

DOI:10.11918/202105060

星载软件的语句级高效更新方法

靳鑫¹,徐清华²,王慧泉¹,金仲和¹

(1. 浙江大学 微小卫星研究中心,杭州 310000; 2. 上海宇航系统工程研究所,上海 201109)

摘要:为减少星载软件在轨更新过程对测控资源的需求量,缩短更新上注时间,提出了一种无需操作系统支持、上注数据量低、版本可灵活控制的语句级软件更新方法。该方法首先利用预编译命令进行程序空间优化,使得不同模块的代码在编译后存储于不同的段中,减少代码修改对加载文件的影响;其次通过逐段求解参考段与更新段之间匹配路径,并基于此进行各段的差异内容提取,大大降低差异补丁规模;然后通过版本引导程序设计,实现对卫星软件版本的灵活控制;最后通过对文件和程序进行可靠性设计,保证整个更新过程的安全性。结果表明:该方法在 ZDPS-3A 卫星平台进行了充分的测试,所提出的更新方法可应用于无操作系统的星载计算机,支持软件状态可回滚且更新过程可靠,同时相比传统的差异内容提取方法,补丁文件的规模降低量均值超过 50.00%,可显著缩短更新上注时间。本方法能够充分满足星载软件低数据量更新的需求,可推广应用至包括微小卫星在内的多种航天器。

关键词:星载软件更新;语句级更新;程序空间优化;编辑距离;匹配路径

中图分类号: TP311 **文献标志码:** A **文章编号:** 0367-6234(2022)12-0038-08

Efficient statement-level update method for on-board software

JIN Xin¹, XU Qinghua², WANG Huiquan¹, JIN Zhonghe¹

(1. Micro-satellite Research Center, Zhejiang University, Hangzhou 310000, China;

2. Aerospace System Engineering Institute of Shanghai, Shanghai 201109, China)

Abstract: To reduce the demand for telemetry, tracking, and command (TT&C) resources in the on-orbit software update process and shorten the update upload time, this paper proposes a statement-level software update method that requires no operating system support, has a small amount of upload data, and can be flexibly controlled. Firstly, precompiled command was used to optimize the program space to make the code of different modules stored in different segments after the compilation, so as to reduce the impact of code modification on the loaded file. Secondly, the matching path between the reference segment and the update segment was solved segment-by-segment, and the difference content of each segment was extracted, which could greatly reduce the size of the differential patch. Thirdly, through the design of version guidance program, the flexible control of satellite software version was realized. Finally, the reliability design of the files and the program was performed, which could ensure safety of the entire update process. The method was fully tested on the ZDPS-3A satellite platform. Results show that the update method proposed in this paper could be applied to on-board computers without operating systems. The software state could be rolled back and the update process was reliable. Besides, compared with the traditional method of extracting different content, the average reduction of patch file size generated by this method was more than 50.00%, greatly shortened the update upload time. Therefore, the proposed method can meet the demand for low volume data update of on-board software and can be applied to other spacecrafts such as micro-satellites.

Keywords: on-board software update; statement-level update; program space optimization; edit distance; path matching

随着微小卫星功能的软件化快速发展^[1-2]及任务难度的升级,星载软件功能出现异常的风险越来越高,软件在轨更新已发展成为星载软件的基础功能之一^[3]。此外,对在轨卫星进行软件更新,还可

实现硬件故障隔离和应用功能扩展,是提高卫星运行安全与经济价值的重要手段^[4]。

由于微小卫星星地测控链路(特别是 S 频段测控)通信速率小、通信时间有限,因此大型星载软件更新通常采用低数据量的差异更新^[5],常用的星载软件差异更新方法主要可分为两种^[6]:1)模块级更新。模块级更新通过将更新模块的目标文件上注,利用星上操作系统动态加载技术进行目标文件替换,达到星载软件更新的目的^[7-8];2)语句级更新。

收稿日期:2021-05-13

基金项目:中央高校基本科研业务费专项资金(2021XZZX038)

作者简介:靳鑫(1990—),女,博士研究生;

王慧泉(1981—),男,研究员,博士生导师;

金仲和(1970—),男,教授,博士生导师

通信作者:王慧泉,hqwang@zju.edu.cn

其基本原理是对比更新版与参考版的加载文件,将不同之处生成补丁上注,星上将上注的差异内容覆盖参考版对应位置内容,完成更新^[9]。相较模块级更新,语句级更新无需操作系统环境,因此在微小卫星中广泛应用。然而,由于代码修改通常会影 响程序空间布局,使得加载文件内容发生大幅变动,进而出现更新数据量激增的现象^[5]。为降低更新数据量,朱虹等^[10]提出了将可更新子程序独立存储,通过地址表进行子程序调用的方法,使得子程序内部修改不影响外部内容。李雁斌等^[11]提出可以将可更新函数进行独立存储,利用钩子函数进行可更新函数寻址调用。这两种预设可更新对象的方法虽易实施,但限制了软件更新范围。李振松等^[12]提出将每个函数看成可更新对象,并逐个分配固定存储区。该方法的优点在于最大程度地限制了代码修改对程序空间的影响,然而该方法不仅难以应对函数新增情况,而且采用的函数级修改方法极易导致存储空间需求激增,需为每个函数预留充足的存储空间,增加内存容量负担,并使得内存空间严重碎片化。

上述语句级更新方法只强调如何通过约束程序空间的方式来降低代码修改对加载文件的影响,却忽略了加载文件的差异生成过程对最终更新数据量

的影响。传统的差异提取方式可称为同地址比较法,即比较参考版与更新版加载文件在相同程序空间地址上的内容,二者不同则认为是差异内容。这种差异提取方式放大了处于低地址空间的更新对整个更新数据量的影响。只约束程序空间而不改进差异提取方式的语句级更新方法,在追求更低更新数据量同时,会极大地限制更新范围或带来严重的内存负担,难以实现星载大规模软件的灵活更新。

为了克服传统语句级更新方法存在的问题,本文在约束程序空间布局的基础上,进一步考虑差异生成过程对更新数据量的影响,提出了一种利用序列间匹配路径进行差异补丁制作的方法。同时本文为了保证星载软件状态可回滚,设计了版本引导程序。实验结果表明,利用本文提出的方法进行软件在轨更新时,更新上注数据量少、过程可靠,且软件状态可回滚。

1 星载软件在轨更新方法

1.1 更新方法概述

一次完整的星载软件语句级更新通常包含地面过程和星上过程两部分,如图 1 所示。

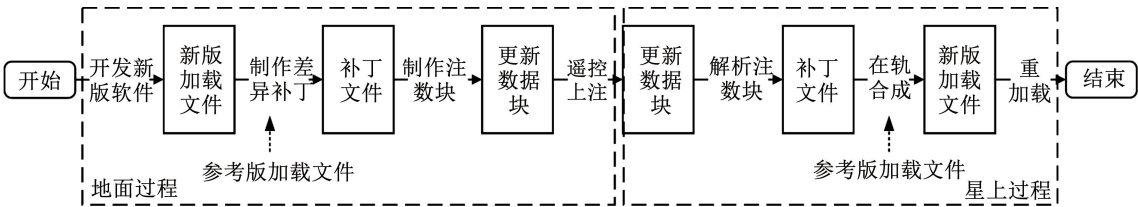


图 1 语句级星载软件在轨更新过程

Fig. 1 Statement-level update process of on-board software

地面过程包含 4 个步骤,分别为:1)开发新版本软件。利用开发工具对软件源码进行修改、编译和链接,生成新版目标可执行文件,并从中提取加载依赖项形成加载文件;2)制作差异补丁。将新版与参考版加载文件按照一定规则进行比对,提取差异内容并制作成补丁文件;3)制作注数块。对补丁文件进行分割与封装,将其转换为满足星地遥控数据格式约束的注入数据块序列;4)遥控上注。通过星地遥控链路,将更新数据块逐块上注至卫星。星上过程包含 3 个步骤,分别为:1)解析注数块。解析并拼接更新数据块,恢复出补丁文件;2)在轨合成。

根据补丁文件内容对参考版加载文件进行修改,形成新版加载文件;3)重加载。对采用非在线更新的星载计算机,还需通过重加载操作实现软件版本切换。

1.2 加载文件设计

加载文件服务于卫星程序装载过程,由可执行文件裁剪而来。在 DSP 软件开发环境 CCS (code composer studio) 中,可执行文件又称 .out 文件,该文件中除包含软件独立运行的所有信息外,还包含用于保持通用性的内容。对采用静态加载的星载计算机,程序装载只关心入口地址和各初始化段内容,因此可将 .out 文件裁剪成图 2 所示文件格式。

入口地址	段总数	段1地址	段1大小	段1内容	...	段K地址	段K大小	段K内容
------	-----	------	------	------	-----	------	------	------

图 2 加载文件格式

Fig. 2 Format of loading file

图 2 中“入口地址”(4 Byte)表示程序运行时的起始地址;“段总数”(4 Byte)表示程序加载时所需的代码段及初始化数据段总数,该值记为 K ;“段地址”(4 Byte)、“段大小”(4 Byte)及“段内容”分别为该数据段/代码段在程序空间中的起始地址、占用量及具体内容。默认设置下,CCS 开发平台生成的初始化段有 5 个: text、switch、vectors、cinit 及 const 段。开发者可以根据实际应用需求,利用 MEMORY 和 SECTIONS 伪指令对链接命令文件(.cmd 文件)重新编辑来自定义程序空间分配情况,然后利用预编译指令 pragma 在源文件和头文件中用设置函数和全局/静态变量的归属段。

在本文提出的更新方法中,要求软件更新时不改变程序空间划分情况,因此参考版加载文件和更新版加载文件中“段总数”和各对应“段地址”相同。

1.3 补丁文件设计

语句级更新方法的核心在于获取参考版加载文件和更新版加载文件之间的差异内容。从图 2 所示的加载文件格式可知,两版加载文件之间的差异主要来自各“段内容”字段,因此高效提取该字段的差异是降低更新数据量的关键。

参考段序列与更新段序列分别设为 $T = t_1, t_2, \dots, t_n$, $S = s_1, s_2, \dots, s_m$, 其中 n, m 分别为参考段和更新段的大小。利用同地址比较法进行差异内容提取时,差异内容规模表示为

$$F(T, S) = F(\{t_i | 1 \leq i \leq n\}, \{s_i | 1 \leq i \leq m\}) = \sum_{i=1}^{\min(n, m)} f(t_i, s_i) + m - \min(n, m) \quad (1)$$

式中 $f(x, y)$ 为差异函数,定义为

$$f(x, y) = \begin{cases} 0, & x = y \\ 1, & x \neq y \end{cases}$$

对星载软件在轨更新而言,差异内容的规模直接决定了更新上注的数据量。若进一步将 T 表示为 $T = t_1, \dots, t_k, t_{k+1}, \dots, t_{k+w}, t_{k+w+1}, \dots, t_n$, 那么利用同地址比较方法得到的差异内容规模与更新方式之间存在如下关系:

1) 替换内容。当更新时将 T 的第 $k+1 \sim k+w$ 字节内容替换成 $x_1, \dots, x_w$, ($t_{k+i} \neq x_i, 1 \leq i \leq w$), 形成的更新段 $S = t_1, \dots, t_k, x_1, \dots, x_w, t_{k+w+1}, \dots, t_n$ 。此时 T 和 S 之间的差异内容规模 $F(T, S)$ 为

$$F(T, S) = F(\{t_i | 1 \leq i \leq k\}, \{t_i | 1 \leq i \leq k\}) + F(\{t_i | k+1 \leq i \leq k+w\}, \{x_i | 1 \leq i \leq w\}) + F(\{t_i | k+w+1 \leq i \leq n\}, \{t_i | k+w+1 \leq i \leq n\}) = \sum_{i=1}^w f(t_{k+i}, x_i) = w \quad (2)$$

可知以内容替换的方式进行更新时,差异内容

规模即为替换内容规模。

2) 新增内容。当更新时在 T 的第 k 字节内容后插入 w 个字节 $x_1, \dots, x_w$, 形成的更新段 $S = t_1, \dots, t_k, x_1, \dots, x_w, t_{k+1}, \dots, t_{k+w}, t_{k+w+1}, \dots, t_n$ 。此时 T 和 S 之间的差异内容规模 $F(T, S)$ 为

$$F(T, S) = F(\{t_i | 1 \leq i \leq k\}, \{t_i | 1 \leq i \leq k\}) + F(\{t_i | k+1 \leq i \leq k+w\}, \{x_i | 1 \leq i \leq w\}) + F(\{t_i | k+w+1 \leq i \leq n\}, \{t_i | k+1 \leq i \leq n\}) = F(\{t_i | k+1 \leq i \leq k+w\}, \{x_i | 1 \leq i \leq w\}) + F(\{t_i | k+w+1 \leq i \leq n\}, \{t_i | k+1 \leq i \leq n-w\}) + F(\emptyset, \{t_i | n-w+1 \leq i \leq n\}) = \sum_{i=1}^w f(t_{k+i}, x_i) + \sum_{i=1}^{n-w-k} f(t_{k+w+i}, t_{k+i}) + w \quad (3)$$

由于 $0 \leq \sum_{i=1}^w f(t_{k+i}, x_i) \leq w$, $0 \leq \sum_{i=1}^{n-w-k} f(t_{k+w+i}, t_{k+i}) \leq n-w-k$, 故有 $w \leq F(T, S) \leq n-k+w$ 。当以内容新增的方式更新时,差异内容规模不小于新增内容规模,规模最大值与参考段内容的长度 n 及新增内容规模 w 成正线性关系,与新增位置偏移 k 成负线性关系。

3) 删除内容。当更新时将 T 的第 k 字节内容后删除 w 个字节,形成更新段 S , 表示为 $S = t_1, \dots, t_k, t_{k+w+1}, \dots, t_n$ 。此时 T 和 S 之间的差异内容规模 $F(T, S)$ 为

$$F(T, S) = F(\{t_i | 1 \leq i \leq k\}, \{t_i | 1 \leq i \leq k\}) + F(\{t_i | k+1 \leq i \leq n-w\}, \{t_i | k+w+1 \leq i \leq n\}) = \sum_{i=1}^{n-w-k} f(t_{k+i}, t_{k+w+i}) \quad (4)$$

可得 $0 \leq F(T, S) \leq n-k-w$, 即当以内容删除的方式更新时,差异内容规模最小值为 0, 最大值与参考段内容的长度 n 成正线性关系,与删除位置偏移 k 和删除内容规模 w 成负线性关系。

根据上述分析可知,利用同地址比较方法进行差异内容提取时,若更新出现内容新增或删除,提取出的差异内容规模受到参考段大小、新增/删除位置偏移及规模相关,易发生数据量激增的现象,不利于星载软件的在轨更新应用。为了克服该缺点,本文引入能够表示序列间相互转换所需的最少编辑操作次数的编辑距离理论^[13],通过计算参考段与序列段间的匹配路径,进行差异内容提取。具体过程如下所述。

序列 T 和 S 之间的编辑距离值 $\text{Lev}(T, S)$ 表示为

$$\text{Lev}(T, S) = \begin{cases} m, & n=0 \\ n, & m=0 \\ \text{Lev}(\text{tail}(T), \text{tail}(S)), & t_1 = s_1 \\ 1 + \min \begin{cases} \text{Lev}(T, \text{tail}(S)), \\ \text{Lev}(\text{tail}(T), S), \\ \text{Lev}(\text{tail}(S), \text{tail}(S)) \end{cases} & \text{otherwise} \end{cases} \quad (5)$$

式中 $\text{tail}(T)$ 、 $\text{tail}(S)$ 分别为序列 T 和 S 除首元素外的尾序列。本文基于动态规划方法计算 $\text{Lev}(T, S)^{[14]}$, 按如下 3 个步骤进行:

步骤 1 建立 $(n+1) \times (m+1)$ 大小的编辑距离矩阵 $D_{(m+1) \times (n+1)} = \{d_{i,j}\} (0 \leq i \leq m, 0 \leq j \leq n)$ 。

步骤 2 设置初项 $d_{0,j} = j (0 \leq j \leq n)$ 和 $d_{i,0} = i (0 \leq i \leq m)$ 。

步骤 3 根据式(6)所示的状态转移公式, 迭代出矩阵 D 中余下元素 $d_{i,j} (0 < i \leq m, 0 < j \leq n)$ 。

$$d_{i,j} = \begin{cases} d_{i-1,j-1}, & s_{i-1} = t_{j-1} \\ 1 + \min\{d_{i-1,j}, d_{i-1,j-1}, d_{i,j-1}\}, & s_{i-1} \neq t_{j-1} \end{cases} \quad (6)$$

式中 $d_{m,n}$ 的值即表示从序列 T 到 S 所需的编辑次数 $\text{Lev}(T, S)$, 算法时间复杂度为 $O(mn)$, 空间复杂度可优化至 $O(\min(m, n))^{[14]}$ 。相比同地址比较法, 基于匹配路径的方法虽然时间复杂度较高, 但该过程只在地面进行, 实际并不影响星载软件的在轨更新应用。此外根据编辑距离的定义可知, 无论更新时发生内容替换、新增还是删除, 所需执行的编辑操作次数均等于更新的字节数, 而编辑操作次数即为差异内容规模, 故本文提出的差异提取方法不会出现更新数据量激增的现象。

为了确定具体编辑位置和编辑方式, 需从矩阵 D 的 $d_{m,n}$ 元素向 $d_{0,0}$ 元素出发, 回溯出序列 T 和 S 的匹配路径^[14], 然后根据匹配路径 $d_{0,0} \rightarrow d_{m,n}$ 上各元素的坐标和取值变化特点生成编辑过程: 1) 当匹配路径呈水平状(行坐标不变)时, 删除序列 T 的当前字节; 2) 当匹配路径呈竖直状(列坐标不变)时, 在序列 T 的当前字节之后插入序列 S 的当前字节; 3) 当匹配路径呈对角状(行、列坐标均变)时, 如果编辑距离值不变, 则无需进行编辑操作, 否则将序列 T 的当前字节改写成序列 S 的当前字节。如图 3 所示展示了序列 $T = \{0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x10\}$, $S = \{0x03, 0x04, 0x06, 0x36, 0x07, 0x09, 0x10, 0x11, 0x12, 0x13\}$ 时的编辑距离矩阵 D 、匹配路径以及 T 到 S 的编辑过程。

由图 3 可知将序列 T 转换为序列 S 需要进行 8 个字节的编辑, 具体编辑过程为: 将序列 T 的第 1、2、5 字节删除, 将第 7、8 字节分别改写成序列 S 的第 4、5 字节, 在第 10 字节之后依次插入序列 S 的第 8、9 和 10 字节。为了保证该编辑过程能够被准确解析, 需在差异补丁中体现对参考内容的编辑位置、编辑方式及编辑后内容; 同时为了减小差异补丁大小, 可将连续字节进行同类型编辑操作的编辑过

程进行合并。因此本文提出的差异补丁的具体格式如图 4 所示。

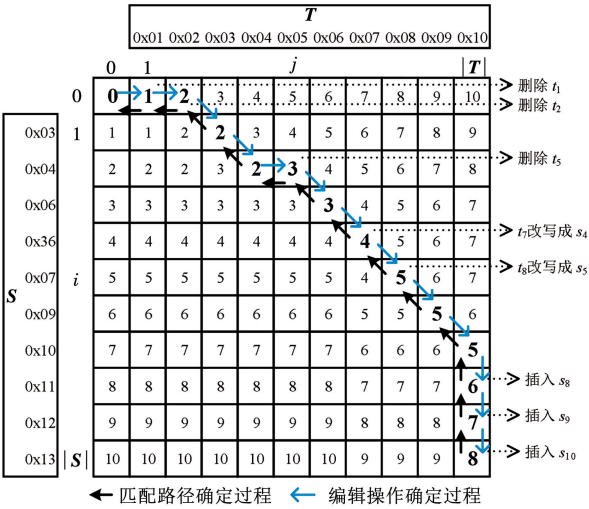


图 3 序列 T 和 S 的编辑矩阵与编辑过程
Fig. 3 Editing matrix and editing process of sequences T and S

地址偏移	编辑类型	编辑长度	更新后内容
------	------	------	-------

图 4 差异补丁格式

Fig. 4 Format of differential patch

“地址偏移”和“编辑长度”分别表示本次编辑的起始地址偏移及编辑字节数, 考虑到编辑距离矩阵求解过程的时间和空间复杂度分别为 $O(nm)$ 和 $O(m)$, 段内容过长时易出现运行时间过长和程序内存不足的问题, 故本文推荐这两个字段均采用 2 字节表示, 对应要求加载文件中各段内容的长度不超过 64 KByte; “编辑类型”(1 Byte)可取值为 0x11、0x22 或 0x33, 分别对应“删除”、“插入”和“改写”操作; “更新后内容”表示插入和改写后的内容。根据该差异补丁格式, 图 3 所示的编辑过程对应的十六进制码流为: $\{0x0000, 0x11, 0x0002; 0x0004, 0x11; 0x0001; 0x0006, 0x33; 0x0002, 0x36, 0x07; 0x000A, 0x22, 0x0003, 0x11, 0x12, 0x13\}$ 。

上述内容描述了参考版与更新版加载文件中, 段内差异补丁制作方法与过程, 而加载文件间可能的差异字段还包括“入口地址”和各“段大小”; 此外为了兼容回滚操作, 补丁文件中需指明补丁类型, 回滚补丁中需给出回滚版本号, 更新补丁中需给出参考版本号和更新版本号。综合上述字段需求, 本文提出的完整补丁文件格式如图 5 所示。

图 5 中“补丁类型”(1 Byte)指明是回滚补丁或更新补丁; 如果是回滚补丁, 则“版本号 1”(1 Byte)表示回滚操作的目标版本号, 补丁文件中无可选字段; 如果是更新补丁, 则“版本号 1”表示参考版本

的特殊字段。首先要求星上收到的第 1 条数据块必须为编号 0 的数据块,原因在于只有编号 0 的数据块中包含补丁类型、版本号和数据块总数等信息;其次要求当前数据块序号 i 应小于总数据块个数 I ;最后还要求补丁文件中“版本号 1”字段(即回滚版本号或参考版本号)在星上已存版本号范围之内,“版本号 2”字段(即更新版本号)等于星上已存最新版本号加 1。

2)数据存储方面。设计使用 NAND Flash 循环存储各版本加载文件,并使用 ReRAM 存储版本管理信息,包括 NAND Flash 中已存各版本加载文件的地址和对应的版本号,以及下次软件加载的版本号。由于差异更新需要基准版本,因此将 NAND Flash 的代码存储区分为不可擦除区和可擦除区,不可擦除区在地面阶段烧入第 0 版本加载文件作为差异更新的可用基准,可擦除区采用循环模式存储在轨生成的各版加载文件。NAND Flash 和 ReRAM 中数据均采用三模冗余存储方式。

3)过程状态监视方面。设计了 8 个遥测量共 18 字节,用于保证地面能够实时掌握在轨更新进度和程序执行状态,进而配合星上过程顺利完成软件更新或回滚。这 8 个遥测量依次为:

- a)合法性检查结果(1 Byte):0x00 合法;0x11 校验码异常;0x22 关键字段提取异常;0x33 版本号 1 异常;0x44 版本号 2 异常;0x55 序号字段异常。
- b)在轨更新运行状态(1 Byte):0x00,完成;0x11,未完成;0x22,数据块非法;0x33,参考版加载文件读取失败;0x44,新版加载文件写入失败。
- c)已存最旧版本号(1 Byte)。
- d)已存最新版本号(1 Byte)。
- e)下次加载版本号(1 Byte)。
- f)所缺数据块条数(2 Byte)。
- g)所缺前 5 个数据块 ID(10 Byte)。

1.6 版本引导程序设计

为了实现图 1 所示的在轨更新星上过程步骤 3),即软件版本切换,本文中设计了版本引导程序,整体流程如图 8 所示。

根据图 8 所示,执行版本引导程序时,首先进行 DSP 和存储器(NAND Flash 和 FeRAM)的初始化;接着从 FeRAM 中读取软件版本管理信息和目标版本号信息,如果这两种信息任一获取失败,则从 NAND Flash 中读取版本 0 的加载文件,否则读取目标版本的加载文件;最后将加载文件内容分段装载至程序空间,并将程序入口地址装载至程序计数器,开始运行卫星程序。

版本引导程序的可靠性设计主要体现为读取

NAND Flash 和 FeRAM 中数据时采用三模冗余操作,且任意数据获取异常时,星上能够加载版本 0 代码回到默认稳定状态。

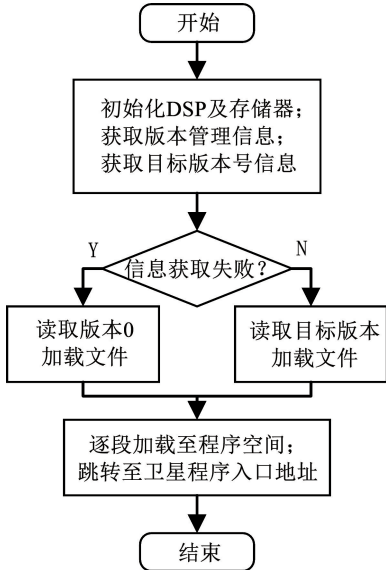


图 8 版本引导程序执行流程

Fig. 8 Operation process of version guidance program

2 实验结果

本文实验基于 ZDPS-3A 卫星星载计算机平台,对所提出的软件在轨更新方法进行试验,主要针对更新/回滚操作执行情况,以及更新数据量等项目进行测试。虽然实验在地面进行,但过程模拟实际在轨状态,数据块均通过地面站以 2 000 bps 的 PCM 遥控模式上注,并在实验开始前将版本 0 的加载文件烧入 NAND Flash 不可擦除区。

2.1 更新与回滚操作实验

实验在版本 0 的基础上进行 15 次修改,形成编号为 1~15 的更新版本。表 1 显示了 24 次更新与回滚操作的实验结果,其中列“Id”为滚回版本号或更新版本号,列“Idr”为参考版本号,列“Idc”、“Idc'”分别为操作前和操作后星上运行的版本号,列“Ido”、“Idn”分别为操作之前 NAND Flash 中可擦除区保存的最旧和最新加载文件版本号。

表 1 数据显示经过 24 次更新/回滚操作实验,更新成功与失败次数分别为 15 次和 2 次,回滚成功与失败次数分别为 5 次和 2 次。进一步分析各次操作结果可知:

- 1)当更新/回滚操作满足在轨更新监视程序的合法性设计时,操作能正常执行;
- 2)执行更新操作时,如果新版加载文件版本号不连续,或参考版本号不在星上存储范围时,星上不执行更新操作,而保持当前版本运行,该现象符合在轨监视程序的可靠性设计;

3) 执行回滚操作时,如果回滚版本号不在星上存储范围时,星上会自动加载版本 0 程序以保证软件安全,该现象符合版本引导程序的可靠性设计。

表 1 软件更新操作执行结果

Tab. 1 Results of software update operation								
序号	操作	Id	Idr	Idc	Ido	Idn	Idc'	结果
1	更新	1	0	0	0	0	1	成功
2	更新	2	0	1	1	1	2	成功
3	更新	3	0	2	1	2	3	成功
4	回滚	0	-	3	1	3	0	成功
5	回滚	2	-	0	1	3	2	成功
6	更新	5	3	2	1	3	2	失败
7	更新	4	3	2	1	3	4	成功
8	更新	5	3	4	1	4	5	成功
9	更新	6	5	5	1	5	6	成功
10	回滚	4	-	6	1	6	4	成功
11	更新	7	6	4	1	6	7	成功
12	更新	8	6	7	1	7	8	成功
13	更新	9	8	8	1	8	9	成功
14	更新	10	9	9	1	9	10	成功
15	更新	11	9	10	1	10	11	成功
16	更新	12	7	11	2	11	12	成功
17	更新	13	12	12	3	12	13	成功
18	回滚	3	-	13	4	13	0	失败
19	回滚	8	-	4	4	13	8	成功
20	回滚	15	-	8	4	13	0	失败
21	更新	14	3	13	4	13	13	失败
22	更新	14	5	13	4	13	14	成功
23	更新	15	5	14	5	14	15	成功
24	回滚	7	-	15	6	15	7	成功

上述实验结果表明本文提出的软件更新方法能够可靠地实现软件更新和回滚。

2.2 差异更新效果实验

由于回滚操作仅需要 1 条注数块,因此本实验仅针对更新操作。实验以传统的同地址比较法为比

较对象,分析了本文提出的利用匹配路径进行差异提取的方法在更新数据量方面的优势。

实验时首先在版本 0 的基础上进行修改,逐步生成 5 个新版本加载文件;然后以版本 0 为参考对象,进行版本 1~5 的更新。如图 9 所示显示了传统差异提取方法和本文差异提取方法在不同版本更新过程中所生成的补丁文件规模。

图 9 表明利用版本 0 进行版本 1~5 的更新时,相比传统差异提取方法,本文方法能够显著减少补丁文件规模。表 2 列出了以不同版本为参考进行版本 1~5 更新时,传统方法与本文方法在更新数据量、注入数据块数量及测控资源需求等 3 个项目上的实验结果。表 2 在计算测控资源需求量时,参考 ZDPS-3A 卫星平均每个测控圈的实际发令情况,设置每个测控圈中可用于上注在轨更新数据块的时长和发令间隔分别为 6 min 和 8 s,因此每圈可上注约 45 条在轨更数据块,当需求量不足整圈时,按整圈向上取整。

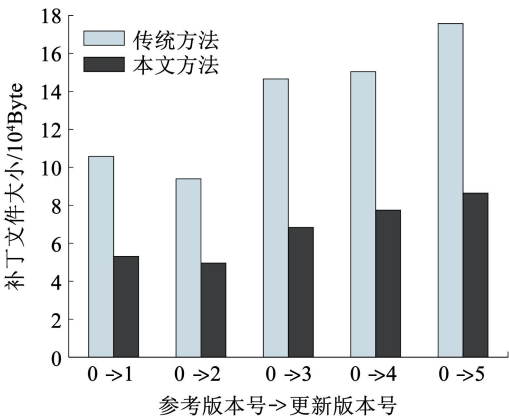


图 9 更新补丁规模

Fig. 9 Update patch size

表 2 不同更新方法实验结果

Tab. 2 Results of different update methods											
Id	加载文件 大小/Byte	Idr	补丁文件大小/Byte			上注数据块总数/条			测控资源需求量/圈		
			传统	本文	减少/%	传统	本文	减少/%	传统	本文	减少/%
0	519 153	-	-	-	-	-	-	-	-	-	-
1	515 921	0	105 758	53 170	49.72	429	216	49.65	10	5	50.00
2	517 225	0	93 964	49 588	47.23	381	201	47.24	9	5	44.44
		1	71 982	19 309	73.18	292	79	72.95	7	2	71.43
		0	146 541	68 374	53.34	594	277	53.37	14	7	50.00
3	516 105	1	170 612	72 847	57.30	691	295	57.31	16	7	56.25
		2	119 582	59 348	50.37	485	241	50.31	11	6	45.45
		0	150 380	77 479	48.48	609	314	48.44	14	7	50.00
4	515 761	1	139 836	53 358	61.84	567	217	61.73	13	5	61.54
		2	89 444	37 498	58.08	363	152	58.13	9	4	55.56
		3	69 269	47 642	31.22	281	193	31.32	7	5	28.57
5	516 081	0	175 728	86 441	50.81	712	350	50.84	16	8	50.00
		1	166 088	63 255	61.91	673	257	61.81	15	6	60.00
		2	118 934	48 815	58.96	482	198	58.92	11	5	54.55
		3	95 728	56 693	40.78	388	230	40.72	9	6	33.33
		4	42 591	20 558	51.73	173	84	51.45	4	2	50.00

表 2 实验数据表明,两种差异更新方法得到的补丁文件规模都远小于加载文件规模,因此上注差异补丁比上注完整的加载文件更适合用于星载软件在轨更新。两种差异更新方法比来看,相比使用传统的同地址比较法进行差异内容提取的方式,使用本文提出的基于匹配路径的差异内容提取方式,平均可降低 53.00% 的补丁规模、52.95% 的数据块上注量和 50.74% 的测控圈需求量,因此本文提出的差异内容提取方法更适合应用于语句级差异更新。

3 结 论

1) 本文提出的星载软件更新方法在程序空间优化的基础上,利用匹配路径进行差异内容提取,达到了降低更新数据量的目的。实验结果表明与传统语句级更新方法相比,本文方法可降低超过 50.00% 更新上注数据量,进而降低超过 50.00% 的测控资源需求量,大大缩短了更新上注时间。

2) 通过版本引导程序设计,可实现星载软件版本灵活控制。

3) 文件及程序的可靠性设计,使得更新过程能够屏蔽非法操作,保证了更新过程的安全性。

参考文献

[1] 白照广. 中国现代小卫星发展成就与展望[J]. 航天器工程, 2019, 28(2): 1
BAI Zhaoguang. Development achievements and prospects of China modern small satellite[J]. Spacecraft Engineering, 2019, 28(2): 1. DOI:10.3969/j. issn. 1673-8748. 2019.02.001

[2] TIPALDI M, LEGENDRE C, KOOPMANN I, et al. Development strategies for the satellite flight software on-board Meteosat Third Generation[J]. Acta Astronautica, 2018, 145: 482. DOI: 10.1016/j. actaastro. 2018.02.020

[3] JI Xinyan, LI Yunze, LIU Guoqing, et al. A brief review of ground and flight failures of Chinese spacecraft[J]. Progress in Aerospace Sciences, 2019, 107: 19. DOI:10.1016/j. paerosci. 2019.04.002

[4] 钱方亮, 林荣锋, 周宇, 等. 一种基于微小卫星系统软件在轨编程功能的设计方法[J]. 计算机应用与软件, 2018, 35(12): 16
QIAN Fangliang, LIN Rongfeng, ZHOU Yu, et al. Design of on-orbit programming for microsatellite system software[J]. Computer Applications and Software, 2018, 35(12): 16. DOI: 10.3969/j. issn. 1000-386x. 2018.12.004

[5] 汪宏浩, 王慧泉, 金仲和. 基于增量链接的可回滚星载软件在轨更新方法[J]. 浙江大学学报(工学版), 2015, 49(4): 724
WANG Honghao, WANG Huiquan, JIN Zhonghe. Rollback-able on-board software upgrade method based on incremental link[J].

Journal of Zhejiang University (Engineering Science), 2015, 49(4): 724. DOI: 10.3785/j. issn. 1008-973X. 2015.04.017

[6] 王绍举, 郑晓云, 徐拓奇. 星载软件在轨重注技术[C]//中国空间科学学会 2013 年空间光学与机电技术研讨会会议论文集. 西安: 中国空间科学学会, 2013: 511
WANG Shaoju, ZHENG Xiaoyun, XU Tuoqi. The technology of re-injection for satellite software on orbit [C]// Proceedings of the 2013 Symposium on space optics and Mechatronics. Xi'an: Chinese Society of Space Sciences, 2013: 511

[7] 史毅龙, 薛长斌. 基于“龙芯”的 VxWorks 系统函数在轨更新研究[J]. 电子设计工程, 2015, 23(21): 106
SHI Yilong, XUE Changbin. VxWorks on-orbit renewal research based on loognson [J]. Electronic Design Engineering, 2015, 23(21): 106. DOI: 10.3969/j. issn. 1674-6236. 2015.21.032

[8] 刘赞, 左小川. 嵌入式软件在线升级系统的设计与实现[J]. 计算机测量与控制, 2015, 23(4): 1425
LIU Yun, ZUO Xiaochuan. Design and implementation of online software updating for embedded system[J]. Computer Measurement & Control, 2015, 23(4): 1425. DOI: 10.3969/j. issn. 1671-4598. 2015.04.108

[9] 徐伟, 朴永杰. 航天相机控制器在轨软件重注[J]. 光电工程, 2013, 40(4): 65
XU Wei, PIAO Yongjie. Re-injection technology for software in aerospace camera controller on orbit [J]. Opto-Electronic Engineering, 2013, 40(4): 65. DOI: 10.3969/j. issn. 1003-501X. 2013.04.010

[10] 朱虹, 王海燕. 一种星载软件在轨编程功能的设计和实现技术[J]. 上海航天, 2004, 21(1): 26
ZHU Hong, WANG Haiyan. The technology of design and realization of on-board programming for AOCS [J]. Aerospace Shanghai, 2004, 21(1): 26. DOI: 10.3969/j. issn. 1006-1630. 2004.01.006

[11] 李雁斌, 吉峰, 黄勇. 一种星载 DSP 软件的在轨编程方法[J]. 制导与引信, 2011, 32(4): 37
LI Yanbin, JI Feng, HUANG Yong. The design of on-board programming for satellite DSP software [J]. Guidance & Fuze, 2011, 32(4): 37. DOI: 10.3969/j. issn. 1671-0576. 2011.04.010

[12] 李振松, 杜建伟, 党纪红. 基于 DSP 平台的航天器软件在轨维护实现方法研究[J]. 空间控制技术与应用, 2017, 43(6): 61
LI Zhensong, DU Jianwei, DANG Jihong. Implementation of on-board software maintenance for spacecraft based on DSP [J]. Aerospace Control and Application, 2017, 43(6): 61. DOI: 10.3969/j. issn. 1674-1579. 2017.06.010

[13] LEVENSHTEIN V I. Binary code capable of correcting deletions, insertions and reversals [J]. Soviet Physics Doklady, 1966, 10(8): 707

[14] NAVARRO G. A guided tour of approximate string matching[J]. ACM Computing Surveys, 2001, 33(1): 31. DOI: 10.1145/375360.375362001

(编辑 张 红)