

基于 Windows XPE 的水下航行器航行操纵系统 多任务软件设计

李洪普¹, 刘卫东^{1,2}, 徐娜¹

(1. 西北工业大学 航海学院, 西安 710072; 2. 水下信息与控制重点实验室, 西安 710072)

摘要: 航行操纵系统是水下航行器控制系统的重要组成部分; 为了保证航行操纵系统的实时性与可靠性, 鉴于其硬件板卡驱动支持 Windows 平台, 故软件采用了一种基于 Windows XPE 实时嵌入式系统的设计方法, 根据操纵要求对航行操纵系统实时多任务进行了划分, 诸如手柄信息处理任务、自动控制任务和网络通信任务等, 介绍了多任务执行实体—线程 (thread) 的优先级的设置和线程同步工具的选取, 并分析了 Windows 的多线程调度策略, 然后详细阐述了航行操纵系统各任务的执行流程以及实现方法; 最后, 对航行操纵系统进行了实验室仿真测试, 结果表明该系统实时性和控制精度达到了设计要求, 操纵系统性能良好。

关键词: 多任务软件设计; Windows XPE; 多线程; SDL; WSAEventSelect 模型

Multi-task Software Design of Windows XPE —based Navigation Steering System for Underwater Vehicle

Li Hongpu¹, Liu Weidong^{1,2}, Xu Na¹

(1. School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China;

2. Science and Technology on Underwater Information and Control Laboratory, Xi'an 710072, China)

Abstract: The navigation steering system is a vital component of control system for underwater vehicle. To ensure the instantaneity and reliability of navigation steering system for underwater vehicle, seeing that some hardware—driver of navigation steering system for underwater vehicle under Windows platform, a software design approach based on Windows XPE is adopted. a detailed classification of multi—tasking system is elaborated emphatically according to the control requirements, such as disposing joystick information task, automatic control task and network communication task etc., the setting of threads' priority and the choice of threads' synchronization tool are introduced, an analysis of Windows' schedule strategy has executed, then the exposition on programming flow and implement methods is carried out. Finally, the simulation—test results in Lab show that the instantaneity and control accuracy of this system can meet the design requirements, and control system performance is good.

Keywords: multi—task software design; Windows XPE; multi—threading; SDL; WSAEventSelect model

0 引言

航行操纵系统是水下航行器 (ROV、UUV 等) 控制系统的重要组成部分, 属于典型的多任务实时系统。其主要负责采集水下航行器状态参数, 控制水下航行器安全、高效航行。性能可靠的航行操纵系统对于水下航行器的整体性能至关重要。传统的水下航行器航行操纵系统一般采用 VxWorks 或 Linux 等嵌入式操作系统作为软件开发平台^[1-3]。

由于本文所涉及的水下航行器航行操纵系统的硬件驱动支持 Windows 平台, 结合对该系统实时性强、可靠性高、控制性能良好的要求, 设计了一种基于 Windows XP Embedded (简称 XPE) 嵌入式系统的航行操纵系统。Windows XPE 是 Windows XP 的组件版本, 系统内核小, 基于 Win32 编程, 系

统可移植性强, 可维护性高。基于该嵌入式系统, 本文对水下航行器的航行操纵系统多任务进行了详细的划分, 通过多线程技术^[4] (multi—threading) 并发式处理各任务, 对航行操纵系统的实时性与高效性进行了实验室仿真测试, 取得了较好的测试效果, 本文就系统结构及原理, 多任务的划分、调度, 设计与运行结果进行详细阐述。

1 系统结构及原理

根据任务功能要求, 航行操纵系统需要实时采集水下航行器的航行状态测量元件 (传感器) 数据, 并能够对水下航行器的执行机构 (艏舵、艉舵和方向舵) 进行安全、有效的控制, 同时还要与显示计算机、操纵手柄以及数字量输入输出模块等进行通信。综上, 绘出航行操纵系统的多任务结构如图 1 所示, 航行操纵系统需要并发处理上述多任务。

软件设计的基本单元是任务。设计原理如下, 将软件功能按要求分解为多个任务, 可以高效地利用系统资源, 加快单个任务的执行速度。但如果分解的任务过多, 会造成任务之间的频繁切换, 再者, 任务的执行实体是线程, 为了协调各线程, 任务之间还要进行同步和互斥机制控制, 系统开销会额外加大, 速度和实时性也会受到影响。因此, 划分任务时必须进行综合的权衡与考虑, 尽可能高效地利用系统资源。

收稿日期: 2015-01-09; 修回日期: 2015-03-13。

基金项目: 国家自然科学基金项目 (61473224); 船舶预研基金项目 (14JZ5.1.6)

作者简介: 李洪普 (1991—), 男, 河南洛阳人, 硕士研究生, 主要从事导航与控制方向的研究。

刘卫东 (1962—), 男, 陕西户县人, 教授, 博士生导师, 主要从事工业自动化, 计算机控制等方向的研究。

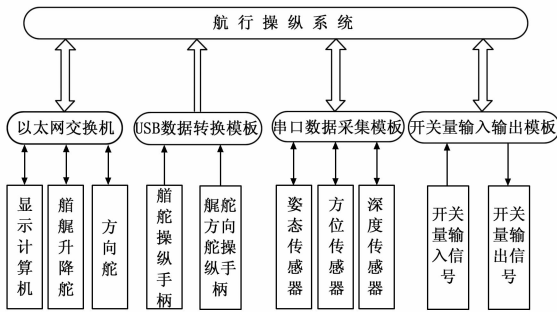


图 1 航行操纵系统多任务结构

根据水下航行器的航行操纵设计要求, 列出各任务的名称和功能如表 1 所示。

表 1 任务名称与功能表

| 任务名称 | 功能 |
|-----------|---|
| 初始化任务 | 初始化各变量, 开关量接口初始化设置, 串行设备接口初始化, 网络接口初始化, 操纵手柄初始化, 设备通信情况检查等。 |
| 串口数据接收任务 | 通过串口接收传感器上传的航行参数并提取出有效数据。 |
| 手柄信息处理任务 | 通过 USB 数据线接收操纵手柄信息, 用于手动操作模式时控制执行机构动作。 |
| 开关量输入输出任务 | 读取操控面板上按、旋钮状态并控制指示灯亮灭指示对应状态。 |
| 自动控制任务 | 在自动控制模式下, 由计算机运算出指令控制执行机构动作。 |
| 网络通信任务 | 利用 socket 套接字与显示计算机和执行机构进行 TCP/IP 通信。 |
| 状态监测任务 | 监测各任务状态, 当某任务出现异常时关闭该任务, 重新建立新任务, 维护系统稳定可靠运行。 |

2 任务调度

水下航行器航行操纵系统多任务软件设计的关键是软件实时性的保证, Windows 的任务调度^[5]是以线程为单位进行, 若要实现高效、合理地调度各任务, 重点是线程优先级的设置、线程间同步机制的选择、线程的调度策略等。下面根据以上 3 个方面进行阐述。

2.1 线程优先级的设置

在 Windows 系统中, 进程是线程的容器, 进程的优先级类默认为标准 (Normal)。线程是可调度的, 每个线程都有一个“优先级”, 范围为 0~31, 等级依次升高。当一个线程刚刚创建时, 它的线程相对优先级 (相对于所在进程的优先级类而言) 总是默认为标准 (Normal) 优先级。若要改变线程的相对优先级, 可以通过调用 SetThreadPriority () 函数来实现, 开发者不用具体设置一个线程的优先级, 操作系统负责将进程“优先级类”和“线程相对优先级”映射到一个具体的优先级上, 映射方式如图 2 所示。设置线程相对优先级应尽量避免使用关键时间 (Time-critical) 和空闲 (Idle) 优先级, 以免影响操作系统性能。

在航行操纵系统中, 经综合考虑, 设置读取串口线程的相

对优先级为高于正常 (Above normal), 向上位机上传传感器数据的线程相对优先级为最高 (Highest), 其它线程默认为标准 (Normal), 确保系统的实时性和稳定性。

表 2 Windows XPE 下的线程优先级映射方式

| 线程相对优先级 | 进程优先级类 | | | | | |
|---------------|--------|--------------|--------|--------------|------|-----------|
| | Idle | Below normal | Normal | Above normal | High | Real Time |
| Time-critical | 15 | 15 | 15 | 15 | 15 | 31 |
| Highest | 6 | 8 | 10 | 12 | 15 | 26 |
| Above normal | 5 | 7 | 9 | 11 | 14 | 25 |
| Normal | 4 | 6 | 8 | 10 | 13 | 24 |
| Below normal | 3 | 5 | 7 | 9 | 12 | 23 |
| Lowest | 2 | 4 | 6 | 8 | 11 | 22 |
| Idle | 1 | 1 | 1 | 1 | 1 | 16 |

2.2 线程间同步机制的选择

在水下航行器航行操纵系统中, 为了保护某些公共资源 (如公共缓冲区的传感器数据) 的完整性和安全性, 必须要求线程互斥访问该公共资源, 即实现线程同步。Windows 提供了多种专用于实现线程同步的内核对象, 如事件 (Event)、等待计数器、信号量 (Semaphore) 以及互斥量 (Mutex) 等, 在此采用信号量的方式实现线程同步, 确保以原子方式实现资源的快速访问。

2.3 线程的调度策略

Windows 线程调度总的算法是基于优先级可抢占算法。同一优先级下按照 FIFO (first in first service) 原则, 系统采用时间片轮询方式执行各线程, 当某个线程正在执行时, 优先级高的线程可抢占其时间片。但系统可以在基本优先级的基础上动态提高或降低线程的优先级来避免“饥饿”现象的出现。

3 任务设计

根据任务所执行的功能, 设计主要任务的程序流程如下。

3.1 初始化任务

初始化任务用于在程序刚启动时, 初始化程序中所有的全局变量, 信号量等, 初始化开关量通信板卡设置、网络通信套接字设置, 初始化操纵手柄等, 并对航行状态测量元件、执行机构、操纵手柄的通信情况进行检查, 然后向上位机上传各设备状态, 执行完成后, 此任务终止。由于此任务较为简单, 故省去程序流程图。

3.2 串口数据接收任务

串口数据接收^[6]任务是航行操纵系统软件设计的一个重点, 水下航行器安装了 7 个传感器, 可分为姿态传感器、方位传感器、深度传感器三大类。系统需要实时采集传感器上传的大量航行参数, 因此开辟 7 个线程分别负责 7 个传感器的数据接收, 为了避免线程阻塞, 确保接收数据的实时性, 串口接收数据采用异步 I/O 方式, 设置串口监听事件以及监听超时间, 在规定时间内收到数据后, 事件对象通知应用程序读取串口输入缓冲区并提取出有效数据, 若监听超时累计达到 3 次则进行异常情况处理, 提取数据后写入公共缓冲区, 另一线程定时读取缓冲区数据并上传。为确保公共缓冲区数据的完整性与安全性, 公共缓冲区采用信号量进行保护, 信号量初的最大资源数设置为 1, 确保每次仅有一个线程对公共缓冲区进行访问, 其程序流程图如图 2 所示。

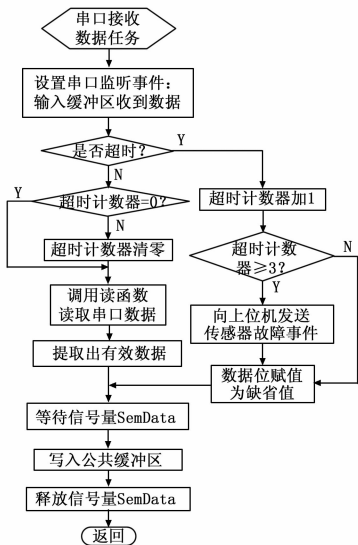


图 2 串口接收任务流程图

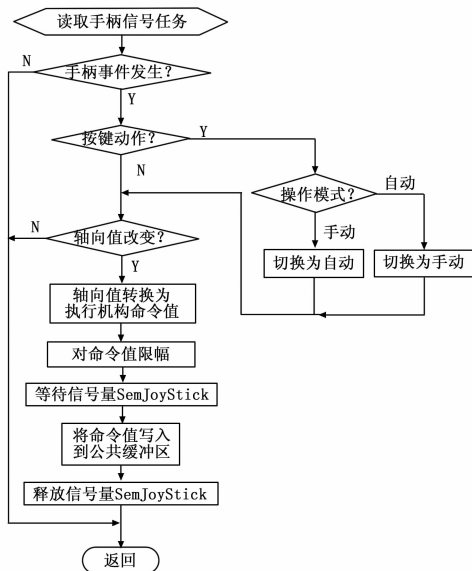


图 3 手柄信息处理流程图

3.3 手柄信息处理任务

在水下航行器航行过程中，手动操作时，需要通过手柄操纵执行机构，控制水下航行器的航向角和深度。读取手柄信息程序采用 SDL (simple directMedia layer) 多媒体开发库进行编写，SDL 是开源的跨平台多媒体开发库，可用于嵌入式系统中，与 DirectX (Direct eXtension, 简称 DX, 微软公司多媒体接口) 相比，具有结构简单，可移植性强的特点，能够满足系统功能要求。首先调用函数 SDL_WaitEvent 等待手柄事件（轴向值改变或者按键动作）发生，当事件发生后，根据事件类型执行相应命令，对手柄轴向值进行限幅是系统的一个考虑细节，在水下航行器航行过程中，为了防止操作人员误操作导致执行机构急剧动作引发危险事故，在发送手柄操纵指令到执行机构之前对指令值进行了一定程度的限幅，确保执行机构渐次达到手柄实际轴向值，保障水下航行器安全稳定运行。执行过程如图 3 所示。

3.4 自动控制任务

自动控制任务用于在水域条件合适的情况下，替代操纵人员操控执行机构，使水下航行器按照指定的航向和深度进行航行。当通过手柄操纵航行器达到理想航向或深度时，按下自动定向按钮或自动定深按钮（位于操纵手柄上），交由计算机根据事先设定的控制算法通过操纵执行机构控制航行器按指定深度和航向航行。水下航行器仿真模型^[7]参照潜艇的 6 自由度模型进行适当简化，其水平面操纵运动线性方程式为：

$$\begin{cases} (m - Y_{\dot{v}})\dot{v} - Y_v v + Y_r \dot{r} + (mV - Y_r)r = Y_{\delta} \delta \\ (I_z - N_r)\dot{r} - N_r r - N_{\dot{v}} \dot{v} - N_v v = N_{\delta} \delta \end{cases} \quad (1)$$

垂直面操纵运动线性方程为：

$$\begin{cases} (m - Z_{\dot{w}})\dot{w} - Z_w w - Z_q \dot{q} - (mV + Z_q)q = Z_0 + Z_{\delta_{\delta}} \delta_{\delta} + Z_{\delta_s} \delta_s + P \\ (I_y - M_{\dot{q}})\dot{q} - M_q q - M_{\dot{w}} \dot{w} - M_w w = M_0 + M_{\delta_b} \delta_b + M_{\delta_s} \delta_s + X_T z_T + M_b + M_{\theta} \theta \end{cases} \quad (2)$$

在式 (1)、(2) 中， m 为潜艇质量， u (纵向速度)、 v (横向速度)、 w (垂向速度) 为航速 V 在动坐标系 ($G - xyz$) 各轴上的分量， p, q, r 分别为横倾角速度、纵倾角速度和偏航角速

度， $\delta, \delta_b, \delta_s$ 分别为方向舵、船舵、艏舵舵角，又水平面运动中的 Y, N 的定义如下，

$$\begin{cases} Y = f_Y(\dot{u}, \dot{v}, \dot{r}, u, v, r, \delta) \\ N = f_N(\dot{u}, \dot{v}, \dot{r}, u, v, r, \delta) \end{cases} \quad (3)$$

垂直面运动中的 Z, M 的定义如下：

$$\begin{cases} Z = f_Z(\dot{u}, \dot{w}, \dot{q}, u, w, q, \delta_b, \delta_s) \\ M = f_M(\dot{u}, \dot{w}, \dot{q}, u, w, q, \delta_b, \delta_s) \end{cases} \quad (4)$$

$Y_{\dot{v}} = \frac{\partial Y}{\partial \dot{v}}, Y_v = \frac{\partial Y}{\partial v}$ ，其他依次类推， I_y, I_z 为潜艇对动

坐标系 ($G - xyz$) y 轴、 z 轴的转动惯量， $P = mg$ 代表潜艇的重量， $M_0 = M(u_0, 0, 0, 0, 0, 0, 0)$ ， X_T 为螺旋桨的推力， z_T 为桨轴线的垂向坐标， M_b 为静载力矩， θ 为纵倾角， $M_{\theta} = -mgh$ (h 为潜艇稳心高)。

控制方法起初阶段采用传统的 PID 方法，深度和航向分开控制，易于操作。

3.5 网络通信任务

网络通信任务也是软件设计的一个重点，负责航行操纵系统与上位机、执行机构之间信息的传送。网络通信具体采用 Windows Sockets 套接字^[8]实现，网络接收在实现过程中，为了确保实时接收上位机或舵机发送的消息，系统采用 Windows Sockets 提供的一种异步 I/O 模型—WSAEventSelect 模型，该模型为非阻塞的。

利用该模型，在接收数据前，创建事件对象，并为套接字注册相应的网络事件（欲接收可读的通知和欲接收套接字关闭的通知）。待事件发生后，检索事件类型，执行相应命令，具体的流程图如图 4 所示。

在运行过程中，考虑到套接字如果 2 个小时内无通信，则系统自动关闭该套接字并回收对应资源，并且交由系统进行保活 (Keepalive) 处理不能及时发现意外中断情况的发生，因此采用通信双方互发心跳包 (heartbeat) 的方式维持网络通信线路的畅通以及意外情况的及时发现。

3.6 状态监测任务

对各任务的执行状态进行实时监测并在执行失败时及时关

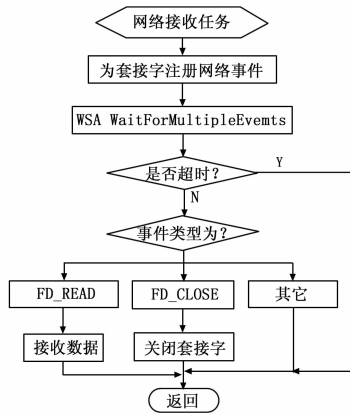


图 4 网络接收任务流程图

闭该任务并重新创建该任务是设计中的一个难点。本方案中, 为每项任务设置一个或多个状态标志位 (0/1: 正常/故障), 当线程运行时, 各任务运行过程中出现异常 (死锁等待或执行失败) 时, 及时将标志位置 1。同时设置单独线程循环扫描各个标志位的状态, 当某一任务状态位出现异常时, 状态监测任务立即终止该任务, 关闭相应句柄或者套接字等, 并重新创建该任务, 包括串口初始化以及网络套接字的重新建立等, 确保在软件可修复的前提下维护系统正常稳定运行。

4 仿真结果

针对本文的水下航行器航行操纵系统, 编写 C 语言代码, 在 VC++ 6.0 软件环境下编译生成可执行文件, 并在实验室搭建半实物仿真回路对航行操纵系统的性能进行多次测试, 记录并保存航行状态数据, 然后利用 MATLAB 软件绘制航行状态参数的变化曲线, 检验操纵效果。航速均设置为 8 Kn, 航向角随方向舵舵角的变化曲线如图 5 所示。

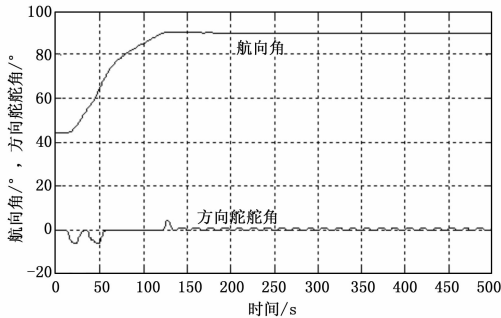


图 5 航向角随方向舵舵角变化的曲线

在图 5 中, 通过操纵手柄操纵方向舵使航向角从初始值 44°迅速达到指定值 90°, 在达到 90°瞬间, 按下手柄上的自动定向按钮, 启动自动控制模式, 由图 5 曲线可以看出, 方向舵迅速打反向舵角抑制航向角由于惯性继续增大, 短时间内即可使航向稳定, 稳定后, 方向舵隔段时间动作一次消除舵角机械死区引起的航向偏差, 维持水下航行器按照指定航向航行。

深度与俯仰角随艏舵、艉舵舵角的变化曲线如图 6 所示。在图 6 中, 操纵水下航行器达到 20 m 深度时, 按下操纵手柄上的自动定深按钮, 艏、艉舵立即联动打反向舵角抑制惯性引起的深度下降, 由图 6 知, 航行器迅速稳定在 20 m 深度航行。在大约 470 s 时, 取消自动定深, 使水下航行器自动下降到

30 m 时, 按下自动定深按钮, 由于之前以自由舵下降, 惯性作用较小, 自动定深后能很快稳定到指令值。在大约 1 000 s 时, 取消自动定深, 手柄操纵打大舵角使水下航行器迅速下潜, 在 37.2 m 深度时再次定深, 航行操纵系统立即打出大反向舵角抑制惯性作用, 使航行器尽可能快地上升, 以此检验误操作情况下航行操纵系统的操控能力。

仿真结果证明, 水下航行器能够及时响应航行操纵系统的指令, 切换到自动操纵模式下后, 能以较快的速度和较小的超调量稳定到指定航向或深度, 最终的稳态误差也稳定在设计要求范围内, 达到了预期效果。

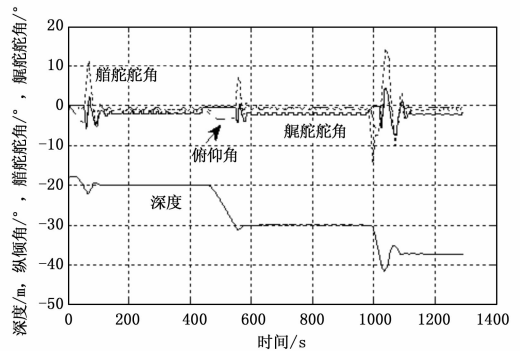


图 6 深度、俯仰角随艏舵、艉舵舵角变化的曲线

5 结论

本文主要研究了基于 Windows XPE 的水下航行器航行操纵系统的软件设计, 根据功能划分任务, 对任务优先级设置和任务间通信机制进行了分析。并着重探讨了多任务系统的任务设计。最后, 通过实验室多次测试证明, 该系统达到了功能要求, 具有良好的运行效果, 对于其他操纵系统的设计与实现有一定的参考价值。

参考文献:

- [1] 李翔宇, 刘卫东, 岳丽娜. PC104 与 VxWorks 在深海 ROV 控制系统的应用 [J]. 声学技术, 2013, 32 (4): 391-394.
- [2] 冯迎宾, 李智刚, 李瑞云. ARM 与 Linux 在深海 ROV 控制系统中的应用 [J]. 测控技术, 2012, 31 (9): 41-44.
- [3] 李洪亮, 侯朝楨, 周绍生. VxWorks 下实时多任务程序的实现 [J]. 微计算机信息, 2008, 24 (7): 90-91.
- [4] Yu C J, Peter P. Adaptive multi-threading for dynamic workloads in embedded multiprocessors [C]. 23rd Symposium on Integrated Circuits and Systems Design, SBCCI' 10. 2010: 67-72.
- [5] Richter J. Windows 核心编程 [M]. 李虎译. 北京: 机械工业出版社, 2008.
- [6] 崔红志, 索岩. 基于 API 的 Win32 串口通信编程技术 [J]. 科技信息, 2010 (7): 80-81.
- [7] 施生达, 潜艇操纵性 [M]. 北京: 国防工业出版社, 1995.
- [8] 孙海民. Windows Sockets 网络开发—基于 Visual C++ 实现 [M]. 北京: 人民邮电出版社, 2008.
- [9] 苏柏光, 王庆生. 基于 XPE 嵌入式煤炭销售票控终端机的研究及实现 [J]. 太原理工大学学报, 2010, 41 (4): 406-410.
- [10] 刘必广. 基于多任务的并行程序设计方法 [J]. 计算机与数字工程, 2010, 12 (38): 157-158.
- [11] 李晓晖, 严卫生, 范欣. 基于 RTX 的水下航行器实时仿真研究 [J]. 计算机测量与控制, 2010, 18 (4): 850-853.
- [12] 孙鑫. VC++ 深入详解 [M]. 北京: 电子工业出版社, 2006.