

# 基于 VXWORKS 系统的热补丁技术研究

姜 文,刘立康

(西安电子科技大学 通信工程学院,陕西 西安 710071)

**摘 要:**软件产品交付用户使用后,需要解决软件产品在使用过程中的各类问题与缺陷。对于 VXWORKS 操作系统开发的软件产品,通常采用热补丁技术,可以在不影响系统正常运行的情况下完成对软件产品错误的修复。为了解决软件在使用过程中的各类问题与缺陷,需要深入研究热补丁技术。结合工作实践,叙述了 VXWORKS 操作系统的特点和热补丁制作过程中的各种相关技术。以 ClearCase 作为配置管理工具,ICP-CI 作为持续集成工具,进行热补丁包制作;详细叙述了基于 VXWORKS 系统的软件热补丁包制作过程,包括热补丁代码的编写和静态检查、模块热补丁文件的制作、热补丁包打包、热补丁版本包的测试与发布。最后介绍了一个典型工作案例。工作实践表明,热补丁技术具有实时修复,且不影响嵌入式系统正常运行的优点,提高了系统可靠性,降低了软件开发成本。

**关键词:**嵌入式操作系统;热补丁;持续集成;静态检查

**中图分类号:**TP311.53

**文献标识码:**A

**文章编号:**1673-629X(2017)03-0018-05

doi:10.3969/j.issn.1673-629X.2017.03.004

## Research on Hot Patch Technology Based on VXWORKS

JIANG Wen,LIU Li-kang

(School of Telecommunication Engineering,Xidian University,Xi'an 710071,China)

**Abstract:**After the software is delivered to customers,all kinds of problems and defects are needed to be solved during the use of software. The software developed by the VXWORKS operating system is usually used the hot-patch technology,which can solve the software errors with the system operated normally. In order to solve kinds of questions and defects,the technique of hot patch is needed to be researched deeply. With the practice of the work,the characteristics of VXWORKS operating system and the types of files during the process of hot patch making are described. With ClearCase as the configuration management tool and ICP-CI as the continuous integration tool to produce hot patches,the process of software hot patches based on the VXWORKS operating system is described in detail,including writing and static checking the code of hot patch,making the hot patch files for modules,packing the hot patches,and testing and releasing the version package of hot patches. Finally,a typical work case is introduced. Practice shows that the technique of hot patch has the advantages of real-time repair,and will not affect the normal operation of the embedded system,improving system reliability, reducing the cost of software development.

**Key words:**embedded operating system;hot patch;continuous integration;static check

## 0 引 言

VXWORKS 是实时嵌入式操作系统软件,可以为开发人员提供高效实时的任务调度、中断管理、系统资源管理和任务间通信。VXWORKS 经过广泛验证,已成功应用在航天、航空、舰船、通信、医疗等关键领域。

软件产品交付用户使用后,需要解决软件产品在使用过程中的各类问题与缺陷<sup>[1]</sup>。对于 VXWORKS 平台开发的软件产品,采用热补丁技术,可以在不影响

系统正常运行的情况下完成对系统错误的修复,从而提高了软件的安全性和可靠性,高质量完成软件维护工作,延续软件的使用寿命,降低软件开发成本。

文中采用软件配置管理工具 ClearCase 和持续集成工具 ICP-CI 进行热补丁包制作。详细叙述了热补丁代码的编写和静态检查、模块热补丁文件的制作、热补丁包打包、热补丁版本包的测试与发布。最后介绍了一个典型工作案例。工作实践表明,采用持续集成

收稿日期:2016-04-17

修回日期:2016-08-11

网络出版时间:2017-02-17

基金项目:国家部委基础科研计划;国防预研基金项目(A1120110007)

作者简介:姜 文(1986-),女,硕士研究生,工程师,CCF 会员,研究方向为图像处理、数据库应用和软件工程等;刘立康,副教授,研究方向为数字通信、图像处理、嵌入式应用技术等。

网络出版地址:<http://www.cnki.net/kcms/detail/61.1450.TP.20170217.1628.022.html>

技术有利于提高软件热补丁制作的效率与质量。

### 1 热补丁相关技术

软件产品交付用户使用过程中会出现的各种问题或缺陷,导致软件产品部分功能不可用,或者影响软件产品的性能,有时软件可能会出现安全问题。需要在原有的软件系统上实施补丁来修复这些问题。

#### 1.1 补丁分类

软件补丁的分类有多种,根据待实施的软件可以分为操作系统补丁、应用软件补丁;根据修复的软件缺陷的危害程度可以分为高危补丁、普通补丁;根据实施补丁的方式可以分为冷补丁、热补丁。

##### 1.1.1 冷补丁

冷补丁<sup>[2-3]</sup>,是修改软件模块的源代码之后,对该模块对应的整个工程文件进行重新编译,将编译之后的所有可执行文件重新加载到设备上执行,在这个过程中需要在设备上停掉原有的软件产品,再重新安装一遍软件产品。

##### 1.1.2 热补丁

热补丁<sup>[4-6]</sup>,是在软件系统正常运行的过程中进行热补丁安装,热补丁的实质就是替换原有软件系统中的一个或多个函数。执行热补丁安装的过程中,并不需要停止软件系统,而且执行热补丁安装的过程操作起来也很快。通常,热补丁适用的软件系统是一个内核非抢占式的实时操作系统,它具有可伸缩、可裁剪和高可靠性等特点,是一个可以同时适用于主流 CPU (如 MIPS、PPC、X86 和 ARM 等) 目标平台的实时操作系统。

#### 1.2 VXWORKS 系统的特点

- VXWORKS 系统<sup>[4,7-8]</sup>提供了以下功能:
- (1)任务管理。
  - 操作系统执行的每个程序都被称之为任务,VXWORKS 系统模块的运行一般是以任务的形式进行调度和运行;操作系统可以动态启动或停止某一指定任务。
  - (2)符号表管理。
  - VXWORKS 系统在软件编译时可以将所有的全局变量和外部函数都编译到一个全局符号表中,软件系统启动后,该符号表可以通过操作系统提供的相关函数通过变量或函数名查找变量存储地址或函数的入口地址。
  - (3)模块动态加载。
  - 操作系统支持将模块软件编译成 ELF 格式的可执行文件,在系统运行过程中动态加载该文件。
  - 由于 VXWORKS 系统具有上述三种功能,有利于采用热补丁的方式修复软件模块存在的各种缺陷。基

于 VXWORKS 操作系统的软件产品热补丁具有继承性,即后一个版本的热补丁需要将前一个版本的热补丁所修改的缺陷收编进来,在热补丁的制作过程中需要注意这一点。

#### 1.3 热补丁制作过程中的文件类型

Tornado 是 VXWORKS 应用系统的集成开发环境。Tornado IDE 采用 C/C++ 语言编程,支持 GNU 的 C/C++ 编译器。GNU 编译器生成的目标文件缺省为 ELF (Executable and Linking Format) 格式<sup>[4-5]</sup>。

##### 1.3.1 目标文件 \*.obj 的文件格式

目标文件 \*.obj 为 ELF 格式<sup>[5-6]</sup>,文件格式如图 1 所示。

文件头	
代码段: 二进制代码指令	
数据段:	
Read-only 的数据	
已经初始化的数据	
还未初始化的数据	
符号段	符号 1 信息
	符号 2 信息
	...
	符号 $N_1$ 信息
重定位段	重定位 1 的信息
	重定位 2 的信息
	...
	重定位 $N_2$ 的信息
段表	代码段的段头
	数据段的段头
	重定位段的段头
	符号段的段头

图 1 目标文件 \*.obj 的文件格式

目标文件包括文件头、代码段、符号段、重定位段和段表。其中,代码段存放二进制代码指令;数据段中分类存储数据;符号段中存放该文件中定义的各个全局函数和全局变量的名字以及它们加载后在内存中的地址;在重定位段中存放引用外部符号“名字”以及“在何处引用”等信息。文件头对段表进行索引,段表中包括代码段的段头、数据段的段头、重定义段的段头,这些段头包含对应段的相关信息。

##### 1.3.2 热补丁文件 \*.pat 的文件格式

热补丁文件 \*.pat 的文件格式<sup>[5-6]</sup>如图 2 所示。在补丁文件中每个补丁单元对应一个补丁函数。每个补丁单元包括代码段、数据段和函数替换表三部分。其中,代码段和数据段存放的是补丁函数的实际指令;函数替换表中包括被补丁函数的地址和补丁函数地址。每个补丁单元对应相应的补丁单元头,补丁单元头中包含补丁函数名称、函数尺寸和函数地址。

##### 1.3.3 文件类型

- 热补丁制作过程中涉及以下文件类型。
- (1) \*.c 或 \*.cpp 文件,采用 C/C++ 语言编写的热补丁源程序。

文件头	
检验信息(CRC)	
单元 1(第 1 个补丁)	函数替换表
	代码段
	数据段
单元 2(第 2 个补丁)	函数替换表
	代码段
	数据段
...	...
单元 N (第 N 个补丁)	函数替换表
	代码段
	数据段
补丁单元 1 头	
补丁单元 2 头	
...	
补丁单元 N 头	

图 2 热补丁 \*.pat 的文件格式

(2)目标文件 \*.obj,采用 gcc 或 g++将 \*.c 或 \*.cpp 文件编译生成的目标文件,具有 ELF 格式的二进制文件。

(3)模块目标文件 \*.elf,将软件模块中的所有目标文件 \*.obj 连接在一起生成模块的目标文件(模块名.elf),通常称为模块基线文件。

(4)热补丁文件 \*.pat,由热补丁函数组成的文件,可以加载到系统中应用。

(5)符号表信息文件 \*.SYM,在目标文件中删除符号信息以外的所有内容得到的文件。文件中以符号为单位存放各个全局函数和全局变量的名字和它们加载后在内存中的地址。

1.4 热补丁包制作流程图

采用持续集成技术制作热补丁,可以高效率和高质量完成热补丁包的制作<sup>[4-6,9-12]</sup>。文中采用的软件配置管理工具是 ClearCase,持续集成工具是 ICP-CI,进行热补丁包制作。热补丁的制作流程如图 3 所示。

2 热补丁代码的编写和静态检查

2.1 热补丁代码编写

制作热补丁之前,开发工程师需要通过软件产品的配置管理工程师(CMO)确认需要制作热补丁的软件代码基线。同时确认目前加载在软件基线上的热补丁包版本。开发人员从版本库下载相关的软件基线代码文件和补丁包代码文件,编写新的热补丁代码。

对于补丁开发首先需要定位存在源代码缺陷函数(因为热补丁的原理就是函数替换),定位之后修改相关的函数源代码,修改函数相应的功能代码。

开发工程师进行补丁开发与通常的软件开发有区别,需要注意如下事项:

- (1)对 \*.h 文件的处理。  
对补丁开发过程中涉及的代码缺陷,定位到需要修改的函数源代码后,修改相应的功能代码时,不能修

改对应的 \*.h 文件中的内容。

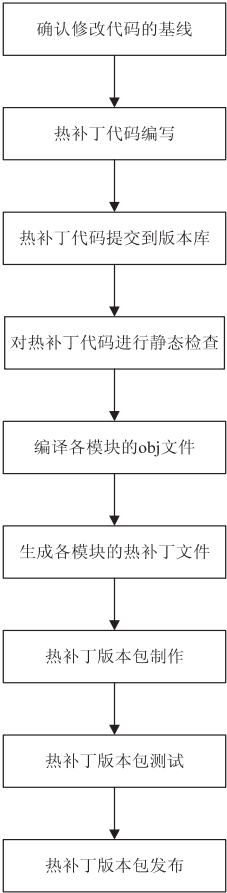


图 3 热补丁的制作流程图

- (2)对函数的处理。  
修改缺陷对应的 \*.c 文件的源代码时,只保留需要修改的函数。去掉所有无关的函数,可以直接删掉或者用 #if 0 #endif 注释掉。
- (3)对全局变量的处理。  
修改缺陷对应的 \*.c 文件时,需要将所有的全局变量都改用 extern 声明,并把在定义全局变量时的赋值都去掉;对于用 const 声明的全局常变量(常变量,只允许读不允许写),如果开发工程师所修改的函数中未用到该全局变量则在该全局变量前加 extern 声明,还要将其后面的赋值去掉;若所修改的函数中用到了该全局变量,就需要将该全局变量改成别的变量名,并重新定义,其初始化与原全局变量一致,同时要在要修改的函数中用新的变量名替换老的变量名;若该全局变量是 static 声明的,如果需要修改的函数中未用到该变量,可直接加 extern 声明。
- (4)对静态函数的处理。  
若修改的函数是静态函数,可通过修改该函数的上层函数,即修改完缺陷对应的函数之后,不能注释掉调用该函数的上层函数,另外,所有的上层函数要调用的静态函数都不能注释掉。
- (5)加入返回补丁版本号的函数。

修改的 \*.c 文件中加入一个返回补丁版本号  
的函数,其返回值为补丁版本号。

开发工程师完成补丁代码编写后,对代码进行单  
元测试,静态走码检查,最后将评审过的代码合入版本  
库。开发工程师向版本库提交代码时,要添加注释、说  
明、CR 单号、修改原因等,以保证可追溯。

### 2.2 ClearCase 版本库的代码更新

持续集成工具 ICP-CI 需要在版本库锁库之后完  
成源代码更新。ICP-CI 工具执行代码更新时,需要编  
写代码更新的批处理脚本 code\_update.bat,把代码更  
新的脚本配置在任务中。

### 2.3 代码静态检查

为了确保热补丁文件的源代码质量,需要对开发  
工程师修改的源代码进行静态检查。采用持续集成工  
具 ICP-CI 调用静态检查工具 PC-Lint 和 Coverity Pre-  
vent<sup>[13]</sup>对补丁代码进行静态检查。PC-Lint 通常用于  
C、C++源代码的静态检查,是最常用的静态检查工具。  
重量级静态检查工具 Coverity Prevent 是检测和解决  
C、C++、Java 和 C#源代码中严重缺陷的领先自动化方  
法。另外,为了提高热补丁的安全性,还需要对补丁源  
代码进行危险函数检查,避免出现诸如 strepy 等函数。  
发现代码问题反馈给开发工程师,其对代码进行修改,  
直到存在的问题全部解决。

## 3 热补丁的制作过程

热补丁分为三个层级:函数热补丁,模块热补丁  
(由多个函数热补丁组成),热补丁包(包含多个模块  
热补丁)。基于持续集成<sup>[7]</sup>的热补丁开发通常是制作  
热补丁包。

### 3.1 模块热补丁目录结构

制作模块热补丁时,需要创建“产品名\_版本号\_  
BASE\VXWORKS\模块名”文件夹,用于存放相关文件  
与热补丁文件。以 UPE 模块为例,文件夹目录结构如  
表 1 所示。

表 1 UPE 模块热补丁文件夹目录结构

目录	子目录	文件夹功能
产品名_版本 号 _ BASE/ VXWORKS/ UPE	obj	存放目标文件 *.obj 和补丁中间文件
	patchbase	存放 UPE 模块的模块基线目标文件 upe.elf
	release pat	存放 UPE 模块的模块补丁文件 upe_patch. pat

### 3.2 模块热补丁文件制作过程

以 UPE 模块为例,叙述模块热补丁文件制作  
过程。

#### 3.2.1 集成到 ICP-CI 工具进行热补丁文件编译

首先创建工程:“产品名\_版本号\_Patch-  
Build”。

在 ICP-CI 的任务管理页面上配置编译任务  
完成热补丁源程序代码编译。配置任务以模块为单  
位,在任务栏上选择“compile”任务。具体配置如表 2  
所示。

表 2 UPE 模块在 ICP-CI 页面上配置“compile”任务

参数名称	参数值	描述
layerpath	/upe	报告分层显示的层次名称,配置格式: “/模块名称”
workingdir	\$ { dir. work } \ VXWORKS \ UPE	设置编译脚本的存放路径
compiler	gcc	编译器配置,支持 msvc, gcc, javac 等编 译器的配置
executable	\$ { dir. work } \ VXWORKS \ UPE \ make _ upe_one_patch. bat	设置编译时需要执行的编译脚本
EAexecutable		设置编译时使用的 EA 加速编译脚本

持续集成工程师编写相关的 \*.ini 配置文件、  
Makefile 文件、补丁编译脚本和 ANT 脚本,启动相关程  
序完成补丁源程序代码的编译工作。将编译生成的  
\*.obj 文件归档到 UPE\obj 目录下。

#### 3.2.2 模块热补丁文件的制作

编译工作完成之后,制作补丁中间文件和模块  
UPE 热补丁文件。

(1)根据 \*.obj 文件的个数,在 UPE\obj 文件  
夹中生成从 HP0001 到 HP000X 的子目录。热补丁中  
间文件制作工具 makepatch.exe 通过补丁目标文件  
\*.obj生成与其对应的热补丁文件、符号表信息文件  
和制作热补丁文件的过程日志文件。\*.obj 文件和生  
成的热补丁中间文件存放在对应的文件夹中。以 HP  
0001 文件夹为例,文件夹中存放的文件如表 3 所示。

表 3 补丁中间文件的文件目录

补丁中间文件 存放目录	补丁中间文件	文件功能
HP0001	patchHP0001.pat	热补丁文件
	patchHP0001.SYM	符号表信息文件
	patchHP0001.txt	制作热补丁文件的过程日志文件
	*.obj	热补丁目标文件

(2)编写批处理脚本,通过热补丁中间文件生成  
模块热补丁文件。由于模块热补丁文件具有继承性,  
即包含上一个版本热补丁修复的产品缺陷,因此对非  
首次编译的热补丁文件而言,需要将前一个热补丁版  
本的热补丁中间文件与本次编译生成的热补丁中间  
文件合并起来生成目前版本的模块热补丁文件。模块  
UPE 的热补丁文件是 upe\_patch.pat。该补丁文件归档  
到 UPE\release 文件夹中。

### 3.3 热补丁打包

软件的所有模块热补丁文件制作完成后,将这些



模块热补丁文件打包。由于热补丁版本包也具有继承性,即后一个版本的热补丁版本包收编包含前一个版本包编译生成的热补丁文件。同时要更新热补丁包版本号。

#### (1) 创建热补丁包文件夹。

热补丁文件制作完成后,需要创建热补丁包的文件夹。将各模块热补丁文件\*.pat 拷贝到热补丁包文件夹的 bin/new 目录下。若模块补丁文件已经更新,拷贝更新后模块热补丁文件;若没有更新,拷贝原来的模块热补丁文件。将其他相关的文件拷贝到相应的目录下。

#### (2) 热补丁包打包。

修改打包配置文件 autorun.ini;填写本次版本号、基线版本号、源文件目录和目的文件目录。运行打包工具 packer.exe 完成热补丁版本包打包。热补丁包制作完成后,补丁版本包的文件名为:产品名 版本号 loadfiles.rar。

## 4 热补丁版本包的测试和发布

热补丁版本包制作完成后由产品经理提交转测试流程。测试经理安排测试工程师与测试环境对补丁版本包进行测试<sup>[14-16]</sup>。

测试工程师根据该补丁版本需要修复的缺陷列表,设计相应的测试用例,测试新增与待改进的特性是否符合设计规格;还要根据基本测试用例完成补丁的基本功能测试,测试工作完成后,测试工程师提供测试报告。

热补丁测试分为三个层次:补丁函数测试、补丁模块测试和补丁包测试。

(1) 补丁函数测试是最基础的测试,补丁函数测试是函数缺陷修复、新增与待改进的特性的测试。对于 UPE 模块,通过基线文件 upe.elf 和四类中间文件(热补丁文件 patchHP000X.pat,符号表信息 patch-HP000X.SYM,日志文件 patchHP000X.txt 和编译目标文件\*.obj)测试补丁函数。通过这四类文件可以方便测试补丁函数,将测试结果及时反馈给补丁源程序代码的开发人员,修改代码存在的各种问题。这四类文件也有利于补丁包测试和补丁模块测试中各种缺陷问题的定位。

(2) 补丁模块测试,对于 UPE 模块,通过基线文件 upe.elf 和 upe\_patch.pat,对补丁模块进行功能、性能和兼容性的综合测试。

(3) 补丁包测试,热补丁包的测试是补丁包加载后综合技术指标的全面测试。

完成热补丁测试工作后,将相关文件和热补丁版本包归档。产品经理发布热补丁包。

## 5 热补丁版本包的安装和管理

热补丁通常有 4 个基本状态:加载,激活,去激活和卸载。基于 VXWORKS 系统的热补丁包可以采用 LMT(Local Maintenance Terminal)客户端工具安装和管理。LMT 工具是指安装了“本地维护终端”的软件组,通过 LMT 工具的 MML(Man-Machine Language)命令,可以对网元进行相应操作和维护。

将热补丁版本包解压之后,使用 FTP 工具,打开客户端工具连接到主版本的单板,执行 MML 命令进行补丁热的安装和管理。执行 LOD TOBAM 将热补丁版本加载到硬盘,执行 LOD TOBP 将热补丁版本加载到 FLASH。加载完成之后激活补丁,执行 CHK VERSION 与 DSP PATCH 查看补丁版本号,表示加载相应的热补丁版本成功。

常用的补丁管理 MML 命令如下:

DSP PATCH 显示补丁

ACT PATCH 激活补丁

DEA PATCH 去激活补丁

CON PATCH 确认补丁

RVW PATCH 删除补丁

## 6 典型案例

某公司采用软件配置管理工具 ClearCase 和持续集成工具 ICP-CI 进行基于 VXWORKS 系统的软件热补丁的制作。

补丁包版本编号为 V2.11.0\_SPC101,共制作了 10 个模块的热补丁文件,收编了若干个来自各局点和内部测试发现的软件产品缺陷,并完成了热补丁包打包以及所有的结果文件归档任务。补丁包发布后在各局点安装使用,解决了存在的问题。

## 7 结束语

热补丁技术为嵌入式系统升级软件和嵌入式系统进行错误动态修复提供了一种技术方法。工作实践表明,热补丁技术具有实时修复,且不影响嵌入式系统正常运行的优点,对于 VXWORKS 系统的软件采用热补丁技术,可以提高软件的安全性和可靠性,同时降低软件开发成本。文中采用持续集成技术开发热补丁包,可以提高软件热补丁的质量和开发效率,可以快速向开发工程师反馈软件源代码的缺陷,便于开发工程师及时修复源代码的缺陷,同时也给项目的管理提供了很好的保证。

### 参考文献:

[1] Grubb P, Takang A A. 软件维护:概念与实践[M]. 韩柯,

(下转第 28 页)

来验证。未来的主要工作是研究基于连续时间的 CT-ZIA 如何转换为有限的域自动机模型,从而实现模型检测。

#### 参考文献:

- [1] 杨志斌,皮磊,胡凯,等.复杂嵌入式实时系统体系结构设计与分析语言:AADL[J].软件学报,2010,21(5):899-915.
- [2] 嵯百田,付秀敏,郑永果,等.基于UML的嵌入式实时系统开发方法[J].信息技术与信息化,2010(1):56-59.
- [3] SAE Aerospace. SAE AS5506:Architecture Analysis and Design Language (AADL) [EB/OL]. 2004. <http://www.aadl.info/aadl/currentsite/>.
- [4] 王立杰,刘昌禄,俞烈彬.基于MDA的MARTE模型形式化转换[J].指挥控制与仿真,2012,34(6):128-133.
- [5] SAE Aerospace. SAE AS5506 annex:behavior\_specification V1.6. [EB/OL]. 2006. [http://www.aadl.info/aadl/documents/Behaviour\\_Annex1.6.pdf](http://www.aadl.info/aadl/documents/Behaviour_Annex1.6.pdf).
- [6] 李振松,顾斌.基于UPPAAL的AADL行为模型验证方

法研究[J].计算机科学,2012,39(2):159-161.

- [7] 倪水妹.面向混成系统的ZIA形式化模型及其自动验证方法研究[D].南京:南京航空航天大学,2014.
- [8] 李广元,唐稚松.带有时钟变量的线性时序逻辑与实时系统验证[J].软件学报,2002,13(1):33-41.
- [9] 狄杨思,曹子宁,王辉.一种基于形式规范ZIA的自动验证方法[C]//火力控制技术/航空电子系统综合技术2011年学术年会.出版地不详:出版者不详,2011:113-120.
- [10] 孙全勇.时间自动机及其应用研究[D].哈尔滨:哈尔滨工程大学,2007.
- [11] 倪水妹,曹子宁,李心磊.带数据约束实时系统的模型检测[J].计算机科学,2014,41(5):254-262.
- [12] Lin Huiming, Zhang Wenhui. Model checking: theories, techniques and applications[J]. Acta Electronica Sinica, 2002, 30(S1):1907-1912.
- [13] 钱俊彦,赵岭忠,古天龙.一种基于时间自动机的时钟等价性优化方法[J].计算机工程,2005,31(18):71-73.
- [14] Henzinger T A, Wong-Toi H. Using HyTech to synthesize control parameters for a steam boiler[M]. Berlin:Springer,1996.

(上接第22页)

- 孟海军,译.第2版.北京:电子工业出版社,2004.
- [2] 姜文,刘立康.现代应用软件的维护与技术支持[J].计算机技术与发展,2015,25(4):116-120.
- [3] 姜文,刘立康.应用软件维护中的补丁开发与管理[J].计算机技术与发展,2015,25(11):11-16.
- [4] 程友清.嵌入式网络设备软件热补丁技术研究[J].微电子学与计算机,2013,30(1):28-31.
- [5] 常莉莉.分布式系统热补丁技术的研究与实现[D].广州:中山大学,2006.
- [6] 雷震宇.嵌入式网络设备在线热升级的研究及实现[D].武汉:武汉邮电科学研究院,2012.
- [7] River W. Vxworks 程序员指南[M].北京:清华大学出版社,2003.
- [8] 陈智育,温彦军,陈琪. Vxworks 程序开发实践[M].北京:人民邮电出版社,2004.
- [9] Mehdi G. Contributors to quality during software maintenance [J]. Decision Support Systems, 1998, 23(4):361-369.
- [10] Hauptmann S, Wasel J. On-line maintenance with on-the-fly software replacement[C]//International conference on config-

urable distributed systems. [s.l.]: IEEE Computer Society, 1996:70-80.

- [11] 姜文,刘立康.Oracle数据库补丁问题研究[J].电子设计工程,2014,22(20):10-13.
- [12] Shah R. Oracle on demand best practices:critical patch update [R]. [s.l.]:Oracle Corporation,2008.
- [13] 姜文,刘立康.基于VxWorks平台的软件重量级静态检查[J].微型机与应用,2016,35(6):79-81.
- [14] 蔡建平.嵌入式软件测试实用技术[M].北京:清华大学出版社,2010.
- [15] Michalik B, Weyns D, Boucke N, et al. Supporting online updates of software product lines: a controlled experiment [C]//2011 international symposium on empirical software engineering and measurement. Washington DC: IEEE Computer Society, 2011:187-196.
- [16] Chen Xuhui, Zhang Dengyi, Yang Hongyun. Research on key technologies of on-line programming in embedded system [C]//2009 third international symposium on intelligent information technology application. Washington DC: IEEE Computer Society, 2009:45-47.