民用机载软件集成过程的技术研究①

宫伟祥[1], 赵　嫚[2]

[1](中电科航空电子有限公司, 成都 611731)
[2](卫士通信息产业股份有限公司, 成都 610041)

摘　要: 民用机载软件的研制以 DO-178B 标准为指导. 与传统的软件工程相比, DO-178B 标准更面向目标和过程. 该标准为各个等级的软件提出了相对应的目标, 申请者需要向局方提供证据以表明研制的软件满足适航目标. 软件的编码和集成过程, 该标准要求集成过程的输出是正确和完整的. 但该标准中并没有提出如何满足这个目标. 本文通过无线电调谐软件对软件编码和集成过程以及对软件编译和链接过程的研究, 提出一种方式来满足 DO-178B 标准的这一目标.

关键词: 机载软件; 集成过程; 适航目标; 无线电调谐; 局方

Research on Integration Process of Civil Airborne Software

GONG Wei-Xiang[1], ZHAO Hua[2]

[1](China Electronics Technology Group Avionics Corporation, Chengdu 611731, China)
[2](Westone Information Industry Inc. Chengdu 610041, China)

Abstract: DO-178B standard has provided the guidelines for Civil Airborne Software Development. Compared with traditional software engineering, DO-178B standard is object-oriented and process-oriented. This standard requires related objects for different level software. The applicant should provide the evidences to the certification authority to show the software complies with the airworthiness objects. As for the software coding and integration process, DO-178B requires the output of software coding and integration process is correct and completed. But this standard doesn't provide how to meet with the objects. By radio tuning software coding and integration process and a research on the compiling and linking process explains how to comply with this object of DO-178B.

Key words: airborne software; integration process; airworthiness object; radio tuning; authority certification

随着 ARJ21 和 C919 等大飞机项目的相继研制, 国内的民机产业得到快速发展, 研制能力得到进一步提升, 民机软件的研制的重要性也越来越得到重视. 目前国内机载软件研制主要参考 RTCA DO-178B[1]标准, 该标准是面向目标和面向过程的. 该标准根据系统安全性的要求对机载软件进行等级划分并为不同等级的软件提出不同数量的研制目标.

非机载软件研制过程中软件开发环境大多采用集成开发环境(IDE)[2], 软件的编译和链接的过程极少被关注. IDE 提供的默认配置、编译和链接参数足以支持编译和链接过程. 但机载软件作为高安全性和高可靠性软件, 软件开发过程应确保源代码被正确编译和链接并且没有引入错误. 这要求软件在进行编译和链接工作时应分析编译器和链接器输出.

DO-178B 标准要求集成过程的输出是正确的和完整的, 但标准中未提供进一步的说明如何实现该目标以及如何提供相应的证据满足目标, 给申请者在机载软件的合格审定造成困扰. 本文将描述 DO-178B 的目标对机载软件编译和链接过程的要求以及如何向局方提供相应的证据满足上述的目标.

1　相关概述

1.1 集成过程

DO-178B 是面向过程的标准, 将软件研制过程划

① 收稿时间:2015-10-29;收到修改稿时间:2015-11-25　[doi: 10.15888/j.cnki.csa.005201]

分为软件计划过程, 软件开发过程以及综合过程三个部分, 其中软件开发过程又细分为软件需求过程, 软件设计过程, 软件编码和集成过程.

集成过程主要活动包括编译、链接和加载数据. 该过程根据提供的源代码生成目标代码, 可执行目标代码. 软件集成活动包括:

① 源代码完成编译、链接产生目标代码和可执行目标代码

② 完成软件集成工作

③ 软件加载到目标机环境用于验证软硬件集成

局方在 SOI#3 阶段将对软件编码和集成过程中的输出数据进行审查. 这一阶段引入错误的过程主要为软件编译过程和链接过程.

1.2 编译过程

编译器是将高级语言翻译成机器语言的工具. 编译器读取源程序, 进行语法和词法分析, 生成汇编代码. 源代码生成可执行目标代码一般需要经过四个步骤:预处理, 编译, 汇编和链接[3], 如图 1 所示.

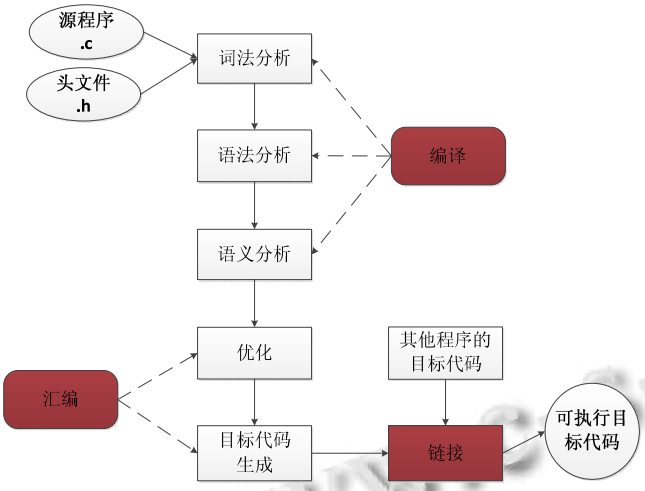

图 1 源代码编译过程

预编译过程将源代码文件和相关的头文件被编译器预编译成一个.i 文件. 预编译过程主要处理那些源代码中以“#”开始的预编译指令, 比如“#include”、“#define”等, 主要的处理规则如下:

① 处理宏定义指令: 展开所有的“#define”宏定义

② 处理所有条件预编译指令: 如“#if”, “#ifdef”等指令

③ 处理“#include”头文件包含指令: 将被包含的文件插入到该预编译指令的位置

④ 删除源文件中的所有注释

⑤ 添加行号和文件名标识: 用于编译时编译器产生调试用的行号信息

⑥ 保留所有的#pragma 编译器指令

汇编过程是将汇编代码生成机器可以执行的指令, 每一个汇编语句计划都对应一条机器指令.

链接主要将各个模块之间能够正确的衔接, 生成可执行的目标文件. 链接的过程包括地址和空间的分配, 符号绑定和重定位[4].

2 解决方案

DO-178B 标准要求软件开发过程中引入的错误被检测并消除. 软件集成过程中引入的错误包括编译器错误, 链接器错误以及加载错误三种, 如图 2 所示.

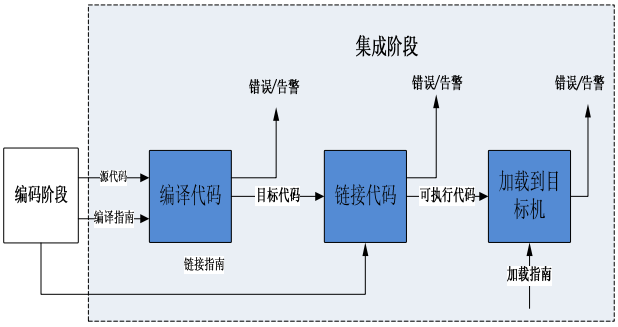

图 2 软件集成过程错误引入说明

软件集成过程的输出是通过开展评审和分析来以保证软件集成过程中引入的错误已经被检测并被消除. 对于集成过程输出的考虑, DO-178B 主要包含以下方面:

① 编译器告警(DO-178C[5]要求)

② 不正确的硬件地址

③ 内存重叠

④ 遗漏软件部件

集成过程的评审和分析工作通过详细检查开发环境代码编译器告警信息, 部件链接、内存映象和加载数据开展.

2.1 告警分析

编译器告警信息分析应首先关注软件编码阶段, 高质量的代码可以减少编译器告警信息的产生. 源代码完成在编译过程中的告警信息应确认和分析.

① 制定软件编码标准

软件编码标准中应针对本项目所需要的编码特性

进行要求. 例如对于系统的标准库使用, 动态内存分配机制, 垃圾回收机制的使用等. 软件编码标准应经过评审, 并向软件开发工程师进行宣贯, 确保软件编码标准的执行性[6].

② 执行软件编码标准

开发者应严格遵循软件编码标准编写源代码. 严格执行编码标准可以使源代码风格统一, 减少编码错误进而降低软件缺陷数量, 降低后续的软件维护难度, 提升软件的质量.

③ 确认和分析编译器生成的告警信息

设置编译器告警级别. 编译时记录软件编译时产生的所有编译器告警信息并对所有的告警进行确认. 编译器产生的告警信息包括编译告警提示和编译错误.

机载软件要求源代码编译过程中不应出现编译错误和编译告警信息. 编译告警和错误信息代表源代码在编译过程中出现故障应修改源代码.

2.2 部件遗漏分析

DO-178B 标准要求检查软件部件存在遗漏的情况应通过分析工程文件的编译链接规则. 通常软件工程的编译和链接通过工程文件进行组织(如 MakeFile 文件). 工程文件定义源代码之间的依赖关系, 实现自动化编译工作. 对于部件遗漏的分析可以通过对工程文件的评审完成, 评审过程中应确认以下几个方面:

① 确认正确的源代码文件被包含在工程文件中

② 源代码文件之间的依赖关系正确

③ 确保所有源文件都生成目标文件

④ 确保目标文件都被链接到可执行目标文件

2.3 内存重叠分析

DO-178B 中内存重叠要求从程序的链接和内存映射分析进行. 验证软件模块都被正确的映射到链接器命令文件(Linker Command File)定义的段(Segment)信息:包括各个段的起始地址, 长度, 结束地址等. 检查内存映射文件以验证以下的内容:

① 链接器命令文件中定义的各个段的最大长度, 属性被链接器正确的创建

② 分配的各个段之间没有地址重叠

③ 每个段实际大小应不大于链接器命令文件中定义的大小

④ 链接器将源代码模块中不同的部分分别映射到正确的段中

⑤ 仅产生来自源文件的目标模块

⑥ 仅产生来自目标模块中的函数, 变量等

通常代码中的数据会存放在以下几个段:

①.text:存放程序代码数据

②.data:存放已经初始化的全局静态变量和局部静态变量

③.bss:存放未初始化的静态全局变量和局部静态变量

④.common:未初始化的全局变量(此段与.bss 重复, 具体的存放段视编译器而定)

可执行代码通过反汇编的方式对各个段内的数据进行分析和确认内存地址无重叠区域, 段大小满足预先分配的要求.

2.4 加载分析

加载分析目的为确认可执行文件被正确的加载到目标计算机中. 加载分析应与链接分析联合开展.

软件加载过程应遵循经过批准的加载规范并且应记录加载日志. 软件的加载过程可以通过Flash烧写的方式也可以通过数据加载方式进行. 加载过程中应检查加载软件的正确性, 包括数据有效性检查(可采用 CRC 校验). 软件正确加载到目标机环境上后需要检查和记录软件的部件号和版本号信息以确认软件正确加载. 加载分析用于分析和确认以下内容:

① 所有的软件部件都构建并且被加载到目标机的正确位置

② 无效的软件没有被加载到目标机

③ 不正确或错误的软件不能执行

④ 加载数据是正确的

3 解决方案

无线电调谐单元软件为 C 级软件. 软件的研制过程分为软件计划过程, 软件需求过程, 软件设计过程, 软件编码和集成过程[7].

无线电调谐软件软件集成过程通过对编译器输出, 工程文件 makefile, 链接器命令文件, 内存映射分析, 软件加载分析确保集成过程满足 DO-178B 的要求. 在软件加载完成后启动软件集成过程评审活动, 为局方提交正式证据.

3.1 编译器告警

编译器告警通过代码符合性检查和检查编译器输出提供相应的证据以满足此要求.

① 代码符合性检查

无线电调谐单元在软件计划阶段制定了软件编码标准, 并将编码标准录入到 TestBed 工具. 在软件正式编译之前, 通过 TestBed 工具对代码规范进行检查. TestBed 可以快速检测出代码中不符合编码规则的源代码. 工具分析后的源代码可极大减少编译器产生的告警信息. 图 3 为工具对代码符合性检查产生的结果信息. 其中标红的部分为不符合项, 应更新对应的源代码.

Number of Violations	LDRA Code	Required Standards	MISRA-C:2004 Code
37	11 S	No brackets to loop body.	MISRA-C:2004 14.8
72	12 S	No brackets to then/else.	MISRA-C:2004 14.9
2	13 S	goto detected.	MISRA-C:2004 14.4
0	15 S	Anonymous field to structure.	MISRA-C:2004 1.2
0	21 S	Number of parameters does not match.	MISRA-C:2004 16.6
8	32 S	Use of continue statement.	MISRA-C:2004 14.5
0	36 S	Function has no return statement.	MISRA-C:2004 16.8
78	37 S	Procedure Parameter has a type but no identifier.	MISRA-C:2004 16.3
0	39 S	Unsuitable type for loop variable.	MISRA-C:2004 13.4
2	41 S	Ellipsis used in procedure parameter list.	MISRA-C:2004 16.1
0	43 S	Use of setjmp/longjmp.	MISRA-C:2004 20.7
18	44 S	Use of banned function or variable.	MISRA-C:2004 20.1,20.4,20.5,20.6,20.10,20.11
3	47 S	Array Bound exceeded.	MISRA-C:2004 21.1
22	48 S	No default case in switch statement.	MISRA-C:2004 15.3
763	49 S	Logical conjunctions need brackets.	MISRA-C:2004 12.1,12.5
18	50 S	Use of shift operator on signed type.	MISRA-C:2004 12.7
3	51 S	Shifting value too far.	MISRA-C:2004 12.8

图 3 代码符合性表

② 检查编译输出信息

无线电调谐单元使用 workbench 编译环境, 源代码通过编码阶段产生的编译指南进行编译, 源代码的编译输出信息如图 4 所示. 所有的编译输出信息应经过分析, 确认编译过程中不存在编译错误和告警信息.

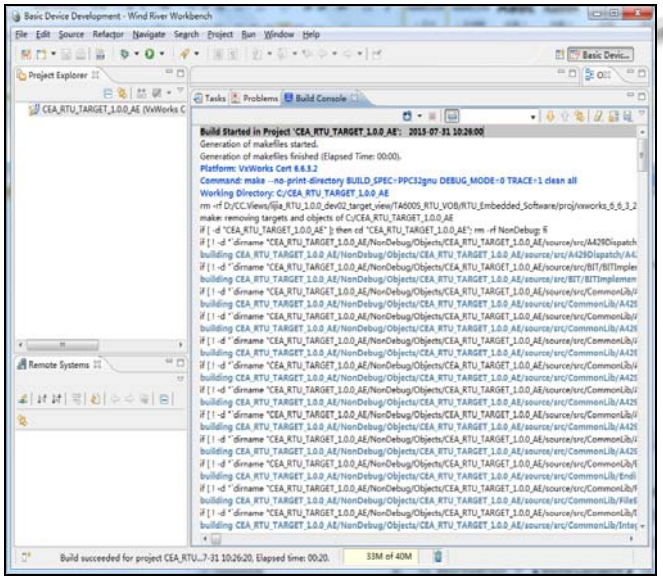

图 4 编译日志信息

3.2 遗漏软件部件

无线电调谐单元编译环境自动生成工程文件 MakeFile 文件. 因此需要对该 MakeFile 工程文件数据进行验证, 以确保该工程文件中包含的本项目所有的源代码文件, 源代码文件生成正确的目标文件, 没有遗漏的目标文件. 该工程文件按照 DO-178B 规定的控制类别进行管理, 该文件通过人工评审的方式进行确认, 评审报告将作为证据提交给局方进行审查. 无线电调谐单元的工程文件的部分内容如图 5 所示.

图 5 工程文件

3.3 内存重叠

无线电调谐单元中采用基于 PowerPC 的计算机, 链接器命令文件采用默认的链接器命令文件, 如图 6 所示.

```
ENTRY(_start)
SECTIONS
{
  .text     0 :
  {

    _VX_START_TEXT = .;
    *(.text)

    *(.text.fast)

    *(.text.init)
    *(.text.*) *(.stub) *(.gnu.warning) *(.gnu.linkonce.t*)
    KEEP(*(.init)) KEEP(*(.fini))
    *(.glue_7t) *(.glue_7) *(.rdata)            /* for ARM */
    *(.mips16.fn.*) *(.mips16.call.*) *(.reginfo)   /* for MIPS */
    *(.rodata) *(.rodata.*) *(.gnu.linkonce.r*) *(.rodata1)

  }
  .sdata2     0 : {
    _SDA2_BASE_ = . + 0x7ff0 ;
    *(.sdata2) *(.sbss2)                /* for PPC */

  }

  .data   0 :
  {

    _VX_START_DATA = .;
    *(.data)
    *(.data.fast)
    *(.data.init)
    *(.data.*) *(.gnu.linkonce.d*) SORT(CONSTRUCTORS) *(.data1)
    *(.eh_frame) *(.gcc_except_table)
    KEEP (*crtbegin.o(.ctors))
    KEEP (*(EXCLUDE_FILE (*crtend.o) .ctors))
    KEEP (*(SORT(.ctors.*)))
    KEEP (*(.ctors))
    KEEP (*crtbegin.o(.dtors))
    KEEP (*(EXCLUDE_FILE (*crtend.o) .dtors))
    KEEP (*(SORT(.dtors.*)))
    KEEP (*(.dtors))
    /* got*, dynamic, sdata*, lit[48], and sbss should follow _gp */
    *(.got.plt) *(.got) *(.dynamic)
    *(.got2)
    *(.lit8) *(.lit4)

  }
```

图 6 链接器控制文件

无线电调谐单元软件编译环境对各个段的大小未进行预先分配. 段以“.text”段从地址 0 开始, 程序代码依次存放在该段中. “.text”段结束后, 其他段依次进行分配. 该链接器控制文件确保段之间不存在地址重叠现象.

各个段之间不存在重叠后, 应检查段内是否存在内存重叠. workbench 开发环境提供了反编译命令“nmppc”, 可显示关于对象文件、可执行文件以及对象文件库里的符号信息. 图 7 为通过使用“nmppc”命令输出的符号信息.

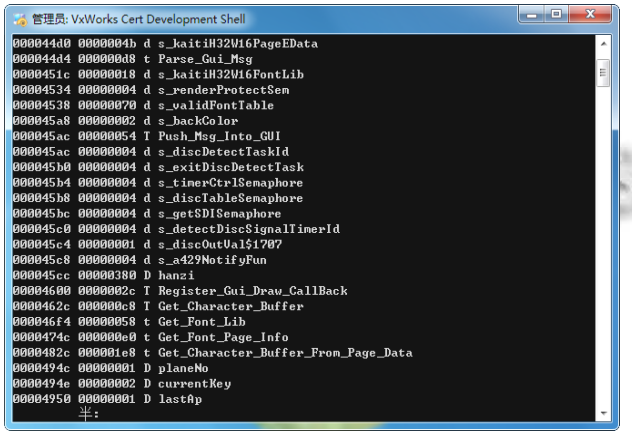

图 7　内存映射分析

检查各个段内的函数或变量地址是否存在重叠现象. 其中各个段的名称如下:

① B:该符号放在 BSS 段中, 通常是那些未初始化的全局变量

② D:该符号放在普通的数据段中, 通常是那些已经初始化的全局变量

③ T:该符号放在代码段中, 通常是那些全局非静态函数

④ U:该符号未定义过, 需从其他对象文件链接

⑤ W:未明确指定的弱链接符号

3.4 加载过程

可执行文件生按照加载指南对加载到目标硬件. 可执行目标文件通过数据加载器软件(该软件满足 ARINC 615A-3[8]标准)加载到目标计算机中. 加载前应记录软件的版本信息. 加载完成后应生成加载日志, 从以下方面对加载过程进行确认:

① 确认加载过程没有出现错误

② 只有规定的文件被加载到无线电调谐设备

③ 可执行文件被加载到制定的位置

④ 可执行文件可以在设备上运行

加载完成后应启动无线电调谐单元设备, 检查软件的显示的版本信息是否与记录的信息一致.

3.5 集成过程评审

完成上述的活动后, 可以针对软件集成过程召开软件集成过程评审会. 该评审会将对集成过程中产生的所有数据进行评审. 评审数据包括软件编码符合性以及编译输出信息, 链接器控制命令文件, 内存映射结果以及加载日志.

上述数据应按照 DO-178B 规定的控制类别实施构型管理并作为证据提供给局方检查. 上述数据与评审报告为 DO-178 中关于软件集成过程的输出是正确的和完整的这一目标提供证据.

4　结语

本论文介绍了一种满足 DO-178B 标准在软件编码和集成过程目标的方法以及针对此目标开展的活动以及提供相应的数据. 通过无线电调谐单元这一实例说明软件集成过程中如何开展的任务活动, 包括编译器告警信息分析、链接和内存映射分析, 加载分析确保该过程中产生的错误都被标识和消除. 经过对该过程产生数据的评审提供了证据.

在无线电调谐单元开展过程中, 软件的集成过程是一个迭代的过程, 需要多次的集成过程的验证活动. 软件集成过程产生的所有数据应严格进行构型管理控制, 为局方提供了可信的证据.

参考文献

1 Software Considerations in Airborne Systems and Equipment Certification. RTCA/DO-178B, Washington: RTCA, Inc, 1992: 21−51.

2 俞甲子,石凡,潘爱民.程序员的自我修养-链接、装载与库. 北京:电子工业出版社,2009.4.

3 杨旭东,火善栋.论 C/C++内存管理中静态区、栈和堆的相互关系.重庆三峡学院学报,2013,23(145):40−42.

4 奚琪,曾勇军,王清贤,吴红水.一种动静结合的代码反汇编框架.小型微型计算机系统,2013,34(10):51−55.

5 Software Considerations in Airborne Systems and Equipment Certification. RTCA/DO-178C, Washington: RTCA, Inc, 2011: 34−51.

6 Rierson L. Developing Safety-Critical Software A Practical Guide for Aviation Software and DO-178C Compliance. Taylor & Francis Group, ISBN: 978-1-4398-1369-0, 2013: 179−181.

7 沈小明,陆国荣,王云明,蔡喁,欧阳坡.机载软件研制流程最佳实践.第一版.上海:上海交通大学出版社,2013.12:75−85.

8 Aeronautical Radio Inc. ARINC 615A-3 Software Data Loader Using Ethemet Interface. ARINC, 2007.