

基于龙芯宇航级芯片的 BSP 开发和移植^{*}

徐 帅^{1,2}, 林宝军^{2,3,4}, 刘迎春^{2,3}, 赵 帅³

(1. 中国科学院空天信息创新研究院, 北京 100094; 2. 中国科学院大学, 北京 100094;
3. 中国科学院微小卫星创新研究院, 上海 201203; 4. 上海科技大学信息科学与技术学院, 上海 201203)

摘 要:随着北斗三号导航卫星系统开始组网运行, 星上系统对星载计算机系统提出了更快的数据传输和运算速度需求。中国科学院研制的北斗卫星采用高运算性能和高数据传输性能的星上计算机系统, 其核心部件全部自主可控, 即以龙芯中科公司生产的龙芯 1E 高性能宇航级处理器芯片为主构建的硬件环境, 以实时操作系统 VxWorks 为软件环境。为适应龙芯中科 1E 系列的新型升级芯片, 本文通过开发 BSP 和串口驱动, 并在设备驱动管理上配置支持 VxBus 型驱动架构, 实现 VxWorks 在新型芯片上的移植和运行, 同时使驱动程序的可靠性、可移植性、独立性等性能得到有效提升。

关键词:BSP 移植; 龙芯; VxWorks; 串口驱动; VxBus

中图分类号:TP303

文献标志码:A

doi:10.3969/j.issn.1007-130X.2020.04.001

Development and transplantation of BSP based on Godson aerospace chip

XU Shuai^{1,2}, LIN Bao-jun^{2,3,4}, LIU Ying-chun^{2,3}, ZHAO Shuai³

(1. Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094;
2. University of Chinese Academy of Sciences, Beijing 100094;
3. Institute of Microsatellite Innovation, Chinese Academy of Sciences, Shanghai 201203;
4. School of Information Science and Technology, Shanghai University of Science and Technology, Shanghai 201203, China)

Abstract: With the Beidou-3 global navigation satellites system starting to operate in network, the on-board system requires faster speed of data transmission and data operation for the on-board computer systems. The core components of on-board computers for Beidou-3 satellites developed by CAS are all made or autonomously controlled by China. The on-board computers adopt the Godson 1E high-performance aerospace-class processing chip produced by Godson Corporation as the main hardware environment, and the real-time operating system VxWorks as the software environment, which constitute an on-board computer system with high performance of data computing and data transmission. In order to adapt to the new Godson 1E chips, this paper not only develop Board Support Package (BSP) and serial port drivers, but also configure the VxBus-type driver architecture on device driver management, in order to realize the porting and operation of VxWorks on the new chips. Meanwhile, the reliability, portability, and independence of the driver are effectively improved.

Key words: BSP transplantation; Godson; VxWorks; serial port driver; VxBus

^{*} 收稿日期:2019-09-26;修回日期:2019-11-26

通信作者:林宝军(linbaojun@aoe.ac.cn)

通信地址:201203 上海市浦东新区张江高科技园区海科路 99 号 4 号楼

Address: Building 4, 99 Haik Road, Zhangjiang Hi-Tech Park, Pudong New Area, Shanghai 201203, P. R. China

1 引言

2018年,北斗三号导航卫星系统完成19颗卫星发射、组网基本系统的建成,可以向全球提供服务^[1]。至2020年底,北斗导航系统将全面建成,30颗卫星在轨组网运行,向全球提供定时、测速、授时的导航功能。北斗三号卫星配备了Ka频段星间链路,1颗卫星可与多颗卫星建链^[2],未来可能还会搭载通信速率更快的激光星间链路,使得星间通信速率从几百Kb/s提升至1Gb/s。随着星间链路的广泛应用,大量的遥控、遥测、导航和测量等数据由星间链路传输至每一颗卫星的星载计算机内进行处理,卫星网络也需要传输和处理网络管理信息,这些都对星上计算机系统的性能提出了更高要求。文献[3]中提出了基于星间链路的集中式自主导航算法,配合星间链路的高速率数据传输,在同等周期下,星载计算机需要处理和存储更多的数据,配备更快的通信接口。这些都对现有CPU的频率、存储、数据接口提出了挑战,软件方面也需要设计复杂算法,搭建高速数据存储交互的基础环境。

2018年中国科协年会中杨长风总设计师提出了北斗导航系统未来瞄准PNT(Positioning Navigation Timing)体系^[4],实现更高精度的导航,信息更多元化,空天地一体化。可以预见,北斗卫星的星载计算机处理能力仍将不断提升。

国产星载计算机的发展比国外起步晚,最早多采用运行能力只有几MIPS的处理器,目前中国科学院研制的北斗卫星在轨星载计算机,采用一种基于核心部件全部自主可控的高运算性能和高数据传输性能的方案,即以龙芯中科公司生产的龙芯1E高性能宇航级处理器芯片为主构建硬件环境,采用32位龙芯2号处理器,2MB Flash和128MB SDRAM存储架构,以1553B总线为通信接口。

星载操作系统经过了由简单顺序执行的无系统模式到功能强大的实时操作系统VxWorks的发展过程,其优越的实时性和可裁剪内核等优势使其在星载操作系统中成为标杆。目前,星载操作系统多运行在较低频率和小存储空间的星载计算机中^[5],在设备驱动方面,外设驱动采用由应用程序直接调用的管理方式,需要设计更多面向高频高速接口的星载计算机操作系统管理方案。

为适应类似新需求,星载计算机将采用升级的新型龙芯1E高性能处理器芯片,通信峰值速率提

升100倍,处理峰值速度提升2倍^[6]。在软件方面,对星上软件和操作系统VxWorks 6.8进行移植,同时为配合卫星网络通信需求,并保证其可靠性、独立性,将外设驱动修改为由内核管理,为多线程多任务运行创建软件基础。

本文将对VxWorks 6.8进行板级支持包BSP(Board Support Package)开发^[7]、移植操作系统到新型1E芯片,在设备驱动开发中采用新型VxBus型架构^[8],并配置操作系统组件。

2 BSP移植背景和需求

新型龙芯1E芯片具有频率更高、处理速度更快等优点,处理器核采用64位双发射龙芯2号GS232,峰值频率由以前的100MHz提升到200MHz,定点峰值性能从200MIPS提升到400MIPS,较上一代芯片处理速度有成倍的提升;SDRAM空间提升至256MB;通信接口方面除传统的1553B外,还配备了SpaceWire高速通信接口,峰值速度从1Mb/s提升到200Mb/s。SpaceWire和DMA相结合,可实现远程访问内存的功能。

软件需采用一个较好的解决方案实现相应的移植开发:

(1) 软件开发环境、编译方法。

星上软件目前开发环境为代码阅读器+命令行编译。该方式有以下缺点:对复杂工程管理不足,对应用程序开发和调试不友好,对VxWorks内核和驱动等组件添加、删除不方便等。并且未来星载计算机发展到多核CPU后,该编译方式不能支持VxWorks对称多处理器SMP(Symmetrical Multi-Processing)编译。对于VxWorks 6.8的编译,风河公司推出WorkBench编译器作为开发工具,该编译器界面友好,方便对BSP^[8]、内核、驱动、应用程序实现相互独立开发,适合管理复杂的星载软件工程。在系统组件方面可以实现一键添加和删除,大大提升了开发效率。

(2) 驱动程序管理方式。

星载计算机驱动程序管理方式分为3类:应用程序直接管理、通过内核调用驱动、通过系统扩展模块调用驱动程序(例如网络)。文献[9]中指出由应用程序直接管理代码效率高,实时性较高,但面向多核多线程运算时这种驱动模式可能会造成缓存数据冲突,产生死锁等问题。本文采用风河公司推出的VxBus型驱动架构管理驱动程序,该架构

对设备管理结构更清晰,开发更简便,可在 WorkBench 编译器中配置,在层次上 VxBus 型驱动和内核、应用程序独立,为其他层次提供统一接口,具有更好的可靠性、可移植性等优点,为应用层开发提供便利。

(3)对新型 1E 芯片硬件环境适配。

本文硬件平台选用新型芯片,在操作系统移植过程中,对硬件环境做相应的适配:配合更大数据存储和计算需求,内存管理增加 VxWorks 任务栈、中断栈空间等 VxWorks 系统管理空间,并采用 VxBus 型驱动管理中断驱动,在系统启动时对相应组件和中断向量表进行配置。为支持自主导航算法中大量浮点等复杂计算和 C++ 编译环境,在 VxWorks 启动阶段配置协处理器开启浮点计算,在操作系统初始化阶段添加 C++ 组件。

3 BSP 介绍及移植思路

3.1 BSP 介绍

板级支持包 BSP,顾名思义,它是应用于操作系统和硬件资源之间的相关软件集合。其主要功能是配置硬件,为操作系统屏蔽硬件,使得上层软件开发不需配置底层,并且在此基础上,BSP 要完成操作系统的引导和启动,外设驱动和内核初始化等任务,使得操作系统可以运行在不同的硬件环境中。从层次上分,BSP 也属于操作系统的一部分,如图 1 所示。

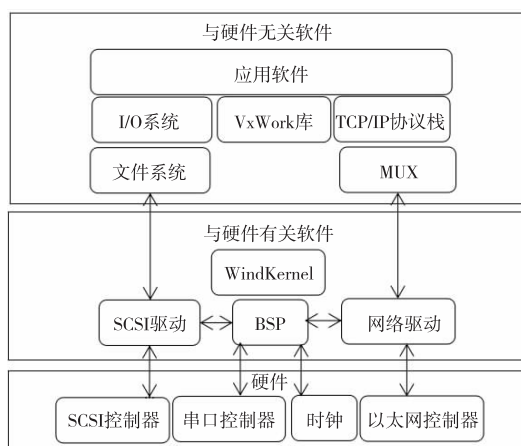


Figure 1 VxWorks operating system structure

图 1 VxWorks 操作系统结构图

3.2 VxWorks 启动流程

考虑到本文开发的 BSP 应用在星载系统中,在保证安全有效存储操作系统镜像前提下,对开发做出以下前提声明:

如图 2 所示,BootLoad 存放在 NandFlash 中,通过 BootLoad 下载并引导操作系统,引导过程中,将操作系统拷贝到 RAM_LOW_ADRS 的位置运行。引导结束后,将执行权交给 VxWorks 操作系统,开始进行系统初始化^[10]。

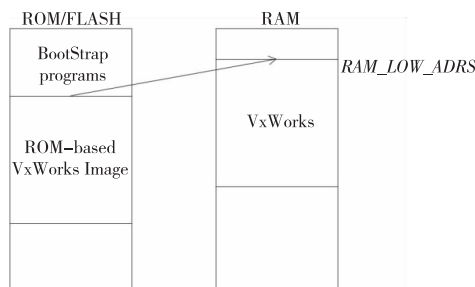


Figure 2 Storage structure of VxWorks

图 2 VxWorks 存储结构

图 3 所示为从函数级描述操作系统启动过程,BootLoad 将程序指针跳转到 RAM_LOW_ADRS 后执行 sysInit(),对 CPU 进行基础初始化,进而在 usrInit()中通过接口 sysHwInit()和 KernelInit()对硬件设备和操作系统内核进行初始化,之后运行 usrRoot()完成内核组件创建并跳到应用层,执行应用程序^[11]。

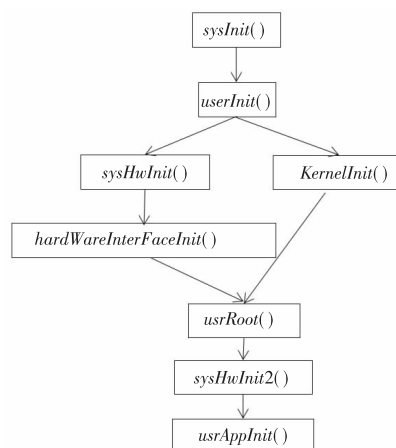


Figure 3 VxWorks startup process

图 3 VxWorks 启动流程

3.3 移植思路

本文采用风河公司推出的 VxWorks 6.8 操作系统,编译环境采用适用于复杂项目管理、具有良好调试功能的 WorkBench 3.2,满足星载系统软件开发需求。在系统启动阶段,启动新型龙芯 1E 芯片的协处理器 1,以增强浮点运算能力,并添加 C++、VxBus 等系统组件,为复杂算法搭建软件基础。采用 VxBus 型驱动架构管理串口等外部设备。在中断管理方面,利用 VxBus 型架构分层管理中中断驱动。将内存分配为 VxWorks 管理和静

态管理 2 部分,并增加 VxWorks 管理区域,充分利用新型 1E 龙芯芯片的存储空间。

4 BSP 移植实践及关键技术

4.1 VxWorks BSP 相关组件编写方案

在 VxWorks BSP 开发中,从文件角度需要编写以下组件:

(1)Config. h。

Config. h 是缺省配置文件,在操作系统启动和初始化过程中,将会根据 Config. h 的参数来对操作系统进行配置。开发人员在对该文件中对 BSP 基本配置选项进行设置,例如系统时钟频率、内存起始地址、Cache 模式等,通过 #define、#undef 的宏命令来对 BSP 的组件进行添加和裁剪。必须的组件配置需要写入该文件中,其他的可选组件可以在 WorkBench 开发中进行添加。

在 Config. h 中更改 RAM_LOW_ADRS 镜像起始地址,通过设置 LOCAL_MEM_SIZE 规定 VxWorks 操作系统管理内存区域大小,并添加对 VxBus 型驱动架构的支持,添加编写的驱动模块,例如串口、中断、时钟。配置操作系统对浮点操作和 C、C++ 的支持。

(2)Makefile。

Makefile 是 BSP 的编译文件,在该文件中指定编译 VxWorks 镜像时所要使用的编译器和函数库。另外需要指定根启动设备 ROM 大小和地址、增加的目标模块、编译应用层软件所需要的源码文件。

(3)sysAlib. s。

在 sysAlib. s 中改写 sysInit(),该程序为 VxWorks 启动的入口程序,其主要功能是初始化协处理器,建立环境启动 usrInit()。下面以伪代码形式介绍本文方案:

```
sysInit(){
    使能协处理器 0,1;
    清空中断;
    清空协处理器 0 的 CAUSE、WIRED 寄存器;
    启动向量入口地址控制,置协处理器 0 ERL 为 1;
    清空 TLB,建立堆栈;
    建立全局偏移量表指针;
    初始化处理器时钟计数器、计时器中断控制器;
    将 usrInit()入栈并执行;
}
```

(4)sysLib. c。

该文件主要提供目标板硬件资源的初始化接口。通过编写 sysHwInit()、sysHwInit2()函数,实现对目标板的硬件初始化。操作系统对 VxBus 驱动的支持也是在这 2 个函数中完成。在 sysHwInit()对 VxBus 架构进行第 1 次初始化,调用 hardWareInterFaceInit()接口,初始化硬件设备表的数据结构,对 VxBus 型设备挂接的虚拟总线 PLB 进行初始化,注册 VxBus 型设备。在 sysHwInit2()中,调用 vxbDevInit()进行设备第 2 次初始化,包括设置设备的寄存器和内存缓存区等,并向应用层提供接口。最后在 vxbDevConnect()中完成中断挂接,至此硬件设备驱动初始化完成。

(5)hwconf. c。

VxWorks 在该文件中定义 hcfDeviceList[]列表来描述 VxBus 驱动架构的所有硬件设备。hcfDeviceList[]列表是由一个个设备结构体构成的,所有与硬件相关的设置都在设备结构体中完成,进而在注册设备驱动时被写入驱动程序中。在设备结构体中需要定义设备寄存器基地址、中断号、设备引脚号等。本文中 ns16550 串口、MIPS 中断、R4K 时钟在 hwconf. c 中注册。

4.2 中断管理方案

本文中硬件环境是 MIPS 架构的龙芯 1E 芯片,对于中断采用 VxBus 驱动的方式管理。在硬件结构上,MIPS 的芯片采取分层次管理。第 1 层为 MIPS 中断控制,共有 8 个类型,第 2 层为龙芯 1E 中断控制,共有 32 个中断^[12],本文中涉及时钟中断和串口中断。在编写软件时,本文采用分层次的方式管理中断。

第 1 层 MIPS 中断控制,可利用自带的 VxBus 型中断控制驱动。在 hwconf. c 中添加所需要控制的中断设备引脚说明,将引脚 2 设置为龙芯 1E 中断,将引脚 7 设置为时钟中断。第 2 层的中断为板级中断,将引脚 11、12 设置为串口 0、1 中断并更新向量表。然后在 hwconf. c 中添加龙芯 1E 的中断:

```
Const Struct intrCtrlInputs mipsLSIntCtrlInputs[]={
    {11,"ns16550",0,0},
    {12,"ns16550",1,0}
}
const struct hcfResource mipsLSResources[]={
    {"regBase",HCF_RES_INT,(void *)TRUE},
    {"input",HCF_RES_ADDR,((void *)mipsLSIntCtrlInputs[0])},
    {"inputTableSize",HCF_RES_INT,((void *)
```

```
NELEMENTS(mipsLSIntCtrlInputs)}}
}
#define cpuNum NELEMENTS(mipsLSResources)
在设备列表中添加中断设备:
{"mipsLSIntCtrl",0,VXB_BUSID_PLB,0,cpuNum,
mipsLSResources}
```

4.3 内存管理方案

从硬件角度看,在 32 位 MIPS 存储器架构中,对程序空间地址映射分为 4 大区域:kuseg, kseg0~kseg2。本文将地址空间的 kseg0 和 kseg1 直接映射到物理地址空间的最低 0.5 GB。考虑到星载软件对实时性和安全性的要求,将内存分为静态管理和动态管理 2 部分。本文将操作系统映射到 kseg0 区域,将 200 MB 分配给操作系统作为动态内存使用,即:0x80000000~0x8C800000 用于系统运行、局部变量存储等。剩下的地址空间作为静态内存使用,重启不刷新,用于存储遥测数据等。

图 4 为用于动态存储的内存空间分配图。在本文中,低端内存区域存放中断向量表、boot-line、exception message 等信息。RAM_LOW_ADRS=0x801000000, VxWorks 区域存放操作系统映像的代码段、数据段和 BSS(Block Started Symbol)段。WDB(Wind deBug)内存池在前期开发时使用,主要用于动态下载目标模块、传递参数等。VxWorks 可用内存区域主要用于动态内存分配(malloc 等)、任务和中断堆栈等。

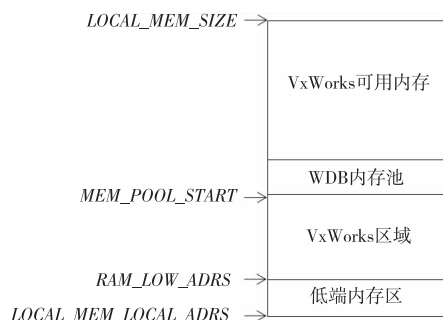


Figure 4 Dynamic memory allocation

图 4 动态内存分配图

静态存储起始地址为 0x8C800000, 至 SDRAM 结束。在操作系统运行时,可由应用程序直接向这段区域写入数据,省去了系统调度等过程,提供了很好的实时性和可靠性。具体的分配策略可根据应用层程序的需要进行分配,以达到最好的内存利用状态。

4.4 时钟管理方案

时钟管理关系着操作系统任务调度、超时处理、时钟计时、时钟中断等。本文中新型龙芯芯片

通过配置 PLL 分倍频参数将系统时钟设置为 180 MHz,其高频 PLL 结构图如图 5 所示,输入的时钟 Fin 经过输入分频后得到 $Fref$ 时钟,该时钟送到倍频器,得到 $Fvco$,然后在输出前除以 1 个分频系数,得到 PLL 输出时钟 $Fout$ 。转换方式如式(1)和式(5)所示:

$$Fout = Fin / ref * ldf / odf \quad (1)$$

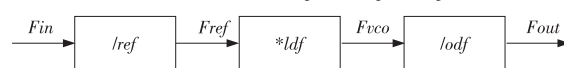


Figure 5 Clock PLL structure

图 5 高频 PLL 结构图

在操作系统中采用 VxWorks 所提供的 MIPS R4K 的 VxBus 型驱动管理时钟。在 Config.h 中包含 #define DRV_TIMER_MISR4K 驱动组件并将系统时钟设置为 180 MHz。在 hwconf.h 中设置时钟参数:最小时钟、最大时钟、CPU 频率等。关键代码如下所示:

```
Const Struct hcfResource r4KTimerDevResources[] {
    {"regBase", HCF_RES_INT, (void *)0},
    {"minClkRate", HCF_RES_INT, (void *)SYS_CLK_RATE_MIN},
    {"maxClkRate", HCF_RES_INT, (void *)SYS_CLK_RATE_MIN},
    {"cpuClkRate", HCF_RES_INT, (void *)180000000},
};
```

```
# define r4TimerDevNum NELEMENTS(r4KTimerDevResources)
```

在设备列表中添加时钟设备:

```
{"r4KTimerDev", 0, VXB_BUSID_PLB, 0,
r4TimerDevNum, r4KTimerDevResources}
```

5 VxBus 型串口驱动编写

VxWorks 系统提供了 VxBus 型驱动来管理 I/O 设备,本文以常用的 ns16550 串口为基础,编写驱动软件。

5.1 串口数据结构定义

编写 VXBNS16550.h 文件,定义串口寄存器地址、中断类型等变量并构建结构体 NS16550VXB_CHAN,把串口抽象成 1 个组件,以方便对设备的操作设置和管理。

在 hwconf.c 中,对串口设备进行基础的初始化定义,并把串口设备添加到 VxBus 设备表中,定义串口寄存器基地址、波特率、时钟频率、寄存器间隔空间,关键代码如下所示:

```

Struct hcfResource NS16550Dev0Resources[] = {
    {"regBase", HCF_RES_INT, 0xb004080},
    {"baudRate", HCF_RES_INT, 152100},
    {"clkFreq", HCF_RES_INT, (void *) BAUD_CLK_FREQ},
    {"regInterval", HCF_RES_INT, (void *) UART_DELTA},
};
# define NS16550Dev0Num NELEMENTS(NS16550Dev0Resources)
Const Struct hcfDevice_hcfDeviceList[] = {
    {"NS16550", 0, VXB_BUSID_PLB, 0, NS16550Dev0Num, NS16550Dev0Resources}
}

```

5.2 串口驱动程序实现

创建 VXBNS16550.c 驱动源码文件, 实现串口设备的初始化、中断以及各种操作, 如图 6 所示。

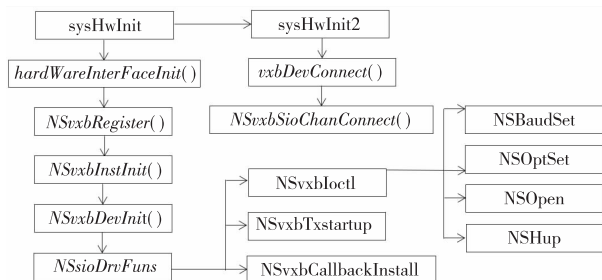


Figure 6 Method of serial port driver

图 6 串口驱动方法图

串口 *NSvxbRegister()*、*NSvxbInstInit()* 2 个函数在内核初始化过程中完成。*NSvxbRegister()* 创建串口设备, 并将串口设备挂接到虚拟总线 PLB。

NSvxbInstInit() 初始化 NS16550VXB_CHAN 数据结构, 将串口的波特率、通道号、时钟频率、串口基地址、寄存器间隔空间参数从 hw-conf.h 中加载到驱动中, 并声明以下 2 个方法:

```

LOCAL device_method_t NS16550VXB_methods[] =
{
    DEVMETHOD(sioChanGet, NS16550SioChanGet);
    DEVMETHOD(sioChanConnect, NS16550SioChanConnect);
};

```

NS16550SioChanGet() 方法为上层提供通过通道号找到串口设备的接口, *NS16550SioChanConnect()* 用于串口中断的挂接和使能。

NSvxbDevInit() 实现 2 部分初始化, 第 1 部分配置 UART 通道, 包括 LCR 线路控制寄存器, 配置端口 8 字节, 1 个停止位无校验, 波特率 115 200, 设置 Modem 控制寄存器为回环模式中连

到 RI 输入。第 2 部分声明串口操作方法, 包括 *NSvxbIoctl()*、*NSvxbTxstartup()*、*NSvxbCallbackInstall()*。

NSvxbIoctl() 为串口控制函数, 包括设定串口波特率、停止位奇偶校验、打开串口、Modem 控制寄存器。*NSvxbTxstartup()* 使能发送的中断。*NSvxbCallbackInstall()* 完成中断后接受和发送的函数连接安装。

在 *sysHwInit2* 中调用 *vxbDevConnect()* 连接和使能串口中断, 将串口中断处理程序 *NSvxbInt()* 挂接到中断向量中。在本文中串口工作模式为中断触发, 在中断处理程序中使用 *NSIntWr()* 和 *NSIntRd()* 对串口循环缓存区进行读写操作。至此, 串口初始化完成, 设备以相应的结构体挂接到 VxBus 设备树中, 并开启中断。

5.3 挂接中间层 TTY

如图 7 所示, 操作系统通过中间层 TTY 管理串口设备: 虚拟设备 TTY 作为 I/O 设备向 I/O 系统注册。在 *usrRoot()* 中将 TTY 设备加入 IO 系统:

```

ttyDevCreate(tyName, sysSerialChanGet(channel-
No), 512, 512);
consoleFd = open(tyName, O_RDWR, 0);

```

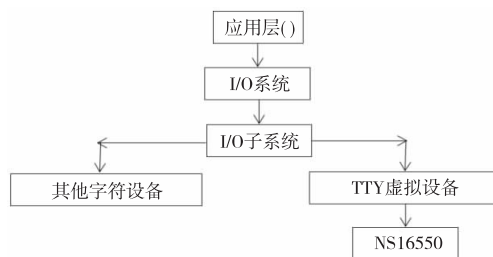


Figure 7 I/O device management

图 7 I/O 设备管理

在 *ttyDevCreate* 函数中, *iosDrvinstall()* 和 *iosDevAdd()* 将串口以 TTY 设备添加进系统驱动表和系统设备表中。用 *open()* 申请 *consoleFd* 为设备文件描述符。至此, 串口设备被挂接到 I/O 系统中, 驱动开发完成。

6 测试方法与结果

6.1 测试环境和方案

本文采用新型龙芯 1E 系列开发板作为验证和测试的目标机, 采用研华工控机作为宿主机。龙芯 1E 串口 0 和工控机相连, 将串口信息发送到工控机中。操作程序镜像通过串口 1 从工控机下载

到开发板中。

测试分为 3 部分:第 1 部分,通过系统运行,在 VxWorks shell 中输入 VxBusShow 检验 VxBus 驱动启动是否成功。第 2 部分,在应用层编写串口收发测试程序,通过串口收发数据各 1 500 次验证 VxBus 型串口驱动在实际应用中的可靠性。第 3 部分,分别在新型龙芯 1E 芯片和在轨星载计算机上创建相同应用程序,检验在复杂运算过程中, BSP 的表现情况和 BSP 性能对比。

6.2 实验过程

(1)在 WorkBench 开发环境中创建 1 个 Vx-Works Image 型工程,选择编辑本文实现的 BSP。编译镜像生成可执行文件 VxWorks。通过串口将镜像下载到开发板中运行。在 shell 中输入 Vx-

BusShow 显示当前系统的驱动和设备,可看到当前 VxBus 串口驱动、中断驱动、时钟驱动成功注册。如图 8 所示。

(2)创建 VxWorks Image 型工程,在 *usrAppInit()* 中创建任务:①通过串口接收数据 1 500 次,并将数据进行比对,输出接收总次数、接收数据成功次数、接收数据错误次数。②通过串口发送数据到宿主机 1 500 次,宿主机接收后在 SSCOM 串口中输出。图 9 为串口接收数据运行结果图。

由图 9 结果得出,分别接收和发送 1 500 次数据,均未出现误码等情况,实践证明 VxBus 型驱动具有很好的可靠性。

(3)创建 VxWorks Image 型工程,在 *usrAppInit()* 中创建 1 个任务:



Figure 8 Diagram of drive device hook up

图 8 驱动设备挂接图



Figure 9 Operation diagram of serial port receiving data

图 9 串口接收数据运行图

```
void usrAppInit(void){
    logMsg("----usrAppInit,----\r\n",0,0,0,0,0,0);
    taskSpawn (" test ", 100, 0x01000000, 2000000,
    (FUNCTPTR)main,0,0,0,0,0,0,0,0,0,0);
}
```

并将任务分别运行在移植成功的新型龙芯 1E 芯片和在轨星载计算机中,将 2 款计算机设置为相同 CPU 频率,通过运行时间对比验证其性能提升。该任务为武汉大学编写的导航卫星自主定轨的模拟程序,用 C++ 语言编写,并用到大量的浮点运算,包括协方差计算,读取存储 12×12 的矩阵,算法复杂度高,可以很好地检测 BSP 的运行状态和进行性能对比。如图 10 所示为模拟程序在目前在轨星载卫星系统中运行截图,如图 11 所示为模拟

程序在新型龙芯 1E 中运行截图。

由运行结果可知操作系统已启动且运行状态良好。运行 1 次模拟程序在轨星载计算机系统的时间为 38 s,在新型 1E 芯片中的时间为 21 s。以运行时间为衡量标准,移植后程序运行速度提升了 81%,性能提升明显,可以满足执行复杂应用的需求。

7 结束语

本文在新型龙芯 1E 开发平台上,针对硬件升级所提出的需求,完整开发了 VxWorks 操作系统的 BSP 包,包括内存分配、中断管理、时钟管理、串



Figure 10 Results of the program running on the on-board computer

图 10 在轨星载计算机中程序运行结果图



Figure 11 Results of the program running on the new Godson 1E

图 11 新型龙芯 1E 中程序运行结果图

口 VxBus 型驱动设计,并基于工程实践的星载计算机,设计了可靠合理的测试方案,测试结果说明操作系统移植成功,计算性能得到了提升,具备承担未来多样化星载软件任务的可行性,同时为未来 SpaceWire 高速通信接口应用开发奠定了技术基础。

参考文献:

- [1] China Satellite Navigation Office. Development of the BeiDou navigation satellite system: 3.0[R]. Beijing: China Satellite Navigation System Management Office, 2018. (in Chinese)
- [2] Xie Jun, Wang Jin-gang. Innovation and technology characteristics of Beidou-3[J]. Space International, 2017, 467(11): 6-9. (in Chinese)
- [3] Tang C, Hu X, Zhou S, et al. Initial results of centralized autonomous orbit determination of the new-generation BDS satellites with inter-satellite link measurements[J]. Journal of Geodesy, 2018, 92(10): 1155-1169.
- [4] Beidou Chief Engineer. Complete the next generation Beidou system constellation network in 2035 [EB/OL]. [2018-05-28]. <http://Baijiahao.baidu.com/s?id=1601659707685806609&wfr=spider&for=pc>. (in Chinese)
- [5] Fan Ling-ling. Design and implementation of high performance computing unit based on Godson processor and its key functions [D]. Shanghai: Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, 2016. (in Chinese)
- [6] Institute of Computing Technology, Chinese Academy of Sciences. Godson 1E processor user manual [Z]. Beijing: Godson Zhongke Technology Company, 2017. (in Chinese)
- [7] Zhou Qi-ping, Zhang Yang. Device driver and BSP development guide under VxWorks [M]. 1st ed. Beijing: China Electric Power Press, 2004. (in Chinese)
- [8] Wind River Systems Inc. . BSP develop's guide, 6.8[Z]. Alameda, CA: Wind River Systems Inc. , 2010.
- [9] An Jun-she, Liu Yan-qiu, Sun Hui-xian. Design and realization of board support package based on RTOS VxWorks[J]. Computer Engineering, 2003, 29(1): 87-88. (in Chinese)
- [10] Huang Chao, Chen Yong, Lin Bao-jun. Research of fault-tolerant booting for on-board computer based on anti-rediation Loongson[J]. Computer Science, 2016, 43(Z2): 532-535. (in Chinese)
- [11] Chen Xue-bing, Shen Yi-nan, Zhang Zhen-hua. Transplantation of VxWorks 5.5 in Godson-2 processor and its performance analysis[J]. Computer Measurement & Control, 2012, 20(9): 2542-2545. (in Chinese)
- [12] Sweetman D. See MIPS run [M]. 2nd ed. San Mateo, CA: Morgan Kaufmann Publishers Inc. , 2006.

附中文参考文献:

- [1] 中国卫星导航系统管理办公室. “北斗”卫星导航系统发展报

告: 3.0[R]. 北京: 中国卫星导航系统管理办公室, 2018.

- [2] 谢军, 王金刚. 北斗-3 卫星的创新和技术特点[J]. 国际太空, 2017, 467(11): 6-9.
- [4] 北斗总师: 2035 年完成下一代北斗系统星座组网[EB/OL]. [2018-05-28]. <http://baijiahao.baidu.com/s?id=1601659707685806609&wfr=spider&for=pc>.
- [5] 范玲玲. 基于龙芯处理器的高性能运算单元及其关键功能设计与实现[D]. 上海: 中国科学院上海微系统与信息技术研究所, 2016.
- [6] 中国科学院计算技术研究所. 龙芯 1E 处理器用户手册[Z]. 北京: 龙芯中科技术有限公司, 2017.
- [7] 周启平, 张杨. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 第 1 版. 北京: 中国电力出版社, 2004.
- [9] 安军社, 刘艳秋, 孙辉先. VxWorks 操作系统板级支持包的设计与实现[J]. 计算机工程, 2003, 29(1): 87-88.
- [10] 黄超, 陈勇, 林宝军. 基于抗辐照龙芯的星载计算机容错启动研究[J]. 计算机科学, 2016, 43(Z2): 532-535.
- [11] 陈学兵, 沈毅南, 张振华. VxWorks 5.5 在龙芯 2 号处理器的移植和性能分析[J]. 计算机测量与控制, 2012, 20(9): 2542-2545.

作者简介:



徐帅(1993 -), 男, 河北沧州人, 硕士生, 研究方向为嵌入式软件。E-mail: xushuai17@mails.ucas.ac.cn

XU Shuai, born in 1993, MS candidate, his research interest includes embedded software.



林宝军(1963 -), 男, 吉林长岭人, 博士, 研究员, 研究方向为计算机控制与应用。E-mail: linbaojun@aoe.ac.cn

LIN Bao-jun, born in 1963, PhD, research fellow, his research interest includes computer control & application.



刘迎春(1975 -), 女, 江苏丹阳人, 博士, 研究员, 研究方向为飞行器总体设计。E-mail: liuyingchun@csu.ac.cn

LIU Ying-chun, born in 1975, PhD, research fellow, her research interest includes aircraft overall design.



赵帅(1987 -), 男, 山东济宁人, 硕士生, 工程师, 研究方向为航天通用产品研发。E-mail: 472230319@qq.com

ZHAO Shuai, born in 1987, MS candidate, engineer, his research interest includes aerospace general product development.