

# 基于嵌入式实时操作系统的弹载计算机软件设计

卢 娥, 贾桂芬, 程越巍, 张志文, 张卫芳

(上海机电工程研究所, 上海 201109)

**摘要:** 嵌入式实时操作系统的引入改变了传统弹载计算机软件开发模式; 通过对我国自主研发的一款嵌入式实时操作系统进行研究, 分析当前战术导弹弹载计算机系统的特征, 对实时操作系统实施裁减定制, 形成了满足弹载软件系统的微内核系统; 围绕着基于该内核系统的应用, 某型弹载计算机软件的设计从系统需求分析、任务划分、任务实现以及最终系统运行结果四方面综合给出了设计方法, 试验证明, 使用操作系统的弹载软件功能、性能满足弹载计算机系统指标要求, 对提高软件的可重用性、可靠性等方面起到重要的作用。

**关键词:** 嵌入式实时操作系统; 微内核; 弹载计算机软件

## Design of Software in Missile-mounted Computer Based on Embedded Real-time Operating System

Lu E, Jia Guifen, Cheng Yuewei, Zhang Zhiwen, Zhang Weifang

(Shanghai Electro-Mechanical Engineering Institute, Shanghai 201109, China)

**Abstract:** Adopting real-time operation system in the missile-mounted computer software changes the traditional development modes. On the base of SZOS for missile system, designing a OS micro-kernel adapt to the practical missile-mounted computer. Subsequently giving the design of software based on the OS from system requirements analyse, task partition, task realization and measurement data of software running results. The system performance of the software based on embedded real-time operating system is well, it has the vital significance in improving the reusability and reliability.

**Keywords:** embedded real-time operating system; micro kernel; software in missile-mounted computer

### 0 引言

操作系统是控制和管理计算机硬件和软件资源, 合理地组织计算机工作流程以及方便用户的程序集合, 通过操作系统, 解决程序运行过程中对计算机系统中各类资源的争夺与共享, 使它们能够得到合理的分配、控制和使用, 以期提高整个系统的利用率<sup>[1]</sup>。

传统的弹载电子信息系统采用“控制流+中断”的嵌入式软件设计思路, 直接对设备物理层实施访问及判断, 协议层和应用任务层融合在一起, 此种设计方法在软件的可重用性、可靠性等方面有很大的弊端。随着弹上电子产品逐渐走向一体化, 弹载计算机所承担的任务更为丰富, 成为一个更为复杂的系统工程, 使得设计者不仅通晓全弹等系统知识, 而且还需具备丰富的计算机软硬件知识和自下而上的程序设计能力。这会大大的增加系统硬件设计和软件设计的周期和难度, 而产品质量的成熟度也难以提高和测试。通过实时操作系统的应用, 可以将软件承制方从复杂的硬件控制中解脱出来, 为软件承制方提供一个虚拟机, 从而更为方便和专注进行软件设计; 对于硬件承制方, 操作系统为计算机系统功能扩展提供了支撑平台, 使硬件系统与应用软件产生了相对独立性, 可以在一定范围内对原硬件模块进行升级或添加新硬件模块, 其结果可以不影响原先使用的应用软件。嵌入式操作系统应用的重要作用是: 屏蔽底层系统软硬件的复杂性、降低软件开发复杂度、实现应用软件

的可重用性、提升硬件资源的使用效率、提高软件的质量与可靠性方面<sup>[2-4]</sup>。

### 1 SZ 嵌入式实时操作系统

SZ 嵌入式实时操作系统 (后文简称 SZOS) 是一款针对星弹筒载航天武器型号的需求为牵引, 采用三层架构思想而实现的微内核操作系统<sup>[5]</sup>。

SZOS 系统代码精简, 使用二值信号量实现任务间的同步和互斥机制, 具有快速、确定的上下文切换及低延迟的中断响应, 能够保存及恢复浮点上下文, 并对窗口上下文溢出等异常及故障进行检测与处理, 实现了任务管理、二值信号量管理、消息队列管理、中断管理、时钟管理、I/O 管理等功能, 从而能满足弹载计算机的功能和性能的要求。SZOS 供弹载系统使用的软件总体结构详见图 1 所示。

针对不同的应用平台, 可以通过对 OS 内核进行适应性裁减和配置, 定制出适用的微内核系统。

### 2 弹载计算机平台 OS 微内核

从弹载计算机系统的应用中来看, 弹载 OS 的多任务并发程序在软件实现上具有以下几个要素: 程序在单处理器上运行; 一组固定数目的任务 (Task); 系统处理时间开销在几个 ms 以内; 每个任务只有单个启用事件 (Invocation event), 启用事件可以是定时的 (针对定时触发的任务), 或者是来自另 1 个任务或系统环境的信号; 任务间仅通过互斥访问的共享数据交互。

SZOS 标准版架构在战术型号中存在冗余的模块。因此, 针对这款标准版的软件架构作了相应裁减, 最终形成适用于弹载系统的多任务微内核的设计。

收稿日期: 2014-02-14; 修回日期: 2014-03-24。

作者简介: 卢 娥 (1977-), 女, 湖北荆州人, 高级工程师, 硕士研究生, 主要从事弹载信息系统和嵌入式软件方向的研究。

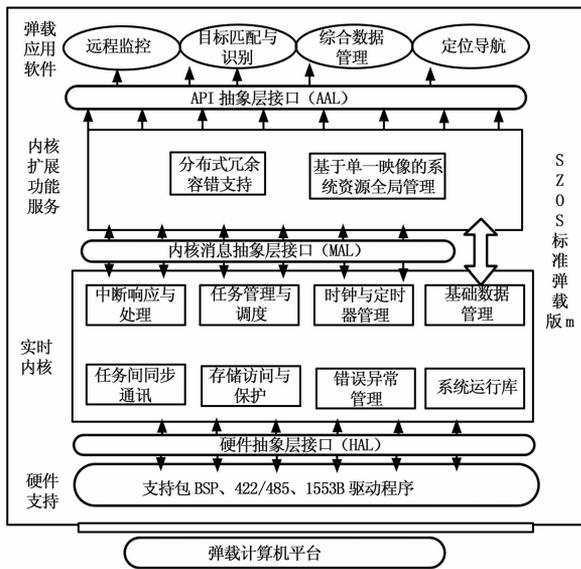


图 1 SZOS 弹载标准版软件架构

### 2.1 OS 微内核

基于弹载计算机系统的 OS 微内核主要包含了包括基础内核“系统初始化”、“任务管理与调度”、“中断管理”，可扩展内核“IO 管理功能”、“消息队列”和“二值信号量”共 6 个功能。

#### 2.1.1 系统初始化

完成从系统上电到系统正常运行需要完成的初始化工作。

#### 2.1.2 任务管理和调度

任务在操作系统中是通过任务控制块 (Task Control Block, TCB) 来描述的，每个任务都有唯一的任务控制块，是任务在系统中存在的唯一标志。内核根据 TCB 对任务进行控制和管理，任务的各种动态特征也通过 TCB 来表现。任务是 OS 运行和调度的最小单位，也是资源拥有者。任务管理负责对系统创建的所有任务进行管理，并调度选择优先级最高的就绪任务获得 CPU 开始执行，具有任务创建、任务挂起、任务唤醒和任务延时 4 个功能模块。

#### 2.1.3 中断管理

中断管理提供的功能包括中断注册、禁止中断和允许中断。应用程序可以通过系统中断管理模块注册自己的中断响应程序。当中断发生时，当前正在运行的任务被停止运行，操作系统的代码会首先接管中断，进行必要的现场保护工作，然后再转给应用程序的中断响应程序。中断响应程序不是任务，不会被切换到其他任务去。中断响应程序没有独立的堆栈，而是统一使用专用于中断响应程序的堆栈空间。如果 CPU 支持高优先级中断的中断嵌套，那么应用程序的中断响应程序运行过程中，可能由于发生了更高级别的中断而被打断，进入另一个中断响应程序。中断响应程序结束后，由操作系统负责恢复现场，回到被抢占的低优先级中断响应程序或被中断的任务继续运行。

#### 2.1.4 IO 管理功能

对硬件平台的外设 (GJB289A、RS-422、CAN、AD、DA、KI/O 等接口)，封装统一 I/O 管理接口，通过标准的 I/O 接口访问底层设备，屏蔽底层硬件细节。

#### 2.1.5 消息队列

消息队列用于任务间传递可变长的消息。

OS 根据不同类型的队列结构，分别抽象出队列头、队列数据节点、队列类型 3 种抽象数据结构。对象类型抽象数据结构是函数指针的集合，函数指针个数与抽象层统一接口数目一致。在系统初始化的时候，根据配置为所需要的队列类型创建数据结构的全局变量，并将该结构中的函数指针分别指向该类型队列操作函数。

#### 2.1.6 二值信号量

信号量有效：信号量状态为“满”。信号量有效表示任务对资源可用。

信号量无效：信号量状态为“空”。信号量无效表示任务对目前资源不可用，需要等待其他另一个任务（或者中断服务子程序）释放该信号量。

信号量是任务间互斥和同步的基础。如果创建一个初始状态为“满”的二值信号量，则这个信号量一般用于任务之间的同步；任务将自己阻塞，等待其他任务通知它数据可用。而初始状态为“空”的信号量，一般用于对共享数据的原子操作，即解决任务间互斥问题。

### 2.2 OS 微内核关键指标

#### 2.2.1 基本性能

(1) 中断响应时间：中断响应时间是指从中断发生起到执行用户中断处理程序的第一条指令所用的时间。它是衡量嵌入式实时操作系统实时性能的最主要、最具有代表性的性能指标。中断响应是系统在最坏情况下的响应中断时间。

弹载 OS 要求中断响应时间不大于 100 μs。

(2) 任务上下文切换时间：任务切换时间，也称上下文切换时间，指从当前运行任务的最后一条指令放弃 CPU 使用权，到下一个就绪任务的首条指令开始执行的时间。任务切换时间分为 3 个部分，即保存当前任务上下文时间、调度程序选中新任务的时间和恢复新任务上下文的时间。切换所需的时间主要取决于保存任务上下文所用的数据结构以及操作系统采用的调度算法的效率。上下文的保存对象为 CPU 寄存器的内容，切换时数据会保存在任务自己的堆栈内。上下文恢复的操作正好相反。保存的寄存器数量越多，任务切换的工作量越大。任务切换时间是系统实时性能的一个重要指标。

弹载 OS 要求中断响应时间不大于 200 μs。

#### 2.2.2 代码量需求

OS 的目标码不超过 30 kbyte。

### 3 弹载软件设计

#### 3.1 硬件配置

弹载计算机选用 TI 公司的 TMS320C6701 芯片作为处理器，工作主频 150 MHz；带有 128 k 双字数据存储器 (SRAM) 和 128 k 双字 Flash 存储器；弹载计算机通过串行异步全双工 RS-422 通讯接口与弹内其余设备进行信息交互。选用 BU61580 总线数据传输的接口芯片，实现与发控系统数据交互功能。

#### 3.2 任务需求

弹载计算机软件需要实现以下功能：按照一定的解算周期接收捷联惯性测量装置的测量信息，进行导航解算和姿态计算，根据选定的制导律实时形成过载指令，装订给导引头相应的数据，完成稳定控制回路算法，输出舵偏角给舵系统，同时还完成与弹上设备、发控设备、遥测设备或数据记录仪等外部

设备的信息交互, 负责弹上设备综合自检等功能。

### 3.3 任务划分

弹载计算机软件基本架构如图 2 所示。利用 OS 开发嵌入式多任务系统程序在系统需求分析明确系统功能之后, 主要任务是将系统功能合理分解, 构造多个不同的任务, 使每个任务负责完成应用要求的一部分功能。

#### 3.3.1 基本原则

任务的划分要遵循以下原则<sup>[6]</sup>。

(1) 功能内聚性: 对于功能联系比较紧密的各工作可以用一个任务来运行。如果用多个任务进行相互之间的消息通信, 将会影响系统效率。

(2) 时间紧迫性: 对于实时性要求比较高的任务, 以高优先级运行, 以保证时间的实时响应。

(3) 周期执行: 对于需要统一周期执行的工作, 应作为一个任务来运行。

#### 3.3.2 弹载软件任务划分

弹载计算机软件按照飞行时序整体逻辑可分为初始化、弹动前处理、弹动后处理 3 个部分。各个部分之间是串行执行关系, 初始化最先执行, 且仅执行一次。弹动前处理和弹动后处理有很多事件, 如接收 IMU 串口数据, 遥测输出等都是同一处理模式, 则可将两部分合并处理;

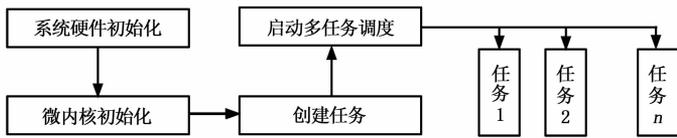


图 2 软件基本架构

纵观各处理事件, 可以分为两类: 周期性事件和随机事件。

周期性事件包括外设的数据采集与输出、数据指令解算形成等任务。结合系统启动主要是由 IMU 每  $T_1$  ms 发送一次数据触发应用程序开始处理, 所以建立最高优先级的 IMU 周期任务 (Task\_IMU), 在这一周期任务中要及时根据 IMU 的信息解算出一系列的控制指令和信息。其中指令解算形成功能需由发射命令触发, 发射前此项功能不参与计算, 主要有发控系统装订的信息进行控制, 发射后此项功能发挥作用。同时导引头通信串口中断事件  $T_2$  ms ( $T_2 > T_1$ ) 发送一阵数据, 由导引头通信串口和导引头事件处理任务共同完成, 则可以成立导引头事件处理任务 (Task\_COM), 设置其优先级为中级。

随机事件主要包括 1553B 命令处理: 接收 1553B 命令, 解析并处理。1553B 通信功能主要是在发射前实现与发射装置的信息交互, 而且其命令和数据更新的间隔  $\Delta T > (20 \times T_1)$  ms, 所以设置其为最低优先级。

具体确立的弹载软件任务划分和数据交互关系详见图 3。

### 3.4 任务实现

任务实现阶段主要是进行详细设计、编码实现和测试, 同时统计时间和空间等性能指标。

#### 3.4.1 系统初始化模块设计

系统初始化模块主要是从系统上电初始化到开始执行用户程序这一段时间, 系统需要完成的初始化工作, 主要包括 CPU 初始化、程序代码段加载、中断向量表初始化、微内核

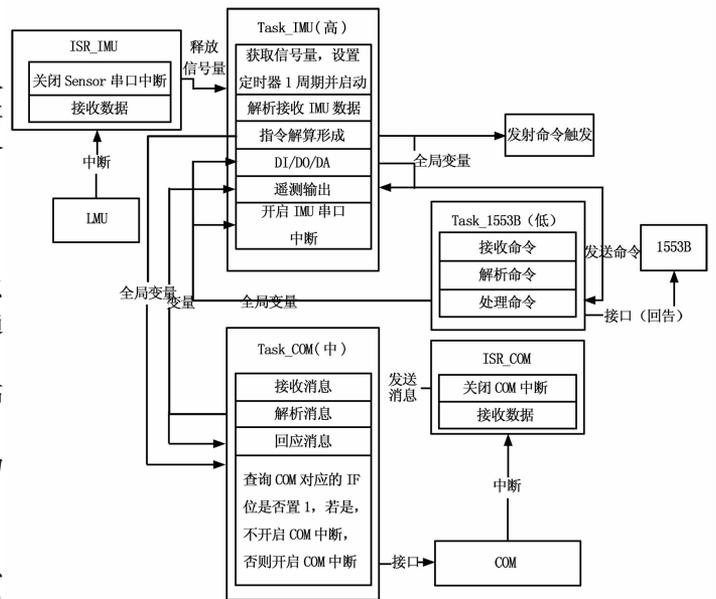


图 3 基于 OS 的某型弹载计算机软件任务划分

初始化、内存管理初始化、系统硬件初始化、系统时钟初始化、用户入口程序调用等过程。

#### 3.4.2 驱动程序设计

对于底层接口进行驱动程序设计, 操作系统中具有打开设备 (open), 关闭设备 (close), 读取设备 (read), 写入设备 (write), 控制设备 (ioctl), 安装设备驱动程序 (io\_DrvInstal), 添加设备 (io\_DevAdd) 等命令; 为提高 OS 在弹载计算机系统运行的效率, 对微内核采用一种相对简单的封装方式, 定义如下结构:

```
typedef struct
{
    FUNCPTR open; /* 函数指针 */
    FUNCPTR close;
    FUNCPTR read;
    FUNCPTR write;
    FUNCPTR ioctl;
    BOOL used;
} DRV_OP;
DRV_OP 1553b; /* 1553b */
...
(1553B.open)();
...
```

按照以上设计方法封装外设接口, 方法简单, 占用空间少, 用户系统开销也比较小。

#### 3.4.3 应用程序设计

##### 3.4.3.1 任务管理

任务管理操作系统主要提供了以下的命令函数: 任务创建 (tsk\_create), 挂起任务 (tsk\_suspend), 就绪任务 (tsk\_resume), 延迟任务 (tsk\_delay)。

以创建软件中 Task\_IMU 任务为例:

```
uint Task_IMU(void);
UINT TskID[5] = {0,0,0};
schar TskStack[9000] = {0};
```

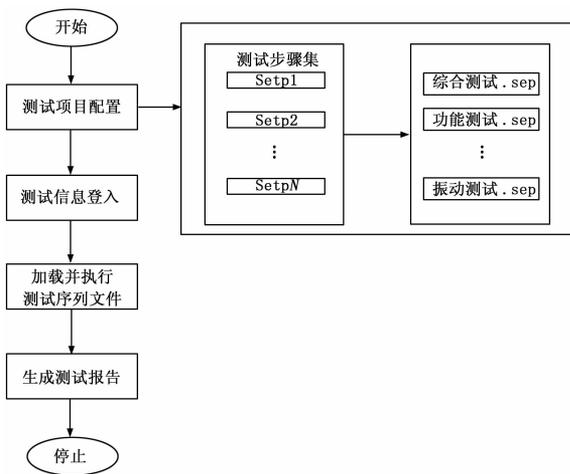


图 3 自动测试系统软件执行流程

#### 4 结论

基于 TestStand 的自动测试系统软件采用 TestStand 构建自动测试系统软件的框架, 通过序列配置定制自动测试流程, 使用 LabVIEW 进行测试代码模块的开发, 操作简单, 使用方便, 有效提高了自动测试系统的测试效率, 减少了软件维护成本, 同时, 该系统软件具有良好的兼容性和灵活性。

#### 参考文献:

[1] 赵 鑫, 等. 基于综合优先级的并行测试调度算法设计及实现 [J]. 计算机测量与控制, 2007, 15 (4): 423-425.  
 [2] 芦 俊, 等. 基于 TestStand 的 DDS 特性参数自动测试管理 [J]. 半导体技术, 2009, 34 (10): 957-959.  
 [3] 于建林, 等. 基于 TestStand 的测试系统及应用 [J]. 国外电子测量技术, 2004, 2: 6-9.  
 [4] 王伟浩, 等. 基于 TestStand 的计量软件介绍 [J]. 计算机技术及应用, 2009, 29 (6): 50-53.

(上接第 2311 页)

```
void Task_IMU(int arg1,int arg2, int arg3)
{
    //do something
}
...
if(ERROR == tsk_create(&.TskID[0], "T1", 5,0, TskStack, sizeof(TskStack), (TSKFUNCPTR)Task_IMU,0,0,0))
    Task_IMU();
```

#### 3.4.3.2 中断管理

在中断服务程序中先关闭该中断, 再在对应的任务中处理完中断事件后开启, 这样保证在一个处理周期中, 若有该中断发生, 不会响应, 避免频繁响应中断造成系统无法处理其他事件, 也无法发送遥测信息。

#### 3.5 试验结果

在某型弹载计算机平台上应用 OS 微内核设计的系统软件, 进行系统级半实物仿真, 弹载计算机、惯测、舵机等弹上系统闭环工作, 通过了半实物仿真验证试验, 具体的测试数据如表 1。

表 1 基于 OS 的弹载计算机软件的性能数据

序号	项目	实测值
1	中断响应时间 < 100 μs	79 μs
2	任务上下文切换时间 < 200 μs	114 μs
3	操作系统 bin 文件 < 30 kB	28 kB
4	整个系统软件 < 128 k 双字	50 k 双字
5	上电软件初始化到用户程序 < 10 ms	6.177 5 ms
6	制导控制指令曲线相似度达到 90%	一致

从表 1 的序号 1~5 的数据上可以看出, 按弹载计算机应

用条件定制的 OS 微内核性能指标满足设计需求, 最终形成的弹上计算机系统软件执行时间、占用空间均满足应用指标要求。从表 1 的序号 6 的数据表明, 弹载计算机系统能运行正常, 运行曲线与数值仿真一致, 证明基于 OS 微内核的弹载计算机软件满足设计要求。

#### 4 结论

通过对嵌入式操作系统和战术导弹弹载计算机系统的研究, 制定出符合弹载软件特性的 OS 微内核系统, 同时在此基础上设计了多任务架构的弹载计算机软件, 屏蔽底层驱动功能, 合理划分系统软件任务项, 充分验证基于嵌入式实时操作系统的战术导弹弹载计算机软件的可行性, 为后续导弹电子设备综合化的多任务软件系统设计奠定了技术基础。相比传统的弹载软件设计方法, 基于实时操作系统的软件设计对提高弹载计算机软件的重用性、降低软件开发的复杂性, 提升系统任务的处理能力具有重要的意义。

#### 参考文献:

[1] 毛伟云. 嵌入式实时操作系统 μC/OS-II 在弹载计算机上的应用 [J]. 火箭与制导学报, 2007, 27 (4): 349-356.  
 [2] 夏克寒, 许化龙. 弹载计算机中的实时操作系统研究 [J]. 微电子学与计算机, 2004, 21 (12): 97-100.  
 [3] 邓 昀. 微内核结构嵌入式实时操作系统的研究与设计 [J]. 微电子学与计算机, 2012, 29 (10): 133-139.  
 [4] 苏玉强, 刘 颖. 基于消息的嵌入式系统应用框架研究 [J]. 计算机测量与控制, 2011, 19 (5): 1142-1144.  
 [5] SZOS 软件用户手册 V3.0 [Z]. 北京: 北京神州航天软件技术有限公司, 2010.  
 [6] 解月江. 基于 SPARC 与 RTEMS 的飞行控制软件开发技术 [J]. 航天控制, 2009, 27 (4): 101-106.