

Linux 系统下远方电能量数据终端配置文件备份及还原^①



蒋世挺, 陈思有, 胡 建

(浙江华云信息科技有限公司, 杭州 310012)
通讯作者: 蒋世挺, E-mail: 7154070@qq.com

摘 要: 随着科技的发展, 电力系统中智能设备的应用场景也变得越来越, 特别是在电力自动化控制和数据采集通信方面, 很多设备平台都采用基于嵌入式的 Linux 操作系统, 以远方电能量数据终端 (ERTU) 为例, 因为其运行过程中需要与各种电能表通信, 又需要支持不同主站系统的数据召唤, 所以各类通信参数就需要以配置文件的形式预置到设备中, 如果设备发生故障, 后续恢复过程中配置文件可能就需要进行重新配置, 不但增加工作量, 而且参数前后一致性也会存在问题. 如果利用 Linux 系统下的脚本程序和外接存储器, 简单快捷的实现配置文件的自动备份与还原, 那么就可以节省现场工程运维工作量, 提升整个工程应用的管理水平.

关键词: Linux; MD5 码; 外接存储器; 远方电能量数据终端; 脚本程序

引用格式: 蒋世挺, 陈思有, 胡建. Linux 系统下远方电能量数据终端配置文件备份及还原. 计算机系统应用, 2021, 30(6): 112-117. <http://www.c-s-a.org.cn/1003-3254/7954.html>

Backup and Restore of Configuration File of Remote Electric Energy Data Terminal in Linux System

JIANG Shi-Ting, CHEN Si-You, HU Jian

(Zhejiang Huayun Information Technology Co. Ltd., Hangzhou 310012, China)

Abstract: Amid the progress in science and technology, the application scenarios of intelligent devices in a power system have been increasing, especially in power automation as well as data acquisition and communication. Many platforms rely on an embedded Linux operating system. An Energy Remote Terminal Unit (ERTU) is taken as an example for analysis. Communication parameters should be preset into the device in the form of configuration files, since ERTU needs to communicate with electric energy meters and supports data calling of different master station systems during operation. If the device fails, the configuration files may require reconfiguration in the subsequent recovery process, increasing the workload and affecting the consistency of parameters. If the automatic backup and restore of configuration files are realized simply and quickly with the script program and external memory in a Linux system, then the workload of field engineering operation and maintenance can be reduced, improving the management level of the whole engineering application.

Key words: Linux; MD5Code; external memory; ERTU; script program

随着科技的发展, 电力系统中智能设备的应用也变得越来越, 特别是在电力自动化控制和数据采集通信方面. 远方电能量数据终端^[1,2]是电力市场技术支

持系统电量计量模块中的厂站端部分. 具有对电能量 (电能累计量) 采集、数据处理、分时存贮、长时间保存、远方传输等功能的设备. 它与电能量计费主站构

① 收稿时间: 2020-09-30; 修改时间: 2020-11-02; 采用时间: 2020-11-17; csa 在线出版时间: 2021-06-01

成电能量计费系统运用于各级调度结算中心对远方电量信息的采集和处理。作为系统中数据采集和上送的关键一环,其硬件平台性能也得到了很大的提升,普遍采用基于嵌入式的 Linux 操作系统^[3,4],其在变电站和电厂的运行数量也变得越来越多。在运行维护的过程中,因为每个变电站和电厂的运行环境不同,导致每个设备运行的参数配置文件^[2]也是各种各样,而一旦装置发生故障,后台维护工具无法连接上运行设备,就需要更换主板或重新升级初始化程序,这时候就需要重新配置系统运行参数文件,一般情况下,要么是工程人员从头开始重新配置,要么就是联系以前配置的人员或向公司询问是否有这个厂站的配置文件备份。无论是哪种方法,都会浪费许多时间,也不能保证其他人手上的备份文件就一定是最新的参数配置文件,从而导致二次调试的工作量发生,提高了运维成本。

本文介绍了在远方电能量数据终端中利用 Linux 系统下的脚本程序和外接存储器,如何简单快捷的实现终端配置文件自动备份与还原的方法,此技术方案的实施可以节省现场工程运维工作量,提升整个工程应用的管理水平。

1 总体思路

远方电能量数据终端采用 Linux 操作系统,同时硬件上设计外接存储器接口,装置运行时插上外接存储器,无论装置是初始化参数配置还是运行时修改参数配置,系统都会把有变动的配置文件自动备份到外接存储器上,如果终端在运行过程中发生故障,后续的故障恢复更换终端时,只要把故障终端的外部存储器拔下,替换到新的终端上,新终端就可以自动运行,不需要重新配置运行参数。

上述过程的实现,除了远方电能量数据终端采用 Linux 操作系统和外接存储器外,配置文件的备份和还原还需要编写相应的功能软件来处理,为了功能软件简单便捷,这个配置文件的备份和还原过程可以采用脚本程序^[5]来完成。因为脚本程序无需编译,就可以在 Linux 系统下运行,所以脚本程序后期的编辑修改和部署都非常容易。工程运维人员可以在研发人员的指导下,用文本编译器就能完成脚本程序的修改和功能扩充。

功能软件总体上可以考虑用脚本语言编写两个脚本文件,分别命名为“脚本一”和“脚本二”,“脚本一”解

决系统启动后,是否根据策略把外接存储器里的备份文件同步到配置文件目录的问题,“脚本二”解决系统运行过程中配置参数改变后,是否把最新配置文件重新备份到外部存储器的问题。然后在系统启动项里添加执行两个脚本的命令,这样等系统启动后就自动实现了远方电能量数据终端与外接存储器之间配置文件的自动更新以及重备份,保证了系统运行和后期运维过程中运行参数的持续一致性。

2 具体实现

从总体思路可以看出,远方电能量数据终端上要实现配置文件的自动备份和还原功能,主要有3部分工作,一是终端硬件平台的选择,需要能方便的支持外部存储器的扩展设计。二是为终端定制合适的 Linux 操作系统。三是设计编写脚本程序。

2.1 硬件平台的选择

近年来,随着国家电网“三型两网”战略的提出,电力系统中很多数据感知层和传输层的智能终端性能都有了很好的提升,ARM 芯片因为高性能、低功耗的优点,已经被广泛应用于电力系统的各类终端平台上。

ARM-Cortex-A9^[5]处理器能与其他 Cortex 系列处理器技术兼容,因此能够很好延用包括操作系统/实时操作系统(OS/RTOS)、中间件及应用在内的丰富生态系统,从而减少采用全新处理器所需的成本。通过首次利用关键微体系架构方面的改进,Cortex-A9 处理器提供了具有高扩展性和高功耗效率的解决方案。本文实例远方电能量数据终端就是采用基于 Cortex-A9 简单双核的飞思卡尔处理器作为硬件平台的核心。

SD^[5]卡是一种基于半导体快闪记忆器的新一代记忆设备,由于它体积小、数据传输速度快、可热插拔等优良的特性,被广泛地应用于便携式设备。因为基于 ARM-Cortex-A9 芯片硬件平台能很好的支持 SD 卡扩展,所以远方电能量数据终端的外接存储器就采用 SD 卡来实现。

远方电能量数据终端的硬件除了上述的核心板和外接存储器外,当然还有包括电源、存储、显示、通信等其它模块,因为本文主要是介绍智能终端配置文件的自动备份和还原方法,所以远方电能量数据终端其它硬件组成不作详述。

2.2 操作系统定制与 SD 卡挂载

适用于嵌入式终端平台的 Linux 操作系统可以根

据终端硬件平台的特性进行适当的裁剪修改,使之能在嵌入式计算机系统下稳定运行.其特点是版权免费,而且性能优异,软件移植容易,有许多公开的代码可以参考和移植,实时性能稳定、安全性好.

如图1所示, Linux 操作系统要能在硬件平台上顺利启动运行,主要做到以下两点工作,一是需要制作 UBoot 引导程序,二是需要定制 Kernel (内核)、文件系统.做到这两点后,操作系统才能够正常启动,应用程序和服务才能在操作系统上顺利运行.

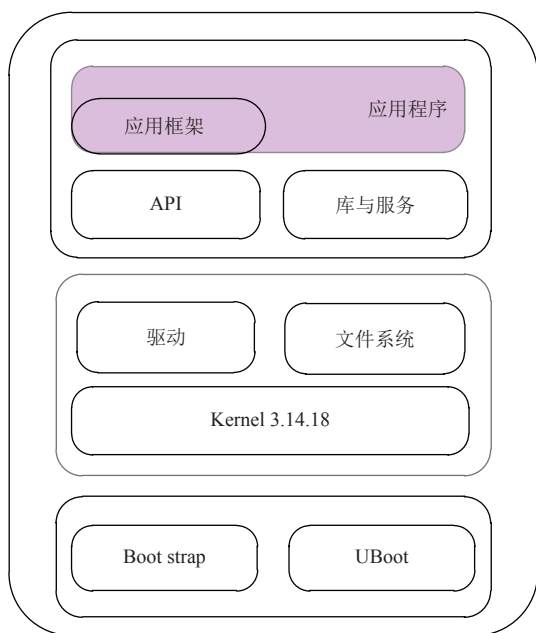


图1 Linux 操作系统架构图

一般采用飞思卡尔芯片的核心板^[6],核心板厂家会提供 u-boot zImage rootfs 三个已经制作好的文件,只要按要求把几个文件烧写到核心板上, Linux 操作系统就能够顺利启动.如果需要自己动手来制作这几个文件,也可以到飞思卡尔的官网上下载符合自己开发板的相关源码,然后在 PC 机上的 Linux 开发环境下利用 ARM 的交叉编译器编译生成,当然实际开发中还会有一些根据自己板卡特点修改源码配置的工作,都可以参考飞思卡尔官方手册.

本文实例远方电能量数据终端,其采用的 Linux 操作系统的内核版本为 Kernel3.14.18,系统主分区文件系统类型为 EXT4,外接存储器 SD 卡因为可插拔特性,为了和 PC 机交互方便,分区文件系统类型采用 FAT32.在嵌入式 Linux 系统内核中因为已经选择了

对 SD 卡支持的相关驱动,所以系统启动后就能自动识别到 SD 卡.

要让应用程序可以对 SD 卡进行读写操作,接下来还需要挂载 SD 卡.通过调试终端用 ls 命令带上参数 (/dev) 查看文件系统设备目录,就能看到 SD 卡的设备文件名为 mmcblk2p1,如何无法确认这个文件名就是 SD 卡,也可以用分区表查看命令 fdisk 带上参数 (-l) 查看分区表信息,我们得到结果如图2所示.也再次印证了 SD 卡的文件名为 mmcblk2p1.然后在文件系统里用 mkdir 命令创建一个备份文件目录/mnt/dst,再在系统启动脚本/etc/rc.local 里添加 mount 命令带参数 (-tvfat /dev/mmcblk2p1 /mnt/dst) 把 SD 卡挂载到备份文件目录了,应用程序也就可以对 SD 卡进行读写操作.

```
Disk /dev/mmcblk2: 14.9 GiB, 15931539456 bytes, 31116288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device           Boot Start      End Sectors  Size Id Type
/dev/mmcblk2p1   8192 31116287 31108096 14.9G  c W95 FAT32 (LBA)
```

图2 SD 卡分区信息

2.3 脚本程序设计实现

因为远方电能量数据终端采用了 Linux 操作系统,终端中配置文件自动备份和还原的软件功能,可以选用 Shell 命令脚本程序来实现^[7,8]. Shell 是一门编程语言,即 Shell 脚本, Shell 是解释执行的脚本语言,可直接调用 Linux 命令.选用 Shell 来实现脚本程序,是因为有以下几个优点:

- (1) Shell 的语法和结构比较简单,易于掌握;
- (2) 学习和使用也比较方便,上手比较容易;
- (3) 常用命令也比较好记,不清楚的情况下,可以通过在 Linux 系统下 man 命令迅速找到相关的说明;
- (4) Shell 是解释型语言,运行之前不需要编译.

在 Shell 脚本程序的设计过程中,判断文件之间是否相同,会用到 MD5 码, MD5 码是一种被广泛使用的密码散列函数产生出来的一个 128 位的散列值,用于确保信息传输完整一致. Linux 系统里有特有的命令 md5sum 来计算每个文件的 MD5 码,文件内容不同所产生的 MD5 码就不同.

明确了脚本程序的开发语言和所要用到的关键技术后,接下来就可以设计两个脚本程序的实现流程,如

图3所示,“脚本一”启动后会先检测SD卡内是否有备份配置文件,如果没有,就直接结束程序.如果有备份配置文件,就判断这个备份配置文件与应用程序配置

文件的MD5码是否相同,如果不相同,则将SD内的备份配置文件拷贝到应用程序配置文件目录进行更新替换,如果相同,就不作处理.

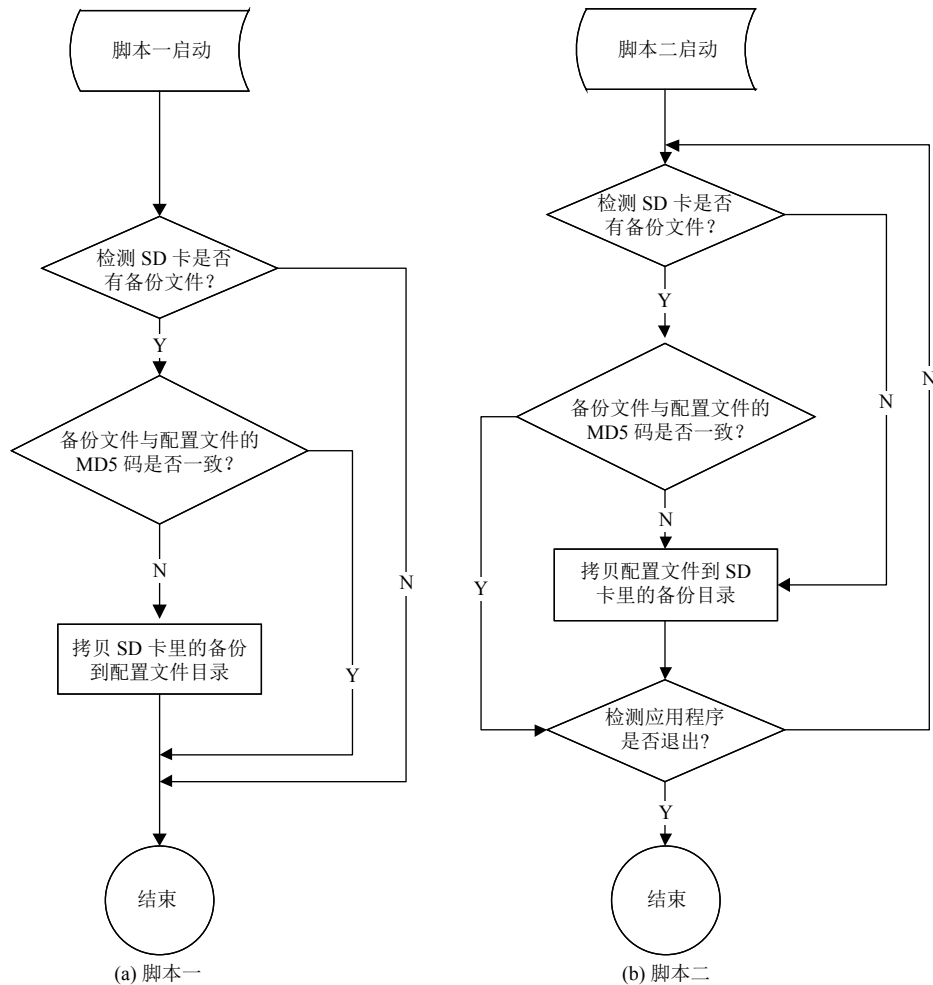


图3 两个脚本程序的实现流程

同样,“脚本二”启动后也会先检测SD内是否有备份配置文件,如果没有,就会拷贝终端的配置文件到SD卡备份,然后检测终端的应用程序是否退出来判断结束自身运行还是再循环.另外,如果检测到SD卡里有备份文件,就判断这个备份配置文件与应用程序配置文件的MD5码是否相同,如果不相同,就将应用程序的配置文件拷贝到外接存储设备内进行备份更新,如果相同,就检测终端的应用程序是否退出来判断结束自身运行还是再循环.

明确了脚本程序的流程,最后就是脚本程度的具体实现,“脚本一”主要功能是把外接存储器里的备份文件同步到配置文件目录,其代码如下:

```

#!/bin/sh
##备份配置路径
cfgpath="/mnt/dst/CFG"
mountpath="/mnt/dst/"
##检测外接存储卡
disknum=$(fdisk -l | grep mmcblk2p1 | wc -l)
mountnum=$(mount | grep mmcblk2p1 | wc -l)
if [ $disknum -ne 1 -o $mountnum -ne 1 ]
then
exit 0
fi
if [ -d $cfgpath ]
  
```



```

then
##检测配置是否为有效配置
cd $mountpath
devnum=0
if [ -f "CFG/ERTU_MG_000.xml" ]
then
devnum=$(cat CFG/ERTU_MG_000.xml | grep
dev_code | wc -l)
fi
if [ $devnum -le 0 ]
then
exit 0
fi
##获取备份配置的 MD5 信息
bakuplist=$(find CFG -path CFG/BAKUP -prune -o
-path CFG/BAKUP102 -prune -o -path CFG/autot* -
prune \-o -type f ! -iname ".*" -a ! -iname datadellist.txt -
print0 | xargs -0 md5sum | sort)
bakupinfo=$(echo $ bakuplist | sed s/[[:space:]]//g)
##获取当前有效配置的 MD5 信息
cd /ERTU/
cfglist=$(find CFG -path CFG/BAKUP -prune -o -
path CFG/BAKUP102 -prune -o -path CFG/autot* -prune \
-o -type f ! -iname ".*" -a ! -iname datadellist.txt -print0 |
xargs -0 md5sum | sort)
cfginfo=$(echo $ cfglist | sed s/[[:space:]]//g)
##判断 MD5, 决定是否执行恢复配置操作
if [ "$bakupinfo" != "$cfginfo" ]
then
mv /ERTU/CFG /ERTU/CFGtmp
cp -a $ cfgpath /ERTU/
rm -Rf /ERTU/CFGtmp
chmod -R 755 /ERTU/CFG/DISP
else
echo "do nothing"
fi
else
echo "do nothing"
exit 0
fi
“脚本二”主要功能是把最新配置文件重新备份到

```

外部存储器, 其代码如下:

```

#!/bin/sh
##备份配置路径
bakpath="/mnt/dst/CFGBak"
cfgpath="/mnt/dst/CFG"
mountpath="/mnt/dst/"
##检测外接存储卡
disknum=$(fdisk -l | grep mmcblk2p1 | wc -l)
mountnum=$(mount | grep mmcblk2p1 | wc -l)
if [ $ disknum -ne 1 -o $ mountnum -ne 1 ]
then
exit 0
fi
while [ true ]
do
if[-f"/ERTU/CFG/ERTU_MG_000.xml" ]
then
devnum=$(cat /ERTU/CFG/ERTU_MG_000.xml |
grep dev_code | wc -l)
if [ $devnum -gt 0 ]
then
##获取备份配置的 MD5 信息
bakupinfo="info"
if [ -d $cfgpath ]
then
cd $mountpath
bakuplist=$(find CFG -path CFG/BAKUP -prune -o
-path CFG/BAKUP102 -prune -o -path CFG/autot* -
prune \-o -type f ! -iname ".*" -a ! -iname datadellist.txt -
print0 | xargs -0 md5sum | sort)
bakupinfo=$(echo $ bakuplist | sed s/[[:space:]]//g)
fi
##获取当前有效配置的 MD5 信息
cd /ERTU/
cfginfo=$bakupinfo
if [ -f "/ERTU/CFG/ERTU_MG_000.xml" ]
then
cfglist=$(find CFG -path CFG/BAKUP -prune -o -
path CFG/BAKUP102 -prune -o -path CFG/autot* -prune \-
o -type f ! -iname ".*" -a ! -iname datadellist.txt -print0 |
xargs -0 md5sum | sort)

```

```

cfginfo=$(echo $cfglist | sed s/[[:space:]]//g)
fi
##判断 MD5
if [ "$bakupinfo" != "$cfginfo" ]
then
##延时判断当前配置 MD5
sleep 2
cfginfo1="info1"
if [ -f "/ERTU/CFG/ERTU_MG_000.xml" ]
then
cfglist1=$(find CFG -path CFG/BAKUP -prune -o -
path CFG/BAKUP102 -prune -o -path CFG/autot* -prune
\ -o -type f ! -iname ".*" -a ! -iname datadellist.txt -print0 |
xargs -0 md5sum | sort)
cfginfo1=$(echo $cfglist1 | sed s/[[:space:]]//g)
fi
##根据 MD5 执行相应操作
if [ "$cfginfo" = "$cfginfo1" ]
then
if [ -d $cfgpath ]
then
if [ -d $bakpath ]
then
rm -Rf $bakpath
fi
mv $cfgpath $bakpath
fi
cp -a /ERTU/CFG $cfgpath
else
echo "wating"
fi
else
echo "do nothing"
fi
fi
##检测应用程序

```

```

mgflag=$(ps -e | grep ERTU_MG | grep -v \<defunct>
| wc -l)
if [ $mgflag -le 0 ]
then
exit 0
fi
sleep 60
do

```

3 结论与展望

在 Linux 操作系统下,借助外部存储器来实现远方电能量数据终端配置文件的自动备份和还原,是一种提高设备运维效率的可行办法,特别是采用 Shell 命令脚本程序,无需编译器编译就可以在 Linux 操作系统下运行,为后期脚本程序的部署和推广都打下了很好的基础。目前应用了此项技术的远方电能量数据终端已经在浙江和华东地区的厂站内试点运行,在工程实施和调试阶段都取得了良好的效果,随着电力系统中使用 Linux 操作系统的智能终端越来越多,应用场景越来越广,此项终端配置文件的自动备份和还原技术应该会有很好的推广价值。

参考文献

- 章坚民,蒋世挺,金乃正,等.基于 IEC 61850 的变电站电能量采集终端的建模与实现.电力系统自动化,2010,34(11):67-71.
- 刘洪波,章坚民,楼海彪,等.Web 技术在变电站数字化 ERTU 配置中的应用.计算机系统应用,2013,22(8):141-144,135. [doi: 10.3969/j.issn.1003-3254.2013.08.029]
- 金伟正.嵌入式 Linux 系统开发与应用.北京:电子工业出版社,2011.
- 徐士强.基于 ARM9 的嵌入式 Linux 系统的研究与应用[硕士学位论文].南京:南京邮电大学,2012.
- 百度百科. <https://baike.baidu.com/item/SD%E5%8D%A1/122767?fr=aladdin>. [2020-09-04].
- Freescale. i. MX 6 BSP Porting Guide. Freescale, 2014.
- 金泰龙. Linux shell 脚本编程入门.金香,译.北京:人民邮电出版社,2014.
- Blum R, Bresnahan C. Linux 命令行与 shell 脚本编程大全.武海峰,译.2版.北京:人民邮电出版社,2012.