

基于动态链接库自动查找的软件发布方案

赵正旭, 刘甜, 张登辉

(石家庄铁道大学 信息科学与技术学院 河北 石家庄 050043)

摘要: 针对传统的软件发布过程中人工定位主程序依赖动态链接库的问题, 提出了一个基于动态链接库自动查找的软件发布方案。该方案利用 perl 脚本对程序所依赖的动态链接库文件进行自动查找, 解决了多路径下同名动态链接库无法正确定位的问题, 并使用开源工具 Inno Setup 实现了大型软件系统的发布。该方案避免了手动查找依赖库的繁琐过程, 减少了人工干预, 降低了开发人员的系统维护成本, 为软件发布提供了一种快速有效的途径, 提高了软件发布的效率和稳定性。

关键词: 软件发布; 软件生命周期; perl 脚本

中图分类号: TP311 **文献标志码:** A **文章编号:** 2095-0373(2014)03-0064-05

0 引言

随着航天事业的发展, 航天软件的规模不断扩大, 载人航天系统中软件系统功能日益复杂, 程序所占用的计算机空间呈现出越来越大的趋势, 对于这种大型的应用软件而言, 大量的分布在计算机不同位置上的文件对软件的发布带来了巨大的困难。航天软件主要指作为型号组成部分的计算机软件, 包括弹载、星载、箭载等飞行器和地面设备上的制导、控制、通讯、遥测、遥控、安全控制、外测、作战管理、测试、诊断、维护软件等^[1]。

当前软件的开发, 大多数是建立在已有的类库或框架上进行二次开发, 这样既可以快速实现客户的要求, 又可以保证程序的鲁棒性。航天可视化操控软件是在 Visual Studio 平台上开发的, 包括任务管理、用户管理、文件管理、操控功能等, 提供了三维信息显示和业务作业平台。这个软件体系结构庞大, 业务逻辑复杂, 还用到了第三方软件 OSG 和 Qt, 作为一个部署于前台图形工作站的依赖多个第三方库文件的大型航天软件, 它的安装部署工作较为频繁, 这对软件的发布带来一定的困难。在对主程序打包时, 查找其依赖的动态链接库就成为一个关键问题。传统的打包过程需要人工查找这些动态链接库并拷贝至可执行文件所在目录, 这种方法增加了系统的维护成本, 而且人工查找的方式不能保证准确性。为了保证版本的统一性和程序的完整性, 实现整个安装程序的打包发布, 保证安装过程中安装、回滚和卸载的统一性, 提供给用户灵活的选择权, 基于 perl 语言提出了一种自动查找动态链接库的方法, 将上述工程自动化, 使系统的发布过程更加简洁。这一技术已应用到嫦娥三号可视化系统的软件发布之中。

1 软件发布研究

目前, 国内外对于制作安装包方面的研究主要热点在于使用第三方软件进行二次开发, 而市场份额最大的操作系统是 Windows, 所以有很多软件是针对 Windows 操作系统的, 它们可以制作出界面友好、功能丰富的安装包。下面介绍几种安装包制作工具及其特点^[2], 见表 1 所示。

DOI: 10.13319/j.cnki.sjztdxxb.2014.03.14

收稿日期: 2013-06-07

作者简介: 赵正旭 男 1960 年出生 教授

基金项目: 国家自然科学基金(60873208)

表 1 常用安装包制作工具及其特点

工具	特点
Install Shield	使用广泛,功能强大,但是过于复杂,对新手来说不容易学习,是收费软件。
Wise Installer	提供多于 20 种的语言包,但是程序依赖的文件需自己查找并添加。
Inno Setup	该软件用 Delphi 编写,支持 pascal 脚本,简单易学。
Setup Factory	屏幕可以重复使用,自动备份功能强大,但是压缩不是很高,创建特殊的快捷方式有点麻烦。
NSIS	NSIS 全称 Nullsoft Scriptable Install System,提供可扩展的插件接口,但是插件太过庞大,占用内存多。
Install Creator	压缩率较高,可以定制工具栏,可以下载文件,但是不够灵活,甚至不能实现某些功能。

另外,有些软件部署工具会集成在应用程序开发平台中,可以较为容易地制作出由本平台开发的应用程序的安装包,因为它会自动去搜索应用软件所需的相关文件,此类平台如 VC++、Visual Basic 和 PowerBuilder 等^[3]。但是有些情况下,开发平台还是会遗漏文件,它们制作出的安装包功能比较简单,安装界面单一,甚至开发商的印记一直保留,无法去除。经过研究,采用 Inno Setup 工具来制作航天可视化操控软件的安装包。

Inno Setup 是一个制作安装包的免费软件,支持 Windows 95、98、2000、2003、XP 等 Windows 所有现有的版本,支持 pascal 脚本,可以快速制作出类似 Windows 2000 风格的安装界面。该软件用 Delphi 语言编写,可以从其官方网站免费下载源程序。该软件简单易懂,主要特点有:支持完全的反安装功能;可以将不同的版本信息进行识别对比,把正在使用的文件替换掉;小巧、简便、精美是它的最大特点;其安装形式灵活多变,甚至超过一些商业的安装软件^[4]。

2 动态链接库查找

动态链接库(DLL)是作为独立的程序模块存在,封装特定功能。系统可以单独测试和编译它们。在运行可执行程序时,一旦系统确定了要调用的 DLL 后,就会分配一些内存来加载它们。这种方式减少了可执行文件的大小和对内存空间的需求,并且多个应用程序可以同时使用这些 DLL 模块。在发布 DLL 文件时,会导入一个对应的 LIB 文件。该文件包含了每一个 DLL 导出函数的可选的标识号和符号名,但是并不含有实际的代码。LIB 文件中也包含了对应的 DLL 文件名(但不是完全的路径名),链接程序将其存储在可执行文件内部。当应用程序运行过程中需要加载 DLL 文件时,系统会根据上述信息定位、加载 DLL,然后通过符号名或标识号实现对 DLL 函数的动态链接^[5]。在相关 DLL 缺失的情况下,程序无法正常运行。

在 windows 系统中,搜索 DLL 的顺序是由注册表键值(HKLM\System\CurrentControlSet\SessionManager\SafeDllSearchMode)决定的:在 vista、xp sp2、server2003 中,这个键值是 1,搜索 DLL 的顺序是:可执行文件所在目录、系统目录、位系统目录、windows 目录、当前进程目录、环境变量 PATH 中的目录。这个值为 1,在 2000 sp4、xp 中,这个键值是 0,搜索 DLL 的顺序是:可执行文件所在目录、当前进程目录、系统目录、16 位系统目录、windows 目录、环境变量 PATH 中的目录。

3 基于动态链接库自动查找的软件发布方案

3.1 动态链接库自动查找方法

如果一个软件实现的功能很复杂,那么在 Visual Studio 平台上开发软件时,就会用到第三方软件进行辅助^[6],所以应用软件所需的文件数量很多,可以将这些文件分为:软件正常运行所需的媒体文件、Visual Studio 所需文件、依赖的第三方库文件(这些第三方库一般以动态链接库的形式存在)。而航天可视化操控软件是在 Visual Studio 平台上开发的一款功能强大的应用软件,并且用到了第三方软件 OSG、Qt,在部署中就容易遗漏。为保证所有文件都包括在部署安装过程之中,使用 Visual Studio 工具组件中的命令行工具 dumpbin 来查看可执行程序的依赖关系。perl 的优势在于它具备非常强大的处理批量数据和文本数据的功能^[7],于是采用 perl 语言编写可以自动查找程序依赖文件并将其打包的程序。部分代码如下:

```
exec("dumpbin /dependents ChangE.exe /out: a.txt");
#逐行读取 a.txt 文件中的内容,并输出可执行程序依赖的文件
```

```

sub readf
{
    open FILE , "d: /a. txt " ;      #打开 a. txt 文件
    foreach ( <FILE > ) {          #逐行读取文件中内容
        if( /dll $ / ) print $ _ , "\n" ; }      #输出动态链接库文件
        close( FILE) ;          #关闭文件
    }
}
#从计算机上查找所依赖的文件
sub wanted
{
    my $ n ;
    my @ filenames ;      #数组存放动态链接库文件名
    foreach $ n ( @ filenames ) {
        if ( $ _ eq $ n ) {
            print $ File : : Find : : name ;      #输出文件的绝对路径
            print " \n" ; }
    }
}
find ( \&wanted , "d: /" ) ;
#把文件复制到指定文件夹
sub copyf
{
    my $ file ;
    my @ files ;      #数组存放文件的绝对路径
    foreach $ file ( @ files ) {
        my $ path = C: /Documents and Settings/Administrator / ;      #目标路径
        copy( $ file , $ path ) ; } #将文件复制到目标路径
}

```

首先在 perl 脚本中调用工具 dumpbin 对可执行程序进行分析 ,并将结果输出到一个临时文件中 ,之后解析这个临时文件得到主程序所依赖的动态链接库。然后综合系统默认的动态链接库加载路径与用户指定的路径准确定位 DLL 文件 ,最后把这些文件复制到指定文件夹。至此 ,制作安装包所依赖的各种文件准备完毕。

3.2 软件发布流程

首先 ,使用 dumpbin 工具查看可执行程序的依赖关系 ,再利用 perl 编写的程序自动查找软件依赖的文件并将其打包。把软件运行所需要的所有文件按照目录结构放到一起 ,包括图片文件、数据库文件、插件等^[8]。

其次 ,运行 Inno Setup 向导 ,开始制作软件包。使用向导设置安装程序的基本信息 ,包括软件名称、版本、软件许可声明、缺省安装路径、安装过程提示信息等 ,在部件中添加所包含的文件组和目录^[9]。把所有需要发布的程序文件、配置文件、动态库文件、资源文件、数据库文件等添加到向导中。

按照提示完成向导后 ,生成安装脚本 ,可在脚本文件中自行添加代码 ,实现多样化的安装。保存并编译脚本 ,生成安装包 ,进行软件发布。软件发布流程见图 1 所示。

3.3 软件发布方案测试

测试环境分别为:(1) Microsoft Windows XP Professional 2002 系统 ,AMD Athlon 64 X2 Dual 处理器。

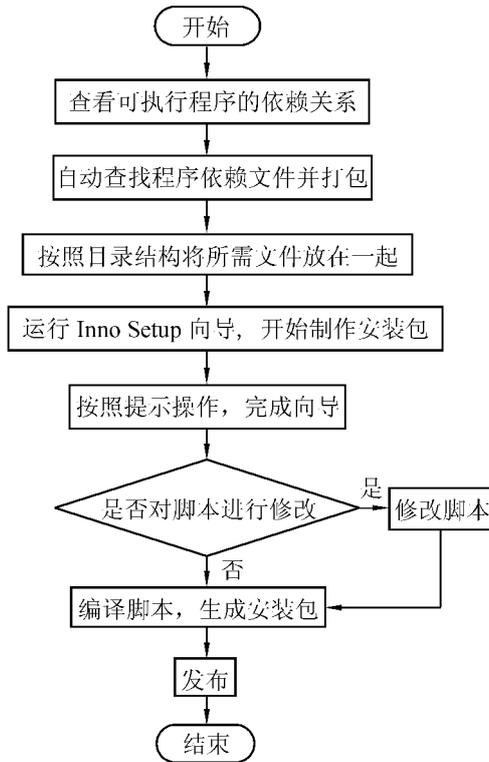


图 1 软件发布流程图

(2) Microsoft Windows 7 系统 ,AMD Athlon 64 X2 Dual 处理器。

生成的软件安装目录如图 2 所示 ,sys_data 存放主程序所依赖的各种数据文件 ,包括各种配置文件、数据库文件、图像文件以及模型文件。DLL 文件夹中存放查找到的主程序所依赖的各种动态链接库文件 ,项目开发时主程序所依赖的第三方库主要是 Qt ,OSG ,下图表明该方案正确查找到了主程序所依赖的各种动态链接库文件。软件的安装界面如图 3 所示。

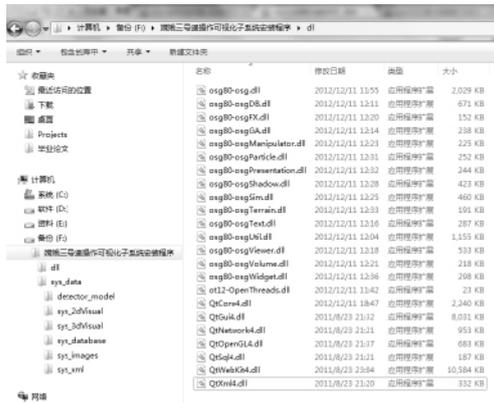


图 2 软件安装目录



图 3 安装软件欢迎页

测试结果表明 ,安装包在测试环境中运行稳定 ,即使存在多目录同名 DLL 文件 ,该方案也能准确定位到主程序所依赖的动态链接库文件 ,达到预期效果。

4 结论

基于开源软件 Inno Setup 与 perl 脚本提出了一种自动化的软件发布方案。利用 perl 脚本强大的文本处理能力与系统粘合力 ,实现了自动查找程序所依赖的动态链接库文件的功能 ,便于可视化操控软件一站式快速部署、管理和发布。实践表明 ,该方案有效的提高了航天可视化操控软件的部署效率 ,降低了系统维护成本 ,为软件发布提供了一种快速有效的途径。

参 考 文 献

- [1]石柱,袁心成,马卫华,等.适用于航天软件开发的可靠性度量[J].航天控制,2004,22(3):87-92.
- [2]赵雨水,左春,杨立,等.软件发布机制体系结构研究[J].计算机工程与设计,2010,31(4):700-705.
- [3]肖宏伟.专家门诊 Visual C++ 开发答疑[M].北京:人民邮电出版社,2003.
- [4]郑江花,秦立刚,张霞.基于 Inno Setup 的 Delphi 应用程序的发布技巧[J].信息技术与信息化,2008(3):76-77.
- [5]李娟,张钦.基于动态链接库的 Visual C++ 混合编程[J].计算机工程与设计,2010,31(9):2140-2143.
- [6]Zhao Zhengxu, Jun Feng, Zhang Zhihua. Complexity Analysis on Engineering Software Data Format Conversion Networks [J]. Complex Systems and Complexity Science, 2010, 7(1):75-81.
- [7]袁洁.基于 perl 的软件多语言支持[D].石家庄:石家庄铁道大学,2012.
- [8]Guo Yang, Zhao Zhengxu, Zhou Yiqi. Complexity Analysis with Function-call Graph on Windows Software [J]. International Review on Computers and Software, 2012, 3(7):1149-1153.
- [9]赵正旭,龙瑞,郭阳,等.工程软件的小世界效应探究[J].石家庄铁道大学学报:自然科学版,2010,23(3):1-6.

Software Release Scheme Based on Automatic Searching for Dynamic Linkable Library

Zhao Zhengxu, Liu Tian, Zhang Denghui

(School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)

Abstract: For the issue of manually locating the dynamic linkable library which the main program depends on during the traditional software release, we propose a software release scheme based on searching automatically for the dynamic linkable library. This scheme searches automatically for the dynamic linkable library which the program depends on through the perl script, solves the problem of the same dynamic linkable library not properly located in the multi-path, and achieves the release of large software system using the open source tool Inno Setup. This scheme avoids the trouble brought by finding manually dependent libraries, reduces the manual intervention, reduces the cost of system maintenance, provides a quick and effective way for releasing software and improves the efficiency as well as stability of the software release.

Key words: software release; software life cycle; perl script

(责任编辑 车轩玉)