骤 S101：将所述 stage1 代码存储于存储介质中预设的第一分区,并将所述 stage2 代码存储于存储介质中预设的第二分区。

[0078] 通常,嵌入式设备中的存储介质采用 NAND Flash。

[0079] 这里,将所述 stage1 代码及所述 stage2 代码在存储介质中分区进行存储,在采用 FOTA 技术对引导程序升级时,所述 stage1 代码是固定不变的,永远不被升级,仅升级可被升级的所述 stage2 代码,因此,在引导程序升级过程中,不会对存储介质中预设的第一分区进行频繁擦写,仅是对存储介质中预设的第二分区进行擦写,这样,能够缩小存储介质中出现问题的存储区域,从而延长存储介质的使用寿命；并且在引导程序运行或升级的过程中,若第一阶段代码检测到第二阶段代码故障,则直接进入引导程序的下载模式,这样,可以挽救嵌入式系统,从而提高嵌入式系统的安全性。

[0080] 优选的,可以对所述 stage2 代码进行备份得到第二阶段备份 stage2backup 代码,并将所述 stage2backup 代码存储于存储介质中预设的第三分区；在引导程序运行或升级过程中,认为所述 stage1 代码肯定是不出现故障的,在所述 stage2 代码出现故障时,可以利用存储于第三分区中的所述 stage2backup 代码恢复故障的 stage2 代码,如此,能够及时地对引导程序运行或升级过程中出现的故障进行恢复,从而提高嵌入式系统的安全性。

[0081] 进一步地,在所述 stage2 代码升级后,利用升级后的 stage2 代码更新存储于第三分区中的 stage2backup 代码。

[0082] 进一步地,将故障的 stage2 代码存储于存储介质中预设的第四分区,以便后续根据故障的 stage2 代码排查问题,且方便问题的定位。

[0083] 基于上述方法,本发明实施例提出了一种引导程序的故障恢复方法,将第一阶段代码和第二阶段代码分别存储于存储介质的第一分区和第二分区中；如图 2 所示,该方法包括：

[0084] 步骤 S200：第一阶段 stage1 代码在完成基本参数设置后,检测第二阶段 stage2 代码是否故障,若检测所述 stage2 代码故障,则转入步骤 S201；若检测所述 stage2 代码无

故障,则转入步骤 S204。

[0085] 步骤 S201:确定是否存在 stage2backup 代码,确定存在第二阶段备份代码时,转入步骤 S202;确定不存在 stage2backup 代码时,进入引导程序的下载模式。

[0086] 步骤 S202:检测所述 stage2backup 代码是否故障,检测到所述 stage2backup 代码无故障时,转入步骤 S203;检测到所述 stage2backup 代码故障时,进入引导程序下载模式。

[0087] 步骤 S203:利用 stage2backup 代码恢复故障的存储于第二分区中的 stage2 代码,转入步骤 S200。

[0088] 步骤 S204:确定是否存在 stage2backup 代码,确定不存在 stage2backup 代码时,跳转到 stage2 代码的入口处,完成剩余的初始化操作;确定存在 stage2backup 代码时,转入步骤 S205。

[0089] 步骤 S205:检测 stage2 代码与 stage2backup 代码是否一致,若检测到 stage2 代码与 stage2backup 代码一致,则跳转到 stage2 代码的入口处,完成剩余的初始化操作;若检测到 stage2 代码与 stage2backup 代码不一致,则转入步骤 S206。

[0090] 步骤 S206:确定 stage2 代码已被升级,利用升级后的 stage2 代码更新存储于第三分区中的 stage2backup 代码。

[0091] 具体的,在步骤 S200 中,所述 stage1 代码在完成基本参数设置后,将所述 stage2 代码从存储介质中读取到内存,采用预设的校验码生成算法生成所述 stage2 代码的校验码,将生成的校验码与预存的 stage2 校验码进行对比,从而检测所述 stage2 代码是否故障,若生成的校验码与预存的 stage2 校验码不相同,则检测所述 stage2 代码故障,转入步骤 S201;若生成的校验码与预存的 stage2 校验码相同,则检测所述 stage2 代码无故障,转入步骤 S204。

[0092] 在步骤 S201 中,确定是否存在 stage2backup 代码,确定存在第二阶段备份代码时,转入步骤 S202;确定不存在 stage2backup 代码时,进入引导程序的下载模式。

[0093] 在步骤 S202 中,采用预设的校验码生成算法生成所述 stage2backup 代码的校验码,将生成的校验码与预存的 stage2backup 校验码进行对比,从而检测所述 stage2backup 代码是否故障,若生成的校验码与预存的 stage2backup 校验码不相同,则检测所述 stage2backup 代码故障,进入引导程序下载模式;若生成的校验码与预存的 stage2backup 校验码相同,则检测到所述 stage2backup 代码无故障时,转入步骤 S203。

[0094] 在步骤 S203 中,利用 stage2backup 代码恢复故障的存储于第二分区中的 stage2 代码,即:将存储于第三分区中的 stage2backup 代码写入故障的 stage2 代码对应的第二分区中,这样能够及时地进行纠错,并对故障的 stage2 代码进行恢复;优选的,将 stage2backup 代码写入故障的 stage2 代码对应的第二分区中后,可以在预先定义的标志位写入表示“引导程序需要重新运行或升级”的标志,当检测到表示“引导程序需要重新运行或升级”的标志时,转入步骤 S200,重新发起引导程序运行或升级流程,并清除该标志。

[0095] 在步骤 S204 中,确定是否存在 stage2backup 代码,确定不存在 stage2backup 代码时,跳转到 stage2 代码的入口处,完成剩余的初始化操作;确定存在 stage2backup 代码时,转入步骤 S205。

[0096] 在步骤 S205 中,首先采用与所述 stage2 相同的校验码生成算法生成所述

stage2backup 代码的校验码,确定所述 stage2backup 代码无故障时,将所述 stage2backup 的校验码与所述 stage2 的校验码进行对比,若所述 stage2backup 的校验码与所述 stage2 的校验码相同,则检测到 stage2 代码与 stage2backup 代码一致,跳转到 stage2 代码的入口处,完成剩余的初始化操作;若所述 stage2backup 的校验码与所述 stage2 的校验码不相同,则检测到 stage2 代码与 stage2backup 代码不一致,此时确定 stage2 代码已被升级,转入步骤 S206。

[0097] 在步骤 S206 中,利用升级后的 stage2 代码更新存储于第三分区中的 stage2backup 代码,即:将升级后的 stage2 代码写入 stage2backup 代码对应的第三分区中。

[0098] 具体实施中,所述 stage2 代码的故障可能是由于在升级 stage2 代码时掉电造成的,在下次启动系统时,首先运行引导程序中的 stage1 代码,对 stage2 代码进行检测,检测到 stage2 代码故障时,能够及时地恢复并挽救所述系统,从而避免了现有技术中出现的系统自举问题;所述 stage2 代码的故障也可能是由对存储介质中的第二分区频繁擦写,造成第二分区出现坏块引起的。

[0099] 具体实施中,所述校验码生成算法可以是外部不可逆推,碰撞成本比较高的算法,比如经过盐化处理的信息摘要算法 MD5。校验操作对性能要求比很高,可以设计适当的校验码生成算法,优选的,可以设计具有一定加密强度的校验码生成算法,这里对校验码生成算法不作具体限定。

[0100] 进一步地,检测到所述 stage2 代码故障后,将故障的 stage2 代码写入存储介质中预设的第四分区,以便后续根据故障的 stage2 代码排查问题。

[0101] 本发明实施例中,将所述 stage1 代码及所述 stage2 代码在存储介质中分区进行存储,即:将所述 stage1 代码存储于存储介质中预设的第一分区,并将所述 stage2 代码存储于存储介质中预设的第二分区;在引导程序运行或升级时,所述 stage1 代码是固定不变的,永远不被升级,仅升级可被升级的所述 stage2 代码,因此,在引导程序运行或升级过程中,不会对存储介质中预设的第一分区进行频繁擦写,仅是对存储介质中预设的第二分区进行擦写;这样,在所述 stage2 代码出现故障时,可以直接进入引导程序的下载模式,能够及时地挽救系统,从而提高嵌入式系统的安全性。

[0102] 在上述一种优选的实施方式中,对所述 stage2 代码进行备份得到 stage2backup 代码,并将所述 stage2backup 代码存储于存储介质中预设的第三分区,在引导程序运行或升级时,认为所述 stage1 代码肯定是不会出现故障的,所述 stage2 代码和所述 stage2backup 代码由软件逻辑保证某一个时间点其中一个是无故障的或完整的,在所述 stage2 代码出现故障时,可以利用存储于第三分区中的无故障的 stage2backup 代码恢复存储于第二分区中的故障的 stage2 代码;如此,能够及时地对引导程序运行或升级过程中出现的故障进行恢复,从而提高嵌入式系统的安全性;

[0103] 进一步地,在所述 stage2 代码出现故障时,若 stage2backup 代码也出现故障,则进入引导程序的下载模式,此时也能够及时地挽救系统。

[0104] 本发明实施例不仅可以应用于引导程序的升级场景,也可以应用于确保引导程序完整性的任意场景。

[0105] 为实现上述方法,本发明实施例还提供了一种存储设备、嵌入式设备,由于该存储

设备、嵌入式设备解决问题的原理与方法相似,因此,存储设备、嵌入式设备的实施过程及实施原理均可以参见前述方法的实施过程及实施原理描述,重复之处不再赘述。

[0106] 如图 3 所示,本发明实施例提供的存储设备,包括:配置单元 300、存储单元 301;其中,

[0107] 所述配置单元 300,用于配置引导程序的第一阶段代码及第二阶段代码;

[0108] 所述存储单元 301,用于将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区;

[0109] 其中,所述第一阶段代码用于完成基本参数设置、对第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行。

[0110] 为了描述的方便,以上所述存储设备的各部分以功能分为各种模块或单元分别描述。以上功能单元或模块的划分方式仅为本发明实施例给出的一种优选实现方式,功能单元或模块的划分方式不构成对本发明的限制。

[0111] 具体实施中,所述配置单元 300,还用于配置所述第二阶段代码的备份代码,得到第二阶段备份代码;

[0112] 所述存储单元 301 还用于:

[0113] 将所述第二阶段备份代码存储于存储介质中预设的第三分区。

[0114] 具体实施中,所述存储设备还包括:

[0115] 更新单元 302,用于在所述第二阶段代码升级后,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0116] 如图 4 所示,本发明实施例提供的嵌入式设备,包括:配置单元 300、存储单元 301、检测单元 400、确定单元 401、故障恢复单元 402;其中,

[0117] 所述配置单元 300,用于配置引导程序的第一阶段代码及第二阶段代码;其中,所述第一阶段代码用于完成基本参数设置、对第二阶段代码进行故障检测及根据检测结果执行相应操作;所述第二阶段代码用于加载内核到内存并运行;

[0118] 所述存储单元 301,用于将所述第一阶段代码存储于存储介质中预设的第一分区,并将所述第二阶段代码存储于存储介质中预设的第二分区;

[0119] 所述检测单元 400,用于在第一阶段代码在完成基本参数设置后,检测第二阶段代码是否故障;

[0120] 所述确定单元 401,用于在检测单元 400 检测到所述第二阶段代码故障时,确定是否存在第二阶段备份代码;

[0121] 所述故障恢复单元 402,用于在确定单元 401 确定不存在所述第二阶段备份代码时,进入引导程序的下载操作模式。

[0122] 为了描述的方便,以上所述存储设备的各部分以功能分为各种模块或单元分别描述。以上功能单元或模块的划分方式仅为本发明实施例给出的一种优选实现方式,功能单元或模块的划分方式不构成对本发明的限制。

[0123] 具体实施中,所述配置单元 300,还用于配置所述第二阶段代码的备份代码,得到第二阶段备份代码;相应的,所述存储单元 301,还用于将所述第二阶段备份代码存储于存储介质中预设的第三分区;

[0124] 所述检测单元 400,还用于检测所述第二阶段备份代码是否故障;

[0125] 所述故障恢复单元 402,还用于在确定单元 401 确定存在所述第二阶段备份代码,并在检测单元 400 检测到所述第二阶段备份代码无故障时,利用所述第二阶段备份代码恢复故障的存储于第二分区中的第二阶段代码;在确定单元 401 确定存在所述第二阶段备份代码,并在检测单元 400 检测到所述第二阶段备份代码故障时,进入引导程序的下载操作模式。

[0126] 具体实施中,所述检测单元 400,还用于检测所述第二阶段代码与所述第二阶段备份代码是否一致;

[0127] 所述嵌入式设备还包括:

[0128] 执行单元 403,用于在检测单元 400 检测到所述第二阶段代码无故障,并在确定单元 401 确定不存在所述第二阶段备份代码时,跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;在检测单元 400 检测到所述第二阶段代码无故障,并在确定单元 401 确定存在所述第二阶段备份代码时,若所述检测单元 400 检测到所述第二阶段代码与所述第二阶段备份代码一致,则跳转到所述第二阶段代码的入口处,完成剩余的初始化操作;

[0129] 更新单元 302,用于在检测单元 400 检测到所述第二阶段代码无故障,并在确定单元 401 确定存在所述第二阶段备份代码时,若所述检测单元 400 检测到所述第二阶段代码与所述第二阶段备份代码不一致,则确定存储于第二分区中的第二段代码已被升级,利用升级后的第二阶段代码更新存储于第三分区中的第二阶段备份代码。

[0130] 在实际应用中,所述配置单元 300、所述存储单元 301、所述更新单元 302 可由位于所述存储设备的中央处理器 (CPU)、微处理器 (MPU)、数字信号处理器 (DSP)、或现场可编程门阵列 (FPGA) 实现;所述配置单元 300、所述存储单元 301、所述更新单元 302、所述检测单元 400、所述确定单元 401、所述故障恢复单元 402、所述执行单元 403 可由位于所述嵌入式设备的中央处理器 (CPU)、微处理器 (MPU)、数字信号处理器 (DSP)、或现场可编程门阵列 (FPGA) 实现。

[0131] 本发明所述的方法并不限于具体实施方式中所述的实施例,本领域技术人员根据本发明的技术方案得出其它的实施方式,同样属于本发明的技术创新范围。

[0132] 显然,本领域的技术人员可以对本发明进行各种改动和变型而不脱离本发明的精神和范围。这样,倘若本发明的这些修改和变型属于本发明权利要求及其等同技术的范围之内,则本发明也意图包含这些改动和变型在内。

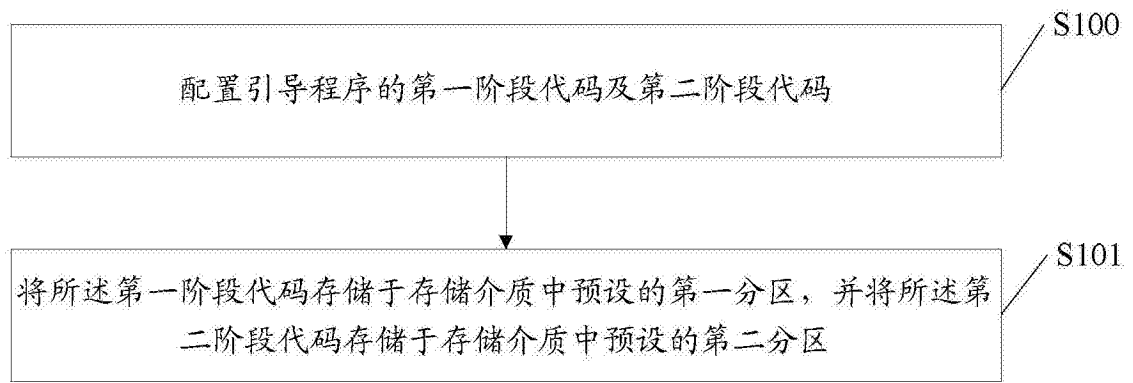


图 1

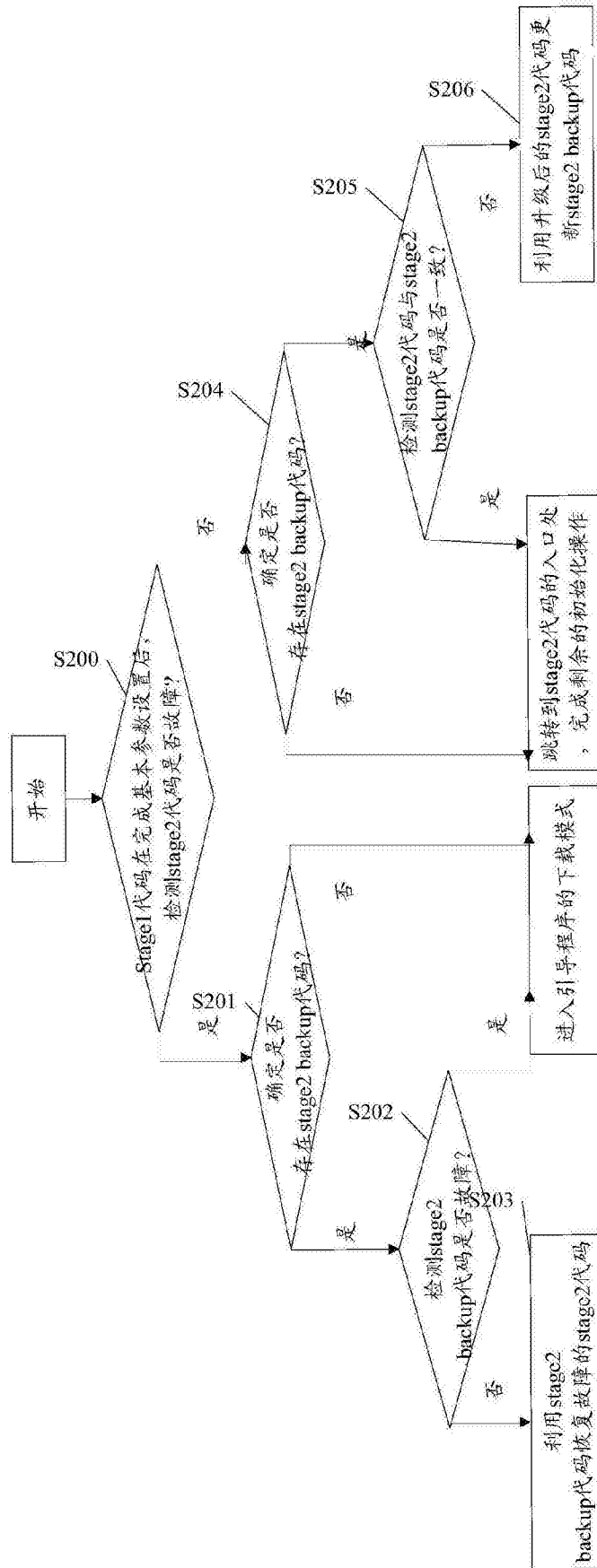


图 2

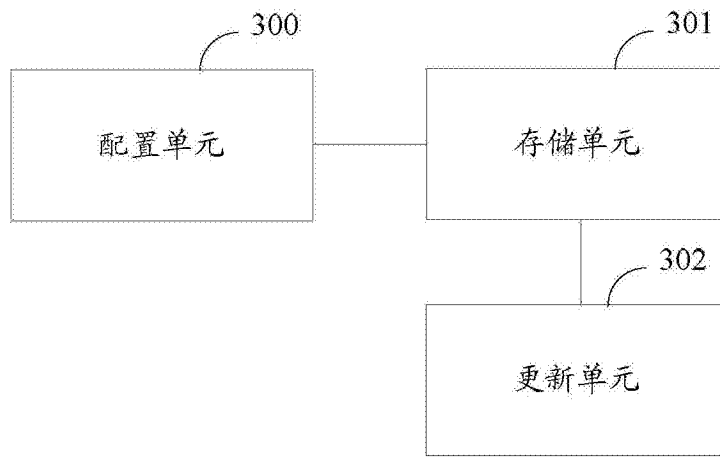


图 3

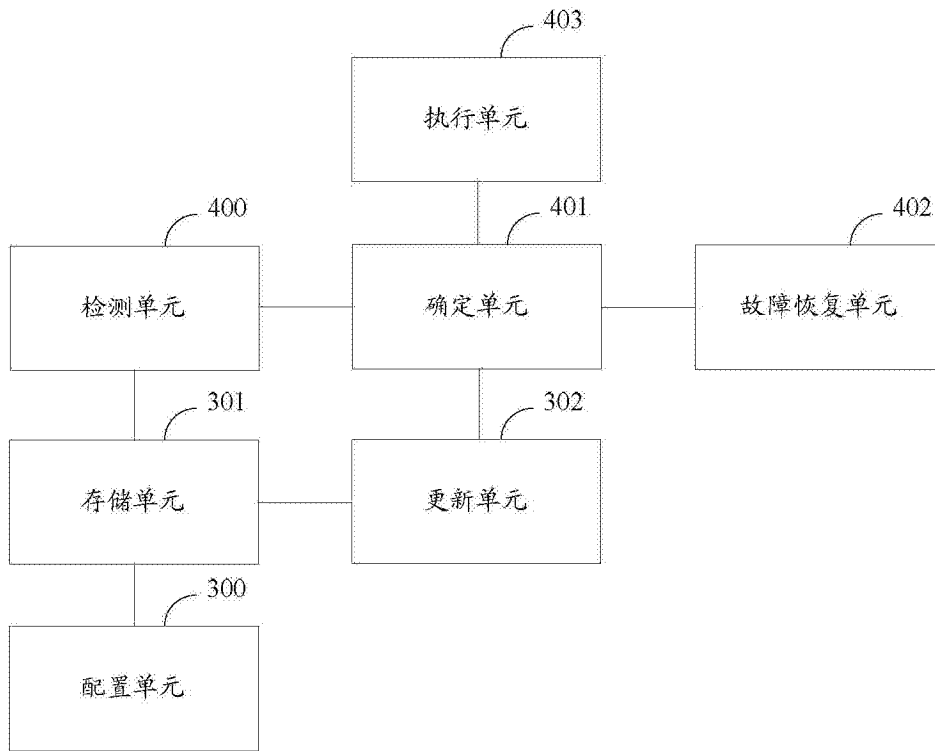


图 4