翼辉信息 ACOINFO

单片机与嵌入式系统应用 Microcontrollers & Embedded Systems

北京航空航天大学出版社 BEIHANG UNIVERSITY PRESS

SylixOS教学实践公开课 第3讲

主　题：**实时操作系统服务机制及其特性**
—— 以SylixOS为例

主讲人：**张凯龙** 教授、博导

西北工业大学—翼辉信息嵌入式操作系统联合实验室 负责人
中国计算机学会嵌入式系统专委 秘书长

内容提纲

- 理解实时计算系统及其本质
- 实现实时计算系统的关键方面
- SylixOS的实时保障机制及其优势

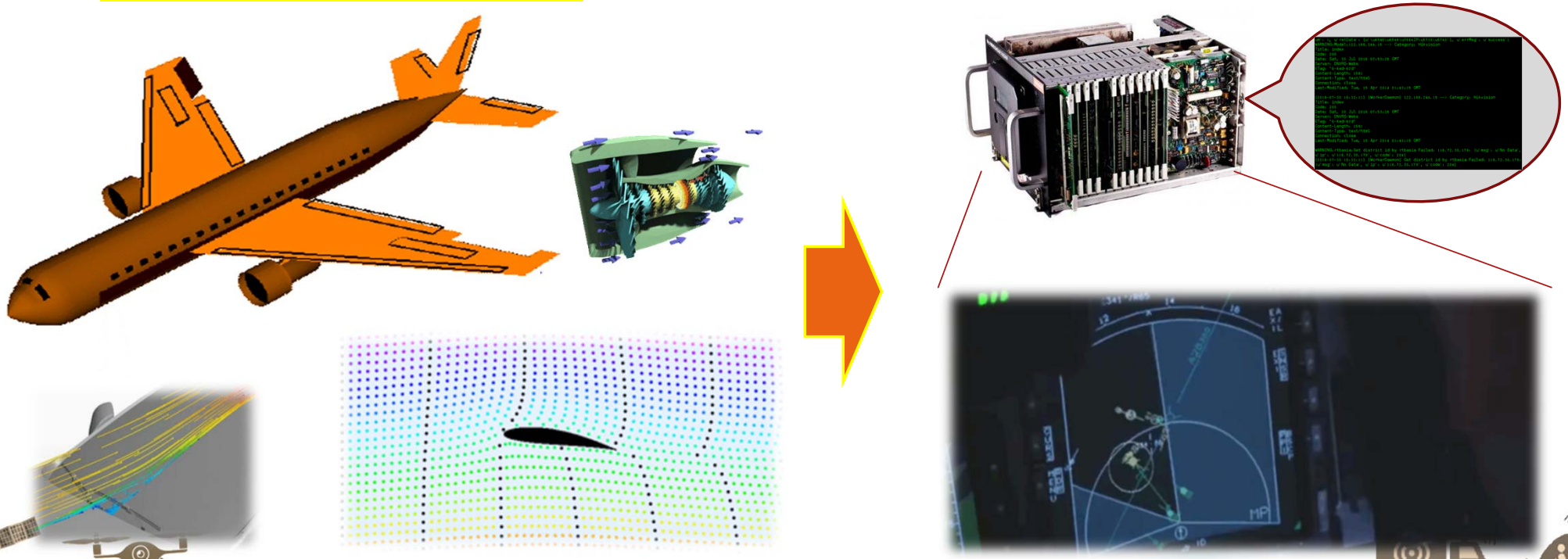
1 理解实时计算系统及其本质

实时（计算）系统：行为过程受环境约束的计算系统

1. 融合于物理环境的计算系统 → 嵌入式系统、信息物理系统
2. 对外界环境做出“快速、及时”反应 → 计算需要满足时间约束

理解本质

- 以环境状态变化为步长、须在限定时间内完成动作的**反应式系统**；
- **一类信息物理系统**：物理环境交互 → 物理规律约束 → 计算时间限制 → **对计算任务的时间约束！**（PS：计算任务是实现计算系统功能的基本对象）



定义与内涵

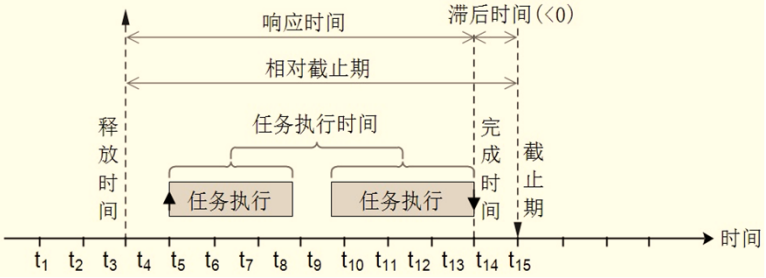
- 与物理环境交互 → 受物理规律约束 → 计算的"时间限定"属性，要求计算结果要满足时间约束；
- POSIX 1003.b 实时性标准中的描述，实时是指系统能够在限定的响应时间内提供所需水平的服务；

系统的计算必须产生正确的结果，称为逻辑或功能正确（Logical or Functional Correctness）；
系统的计算必须在预定的时间内完成，称为时间正确（Timing Correctness）

- **不论任务什么时候到来，其都能在可估计的时间内响应和完成。**

实时计算的实现

- 任务的时间属性 — 截止期；
- 复杂系统 → 实时操作系统是关键。



截止期引出的任务调度特性

- （任务）可调度（Feasible schedule），对于每一个具有截止期要求的任务，不论何时，只要在释放时间时（后）启动，就都能够在截止期之前完成；
- 可调度性（Schedulable），对于一个调度算法，一组任务总是有可行的调度方案；如果一个系统中的所有任务都是可调度的，就可以说这个系统就是实时的；
- 调度优化（Optimal Schedule），只要存在可调度的方案，调度算法总是可以找到这个可行的调度序列；
- 错失率（Miss rate），已执行任务中，完成执行但超过截止期的任务所占的比例；
- 丢失率（Loss rate），丢弃的任务所占的比例；
- 失效率（Invalid rate），等于"错失率+丢失率"。

2 实现实时计算系统的关键方面

问题 与 对策

总目标：让紧急任务尽可能先执行且尽可能“快”地完成执行

翼辉信息 ACOINFO

分解的因素

① 任务可以执行的条件是什么？

- ✓ 得到所需的资源→转入就绪队列→择机执行

② 如何让紧急任务优先执行？

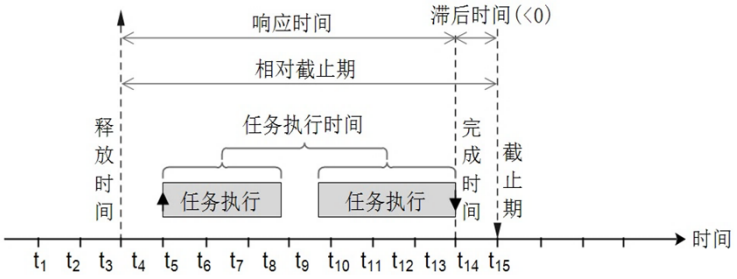
- ✓ 给紧急任务分配高优先级
- ✓ 抢先式调度机制→静态、动态调度策略

③ 如何保证截止期不被错过？（参考问题1）

- ✓ 尽快赋予所需的所有资源
- ✓ 中断、事件的响应时间可预测
- ✓ 可调度性设计与验证

④ 延伸问题

- ✓ 基于实时操作系统设计的系统一定实时吗？
- ✓ 实时操作系统如何能为紧急任务尽快分配所需的资源？（能否做到随需随分配？）



任务调度

- 是任务管理的重要功能，由内核中的任务调度器根据具体的调度算法和策略对就绪队列中的任务进行 选择 和 切换。
- 单调速率调度（Rate Monotonic Scheduling, RM），衍生的单调截止期调度（Deadline Monotonic Scheduling, DM）

算法的核心思想是，根据周期为每个任务分配一个静态优先级，任务周期越短（即执行速率越快），优先级越高，进而执行基于优先级的抢先式动态调度。

- 动态优先级调度算法，任务的优先级在系统运行过程中随着某些因素动态变化。
- 最小截止期的任务优先级最大，截止期越大，任务的优先级越低。
- 不仅适用于周期性任务，也可用于非周期任务，但开销较大。
- 只要是其他算法可调度的任务组，EDF就能调度。

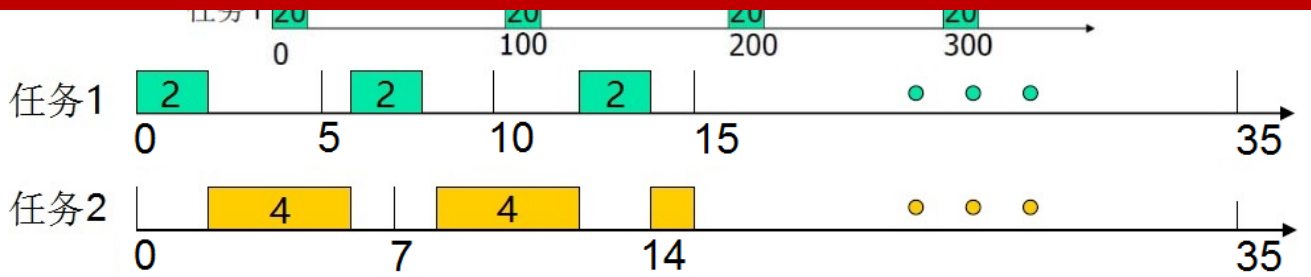


图8.20 两个周期任务上的RM调度过程

- 为周期性任务解决多任务调度冲突的一种非常好的方法是速率单调调度RMS（Rate Monotonic Scheduling），RMS基于任务的周期指定优先级。
- RMS主要用于工业控制、工业机器人等对实时性要求高，周期任务抖动小的周期任务。
- SylixOS已成功应用于新松、哈工大、埃斯顿、纳博特等多家工业机器人企业的项目和产品的控制器上。

SylixOS RMS性能测试

SylixOS RMS特点：

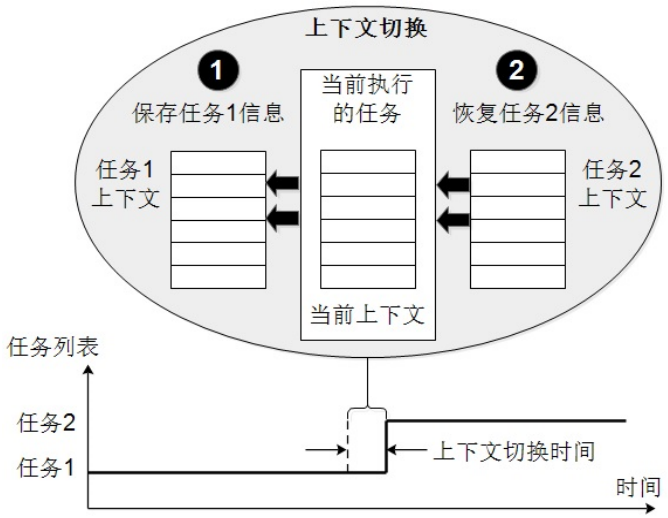
- 获取CPU频率来获取高达ns级的精度（需高精度时钟）。
- 支持周期偏离自动矫正，感知错误。
- 解决周期调度稳定，精确问题，周期任务抖动小。

SylixOS RMS 性能测试

- 抖动测试环境由PC、运动控制器和电机组成，如右图所示。
- 实测结果：2000us 周期任务抖动在 5us 内。

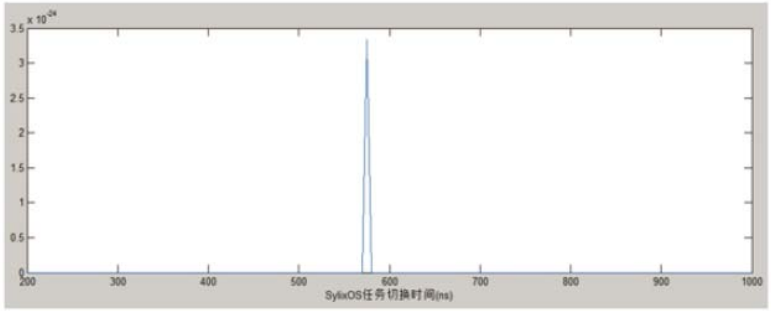
```
00027cb7: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/2000.000/2001.928
00028487: ********************************************************************************
00028487: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
00028c57: ********************************************************************************
00028c57: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
00029427: ********************************************************************************
00029427: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/2000.000/2001.928
00029bf7: ********************************************************************************
00029bf7: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
0002a3c7: ********************************************************************************
0002a3c7: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
0002ab97: ********************************************************************************
0002ab97: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/2000.000/2001.928
0002b367: ********************************************************************************
0002b367: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
0002bb37: ********************************************************************************
0002bb37: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/2000.000/2001.928
0002c307: ********************************************************************************
0002c307: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/1999.999/2001.928
0002cad7: ********************************************************************************
0002cad7: PerfMsmt 'Cycle Time            ' (min/avg/max) [usec]: 1999.152/2000.000/2001.928
0002cad7: ################################################################################
0002cad7: Test RMS PASS. RMS performance is pefect A+, jitter TIME less then 5 us.
0002cad7: ################################################################################
```

任务调度：CPU环境中任务的换入换出 → 上下文切换



任务切换时间：

同样在 Cortex-A8 处理器上进行测试，每秒进行任务切换至少 100 次，总计进行 12 万次任务切换。统计结果如下：



12万次切换	时间
最大任务切换时间（ns）	890.0
最小任务切换时间（ns）	470.0
平均任务切换时间（ns）	577.1

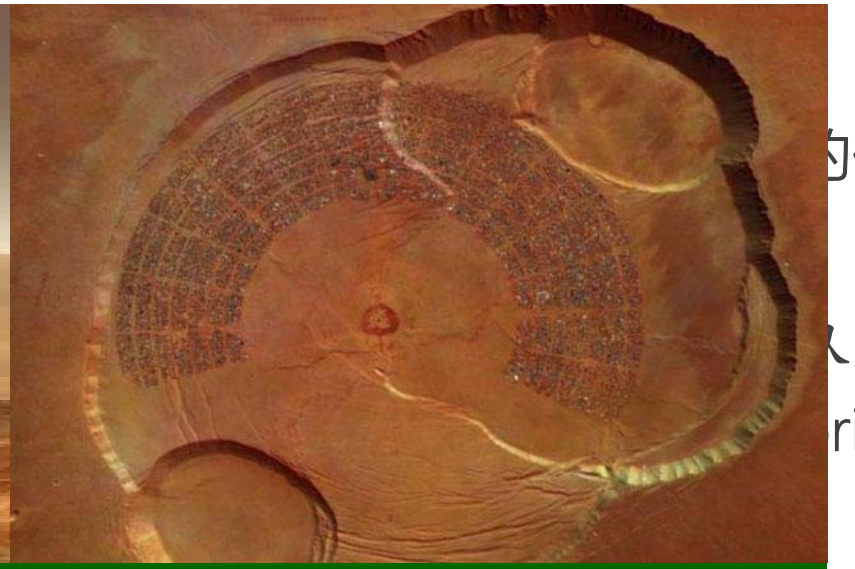
基于龙芯LS2K1000实测

任务切换时间测试	avg time/ns	max time/ns	min time/ns
testPreeptive	4625	15336	3336

关键机制2：资源管理优化 → 减少紧急任务被阻塞时间

优先级翻转

- 理想地，基于 的任务优先执行；
- 实际中，常出 从而导致高优先 rity Inversion)



1996年12月4日 美国NASA发射“火星探路者”

1997年7月4日在火星表面登陆，之初的一段时间，探测器的工作比较稳定，但随后系统出现了频繁的复位、数据丢失现象。

低优先级的数据采集任务执行期间，高优先级的总线管理任务阻塞等待数据，此时到来的中优先级数据通信任务会抢先执行；由于数据通信任务的执行时间较长，因此低优先级的数据采集任务和高优先级的总线管理任务长时间不能执行。系统看门狗发现总线长时间无响应，“认为”系统发生了严重的错误，自动复位。

解决优先级翻转问题——方法1 ★★★★★

优先级继承协议 Priority Inheritance Protocol, PIP

- 核心思想：优先级翻转问题发生时，让持有共享资源的低优先级任务获取被阻塞高优先级任务的优先级，以尽快执行并释放共享资源，进而使高优先级任务能够得到快速响应。
- **方法举例**
  - SylixOS中默认采用优先级继承，LW_OPTION_INHERIT_PRIORITY；
  - μC/OS III中使用OSMutexCreate()创建的互斥信号量默认就具有优先级继承的属性；
  - VxWorks中使用semMCreate()函数以及SEM_Q_PRIORITY 和SEM_INVERSION_SAFE属性，才可创建具有优先级继承能力的互斥信号量；

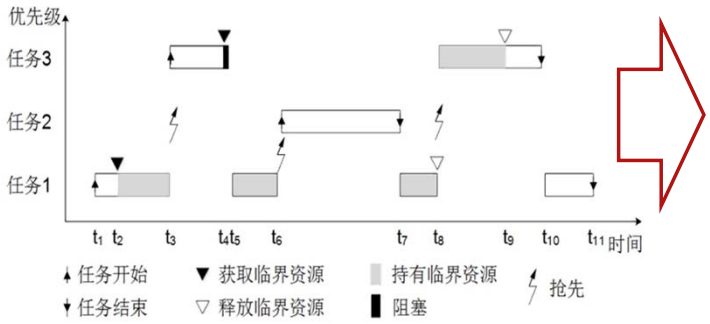


图8.15 优先级翻转示例

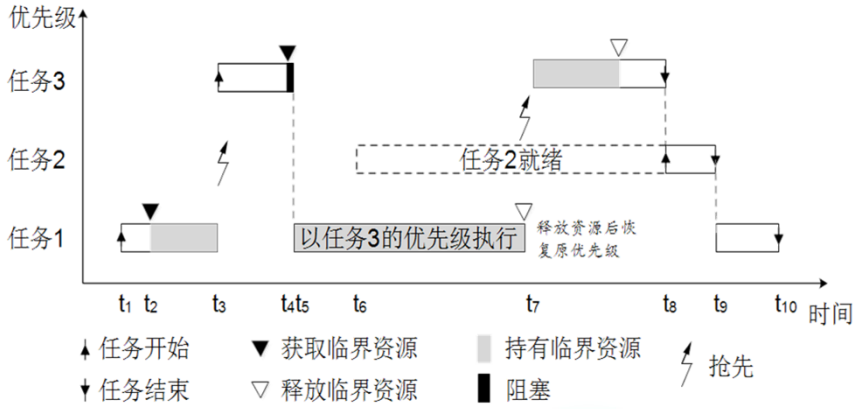
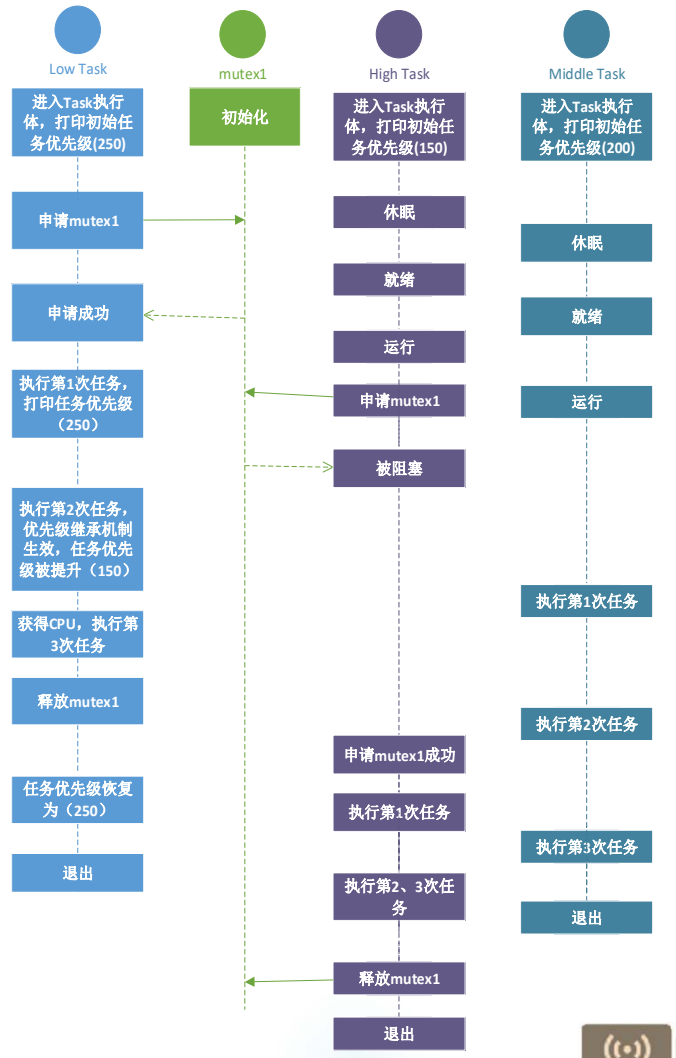


图8.16 优先级继承方法示例

SylixOS 优先级继承示例

- 创建优先级低、中、高三个任务：low task 、 middle task、high task ；
- low task和high task任务依赖资源互斥锁mutex1；
- low task首先就绪并被调度运行。

- POSIX标准中优先级数值越大，级别越高，而SylixOS则相反。



解决优先级翻转问题——方法2 ☆☆☆

优先级天花板协议 Priority Ceiling Protocol，PCP

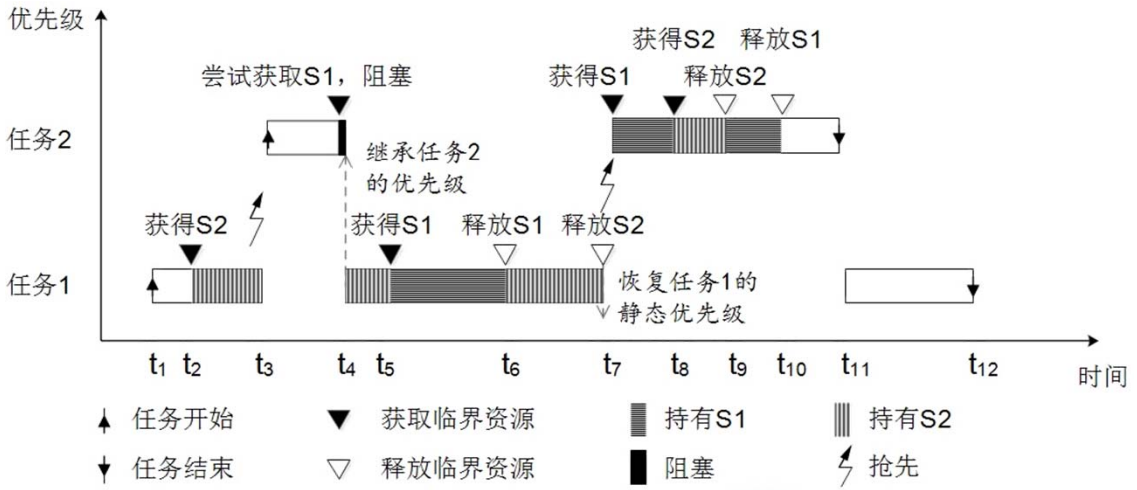
- 优先级天花板（Priority Ceiling）意味着要为每个临界资源赋予一个优先级；
- 基本的优先级天花板协议可描述为：
  1. 任务在临界资源以外时，以原有的优先级运行；
  2. 当一个任务t_i尝试获取一组所需临界资源中的一个资源S时
     - ✓ 如果任务t_i的优先级严格大于已被其他任务所持有临界资源的优先级天花板，任务将能够获得该临界资源；
     - ✓ 否则，任务t_i被阻塞，而且持有共享资源的任务继承任务t_i的优先级。

要理解这个较为抽象的优先级天花板协议，一定要记住“高优先级任务抢先执行”这个前提；不仅任务有优先级，资源也有优先级。
协议的第二条是核心，不但涉及任务优先级的管理，还包括了对死锁和阻塞传递的预防。

➢ 原始天花板优先级协议（Original Ceiling Priority Protocol, OCPP）

- 每个任务都有一个默认的静态优先级；
- 各共享资源分别有一个静态天花板优先级，设定为使用该资源的任务中的最大优先级；
- 任务有一动态优先级，是其静态优先级和继承自阻塞者任务的优先级中的最大者；
- 约定：当一个任务申请一个资源时，如果其动态优先级高于任何被其他任务持有的资源的优先级天花板时，该任务将能够获得所申请资源。

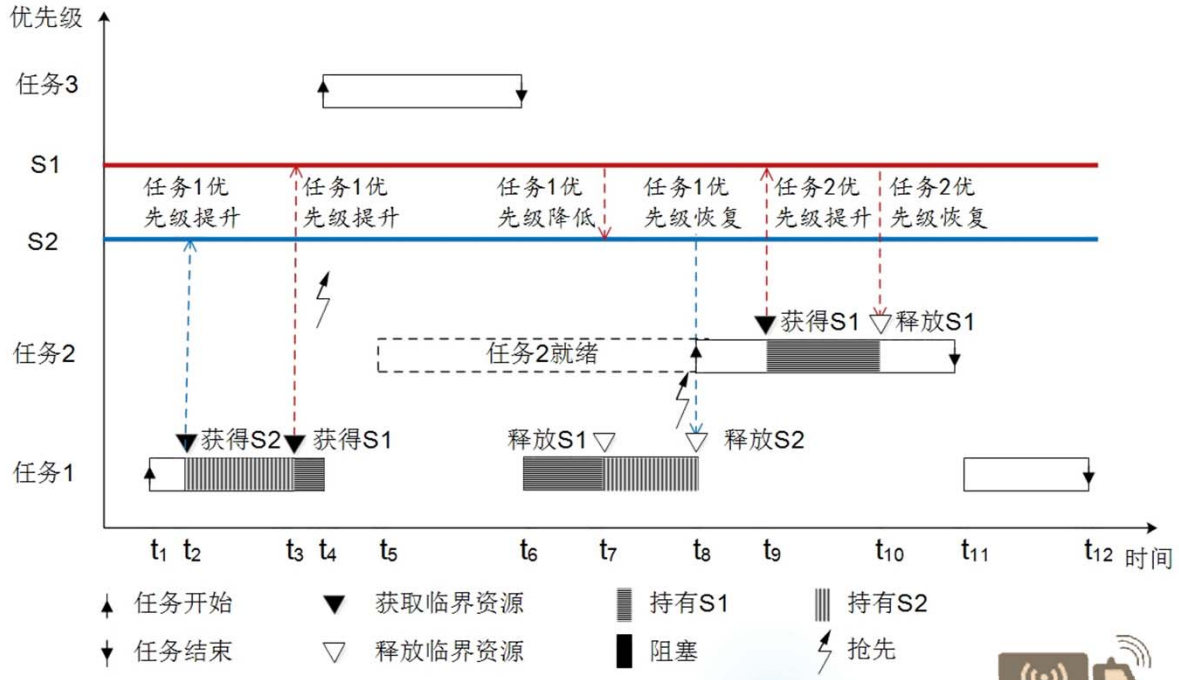
假设任务2的优先级高于任务1的优先级，两个任务都使用资源S1和S2，那么，两个临界资源的优先级天花板都将等于任务2的优先级。



立即天花板优先级协议（Immediate Ceiling Priority Protocol, ICPP）

- 和OCPP中的约定相同，任务有一个默认的静态优先级，同时每个资源有一个静态的优先级天花板；
- 不同的是，任务的动态优先级是其静态优先级及其所持有所有资源的优先级天花板中的最大值。

注意，该协议中任务的动态优先级管理更为简化，仅依赖于所持有的资源，不再从其他任务继承。



3.2 提高优先级

低优先级的任务占有资源，且高任务的优先级请求资源时，会尝试提高低优先级任务的优先级，在申请资源时调用_EventPrioTryBoost函数尝试提高任务优先级。

```
#include <SylixOS.h>
VOID  _EventPrioTryBoost (PLW_CLASS_EVENT  pevent, PLW_CLASS_TCB  ptcbCur)
```

函数_EventPrioTryBoost原型分析：

- 参数pevent为资源所属的事件控制块；
- 参数ptcbCur为当前任务控制块。

该函数根据互斥量属性块的属性设置，确定选用优先级继承策略_SchedSetPrio函数改变任务的优先级。

3.3 恢复优先级

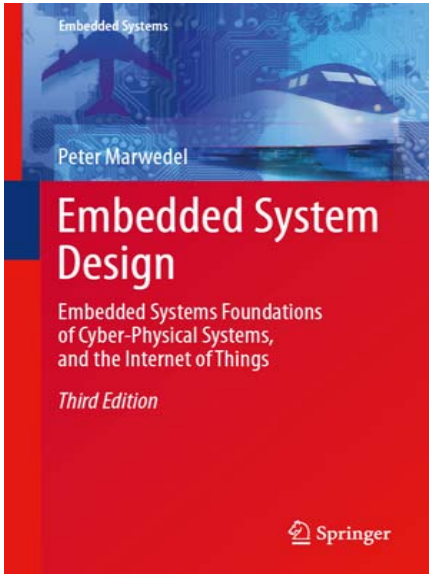
低优先级任务释放互斥量时，调用EventPrioTryResume尝试恢复任务的优先级。

```
#include <SylixOS.h>
VOID  _EventPrioTryResume (PLW_CLASS_EVENT  pevent, PLW_CLASS_TCB  ptcbCur)
```

函数_EventPrioTryResume原型分析：

- 参数pevent为资源所属的事件控制块；
- 参数ptcbCur为当前任务控制块。

解决优先级翻转问题的其他方法



关键方面3：确定、快速的中断响应 → 时间正确性

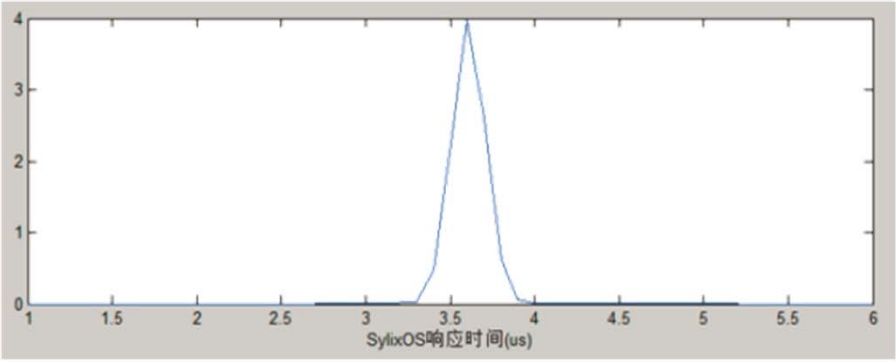
中断延迟与中断响应时间 → EOS内核机制的优化设计

- 中断延迟是指从中断发生到中断处理程序开始执行所经历的时间间隔；
  中断延迟包括：最大中断禁止时间和系统为ISR保存、加载寄存器和数据的时间。
- “中断延迟+现场保护时间”是中断响应时间。

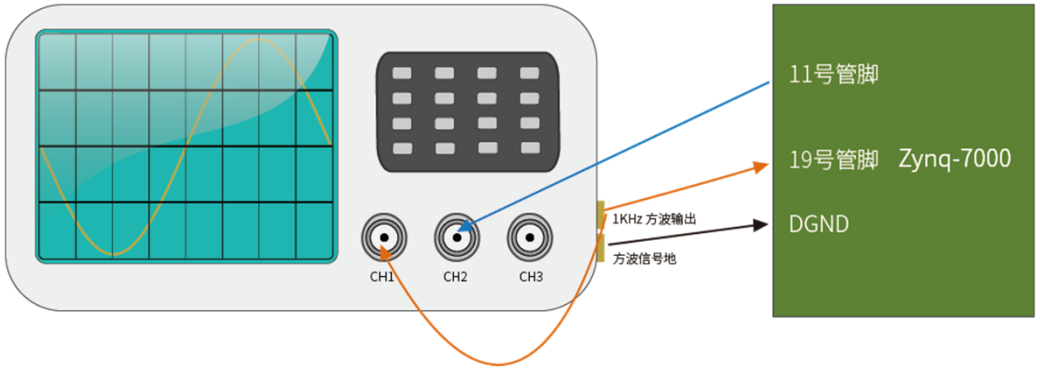
中断响应时间：

翼辉信息对 SylixOS 的中断响应时间和任务切换时间进行了测试。在 Cortex-A8 处理器上对 120 万次中断响应时间进行统计，统计结果如下：

120万次中断	时间
最大响应时间（us）	4.100
最小响应时间（us）	2.900
平均响应时间（us）	3.612



中断响应能力测试



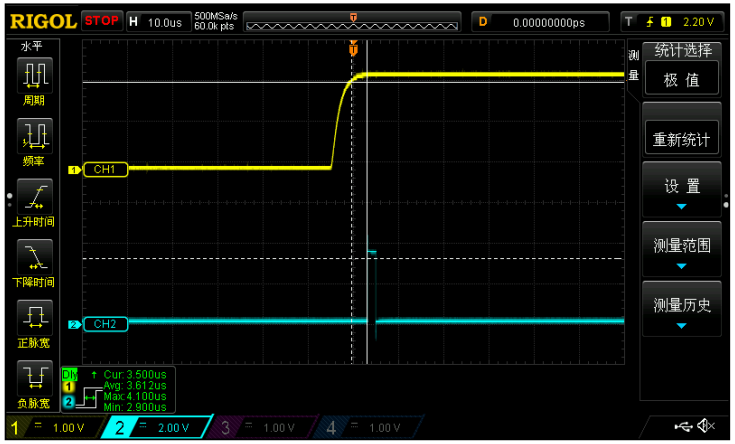
- 配置目标设备（Zynq-7000）的19号管脚为外部中断模式并将11号管脚配置为输出模式；
- 使用示波器自带的1KHz方波作为目标设备（Zynq-7000）外部中断触发源并连接19号管脚；
- 将目标设备（Zynq-7000）的11号管脚接入示波器的输入通道（CH2）；
- 进行120万次测试并统计中断响应时间。

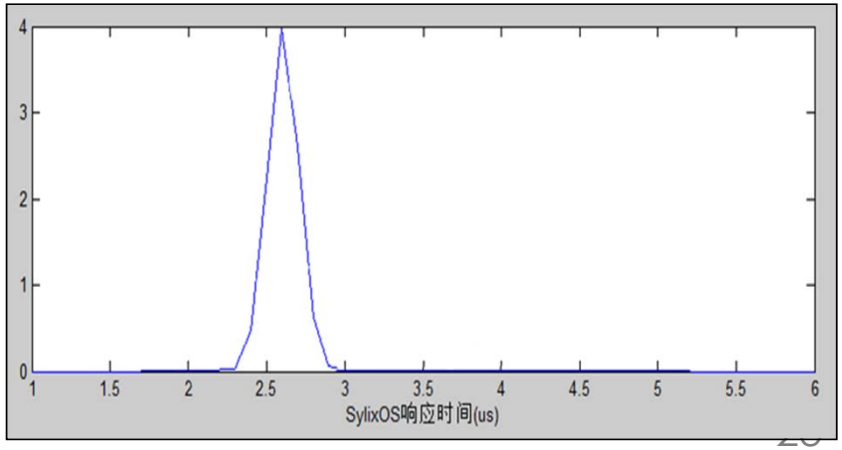
示波器显示结果

120万次正态分布图

120万次统计结果

120万次中断	时间
最大响应时间	3.670
最小响应时间	2.310
平均响应时间	2.987





关键方面4：高可靠 → 逻辑正确性

翼辉信息 ACOINFO

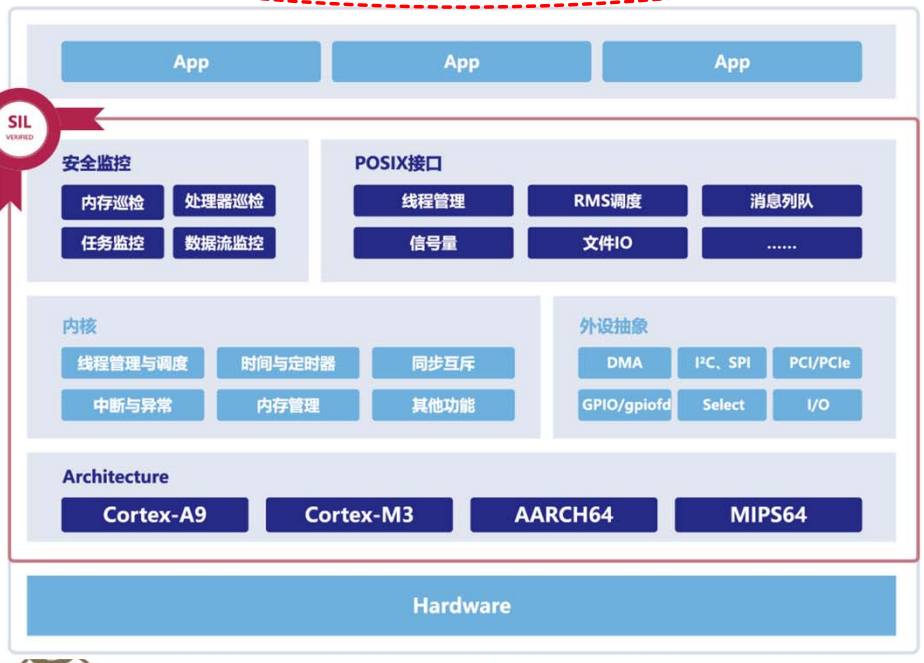
SylixOS的长期应用检验，已通过SIL安全认证

SIL: Safety Integrity Level
安全完整性等级

系统安全点加固
内核关键数据采用汉明码定义
内存巡检
数据流监控
任务监控

安全加固

解决单粒子翻转导致的系统下安全运行
降低内存故障导致的系统风险
保障程序执行逻辑安全
保障程序执行时间安全



DAkkS Deutsche Akkreditierungsstelle D-ZE-11190-01-00

TÜV SÜD

CERTIFICATE

No. Z10B 113242 0002 Rev. 00

Holder of Certificate: BEIJING ACOINFO TECHNOLOGY CO., LTD
Building No.12, Zhongguancun Cuihu Technology Park
Haidian District
100095 Beijing
PEOPLE'S REPUBLIC OF CHINA

Factory(ies): 113242

Certification Mark:

Product: Software, Operating Systems
Real Time Operating System

Model(s): SylixOS Safe Certified

Parameters: Safety Parameters: SIL 3 IEC 61508-3
SIL 4 EN 50128
ASIL D ISO 26262

Tested according to: EN 50128:2011/A2:2020
IEC 61508-1:2010
IEC 61508-3:2010
ISO 26262-2:2018
ISO 26262-6:2018
ISO 26262-8:2018

The product was tested on a voluntary basis and complies with the essential requirements. The certification mark shown above can be affixed on the product. It is not permitted to alter the certification mark in any way. In addition the certification holder must not transfer the certificate to third parties. See also notes overleaf.

Test report no.: BB98313T, V1.0
Valid until: 2027-03-16

Date, 2022-03-21 (Guido Neumann)

Page 1 of 1
TÜV SÜD Rail GmbH • Certification Body • Barthstraße 16 • 80339 München • Germany

ZERTIFIKAT ◆ CERTIFICATE ◆ 認證證書 ◆ СЕРТИФИКАТ ◆ CERTIFICADO ◆ CERTIFICAT

翼辉信息 ACOINFO

SylixOS™
embedded

自主实时操作系统引领者

SylixOS Source v1.4.3

清华大学
http://coop.tunahua.edu.cn/git

SylixOS IDE v3.2.5

北京航空航天大学
http://www.ldmc.buaa.edu.cn/sylixos

IBM OpenSSL SQLite Atmel python EMBEDTHIS LOONGSON 龙芯 intel 君正 Ingenic NXP unixODBC Qt freescale SAMSUNG FastDB TEXAS INSTRUMENTS XILINX Espruino μGFX Lua GoAhead Phytium 飞腾 国微电子

军用案例


弹载制导与控制系统


星载任务计算机系统


潜艇电力控制系统


轮式装甲车全车电子监控系统


远程火箭炮制导系统

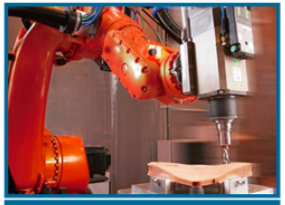

多轴无人机

民用案例

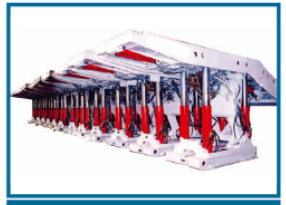

家用新能源发电系统

ABB 电气火灾报警系统


机房动力环境监控系统


多轴联动机器人


世博会超级电容车


煤矿智能液压支架

3 SylixOS的实时保障机制及其优势

国内实时操作系统发展

起步较晚 (2000 年后)

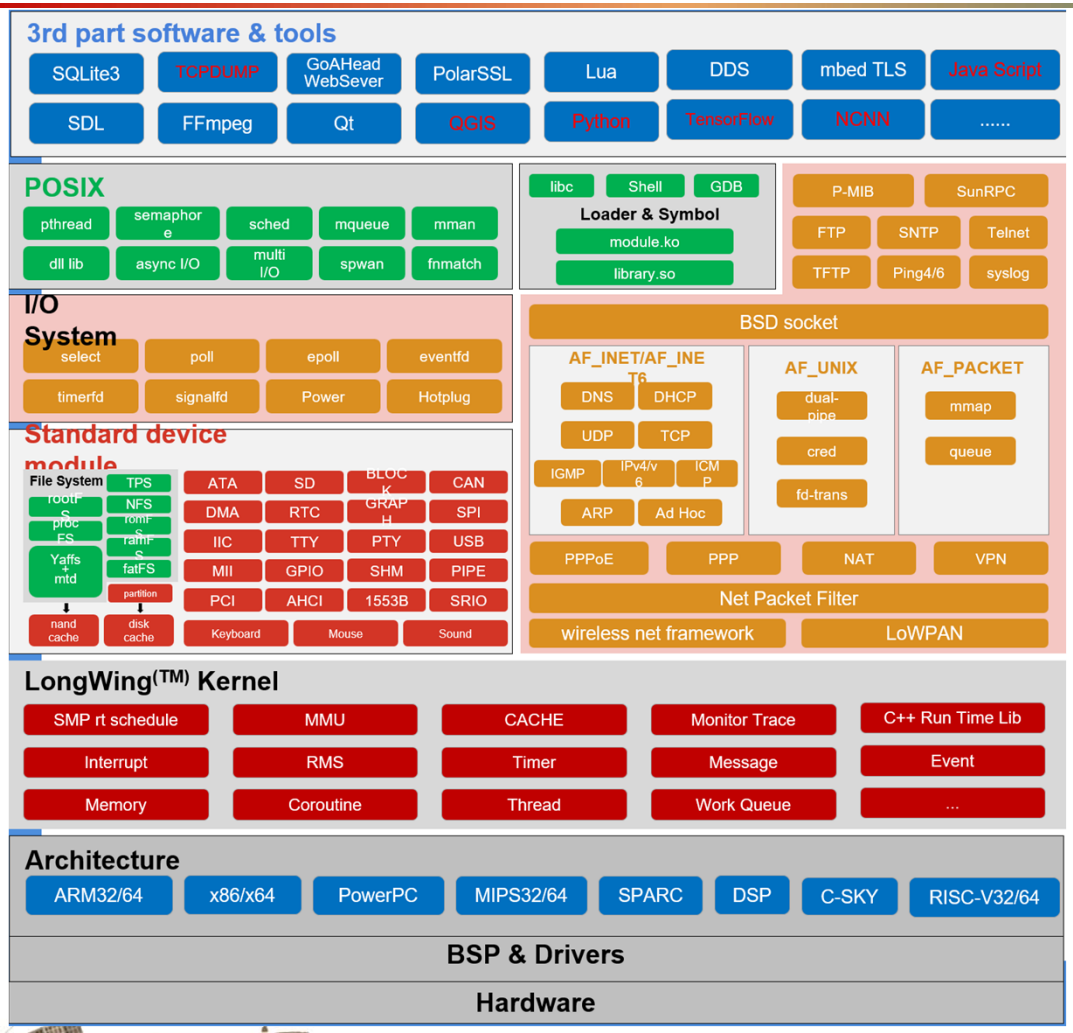
- 早期单片机不需要操作系统
- 高性能嵌入式芯片在国内普及较晚

内核功能单一

- 多仅实现任务调度, 功能不完备
- 目前SylixOS是国产唯一大型实时操作系统

缺乏形成大型实时系统的环境

- 复杂软件的开发能力受限



- 全面支持八大架构，新增LoongArch、DSP、SPARC、C-SKY、RISC-V32/64；
- 支持TCPDUMP、Python、JavaScript、TensorFlow、NCNN、QGIS等中间件；
- 图形化集成开发环境；
- ……
- 良好的实时性能！

系统功能	SylixOS	VxWorks	RTEMS	QNX
内核抢占	■	■	■	■
优先级	256	256	256	256
优先级继承	■	■	■	■
任务数量	无限	无限	无限	无限
进程支持	POSIX 进程	RTP 进程	□	POSIX 进程
协程（纤程）	■	□	□	□
RMS 调度	■	□	■	□
动态装载	■	■	□	■
异步IO	■	■	□	■
自组网协议	MAODV	□	□	□
UNIX 域协议	■	□	□	■
内置热插拔	■	□	□	□
高速定时器	■	■	□	□

SylixOS 三大特点

与 

一样的开发体验

与 

一样的兼容接口

与 

一样的实时性能

SylixOS 应用编程接口符合 IEEE、ISO、IEC 相关操作系统编程接口规范，兼容 POSIX 1003.1b 实时编程标准。基于 Linux、VxWorks 操作系统的应用程序，可以方便快捷地移植到 SylixOS 系统上运行。

工信部实时性测试

编码：CSIP-RF-12-032-S-V4.3　　编号：SP-2015-SW-0065-TR-11

SylixOS 实时性

测试报告

编制：　　2015 年 12 月 31 日

审核：　　2015 年 12 月 31 日

批准：　　2015 年 12 月 31 日

工业和信息化部软件与集成电路促进中心

赛普评测中心

工业和信息化部软件与集成电路促进中心赛普评测中心　　SP-2015-SW-0065-TR-11

（三） 多核无压力测试

使用多核测试开发板 E9（SylixOS STABLE-1.1.1），通过测试工具 Cyclictest 进行 1000000 次循环测试，设定调度的间隔时间为 1ms，对 1000us 以下的延迟进行统计，比较真实时间与计算时间的差值来获得实际过程中的延迟时间，通过延迟时间的大小来进行操作系统的实时性测试，测试结果数据如下：

延迟时间	值			
	CPU#0	CPU#1	CPU#2	CPU#3
最大延迟时间	13us	10us	6us	9us
最小延迟时间	2us	1us	1us	1us
平均延迟时间	4us	3us	2us	2us

（四） 多核有压力测试

使用多核测试开发板 E9（SylixOS STABLE-1.1.1），通过 Hackbench 不断地进行进程间通信以及进程创建，从而提供一个压力环境，通过 Cyclictest 测试工具进行 100000 次循环测试，设定调度的间隔时间为 1ms，对 1000us 以下的延迟进行统计，比较真实时间与计算时间的差值来获得实际过程中的延迟时间，通过延迟时间的大小来进行操作系统的实时性测试，测试结果数据如下：

延迟时间	值			
	CPU#0	CPU#1	CPU#2	CPU#3
最大延迟时间	17us	19us	16us	25us
最小延迟时间	2us	2us	2us	2us
平均延迟时间	5us	5us	5us	5us

详细结果参见本报告测试内容。

工业和信息化部软件与集成电路促进中心
赛普评测中心
2015 年 12 月 31 日

地址：北京市海淀区羊坊店东路 5 号博雅园 10 号（100038）　　第 2 页 共 9 页
电话：010-63951881-82128155　　传真：010-63973485
E-mail：testlab@csip.org.cn

2015 年12 月31 日，工业和信息化部赛普评测中心（简称“赛普评测中心”）对 SylixOS 实时操作系统进行了实时性测试，并出具了测试报告。此报告是工信部第一份关于操作系统实时性的测试报告。

赛普评测中心的整个测试过程依据标准（规范）GB/T 25000.51-2010《软件工程软件产品质量要求与评价（SquaRE ）商业现货（COTS）软件产品的质量要求和测试细则》，分别对 SylixOS 在单核无压力、单核有压力、多核无压力和多核有压力四种情况下的实时性进行了测试（激励与测量时间计入延迟时间）。

（一）单核无压力测试

延迟时间	值
最大延迟时间（μs）	12
最小延迟时间（μs）	1
平均延迟时间（μs）	2

（二）单核有压力测试

延迟时间	值
最大延迟时间（μs）	26
最小延迟时间（μs）	2
平均延迟时间（μs）	3

（三）多核无压力测试

延迟时间	值			
	CPU#0	CPU#1	CPU#2	CPU#3
最大延迟时间（μs）	13	10	6	9
最小延迟时间（μs）	2	1	1	1
平均延迟时间（μs）	4	3	2	2

（四）多核有压力测试

延迟时间	值			
	CPU#0	CPU#1	CPU#2	CPU#3
最大延迟时间（μs）	17	19	16	25
最小延迟时间（μs）	2	2	2	2
平均延迟时间（μs）	5	5	5	5

SylixOS为什么是一款优秀的实时EOS?

翼辉信息 ACOINFO

开源实时操作系统，优势突出!

1. 适用嵌入式开发

嵌入式系统的开发工作主要是在非标准硬件平台上开展的，基于开源系统，将使系统移植和定制化开发更加容易。

2. 提高系统可靠性

嵌入式系统的首要要求是安全、可靠。开源系统的安全性和可靠性更容易验证，代码允许公众审查，其 Bug 也易于发现和修补，代码质量更有保障。

3. 降低使用风险

用户可以获取系统源代码，培育自己的团队对系统进行维护，不需要担心操作系统原有版本升级后，出现旧版本系统无人维护等风险。

4. 便于故障定位

嵌入式系统在开发过程中，很容易出现图形显示、网络通讯、外设异常等故障，开源系统可避免闭源系统带来的故障定位难、排查周期长、影响研发进度等问题，提高故障定位的效率。

5. 技术透明度高

开源系统的发展由社区推动，用户可以随时获取到最新信息，甚至参与到系统的演变中，系统的发展不再受限于一家公司的意愿，用户可以了解系统的未来发展规划和方向。

6. 应用场景丰富，得到充分验证


ACOINFO
北京翼辉信息技术有限公司
西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY
西工大 – 翼辉
嵌入式操作系统联合实验室

翼辉信息 ACOINFO

单片机与嵌入式系统应用 Microcontrollers & Embedded Systems

北京航空航天大学出版社 BEIHANG UNIVERSITY PRESS

谢谢！

kl.zhang@nwpu.edu.cn