维普资讯 http://www.cqvip.com

计算机科学2001Vol.28№.6

Ada语言实时绑定库的设计与实现

The Design and Implementation of Ada Real-Time Bindings

郭 兵 熊光泽 陈 宇

(电子科技大学计算机科学与工程学院 成都610054)

Abstract Ada language possesses very high dependability and real-time. When a large, complex embedded syetem is developed in Ada, the running support of RTOS(Real-Time Operating System) is required, that is to say, Ada real-time bindings is required. This paper emphasizes to research the issue of design and implementation of Ada bindings on a RTOS, including its structure, features and key technology in its implementation.

Keywords Ada, Bindings, RTOS, Embedded system development

1 引言

嵌入式实时系统(下称嵌入式系统)的开发一般按照软件工程生命周期的瀑布模型来进行,并可考虑快速原型方法。在嵌入式系统的代码实现过程中,选择编程语言成为关键,因为它关系到所选语言的语言特征能否很好地支持嵌入式系统的有关特性,即该语言是否能够支持嵌入式系统设计的6个重要准则:安全性、可靠性、适应性、简单性、可移植性和有效性[2]。

Ada语言最初是由美国国防部在八十年代初主持开发的一种面向对象编程的语言,主要用于实现大型、复杂的高可靠性的软件系统,除了用于实时系统外,还在其他许多领域中得到广泛使用,包括大范围的信息系统、分布式系统、科学计算和系统编程等方面。Ada 95是ISO和ANSI于1995年制定的最新Ada标准,与Ada 83标准兼容,是第一种国际标准的面向对象编程语言。Ada语言对嵌入式系统开发的支持主要体现在它既满足上述嵌入式系统设计的6个重要准则,还满足用软件工程方法进行软件系统工程化开发的要求,如支持大型程序开发、可靠性和安全性设计、异常处理、并行程序设计、同步通信、实时构造、任务调度、分布式系统支持及低级编程等。因此,Ada语言目前应用非常广泛,美国陆海空三军都积极采用Ada语言,仅美国国防部的作战软件就至少有5000万行程序是用Ada语言编写的。同时,许多国家的国防、航空和航天等部门也广泛使用Ada语言,将Ada语言作为嵌入式软件开发的标准语言,如欧洲航空工业公司生产的A340大型客机,使用了115个数字处理单元,有近20Mb的目标代码,其中大部分代码采用Ada语言编码[7]。

Ada语言用于嵌入式系统开发,运行时需要实时操作系统(RTOS)的支持,以提供一个多任务的运行环境。CRTOS是我们在"八五"、"九五"国防科技预研成果的基础上发展起来的一种国产实时操作系统[3],在军事电子、信息家电、移动计算设备等领域有着成功的应用和良好的发展前景。本文以CRTOS为例,研究实现Ada语言与实时执行体的绑定库(Bindings)(下称Ada语言实时绑定库)技术。在CRTOS上实现Ada语言的绑定库,使CRTOS支持Ada语言开发,对于提高我国嵌入式应用的水平具有重要的意义。

2 Ada语言实时绑定库的设计

2.1 概述

函数库是一种由子函数组成的预先编译好并经过测试的函数集合,提供了许多可重用的公用函数代码。函数库通常以文件的形式存放,库文件是一个可供不同程序使用的共享文件,且是可以直接调用的二进制文件。函数库可分为静态函数库和动态函数库两种。静态函数库是在应用程序编译链接时,将库函数的函数体链接到应用程序中生成可执行程序;而动态函数库是在应用程序运行过程中,需要调用库函数时才将函数体动态链接到应用程序中,这样使得一个函数体代码能够被多个应用程序同时共享调用,在内存中只需保留一份拷贝,从而节省了占用的内存存储空间,但牺牲了应用程序的一些实时特性。函数库在计算机软件

郭 兵 博士研究生,主要研究方向为嵌入式系统软件开发环境和中间件技术。熊光泽 博士生导师,主要研究方向为实时计算机系统及其应用。陈 宇 博士研究生,主要研究方向为实时操作系统及其应用。

维普资讯 http://www.cqvip.com

领域中使用非常广泛，如操作系统的系统服务以函数库的形式提供给上层应用程序的编程接口（API）、编译系统提供的通用函数、编程语言的 RTL（RunTime Library 运行支持库）等。Ada 语言实时绑定库属于编程语言的 RTL，是 Ada 语言的一些函数（包括 I/O 函数、内存管理函数、任务管理函数等）与低层实时操作系统 API 的接口函数。它实现了编程语言的函数与操作系统 API 的映射（如图1所示），应用程序通过 Ada 函数调用获得实时操作系统的支持。

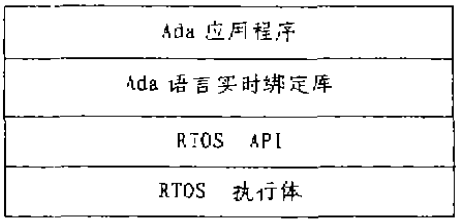

图1 Ada 语言实时绑定库与 RTOS、Ada 应用程序的关系

由于 Ada 语言实时绑定库是介于 Ada 应用程序和 RTOS API 的中间层，其设计和实现与 Ada 语言和 RTOS 紧密相关。因此，下面对 Ada 语言及 CRTOS 相关的特点做简单的描述。

Ada 95是在 Ada 83的基础上发展起来的最新标准，Ada 95的改进之处有[5]：

·接口：通过 pragma 等语句用一种更灵活的方式实现了与其它语言应用程序的接口。

·编程扩展：通过包（Package）和私有类型（Private Type）提供抽象的能力，提高了代码的可重用性。

·程序库：Ada 83程序库的问题是作用范围的限制，如两个库程序包不能共享一个私有类型，这妨碍了程序包间的共享，Ada 95采用层次库结构解决了这个问题。

·任务：一个任务单元由任务规范说明和任务体组成，在语言一级描述任务创建、任务删除、任务挂起、设置任务的优先级、任务的通讯和同步机制、保护对象、定时等操作，同时，采用任务会合点（Rendezvous）机制实现多个任务间的同步和通讯，这使得 Ada 语言非常适用于实时系统。

·可靠性：强类型等特点保证了 Ada 应用程序的可靠性，编译时尽可能检测到所有错误，其他未在编译时检测到的错误也将在运行约束下被发现，与 C 和 C++语言相比，大大提高了应用程序的可靠性。

·符合软件工程的要求：Ada 95是面向对象的语言，语法规范、严谨，符合软件工程的要求，适合开发大型、复杂的应用软件。

CRTOS 是一个强实时、可抢占、多任务的实时操作系统，其特点有基于优先级的抢占式调度、多任务能力、优先级继承、任务间通讯和同步机制、动态内存分配和灵活的用户配置能力等[1]。CRTOS 和大多数 RTOS 一样，其 API 用 ANSI C 语言实现，具有紧凑、高效的特点，但不符合 IEEE POSIX 标准1003.1b。因此，在 CRTOS 上设计和实现 Ada 语言绑定库，将参照 POSIX 标准《P1003.5：Ada binding to POSIX.1》的思想和结构，部分符合 P1003.5标准[4,5]。

2.2 特点

由于 Ada 语言实时绑定库是在大型、复杂、高可靠的嵌入式系统中使用的函数库，比一般的函数库更强调以下主要特性[6]：

·实时性和嵌入性：由于 Ada 语言实时绑定库是介于 Ada 应用程序和 RTOS API 的中间层，其性能好坏直接影响到整个系统的性能。因此，其代码实现应该紧凑、高效，满足实时性和嵌入性的要求。

·可靠性：Ada 语言多用于高可靠的嵌入式系统，Ada 语言实时绑定库的代码实现应该满足可靠性要求。

·可重入性：允许多个进程使用同一个地址空间的代码和数据，同时保证数据在进程调用期间的完整性。

因此，Ada 语言实时绑定库是一个用 Ada 语言实现的实时执行体静态函数库，作为 P1003.5标准的子集，以紧凑、高效的方式，用 Ada 语言实现对 RTOS API 的进一步封装和转换。同时，结构灵活，具有易于扩充、易于维护、易于移植的特点。

2.3 结构

Ada 语言实时绑定库分为 Ada 语言运行支持接口（Ada RunTime Interface）和低层接口（Low level Interface）两部分[5]（如图2所示）。低层接口主要实现 Ada 语言与 CRTOS API 的接口，将 CRTOS API 引入 Ada 语言中，作为 Ada 语言的外部函数使用。该接口由一系列的 CRTOS 程序包说明（CRTOS Package Specifications）和程序包函数体（Package Body）组成，包括 task、heap、partition、semaphore、mailbox、IO、signal、interrupt 等程序包及其子程序包（Subprogram Package Body），是一个层次化结构的函数库。

CRTOS 程序包说明定义了 CRTOS 运行环境的基本信息，包括任务操作、输入输出、内存管理、任务间的通信与同步等。CRTOS 程序包说明尽可能一一对应于 CRTOS 的 C 语言头文件，如 CRTOS 程序包说明文件 CRTOS-task.ads 对应于 C 语言的头文件 task.h，这样有助于具有 C 语言开发经验的程序员理解。其中 CRTOS 的对象、变量类型、函数、宏与 Ada 的对象、变量类型、子程序间也尽可能一对一地实现直接

维普资讯 http://www.cqvip.com

转换，但为了区别 Ada 变量类型与 CRTOS 变量类型，在 CRTOS 变量类型名前加上 CRTOS 前缀，如 CRTOS 变量类型 signed16转换为 Ada 变量类型 CRTOS. signed16；如果不能直接转换，则需要加入中间层，进行间接转换。

CRTOS 程序包函数体主要用 Ada 语言实现了对 CRTOS 提供的120多个系统调用的进一步封装，两者之间的转换大量使用了 pragma import 和 pragma convertion 等语句。这两条语句主要用于实现子程序和对象的转换，当然也支持其它实体的转换，pragma import 用于将其它语言定义的函数直接引入到 Ada 程序包中，使 Ada 语言能够调用其它语言的子程序或访问它的变量；而 pragma convertion 用于指定 Ada 实体和其他语言之间的转换，它主要是用于转换类型和回调子程序。对于子程序调用的格式，大多数实时运行库用 pragma import(C,…)或者 pragma import(Stdcall,…)调用转换，前者主要用于和 C 语言之间的接口转换，函数参数从右往左压入堆栈；Ada 语言实时绑定库主要使用后者，因为 Ada 语言实时绑定库使用静态方式调用 CRTOS API。

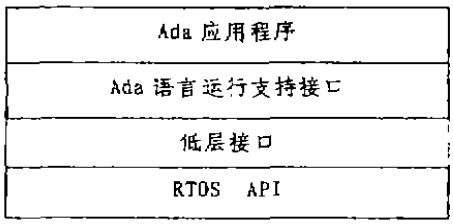

图2 Ada 语言实时绑定库的结构

Ada 语言运行支持接口主要实现用 Ada 语言封装的 CRTOS API 与 Ada 语言函数的接口，由 Ada、System、Interface 等程序包及其子包组成，也是一个层次化结构的函数库。该接口主要将 Ada 语言设施(Facility)的特性映射到 CRTOS 上，以实现正确的 Ada 语言语义(Semantics)、语法(Syntax)和语用(Pragmas)，如任务(Task)是 Ada 语言中最具特色的一个语言设施，作为 Ada 语言中的一个对象，通过任务规范说明(Task Specification)和任务体(Task Body)定义一个任务，任务具有属性(如 Task_ID、优先级等)和操作(如初始化、取消、激活等)。Ada 任务模型的多数属性和操作能够一一映射到 CRTOS 的任务模型上，但有些任务属性和操作(如 Defer_Abort)CRTOS 任务无法支持。对此，要做一些特殊的处理，对于无法支持的任务属性一般设置为某种安全的初始值，对于无法支持的任务操作一般仅有一个简单的返回值，没有任何处理动作，用户编程时应该注意到这些限制。

3 关键技术分析

GNAT(GNU New York Ada Translator)是由美国纽约大学开发的符合 Ada 95标准的 Ada 语言编译器，是一种遵守 GPL 协议的自由软件，已经被许多公司成功地用于商业开发。由于 GNAT 作为 GNU 软件，源码开放，一方面便于我们深入掌握有关 Ada 编译器的关键技术，提高 Ada 语言实时绑定库的安全性和可靠性，另一方面其长期发展也有着坚实保障。因此，我们采用 GNAT 的 Ada 语言函数库与 CRTOS API 实现 Ada 语言实时绑定库[6]。

3.1 Ada 语言与 C 语言的数据类型转换

CRTOS 是用 C 语言编程实现的，并以 C 语言函数库的形式为用户提供系统调用的应用编程接口(API)。C 语言也是一种结构化设计语言，并以其灵活、易用、高效而著称，与 Ada 语言相比，C 语言的严谨性明显不够，不具备强类型语言特征。

因此，为了用 Ada 语言实现 CRTOS 系统调用，必须解决 Ada 语言与 C 语言的数据访问接口、程序调用接口和数据类型转换等问题。由于 Ada 语言是强类型语言，必然要求接口函数类型和数据结构与 C 语言中的相应类型和结构一致，如图3所示[3]。

数据类型	Ada 95	ANSI C/C++
整型	Integer	Int
长整数	Integer	Long
短整数	Short_integer	Short
字符型	Character	Char
浮点型	Float	Float
双精度型	Long_float	Double
微整型	Ting_integer	(char)

图3 Ada 语言与 C 语言数据类型对照转换表

3.2 CRTOS API 转换示例：任务管理

在 CRTOS 的头文件 task.h 中定义了创建任务的函数 tcreate，函数原型为：

```
signed8 tcreate(signed16 tid,signed16 depth,signed16
    priority, unsigned32 stack_start, unsigned32 ad-
    dress);
```

数据类型 signed8、signed16、unsigned32 在 CRTOS 的头文件 typedef.h 中定义为：

```
typedef signed char   signed8;
typedef signed short  signed16;
typedef unsigned long  unsigned32;
```

将 tcreate 转换为 Ada 函数，其程序包说明和程序包函数体分别如下所示：

在文件 CRTOS-typedef.ads 中，程序包 CRTOS.typedef 的 signed8、signed16、unsigned32的说明信息

维普资讯 http://www.cqvip.com

为：

```
type signed8 is new Interfaces.C.signed char;
type signed16 is new Interfaces.C.signed short;
type unsigned32 is new Interfaces.C unsigned long;
subtype CRTOS.signed8 is signed8;
subtype CRTOS.signed16 is signed16;
subtype CRTOS.unsigned32 is unsigned32;
```

在文件 CRTOS-task.ads 中，程序包 CRTOS.task 的 tcreate 的说明信息为：

```
function tcreate (tid: CRTOS.signed16, depth: CRTOS signed16, priority: CRTOS.signed16, stack
    start: CRTOS.unsigned32, address: CRTOS unsigned32);
return CRTOS.signed8;
pragma Import(Stdcall, tcreate, "tcreate");
```

在文件 CRTOS-task.adb 中，程序包 CRTOS.task 的 tcreate 的函数体为：

```
function tcreate (tid: CRTOS.signed16, depth: CRTOS.signed16, priority: CRTOS.signed16, stack_
    start: CRTOS.unsigned32, address: CRTOS.unsigned32);
return CRTOS.signed8;
begin
  return tcreate (tid, depth, priority, stack start, address);
end tcreate;
```

3.3 Ada 任务的创建过程

在 Ada 应用程序中，使用 task type 和 task body 语句定义一个任务，如在程序包说明文件(*.ads)中声明任务：

```
task type<task_identifier>[is task_definition];
  entry<task_entrypoint>;
end[task_identifier];
```

在程序包函数体文件(*.adb)中描述任务执行的代码：

```
task body<task_identifier> is
    <declarative_part>;
begin
    <handled_sequence_of_statements>;
end[task_identifier];
```

当应用程序运行创建该任务时，首先调用 Initialize ATCB 函数，设置 Ada 任务控制块(Ada Task Control Block)等参数，接着调用 create_task 函数，将 Ada 任务的参数转换为 CRTOS 任务的相应参数，最后调用 CRTOS API tcreate 函数，创建一个 CRTOS 任务，由 CRTOS 内核对该任务进行调度管理。

结论 Ada 语言用于嵌入式系统的开发，需要 RTOS 提供实时运行支持。我们在 CRTOS 实时执行体上设计和实现的 Ada 语言绑定库，满足了整个系统的实时性和嵌入性等要求，其体系结构、设计特点和实现技术具有一定代表性和先进性，易于扩充，易于移植到其它 RTOS 上，对于提高国内军用及民用高端嵌入式应用的水平具有积极的推动作用。

参考文献

1 徐仑峰，熊光泽，刘锦德. 超微内核嵌入式实时操作系统的设计. 计算机科学，1998，25(3)
2 蔡建平，等. 实时嵌入式系统与 Ada 支持. 见：第八届抗恶劣环境计算机学术年会论文集
3 魏杰. Ada 语言在大型软件工程中的应用 见：第八届抗恶劣环境计算机学术年会论文集
4 IEEE Computer Society. IEEE 1003 5: Ada binding to POSIX. 1, July 1998
5 Baker T P, Giering E W III. The Gnu Ada Runtime Library (GNARL): Design and Implementation. Available at http://www.cs.fsu.edu
6 Giering E W, Mueller F, Baker T P. Features of the Gnu Ada Runtime Library. http://www.cs.fsu.edu
7 Teleogic Tau SCADE Available at http://www.csverilog com

(上接第122页)

应用程序的执行环境，比如需要考虑具体的对象概念如何映射为操作系统中的进程或线程等概念，对象间的并发机制在实际系统中的实际实现等。大体上说可按照以下步骤完成：

(1)使用自动代码产生工具来从 SDL 设计中产生代码；

(2)把产生的代码用于它的模拟执行环境，比如考虑外部的信号处理等；

(3)通过使用实时操作系统和交叉编译技术产生在相应硬件上可执行的代码；

(4)在目标环境里实现和执行测试用例(来自系统设计阶段的设计交互模型)。

我们将在前阶段产生的 SDL 模型上使用 SDL 支持工具，产生完整的目标代码，同时支持模拟执行、验证和测试，限于篇幅关系，目标代码和测试代码暂不给出。

总结 RCSM 方法主要面向实时通信软件系统的开发。目前 UML 模型和 SDL 模型之间的转换还需要人工的帮助，我们将逐步改进以支持从 UML 到 SDL 模型以及各个层次模型之间的智能转换工具。RCSM 的最终目标是以面向对象的方式完成实时系统开发中的从需求分析到实现和测试阶段所有模型的创建和转换工作。

参考文献

1 Rational Software Corporation. UMLV1.3 Specification, 1999
2 Lyons A UML for Real-Time Overview. Available at: http://www.Objec Time.com
3 Ellsberger I, Hogrefe D, Sarma A. Chapter 5. SDL formal object-ortented language for communicating systems. Prentice Hall, 1997
4 ITU Recommendation Z.105, SDL combined with ASN.1 (SDL/ASN.1), 1995
5 ITU Recommendation Z.120, Message Sequence Charts (MSC)