

MICAPS 数据文件存储系统的设计与实现

卢绍宗, 鲁玲 (北海市气象局, 广西北海 536000)

摘要 按照气象台站存储管理 MICAPS 数据文件、构建本地气象数据中心、促进气象业务发展的要求, 基于模块化设计原理, 设计了一种 MICAPS 数据文件存储系统。系统应用面向对象技术、线程技术、.NET 框架等编程实现 MICAPS 数据文件的自动化存储, 对资料进行逐日压缩备份。系统包括自动压缩备份模块和手动压缩备份模块, 在自定义类中调用 WinRAR, 然后在主程序中调用该类将 MICAPS 数据文件压缩为 RAR 文件, 既节省服务器存储空间, 又方便日后使用和管理。系统采用 C# 编程实现, 在用户层面和结构层面都有良好的表现。

关键词 MICAPS; 数据文件; 存储; 压缩备份

中图分类号 S126 文献标识码 A 文章编号 0517-6611(2015)31-362-03

Design and Implementation on Storage System of MICAPS Data Files

LU Shao-zong, LU Ling (Beihai Meteorological Bureau, Beihai, Guangxi 536000)

Abstract Based on the requirements for storage management of MICAPS data files, construction of local meteorological data center and promoting development of meteorological business, according to the modular design principles, a storage system of MICAPS data files is designed, the system takes advantage of object-oriented programming, threading technology and .NET Framework to achieve the automatic storage of MICAPS data files. The system can compress and backup files daily, including automatic and manual compress and backup modules, the system is devised to call WinRAR in self-defining class. The main program then call the class to compress MICAPS data files to RAR files. This can save disk space of the server. It is convenient for using and managing the files in the future. The system is implemented by C# programming language and proved to be good in the levels of user and structure.

Key words MICAPS; Data files; Storage; Compress and backup

随着气象现代化事业的发展, 对气象业务信息化也提出了新的要求, 构建本地气象数据中心是一项重要的工作, 可进一步实现气象资料自动化存储和数据共享, 提高气象业务数据的利用率, 促进业务发展。

MICAPS 系统是气象台站中短期天气预报业务的主要工作平台^[1]。目前中国气象局新一代卫星广播系统 CMACast 大幅增加了气象资料广播的种类和数量, 地市级气象台每日接收的卫星广播数据超过 200 G, 生成的 MICAPS 产品数据文件超过 20 G。由于本地 MICAPS 服务器存储空间有限, 需要定时删除数据文件以保证系统的正常运行。在专用服务器上及时备份 MICAPS 产品数据文件, 保证资料的完整, 积累气象档案, 构建本地业务数据应用环境, 是提高天气预报准确率、增强气象台站业务、科技服务和科研能力的基础, 对提高工作效率也有很大帮助^[2]。为实现 MICAPS 产品数据文件自动化存储模式并保障数据安全, 笔者设计了一种 MICAPS 数据文件存储系统, 应用面向对象编程技术、线程技术、.NET 框架等实现 MICAPS 服务器产品数据文件在另一台专用服务器上的自动化存储, 逐日定时分类压缩过去 24 小时的 MICAPS 产品数据文件。该存储专用服务器上的压缩文件可以通过网络共享或 FTP 下载的方式实现对气象业务及科研工作的支持。

1 系统设计

1.1 系统功能设计 对系统功能进行设计时按照“高内聚、低耦合”的原则, 采用模块化设计, 将系统划分为 2 个功能模块, 即自动压缩备份 MICAPS 数据文件模块和手动压缩备份 MICAPS 数据文件模块。使用 WinRAR 创建 RAR 格式的压

缩文件能提供较好的压缩率, 并允许物理受损的数据的恢复。RAR 格式能锁定重要的压缩文件, 防止意外更改。此外, RAR 格式可以管理的文件大小几乎是没限制的。因此, 系统采用 WinRAR 软件压缩保存 MICAPS 产品数据文件, 便于日后对文件的管理和维护。

正常情况下, 系统在每天设定的时间分别对前 24 h 的 8 个 MICAPS 产品数据目录 surface、high、physic、fy2、japan、japan_thin、ecmwf、ecmwf_thin 下的文件进行自动压缩备份, 生成 8 个 RAR 文件。当自动备份失败时, 可以运行手动备份程序完成相关目录文件的备份。

自动压缩备份 MICAPS 数据文件模块工作流程见图 1。

1.2 系统配置 在存储专用服务器上安装运行 MICAPS 数据文件存储系统软件。该服务器型号为 DELL PowerEdge R730, 磁盘阵列级别设为 RAID5, 安装 Windows Server 2008, 存储空间约为 5.4 TB, 使用 NTFS 磁盘格式并选择压缩驱动器以节约磁盘空间。该服务器与本地 MICAPS 服务器位于同一个局域网内。

2 系统实现

系统开发平台选择 Visual Studio 2010 集成开发环境, 在 .NET 框架下, 使用面向对象的程序设计语言 C# 进行开发。

2.1 自动压缩备份 MICAPS 数据文件模块 自动压缩备份 MICAPS 数据文件模块基于定时器触发运行。在存储专用服务器上本地 MICAPS 服务器产品数据目录映射为“X:\”, 在系统中将待备份文件目录设置为“X:\”。在该模块中设计有 2 个类, 分别完成 2 个功能。Syslog 类实现写日志功能, CWinRAR 类实现文件压缩备份功能。在模块中声明 3 个全局变量 format1、zippath1 和 myThread。将字符型变量 format1 初始化为“yyyyMMdd”, 用于表示日期格式。zippath1 是一个 int 类型的一维数组, 该数组中包含 8 个元素, 初始化

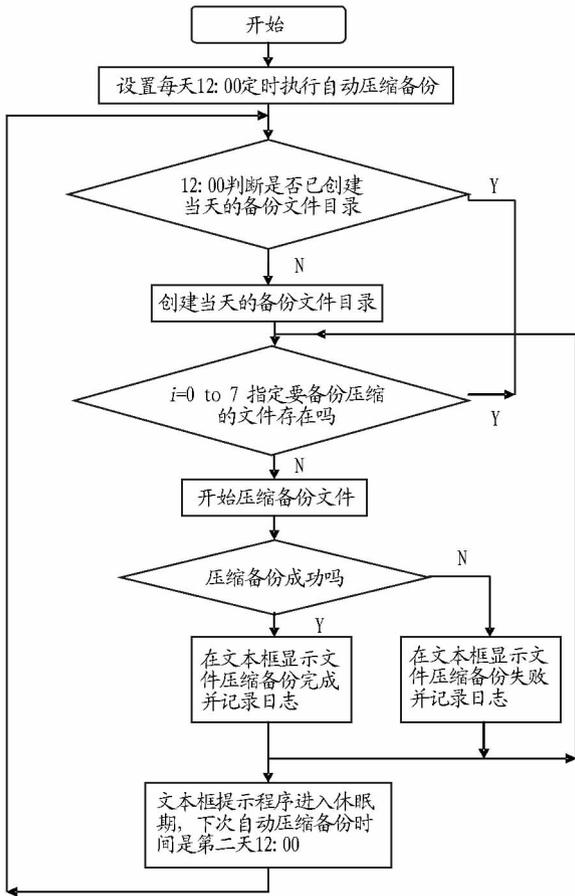


图1 自动压缩备份 MICAPS 数据文件模块工作流程

其中的元素值分别为 surface、high、physic、fy2、japan、japan_thin、ecmwf、ecmwf_thin,代表需要压缩的文件夹名称。myThread 是程序中将要创建的线程。

在模块中添加 timer 控件并定义为 timer1,用于定期引发事件。将 timer1 的 Interval 属性设置为 86 400 000 ms。定义方法 SetTimer1(),在该方法中设置每天 12:00 定时执行压缩备份操作。此外,模块还定义了 RunFileBak()方法,该方法实例化 CWinRAR 类的一个对象,并使用该对象名调用 RAR 方法,实现文件自动压缩备份并在文本框中给出相应的运行信息,说明系统当前的运行情况。因为程序执行打包压缩操作比较费时间,会导致用户界面长时间处于停滞状态,因此在模块中创建一个线程去执行压缩操作,避免出现用户界面停滞的情况。

正常情况下,文件压缩完成后,在当天的备份目录下将生成 surface_YYYYMMdd.RAR、high_YYYYMMdd.RAR、physic_YYYYMMdd.RAR 等 8 个文件(“YYYYMMdd”表示当天的年月日格式)。

完成写日志功能的 Syslog 类的关键代码如下:

```

public static void Add_log(string LogInfo)
{
    StreamWriter sw = File.AppendText(@"e:/log/SysLog_
cf.log");
    sw.WriteLine(DateTime.Now.ToString() + " : " + Log-

```

Info);

```

sw.Close();
}

```

完成文件压缩备份功能的 CWinRAR 类需要引用 Microsoft.Win32 命名空间,该命名空间提供了 2 个用于注册表操作的类,即 Registry 类和 RegistryKey 类。在程序设计中,通过 Registry 类找到注册表中指定的主键,再通过 RegistryKey 类打开指定的子键并获取该子键的键值^[3]。CWinRAR 类的关键代码如下:

```

public bool RAR (string path, string rarPath, string rar-
Name)
{
    bool flag = false;
    string rarexe;
    RegistryKey regkey;
    Object regvalue;
    string cmd;
    ProcessStartInfo startinfo;
    Process process;
    try
    {
        regkey = Registry.ClassesRoot.OpenSubKey (WinRAR_
KEY);
        regvalue = regkey.GetValue("");
        rarexe = regvalue.ToString();
        regkey.Close();
        rarexe = rarexe.Substring(1, rarexe.Length - 7);
        Directory.CreateDirectory (path);
        cmd = string.Format("a {0} {1} - r - ibck - tn24",
rarName,
path); //在后台压缩过去 24 小时的文件(含子目录下
的文件)
        startinfo = new ProcessStartInfo();
        startinfo.FileName = rarexe;
        startinfo.Arguments = cmd;
        startinfo.WindowStyle = ProcessWindowStyle.Hidden;
        startinfo.WorkingDirectory = rarPath;
        process = new Process();
        process.StartInfo = startinfo;
        process.Start();
        process.WaitForExit();
        if (process.HasExited)
        {
            flag = true;
        }
        process.Close();
    }
    catch (Exception e)

```

```

}
throw e;
}
return flag;
}

```

2.2 手动压缩备份 MICAPS 数据文件模块 在该模块中有 2 个类, Syslog 类实现写日志功能, CWinRAR 类实现文件压缩备份功能。该模块使用复选框控件监控相关文件目录是否被选中, 对选中的控件所代表的目录在状态栏中显示。如果代表“全选”的复选框控件被选中, 点击“手动备份”按钮后将对 8 个目录下的文件进行压缩备份。FileBak1() 方法对复选框控件的 CheckState 属性进行遍历, 如果复选框控件被选中, 则调用 Filebakcom 方法对相关目录进行压缩备份。FileBakcom 方法实现具体的压缩备份过程, 在该模块中创建一个线程去执行压缩备份操作。

2.3 用户界面 自动压缩备份 MICAPS 数据文件模块用户界面见图 2, 手动压缩备份 MICAPS 数据文件模块用户界面见图 3。

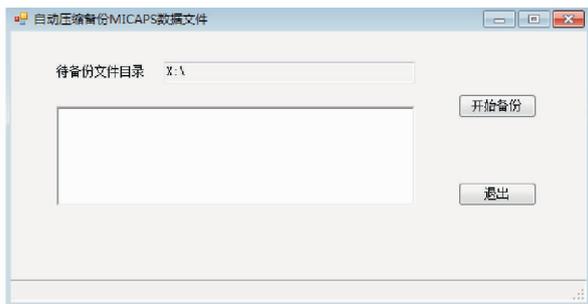


图 2 自动压缩备份 MICAPS 数据文件模块用户界面



图 3 手动压缩备份 MICAPS 数据文件模块用户界面

3 系统采用的关键技术

3.1 线程的调度 Thread 类位于 System.Threading 命名空间下, 主要用于创建并控制线程、设置线程优先级并获取其状态。

创建一个线程, 首先要声明该线程并为其提供线程起始点处的方法委托。Thread 类具有接受一个 ThreadStart 委托或 ParameterizedThreadStart 委托的构造函数, 该委托包装了调用 Start 方法时由新线程调用的方法, 用来使线程被安排执行。当处于运行状态下的线程调用 Thread 类的 Suspend 方法时, 该线程处于等待状态。进入等待状态的线程必须调

用 Resume 方法才能被唤醒, 终止线程可以使用 Abort 方法来实现^[4]。

Control 类的 Invoke 方法可提供让其他线程更新 UI 线程控件的机制。自动压缩备份 MICAPS 数据文件模块中, 自定义方法 SetFileBak 用于实现线程的托管, 自定义方法 RunFileBak 用于实现文件的自动压缩备份在线程中执行。在线程 myThread 调用的 SetFileBak 方法里, 启动 Control.Invoke 方法, Control.Invoke 方法调用委托方法 RunFileBak(), 可在拥有控件的基础窗口句柄的线程上执行指定的委托^[4]。

程序关键代码如下:

```

private void button1_Click(object sender, EventArgs e)
{
    Button1.Enabled = false;
    this.toolStripStatusLabel1.Text = "程序开始运行";
    SetTimer1();
    myThread = new Thread(new ThreadStart(SetFileBak));
    myThread.Start();
}

public delegate void FileBak();
public void SetFileBak()
{
    this.Invoke(new FileBak(RunFileBak));
}

```

3.2 类的使用 DirectoryInfo 类和 Directory 类可以方便地对目录进行操作, 而 FileInfo 类用于对文件进行操作。这 3 个类都位于 System.IO 命名空间下。该系统使用这 3 个类分别实现目录创建、判断要创建的文件夹是否存在、判断指定的文件是否存在。此外, 在自定义类 Syslog 中实例化 StreamWriter 类对象, 可以使用 I/O 流操作日志文本文件^[4]。

4 结语

该系统能实现 MICAPS 产品数据文件的存储, 界面简洁友好, 性能稳定可靠。当有新的 MICAPS 数据文件项目需要备份时, 可向 zippath1 数组及 zippath2 数组添加相应的目录元素, 程序作较小的改动即可实现新增加目录的备份, 代码易于维护。压缩备份后每天生成的 RAR 文件大小总量不到原 MICAPS 数据文件的 20%, 既节省服务器存储空间, 又方便日后管理, 有较好的应用前景。

参考文献

- [1] 李月安, 曹莉, 高嵩, 等. MICAPS 预报业务平台现状与发展[J]. 气象, 2010, 36(7): 50-55.
- [2] 姚愚, 闫丽萍, 常有礼. 调用 WinRAR 压缩备份 Micaps 数据文件[J]. 成都信息工程学院学报, 2013, 28(3): 223-227.
- [3] 明日科技. C#从入门到精通[M]. 北京: 清华大学出版社, 2012: 428-435.
- [4] 王小科. C#开发实战宝典[M]. 北京: 清华大学出版社, 2012: 410, 412, 426, 516-519, 534.