

TOPPERS/ASP内核介绍与使用

版本：0.02

任 慰

华中科技大学
瑞萨高级嵌入式控制器实验室

2011 年 12 月 8 日

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

The Real-time Operating Nucleus

Industrial TRON

- TRON是一项开放式的实时操作系统内核设计项目,1984年由坂村健¹教授发起,宗旨是为全社会的需要开发一套理想的计算机结构和网络。
- ITRON为TRON计划中的一部分,是有关嵌入式实时操作系统(RTOS)的一系列标准规范
 - TRON计划中的主要代表部分(BTRON, CTRON, JTRON等项目很早就已停止)
 - 开源化,弱标准化,主要针对小型嵌入式系统
 - 目前已经有4代,最新一代 μ ITRON4.0与1999年发布
- 世界上应用最多的操作系统之一,出现在数以百万计的电子设备上(或许)

¹坂村健,东京大学教授,日本著名计算机结构专家,同时也是高田广章教授的导师

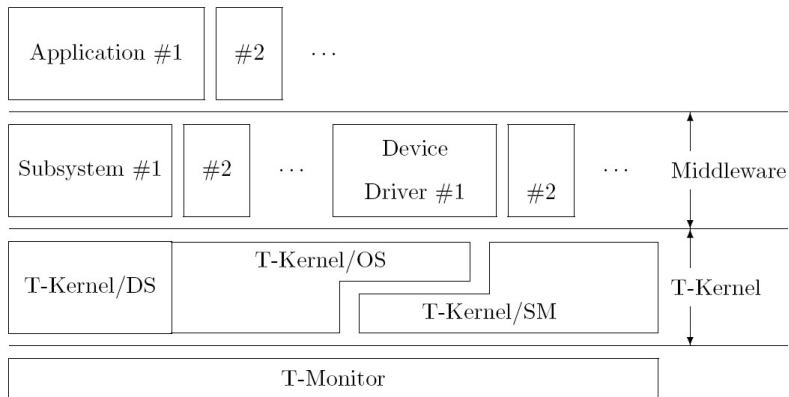
- 成也弱标准化，败也弱标准化。弱标准化导致过多的多样性
- 绝大多数成果都是以日语的形式存在，国际化不足
- μ ITRON已经有10年未更新，已近跟不上技术的发展
- TRON组织已于2010年解散，资产由T-Engine继承
- ITRON的两大后继: TOPPERS Project与T-Engine Forum

- 成也弱标准化，败也弱标准化。弱标准化导致过多的多样性
- 绝大多数成果都是以日语的形式存在，国际化不足
- μ ITRON已经有10年未更新，已近跟不上技术的发展
- TRON组织已于2010年解散，资产由T-Engine继承
- ITRON的两大后继: TOPPERS Project与T-Engine Forum

- 成也弱标准化，败也弱标准化。弱标准化导致过多的多样性
- 绝大多数成果都是以日语的形式存在，国际化不足
- μ ITRON已经有10年未更新，已近跟不上技术的发展
- TRON组织已于2010年解散，资产由T-Engine继承
- ITRON的两大后继: TOPPERS Project与T-Engine Forum

- “为了实现更为理想的实时操作系统的嵌入式计算结构，TRON项目计划开始了一次新的革命——启动了T-Engine项目计划“(口号)
- 由标准化硬件结构 (T-Engine) 与标准开源实时操作系统核心 (T-Kernel) 组成的嵌入式系统的开放式标准平台
- T-Kernel源自 μ ITRON3.0, 采用了BTRON、CTRON上的部分成果, 包含
 - 标准T-Kernel(针对32bit带MMU的系统)
 - μ T-kernel(针对不带MMU的系统, 如16 bit MCU, ARM7TDMI, ARM Cortex-M3)
 - p-kernel, n-kernel (针对更小规模的系统,8bit系统)
 - 多核扩展,AMP与SMP

T-Engine的结构



来源: T-engine手册

T-Kernel与 μ ITRON4.0不同之处

功能	μ ITRON4.0	T-Kernel	功能	μ ITRON4.0	T-Kernel
Task管理功能	○	○	时间管理功能		
Task同期功能	○	○	系统时刻管理	○	○
Task例外处理	○	○	定周期处理	○	○
同期通信			Alarm处理	○	○
semaphore	○	○	Overrun处理	○	x
event flag	○	○	系统状态管理	○	○
data queue	○	x	中断管理	○	○
mailbox	○	○	服务调用管理	○	x
扩展同期通信			系统构成管理	○	○
Mutex	○	○	Subsystem管理	-	○
message buffer	○	○	System内存管理	-	○
ランデブ	○	○	Address空间管理	-	○
内存池功能			device管理	-	○
固定长内存池	○	○	I/O port管理	-	○
可变长内存池	○	○	省电模式管理	-	○

来源:商国平, ITRON与T-kernel的差异

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

TOPPERS是什么

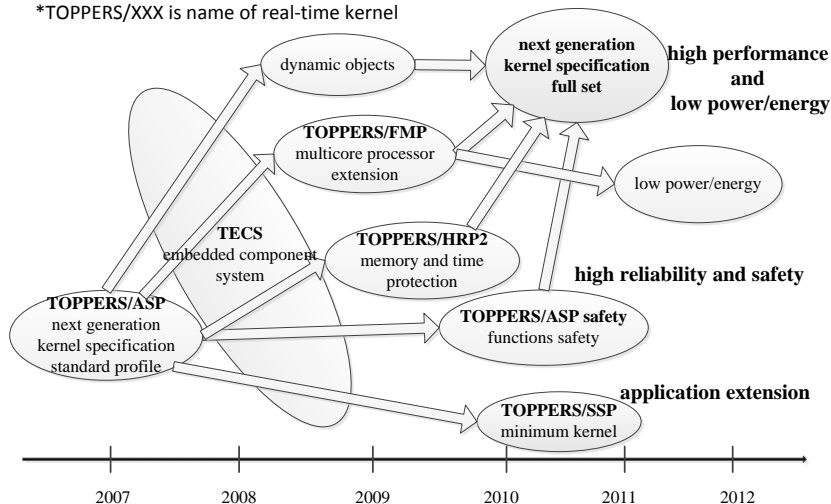
- TOPPERS=Toyohashi Open Platform for Embedded and Real-Time Systems
 - 由高田广章教授在丰桥科技大学创立,2003起NPO化
 - <http://www.toppers.jp>
 - 由产(产业)、学(学校)、官(政府、研究机构)的团体或个人组成
- 宗旨
 - 以ITRON现有的几十年成果为基础, 针对嵌入式系统开发包括RTOS在内多种多样的开源软件并促进其应用
 - 设计和构建嵌入式系统的平台
 - 在嵌入式系统领域, 构建一个如linux那样广泛应用的开源OS!

TOPPERS的主要目标

- 开发现代实时操作系统的决定版
解决或减轻ITRON过多的重复投资和过多的多样性
- 研究下世代的实时操作系统技术
针对嵌入式系统的要求和趋势，继承ITRON的成果，开发下一代RTOS
(不是重复设计一个类似linux的OS)。通过开源化来集结各方力量
- 研究嵌入式软件开发技术和工具
有力地支援高效开发高品质的嵌入式系统
- 推动嵌入式系统开发人才的培养
开发以开源软件为基础的课程或教材，并营造一个良好的氛围

TOPPERS 路线图

*TOPPERS/XXX is name of real-time kernel



主要开发成果（第一代内核）

TOPPERS/JSP 遵循 μ ITRON4.0 Standard Profile的实时内核

TOPPERS/FI4 完全遵循 μ ITRON4.0 Standard的实时内核

TOPPERS/ATK1 面向汽车电子遵循OSEK/VDX标准的实时内核

TOPPERS/FDMP 针对功能分散性多处理器的实时内核(分布式系统)

TOPPERS/HRP 针对带有内存保护功能的(MMU/MPU)的嵌入式系统,并由JAXA实施验证

主要开发成果（新一代内核）

TOPPERS/ASP 在JSP基础上进行了大量改进的实时内核，整个TOPPERS Project的基础

TOPPERS/FMP ASP内核针对多处理器的扩展，接替FDMP内核

TOPPERS/ASP Safety 符合IEC61058-3标准的安全内核

TOPPERS/HRP2 第二代HRP

TOPPERS/SSP 最小内核，面向8位/16位系统，用于无线传感器（?）

主要开发成果（中间件）

TECS 嵌入式组件系统，符合组件化设计的趋势

TINET 遵循ITRON TCP/IP标准的TCP/IP协议栈，同时支持IPv4和IPv6

FatFS for TOPPERS 支持FAT16/32的文件系统，适合小型系统使用

Tethys USB 符合USB2.0的面向嵌入式系统的USB Host Stack（正在开发中，尚未release）

SafeG 嵌入式虚拟机主要利用ARM的TrustZone技术

TLV TraceLogVisualizer，提供OS级的可视化调试支持

TTSP TOPPERS Test Suite Package，用于软件测试

CAN/LIN 针对CAN/LIN总线的中间件

- 在TOPPERS/JSP基础上改良而来 (JSP²内核为一个完全遵循 μ ITRON4.0的RTOS内核)
- 在软件的可重复性、可靠性、可扩展性和安全性上有许多改进
- 整个TOPPERS Project中的基石
- 符合 μ ITRON4.0规范的样板性的实现(ITRON RTOS的决定版)
- 整体内核(monothlic),与 μ C/OS-II一个级别

²JSP: Just Standard Profile, TOPPERS Project的最初成果

设计与实现的方针

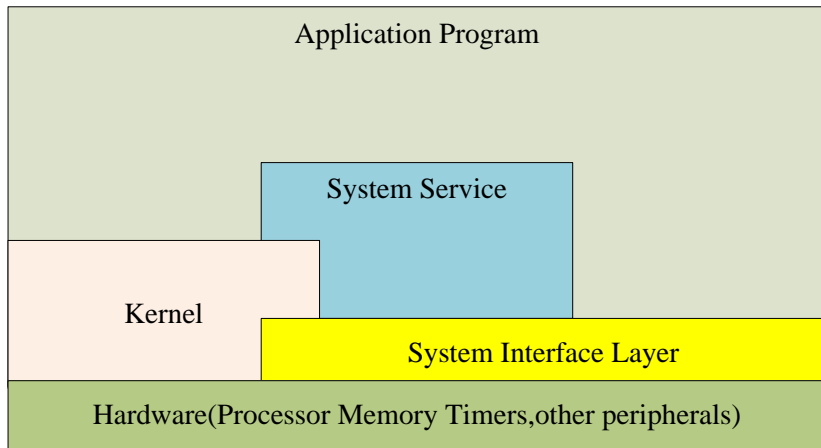
● 设计方针

- ① 重视代码的易读性和可改造性
- ② 易于移植于新的架构平台的构造(μ C/OS的移植太简陋)
- ③ 易于check的构造(参数验证, 配置器的第三阶段)
- ④ 执行性能与内存使用量的综合考虑
- ⑤ 易于扩展

● 实现方针

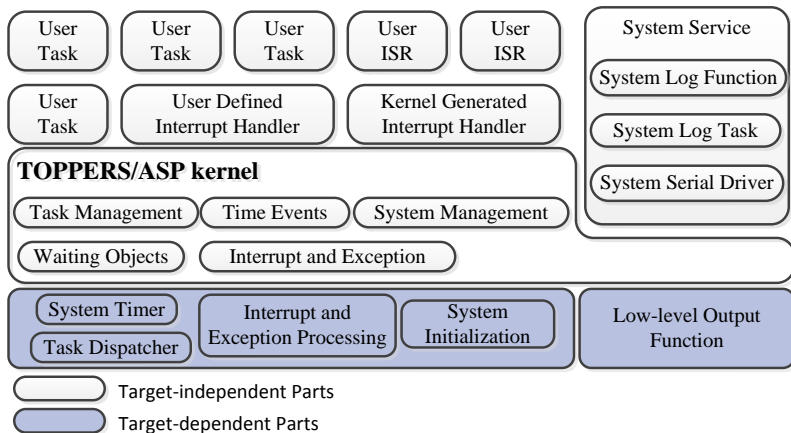
- 统一链接, 整体内核(monolithic), 以减少调用花费
- 内核的library化, 深度条件编译
- 目标依赖部分与目标非依赖部分的明确分离
- RAM使用量抑制

ASP的结构-总体



软件平台 = 内核 + 服务 + 驱动

ASP的结构-内部



ASP的功能

● 任务管理

- 最大任务数 2^{32} ，默认最大优先级16级（可扩展到256级）
- 任务ID、堆栈自动分配（配置器），支持同优先级任务，1.7版本后，开始支持动态创建
- 优先级调度，合作调度（Cooperating Scheduling），位图调度器
- 任务状态数5
- 任务异常/signal

● 同步与通信

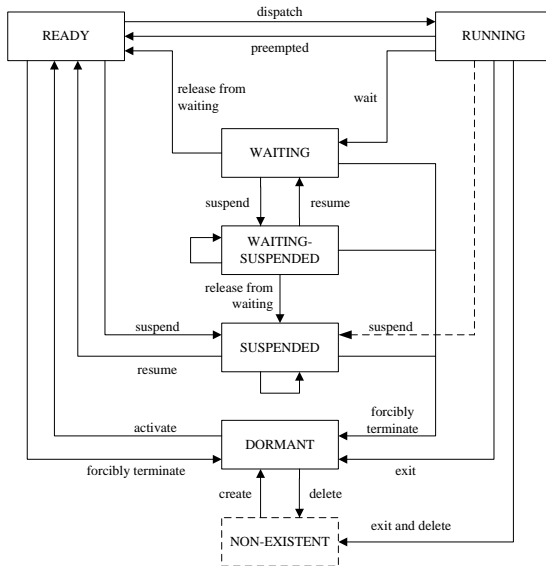
- 信号量（Semaphore）/互斥量（Mutex），支持优先级置顶（Priority Ceiling Policy）
- 邮箱，易于封装
- 优先级数据队列
- 事件标志

- 内存管理
 - 固定长内存池
- 时间事件管理
 - 循环时间事件 (cyclic handler)
 - 单次时间时间 (alarm handler)
 - 超时运行定时器 (overrunhandler)
- 中断与异常
 - TOPPERS标准中断模型 (后述)
 - CPU异常中断处理
- 中间件
 - TCP/IP协议栈-TINET
 - FAT文件系统-FatFS for TOPPERS
 - CAN/LIN..., Tethys-USB

ASP与 μ C/OS-II的比较

	TOPPERS/ASP	μ COS-II(V2.8.6)
Royalty-Free	YES, TOPPERS license, GNU/GPL license	NO, the kernel just can be used for education
MaxTasks	unlimited	256
MaxPriorities	16(default)	256
Tasks on one priority	unlimited	1
Preemptive Scheduling	YES, based on priority	YES, based on priority
Cooperative Scheduling	FCFS for tasks with same priority	NO
Synchronization	semaphore, eventflag, mutex	semaphore, eventflag, mutex
Communication	mailbox, dataqueue, priority dataqueue	mailbox, message queue
Interrupt Management	TOPPERS Standard Interrupt Model	no standard model just simple processing
Memory requirement	10-100KB ROM and RAM	several KB ROM and RAM(minimum)
CPU support	ARM, Japanese CPUs	many
Porting	complicate, require auxiliary tools	easy, just 3 files
Time Events	support alarm, cyclic	support alarm and cyclic from V2.8.6
System Service	Log Service(default)	no, just kernel
Scalability	good, the basestone of TOPPERS Project,	poor
Middlewares	FS, TCP/IP, CAN	FS, TCP/IP, USB, GUI, Modbus...
Documentation	fruitful, but only support Japanese	many textbooks, papers, applications

ASP的任务状态转换



1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

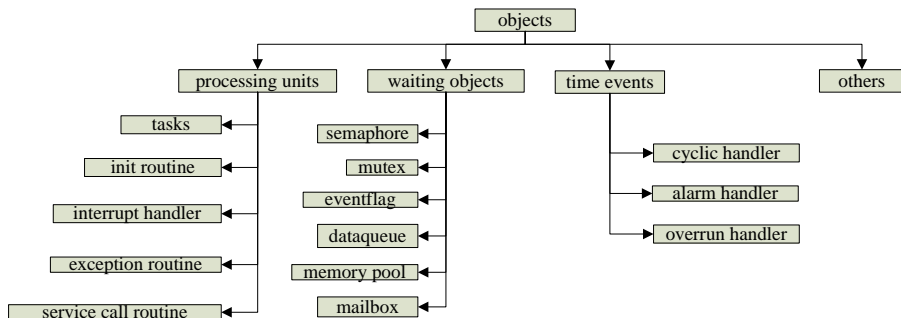
- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

总体介绍

- 一个如果真正掌握的话，十分强大、且人性化的内核
- 基本概念参照 μ ITRON4.0标准（英文）
- 三大特点
 - 面向对象
 - 静态配置
 - TOPPERS标准中断模型
 - 高质量的代码

- 内核和系统服务中可以作为管理对象的软件资源称之为Object。
Object以ID作为区分，有其自身属性
- 与具体程序相关联的object称之为处理单位（ Processing Units）
- 用于任务与任务之间，任务与其他处理单位之间的同步和通信用对象，统称之为等待对象（Waiting Objects）
- 与时间相关的对象称之为时间事件（Time Events）

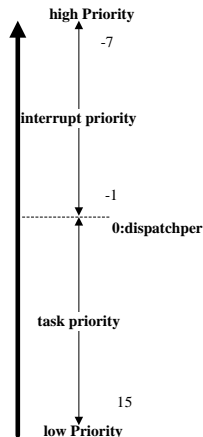
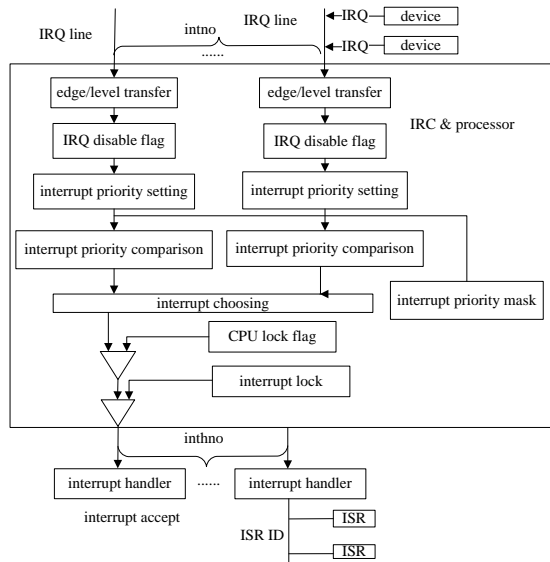
对象组成



静态配置

- 常见的RTOS如 μ C/OS-II, RTEMS, FreeRTOS, 采用C语言宏定义来配置系统。
- TOPPERS/ASP中所有系统资源如任务、信号量等等都通过配置器静态产生，类似与OSEK的OIL。
- 配合library化，可以实现系统自动裁剪，函数级编译链接。
- 优点
 - 减少初始化的工作量，加快启动速度
 - 提高资源的利用效率，不链接没有用到的代码或资源
 - 交叉验证，可有效提高可靠性和安全性
- 缺点
 - 运行时，如需要新的资源时，不灵活，不能复用
 - 单独一套静态配置API，提高了系统的复杂度

TOPPERS标准中断模型



TOPPERS标准中断模型的特点

● 中断处理模型概念图

- 提高中断相关代码的再利用性
- 如前图所示，所有的功能都可以通过硬件实现
- 无法硬件实现的部分可以通过软件模拟实现
- 支持中断嵌套，统一的优先级概念，任务优先级 大于0，调度器优先级为0，中断优先级为负

● 与 μ ITRON4.0不同之处

- 引入了中断优先级屏蔽 (ipm)
- 中断属性标准化，处理流程标准化

● 代价

- 时间上的一定花费，但很小，尤其对于向量中断 控制器 (Vector Interrupt Controller)
- 提高了移植的困难度，尤其是需软件模拟相应功能时

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

文件布局

- 目标依赖(target dependent)

- target 所有与硬件相关的文件,内部文件夹按target_compiler命名,如at91skyeye_gcc
- arch 为了提高代码的复用率,同一架构的共通部分放在该文件夹下,包括编译器共通部分,如arm, gcc

- 非目标依赖(target independent)

- kernel TOPPERS/ASP具体实现,如任务管理,信号量管理等等
- library TOPPERS/ASP常用的库,如printf, histogram, 字符串处理等等
- syssvc 系统服务,目前主要是串口输出服务
- include 常用的头文件,对各部分都共享

- 其他

- pdic 比较常见的驱动程序,如串口驱动 (prime Cell)
- cfg 配置器源码和二进制所在文件夹
- doc 说明文档
- utils 辅助工具,主要是一些perl脚本,如rename, genoffest

从sample开始

sample例程 TOPPERS/ASP的范例，大多数ASP的特点都可以从其中体现。

1个主任务maintask接收信息,3个初始任务task共享同一代码,1个日志任务用于输出

● 必备工具

- GNU开发环境Cygwin/Linux(Unix)
- gcc 3.43工具链和Boost库,支持EUC-JP的编辑器
- CodeSourcery的arm-none-eabi-xxx的工具链

● 基本流程

- ① 从toppers.jp下载asp for at91skyeye包,并解压缩到某个文件夹。
- ② 进入cfg文件下, 编译配置器(configurator),./configure(检查编译环境),make。
cfg.exe/cfg在cfg/cfg目录下
- ③ 在toppers根目录下, 建立obj文件夹作为编译目录。在obj下, 运行../configure -T at91skyeye_gcc生成工程
- ④ obj下有sample.c, sample.h, sample.cfg三个文件。make。
- ⑤ asp.exe为ELF格式的目标文件, kernel_cfg.c/.h为自动生成的配置文件

sample例程的主要功能

- 所用到的资源：1个接受用户命令的main_task (task ID: MAIN_TASK, 优先级: MAIN_PRIORITY), 3个并列执行的任务以及相应的任务异常处理函数 (task ID: TASK1~TASK3, 初始优先级: MID_PRIORITY)。1个执行周期为2秒的周期定时器 (cyclic handler ID: CYCHDR1), 1个单次定时器 (alarm handler ID: ALMHDR10), 1个CPU异常处理函数 (cpu exception handler, ID: CPUEXC1)。
- 功能：并列执行的3个任务，每执行完task_loop回循环后，输出一条信息表示任务正在执行。定时每执行一次就会更改HIGH_PRIORITY, MID_PRIORITY和LOW_PRIORITY这3个优先级就绪队列上就绪任务的顺序。周期handler初始为停止状态。main_task等待接收用户命令，并执行相应的功能，等待期间切换为3个并列执行任务。如输入' Q' 或CTRL+C, 则整个sample program退出

sample例程的命令列表

命令	功能	命令	功能
1	切换对象任务为TASK1（初期设定）	2	切换对象任务为TASK2
3	切换对象任务为TASK3	a	通过act.tsk产生一个启动对象任务请求
A	通过can.act取消对象任务的启动请求	e	在对象任务中通过ext.tsk自行中止
t	通过ter.tsk强制终止对象任务	>	改变对象任务优先级为HIGH_PRIORITY
=	改变对象任务优先级为MID_PRIORITY	<	改变对象任务优先级为LOW_PRIORITY
G	通过get.pri读取对象任务的优先级	s	通过slp.tsk使对象任务进入等待状态，等待被唤醒
S	通过tslp.tsk(10秒)使对象任务进入等待状态，10秒后超时自动唤醒。	w	通过wup.tsk生成一个对象任务唤醒请求
W	通过can.wup取消对象任务唤醒请求	l	通过rel.wai强制解除对象任务等待状态
u	通过sus.tsk强制挂起对象任务，进入SUSPENDED状态	m	通过rsm.tsk解除对象任务的SUSPENDED状态
d	通过dly.tsk(10秒)，延迟10秒执行对象任务	x	通过ras.tex(0x0001)引发对象任务异常处理函数
X	通过ras.tex(0x0002)引发对象任务异常处理函数	y	通过dis.tex禁止对象任务的exception handler
Y	通过ena.tex使能对象任务的任务异常处理函数	r	更改HIGH_PRIORITY, MID_PRIORITY和LOW_PRIORITY这3个优先级就绪队列上就绪任务的顺序
c	使能周期定时器	C	停止周期定时器
b	设定单次定时器5秒后执行	B	停止单次定时器
z	在对象任务中引发CPU Exception（中止任务）	Z	在对象任务中并且在CPU_LOCK状态下引发CPU Exception（中止整个sample program）
q	设定日志信息优先级为LOG_NOTICE（LOG_INFO级别的信息不输出）	v	设定日志信息优先级为LOG_INFO（默认）
V	通过get.utm读取两次系统评价时钟SYSUTM		

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

配置器-configurator

configurator TOPPERS/ASP中所有的配置都必须由配置器来完成。配置器读取并分析配置文件(.cfg)，生成相应的系统构成文件。配置过程最少有3个阶段

● 第一阶段

- 输入: kernel_api.csv, kernel_def.csv, sample.cfg(包括在此之下的.cfg)
- 输出: cfg_out1.c
- 说明: 生成并计算后面阶段需要使用的参数, 包括静态API需要的参数。生成ASCII格式的Motorola S文件cfg_out1.srec。符号列表cfg_out1.syms文件以定位symbol在.srec中的位置。非.syms(.map)或非S格式(.bin)必须经过转换

● 第二阶段

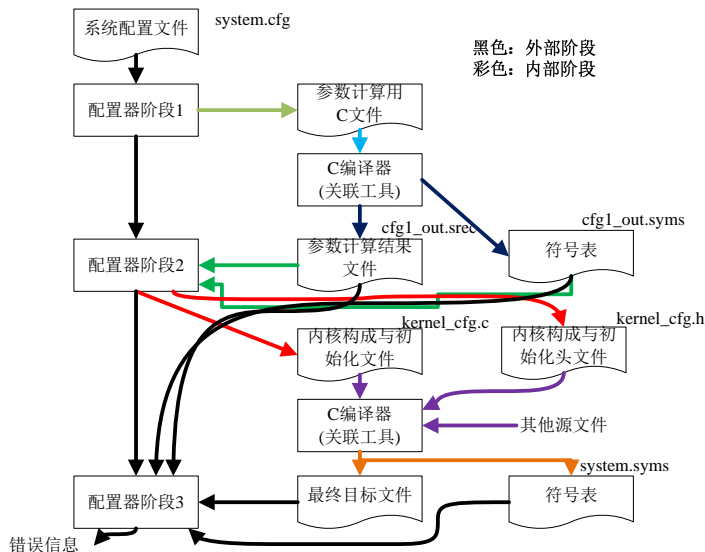
- 输入: cfg_out1.srec, cfg_oout.sym, target.tf, kernel_o.csv, sample1.cfg
- 输出kernel_cfg.c, kernel_cfg.h, 以及可能一些其他的文件(汇编文件)
- 说明: 生成具体的系统配置文件, 包好各object的实现、ID和中断向量表等等。具体依据.tf文件生成。配置其中实现了一个宏解释器的功能

● 第三阶段

- 输入:xxx_check.tf, 符号表, S格式的目标文件
- 输出: 错误信息
- 说明: 根据模板文件, 对最终目标文件进行检查如堆栈溢出, 内存对齐等等

● 第四阶段, 待扩展

配置器的三个阶段



配置器实例

配置脚本

```
CRE_TSK(TASK1, { TA_NULL, 1, task, MID_PRIORITY, STACK_SIZE, NULL });  
DEF_TEX(TASK1, { TA_NULL, tex_routine });  
CRE_CYC(CYCHDR1, { TA_NULL, 0, cyclic_handler, 2000, 0 });  
CRE_ALM(ALMHDR1, { TA_NULL, 0, alarm_handler });
```

配置实现

```
static STK_T _kernel_stack_TASK1[COUNT_STK_T(LOGTASK_STACK_SIZE)];  
const TINIB _kernel_tinib_table[TNUM_TSKID] = {  
  { (TA_NULL), (intptr_t)1, (task), MID_PRIORITY, STACK_SIZE, _kernel_stack_TASK1, (TA_NULL), (tex_routine) },  
};  
...  
const CYCINIB _kernel_cycinib_table[TNUM_CYCID] = {  
  { (TA_NULL), (intptr_t)0, (cyclic_handler), (2000), (0) } };  
const ALMINIB _kernel_alminib_table[TNUM_ALMID] = {  
  { (TA_NULL), (intptr_t)0, (alarm_handler) } };  
};
```

配置脚本-configure

`configure` 用于生成工程的perl脚本，类似一个应用程序

● 参数

- `-T <target>` 目标名，制定BSP
- `-A <appname>` 应用程序名，默认为sample
- `-a <appdir>` 应用所在路径
- `-U <appobjs>` 应用中其他目标文件(.o),可以有多个
- `-L <kernel_lib>` libkernel.a 所在文件夹（如果省略，则重新编译）
- `-f` 是否以函数为单位编译内核
- `-D <srcdir>` 内核源文件所在目录
- `-t <templatedir>` 模板文件夹所在目录（默认为sample）
- `-m <makefile>` makefile的名字默认为Makefile
- `-d <dbgen>` 运行环境
- `-r` 是否使用trace log
- `-p <dir>` perl解释器所在路径
- `-g <dir>` 配置器所在路径

● 例子 `../configure -T dve68k_gcc -D GDB_STUB -A perf1 -a ../test -U histogram.o`

- extension(ASP扩展包, 自1.6后加入)
 - mutex的实现
 - overrunhandler的实现
 - pri_level 中断优先级从16级变成256级
- utilis(脚本工具)
 - genoffset 计算成员在结构体中的偏移量, 供汇编文件使用(offset.h)
 - genrename 生成重命名头文件target_rename.h和target_unrename.h
 - makedep 生成依赖文件,编译时使用
 - makerelease 发布时使用
- test(测试包)
 - 基本的测试程序, 包括常用api等等
 - 用定时器或cycle counter重复测量多遍, 并以直方图的形式显示结果
 - 自动化的测试, 测试生成器

总结-大致开发流程

- 为应用准备配置文件(.cfg)，确定所需要的资源
- 开始编写编写源代码
- 调用配置脚本，根据Makefile模板生成相应的工程Makefile，并修改
- 编译，链接
- 调试

1 介绍

- TRON与ITRON
- TOPPERS 计划

2 TOPPERS/ASP内核

- ASP内核的特点
- 如何使用ASP
- 更加深入一点
- 在skysye上的仿真

嵌入式仿真器

ARMulator ARMulator是一个由ARM公司推出的集成开发环境ADS中提供的指令集模拟器。ARMulator 不仅可以仿真ARM处理器的体系结构和指令集，还可以仿真存储器和处理器外围设备。

Skyeye **推荐!** SkyEye是一个主要针对ARM并支持多种CPU的指令集仿真器为开源软件。SkyEye的目标是在通用的Linux和Windows平台上实现一个纯软件集成开发环境，模拟常见的嵌入式计算机系统，并可对它们进行源码级的分析和测试。

TOPPERS/ASP对SkyEye支持比较好

QEMU **推荐!** QEMU是一套由Fabrice Bellard所编写的模拟处理器的自由软件。Google选择QEMU作为Android的仿真器 **优点:支持比较多的架构和较新的处理!(ARM Cortex-A8)**

skyeye 中 ASP

- 编译Skyeye，与一般的开源项目一致，先configure再make。
- TOPPERS对skyeye1.2.4进行了改造，skyeye with device manager。对at91支持比较好。
- **skyeye -e asp.exe -c skyeye.conf**, skyeye.conf为仿真配置文件

skyeye.conf

```
cpu: arm7tdmi
mach: at91
mem_bank: map=M, type=RW, addr=0x00000000, size=0x08000000
mem_bank: map=I, type=RW, addr=0xf0000000, size=0x10000000
cycle_counter: base=0xfffffc0
#net: type=cs8900a, base=0xfffa0000, size=0x20,int=16, mac=0:4:3:2:1:f, ethmod=tuntap, hostip=10.0.0.1
#net: type=cs8900a, ethmod=tuntap, hostip=10.0.0.1
#uart: mod=stdio
#log: memlogon=1, logfile=log.txt
#dbct: state=on
```