基于分区操作系统的目标机交互工具的研究

张钰尧[1] 张 前[1] 郭 锋[1]
ZHANG Yuyao ZHANG Qian GUO Feng

摘 要 现阶段综合化航空电子系统的应用领域和场景日益广泛，在地面状态部署或调试应用软件时，主机端与目标机端的交互需求也随之增加。综合化航空电子系统通常使用基于 ARINC653 标准的分区操作系统来完成软件时间和空间上的隔离，分区操作系统中采用系统态和用户态的权限划分，从而保证系统程序与应用软件的访问和执行特权。主机端与目标机端的交互使用目标机交互工具。在介绍系统态与用户态的基础上，阐述系统态下目标机交互工具的功能及基本设计原理，并提供一种在用户态下进行目标机交互的解决方案。基于多个项目对该方案进行了实验，验证了其可行性。

关键词 分区操作系统；目标机交互；用户态；系统调用；虚中断；嵌入式；ARINC653 标准

doi：10.3969/j.issn.1672-9528.2024.02.017

0 引言

在航空电子技术高速发展的今天，系统的设计和集成复杂度大幅增加，采用模块化、综合化的航空电子系统（integrated modular avionics，IMA[1]）已成为航空电子领域的主流趋势。航空电子系统中应用程序复杂度、交互性的提升，导致在开发、调试和集成应用程序时，主机端和目标机端的交互需求也随之增加。ARINC653 标准即是针对综合化航空电子系统的要求而提出的应用程序接口规范，其中也提出了分区的概念。航空电子领域中常用的嵌入式分区操作系统通常可划分为三个部分：核心操作系统、分区操作系统以及用户分区应用。一般采用系统态和用户态的权限划分，来保证系统程序与应用软件的访问和执行特权。主机端与目标机端的交互通常采用目标机交互工具。目标机交互工具一般具有通过串口的交互，查看目标机上任务运行状态，查看目标机内存空间、主动调用、执行函数等功能。

本文将基于嵌入式分区操作系统，研究系统态和用户态之间的通信访问方式，结合系统态下的目标机交互工具的设计实现与基本功能，提出一种在用户态下进行主机端和目标极端交互的解决方案，提升航空电子系统应用软件开发、调试的灵活性。

1 分区操作系统及两态

分区操作系统是满足 ARINC653 标准的嵌入式操作系统。ARINC653 标准作为航空电子应用软件接口的工业标准，不仅定义了应用软件与操作系统之间的接口规范，还对航空电子系统中操作系统的功能要求及安全要求进行了定义。ARINC653 标准中定义了系统中的任务之间，时间、空间相互隔离的概念，用于确保系统中关键任务的安全隔离与调度。

遵循 ARINC653 标准的嵌入式分区操作系统一般分为三个部分：核心操作系统、分区操作系统以及用户分区应用。其中核心操作系统运行在系统态[2]下，而分区操作系统及用户分区应用则运行在用户态下，如图 1 所示。系统态与用户态则被称为嵌入式分区操作系统下的两态。

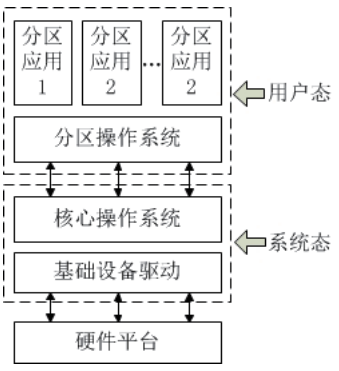

图 1 嵌入式分区操作系统结构

在系统态与用户态下，对于特权指令与地址空间的使用、访问和执行权限有所不同。特权指令是指一类在系统态下具有读、写或者执行权限，而在用户态下不具备这些权限的特殊指令。特权指令一般与硬件设计有较强相关性，不同的硬件体系架构中，有该体系架构特有的特权指令。在嵌入式分区操作系统中，如果用户态想要执行或访问一些依赖特权指

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710071

令实现的功能，就需要基于系统态与用户态之间的通信访问机制。

在嵌入式分区操作系统中，通过硬件的存储管理单元（memory management unit，MMU）来完成虚拟地址与实际物理地址的映射关系。操作系统启动，MMU 初始化构建对应的页表映射关系后，程序所运行的地址均为虚拟地址空间。依赖 MMU 建立核心操作系统、分区操作系统与各个分区应用所运行空间虚拟地址和物理地址的映射关系，可以将各个分区应用运行的相同虚拟地址，映射到不同物理地址空间上，保证各个分区应用间的空间隔离。同时对于用户态与系统态下的空间，依赖 MMU 对虚拟地址设置不同的访问权限，保证系统态与用户态下的空间隔离。

基于系统态与用户态之间，特权指令和地址空间使用、访问和执行权限不同的特点，在嵌入式分区操作系统下，完成系统态与用户态的交互，就需要特定的操作方案。通常情况下，在嵌入式分区操作系统中，通过系统调用机制来完成用户态访问系统态中的功能接口，通过虚中断机制来完成系统态通知用户态的事件触发，使用共享数据机制来完成系统态与用户态之间的直接数据交互。

1.1 系统调用

前文提到，由于用户态下不能使用特权指令，同时用户态与系统态下对于地址空间的读写权限不同，而分区操作系统及分区应用程序都运行在用户态下，那么分区应用程序想要操作硬件外设，或者访问核心操作系统提供的函数接口，就需要依赖系统调用 [3]。

用户态访问系统态的设备或功能接口时，通过执行一条处理器提供的指令，触发相应的处理器异常，处理器接收到该异常信号后，启动对应的异常处理程序，从用户态切换至系统态，完成用户态所请求的功能，而后会返回用户态继续执行，这一流程就是系统调用的通用流程 [4]，如图 2 所示。

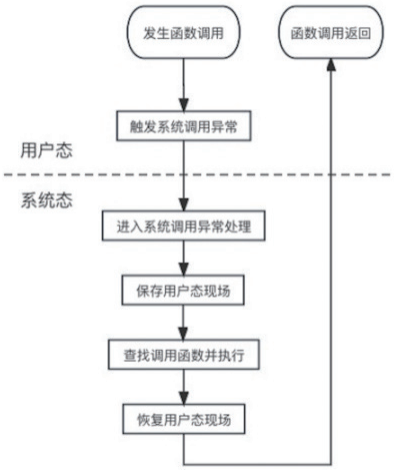

图 2 系统调用通用流程

在这个流程中，具备三个关键的实现点。首先，处理器需要具有一条能从用户态切换至系统态的指令，并且能向系统态发出处理该指令的信号。其次，不同的硬件平台被选用作为系统调用的指令并不相同，在 PowerPC 平台上，采用 SC 指令，在 ARM 32 位平台上，采用 SWI 指令，在 ARM 64 位平台上，采用 SVC 指令或 HVC 指令，在 MIPS 平台上，采用 SYSCALL 指令。然后，核心操作系统处理系统调用指令产生的信号，完成用户态程序提出的请求。最后，核心操作系统完成对应处理后，从系统态切换至用户态，继续执行用户态程序。

除了分区操作系统或分区应用中通过系统调用，使用核心操作系统提供的基础服务外，核心操作系统中可能存在着一些用户扩展的服务，需要被分区操作系统或分区应用使用。因此，核心操作系统需要提供系统调用扩展注册机制，将扩展的服务接口，注册至系统调用的列表，允许在用户态下调用系统态下扩展的服务程序 [5]。

1.2 虚中断

系统调用解决了用户态下的分区操作系统及分区应用访问系统态下的硬件设备接口或核心操作系统服务的权限问题，那么系统态下的程序需要通知用户态下的程序有事件发生，就需要依赖虚中断机制 [6-7]。虚中断属于一种软件信号，用于将系统态下的事件或通知，跨权限地投递至用户态下的分区操作系统或分区应用中，是一种从系统态到用户态的基本通信方式。

嵌入式分区应用软件中，使用虚中断机制的场景主要有三种，如图 3 所示。

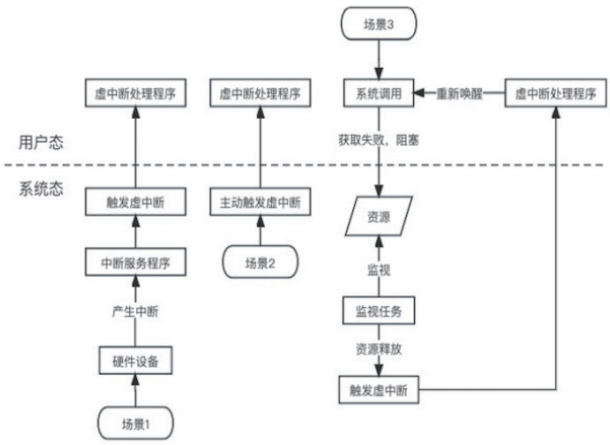

图 3 虚中断应用场景

第一种为处在系统态下的硬件中断无法直接投递至用户态下被相应的服务程序处理，但各类硬件中断又不得不通知用户态下的程序或用户态下的程序需要依赖该硬件中断实现相应的逻辑功能；第二种为系统态下不依赖外部硬件设备产

生的中断，而是主动向用户态下通知某些事件的发生，需要用户态程序执行对应的处理操作；第三种为由于系统态下的任务无法获得某种资源而发生阻塞性系统调用时，系统态下会启动特殊的任务等待资源的释放，当系统态下的特殊任务获得相应资源后，需要通知用户态下的程序继续执行后续的处理操作 [8]。

上述虚中断的应用场景中，第一种与第二种应用场景均为主动触发。例如当硬件设备中断产生时，系统态下的核心操作系统响应该硬件中断，进入对应的中断服务程序中，如果该硬件中断需要通知用户态下的分区操作系统或分区应用，则向用户态下投递异步事件。若不存在硬件设备中断或系统态事件主动向用户态投递，而是由分区操作系统或分区应用确定何时触发虚中断，那么则需要在系统态下注册用于发送虚中断的系统调用服务程序，并在用户态下注册对应的虚中断服务程序，同时在用户态合适的时机下使用系统调用，进入系统态下的系统调用服务程序中，触发虚中断的发送，使系统态被动地向用户态投递虚中断事件，此时位于用户态下注册的虚中断服务程序响应虚中断并进行处理。

2 系统态目标机交互工具

在嵌入式系统的应用软件的开发过程中，通常情况下，都是采用在开发主机端编写、编译开发代码，再将结果文件运行在嵌入式硬件设备上的开发部署模式，这里的嵌入式硬件设备即为目标机设备。在此开发部署模式下，想要监视目标机设备上程序的运行状态，就需要在开发主机上输入命令并发送给目标机，然后由目标机上的程序执行这些命令 [9]。但该模式的局限性在于，要想实现对目标机上程序运行状态的监视，就无法脱离主机端的开发环境。

为了解决这种开发部署模式的局限性，便于嵌入式操作系统及应用软件在运行过程中可以脱离主机端开发环境进行简单的监视与维护，例如查看当前运行任务的列表（包括任务 ID、任务入口函数、任务栈大小、任务栈使用率、任务优先级、任务当前状态、任务所需资源等），对指定任务进行简单控制（包括停止、挂起、解挂等），查看信号量列表，查看消息队列列表，查看、修改内存区域的值，调用扩展函数接口等，就需要在目标机上实现目标机交互的功能，通过目标机交互命令对目标机运行程序进行监视与维护 [10]。

目标机交互工具的主要功能在系统态下实现，依赖目标机设备上的系统态操作系统命令运行操作环境，支持在主机端仅有串口终端工具，无开发环境的条件下，对目标机运行程序的状态进行查询、介入和修改。目标机交互工具包括命令行编辑器、命令行词法语法解析执行器、命令异常处理以及目标机交互核心任务四个基本组成部分。目标机交互工具的基础功能有函数调用、任务信息的查看、内存查看、内存修改、创建任务、删除任务、挂起任务、解挂任务、周期性运行任务，同时提供对内存进行反汇编查看代码内容以及查看任务调用栈的两个基本调试命令。

目标机交互工具能够读取目标机的串口终端或将脚本文件作为输入。目标机交互工具能够对输入进行解析，得到需要执行的表达式，同时可以对不合法的词法、语法规则显示错误报告。目标机交互工具能够执行解析出的表达式，对于执行出现的异常，可以进行对应的异常处理。出于安全性考虑，目标机交互工具中实现了对目标机终端误触输入的处理分支，其处理流程如图 4 所示。

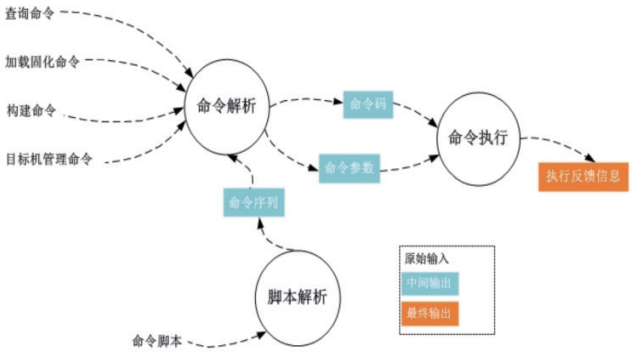

图 4 目标机交互工具功能基本流程

嵌入式分区操作系统系统态下的目标机交互工具常见命令如表 1 所示。

表 1 目标机交互工具常见命令

命令名	命令功能描述
help	输出目标机交互命令列表
printLogo	打印目标机交互工具 Logo
version	查看目标机交互工具版本号
i	显示当前任务信息（如任务名、任务 ID、优先级、栈大小、当前运行状态等）
sp	创建任务，缺省任务优先级 100
td	删除任务
ts	挂起任务
tr	恢复任务
d	显示内存数据
m	修改内存数据
period	创建周期任务

3 用户态目标机交互工具

由于目标机交互工具的核心功能实现均在系统态下完成，并且用户态下无法直接访问系统态下的资源，那么如何在用户态下的分区应用中使用目标机交互工具就是本文主要研究及解决的问题。

因为分区应用与目标机交互工具基础功能分别隶属于用户态与系统态，要实现跨态的通信交互，就需要依赖第 2 章中讲述的两种通信方式。以使用目标机工具中的函数调用功能，调用分区应用中的接口为例，结合虚中断的机制，实现一种用户态下目标机工具使用的解决方案，具体操作步骤如下。

首先，配置 SPARE 空间分区。为了保证在分区应用启动后，系统态下的目标机交互工具可以被执行，因此需要在将要使用目标机交互工具的分区应用所在的调度表上配置 SPARE 空闲分区，同时给 SPARE 空闲分区配置适当的运行时间。

然后，是系统态下的处理。在系统态下创建一个任务 usrShell 并挂接到目标机交互工具函数调用功能对应的函数接口表中。目标机交互工具的函数待用功能允许传递 22 个参数，因此该任务具有 22 个输入参数，用于向对应分区投递虚中断，并解析、构造分区应用中函数接口所需要的参数，包括最多 20 个参数。usrShell 任务的 22 个输入参数中，参数 1 用作传递将要进行虚中断投递的分区号，参数 2 用作分区应用函数接口编号，第 3 个至第 22 个参数可以传递分区应用函数接口的具体参数。由于虚中断投递仅能传递三个分区应用函数接口所使用的参数，当分区应用中函数接口参数超过三个时，就无法通过虚中断剩余的参数进行传递。因此设计用户态目标机交互的传参结构体 USRSHELL_ARG，解析 usrShell 任务的输入参数，构造该结构体，将该结构体地址作为虚中断投递的参数传递至用户态。传参结构体 USRSHELL_ARG 需要在系统态下构造，并传递至用户态，涉及系统态与用户态之间的数据传输，因此需使用到共享数据区机制，配置一段 shareIO 的空间，将该空间系统态与用户态均可读写访问，用作存放 USRSHELL_ARG 传参结构体数据。

最后，是用户态下的处理。创建一个用于分发虚中断事件的进程 usrShellSwitch，这里的虚中断事件，即为系统态下投递的目标机交互工具中的分区应用函数接口调用事件。创建一个用于激活分发进程的二值信号量 usrTshSem，创建一个用于传递虚中断上报数据的黑板 usrTshBlack，同时注册系统态下 usrShell 任务触发的虚中断的服务程序 usrShellHandler。这些资源需要在分区应用正式启动前准备完成。

完成上述设计实现后，依赖目标机交互工具进行用户态下函数接口调用的具体操作流程如下。通过串口输入，调用目标机交互工具函数调用接口表中注册的系统态下的 usrShell 任务，携带参数 1 为调用函数所在的分区号，参数 2 为分区应用中需要在目标机交互工具中调用的函数接口编号，其他参数为要被调用的分区应用接口所需的参数。该任务中触发虚中断，进入用户态下的虚中断处理程序 usrShellHandler。虚中断处理程序 usrShellHandler 将虚中断传递的用户态目标机交互的传参结构体 USRSHELL_ARG 写入黑板 usrTshBlack，并释放二值信号量 usrTshSem。usrShellSwitch 进程中等待二值信号量 usrTshSem 的释放，当获取到二值信号量 usrTshSem 时，即从黑板 usrTshBlack 中取得虚中断传递的传参结构体 USRSHELL_ARG，解析出分区应用函数接口编号以及相关参数，进行分发，实现用户态下目标机交互工具的使用。

4 总结

本文分析嵌入式分区操作系统的基本组成，以及核心操作系统与分区操作系统及分区应用在系统态与用户态之间的通信模式，基于系统态下的目标机交互工具模型，依赖通信模式中的虚中断机制和共享数据机制，研究设计用户态下的目标机交互工具，实现目标机端对分区应用函数接口的调用运行。本文描述的用户态目标机交互工具解决方案，扩展了嵌入式分区操作系统下目标机交互工具的能力，可以在脱离主机端开发环境的条件下，灵活、动态地控制目标机端用户态下分区应用函数的运行。

参考文献：

[1] 齐晓斌，田丹，麦先根，等．基于分区操作系统的实时监控工具的研究与实现 [J]. 航空计算技术，2014, 44(6):92-94+99.

[2] 仝敏，张东．嵌入式实时多分区操作系统两态访问的研究 [J]. 航空计算技术，2014, 44(6):88-91.

[3] 卫一芃，李运喜．嵌入式实时操作系统中系统调用方法的设计与实现 [J]. 电子设计工程，2011, 9(13):42-45.

[4] 陈云龙，曲波．小型微内核操作系统内核模型设计与实现 [J]. 赤峰学院学报：自然科学版，2011, 27(8):58-59.

[5] 张东，仝敏．嵌入式实时操作系统两态模式下的交互方法的研究 [J]. 信息通信，2018(1):57-59.

[6] 徐晓光，叶宏．分区间通信在航空电子系统中的设计与实现 [J]. 航空计算技术，2005(1):45-47.

[7] 方敏，计算机操作系统 [M]. 西安：西安电子科技大学出版社，2004.

[8] 周霆，李运喜．分区操作系统虚拟化中断处理方法研究 [J]. 科技风，2017(5):89-90.

[9] 魏国，张旻．一种嵌入式系统增强开发工具与目标机交互的方法 [J]. 信息通信，2019(6):96-97.

[10] 田丹，麦先根．嵌入式软件通信架构研究 [J]. 信息通信，2018(1):161-163.

【作者简介】

张钰尧（1992—），女，陕西咸阳人，硕士研究生，工程师，研究方向：嵌入式体系结构。

（收稿日期：2023-11-30）