

***Desenvolvimento de um
Sistema Operacional Orientado a Objetos
para uso em Sistemas Embarcados***

Rafael de Góes e Douglas Renaux

Objeto

- Núcleo Operacional para Sistemas Embarcados
- Arquitecturas RISC de 32 bits

Características Desejadas

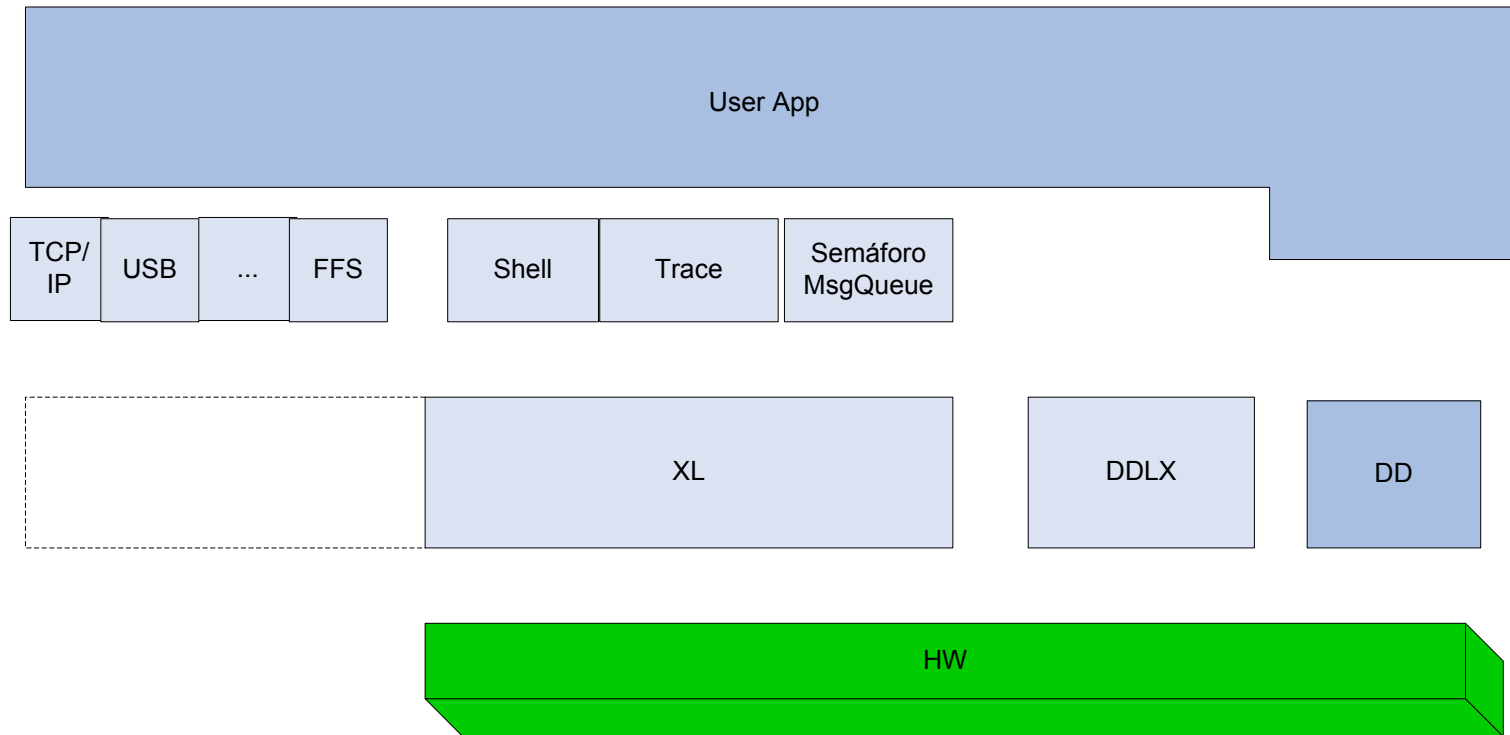


- Orientação a Objetos
 - Desenvolvimento em C++
 - Uso extensivo de construtores/destrutores – configuração
 - Uso de herança – DDLX
- Tempo Real
 - Caracterização temporal
 - Regiões onde as interrupções estão desabilitadas
 - Uso eficiente dos recursos computacionais
- Embarcado
 - Desempenho
 - Flexibilidade
 - Footprint

Histórico do Desenvolvimento

1992	RTSS para RTX-Parlog
1994	PET (x86) uso acadêmico
1995	PET (80186) – uso em projetos em parceria
1998	PET (V55)
2001	PET ARM
2002	PET PowerPC
2003	Desenvolvimento do “X Real-Time Kernel”
2004	Desenvolvimento de produtos
2005	USB, TCP/IP, FFS
2006	Versão 2.0

Caracterização (v1.0)



Gerenciamento de Tarefas



CreateThread	Criação de tarefa rotina principal, nome, tamanho pilha, prioridade
KillThread	Destruição de tarefa
RestartThread	Reinício (equivalente a reset)
GetTId	Identificação de uma tarefa
GetMyTId	Identificação da própria tarefa
GetTaskName	Obtenção do nome
Yield	Libera o processador para tarefas de prioridade \geq
ChangePriority ReturnToOriginalPriority	Alteração do nível de prioridade de uma tarefa
Escalonamento	Seleção da próxima tarefa a executar
Chaveamento de Contexto	Salvamento e recuperação das informações de contexto de cada tarefa

Temporização de Tarefas



GetTime	Leitura do relógio do kernel (resolução ns)
SleepFor SleepUntil	Suspensão por tempo
MsgAt	Programação de mensagem assíncrona temporizada
PeriodicMsg	Programação de mensagem periódica assíncrona
ClearMsg	Desprogramação de mensagem temporizada
SetTimeRef	Estabelece o offset entre o relógio interno do X e o do RTC

Comunicação e Sincronização



Receive	Recebimento de mensagem síncrona ou assíncrona (suspende se não houver mensagem)
Send	Envio de mensagem síncrona
Reply	Envio de resposta a uma mensagem recebida por Send
Put	Envio de mensagem assíncrona (capacidade do buffer de recepção = 16 msg)
CheckForMsg CheckPutQueue	Verificação da presença de mensagens e do estado do buffer de mensagens assíncronas

Message Queues

Put	Envio de mensagem
Get	Recebimento de mensagem
PipeId	Leitura do identificador da MessageQueue
Size	Leitura da capacidade
IsEmpty	Verificação de MessageQueue vazia
IsFull	Verificação de MessageQueue cheia

Interrupções

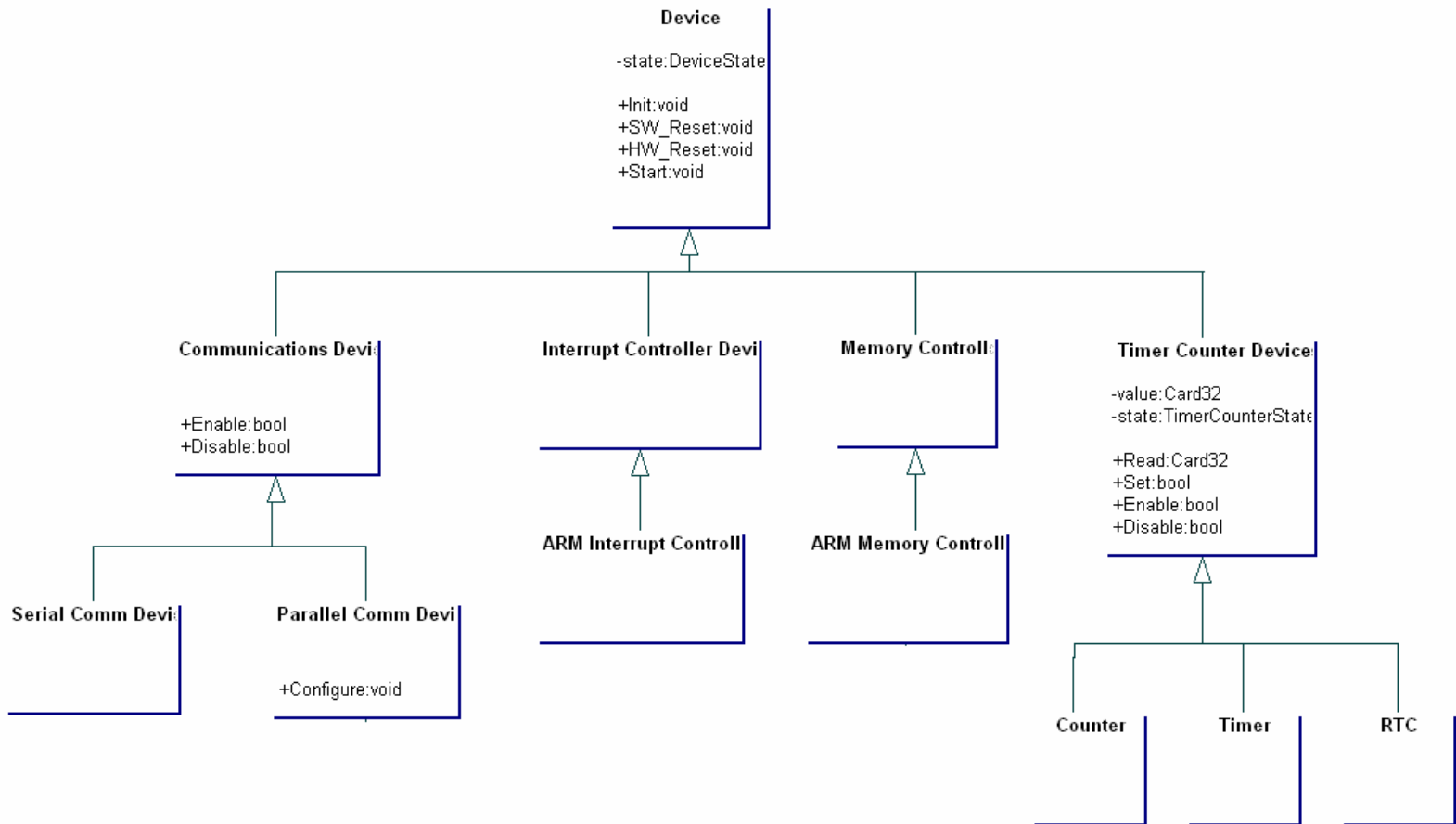
RegisterISR	registro de uma função como rotina de atendimento
UnregisterISR	apaga registro anterior
UnmaskHWInt	habilita determinado canal de interrupção
MaskHWInt	desabilita determinado canal de interrupção
ConfigHWInt	configura a forma de funcionamento de determinado canal de interrupção
InterruptLock	Bloqueia interrupções
InterruptUnlock	Desbloqueia interrupções

Semáforos

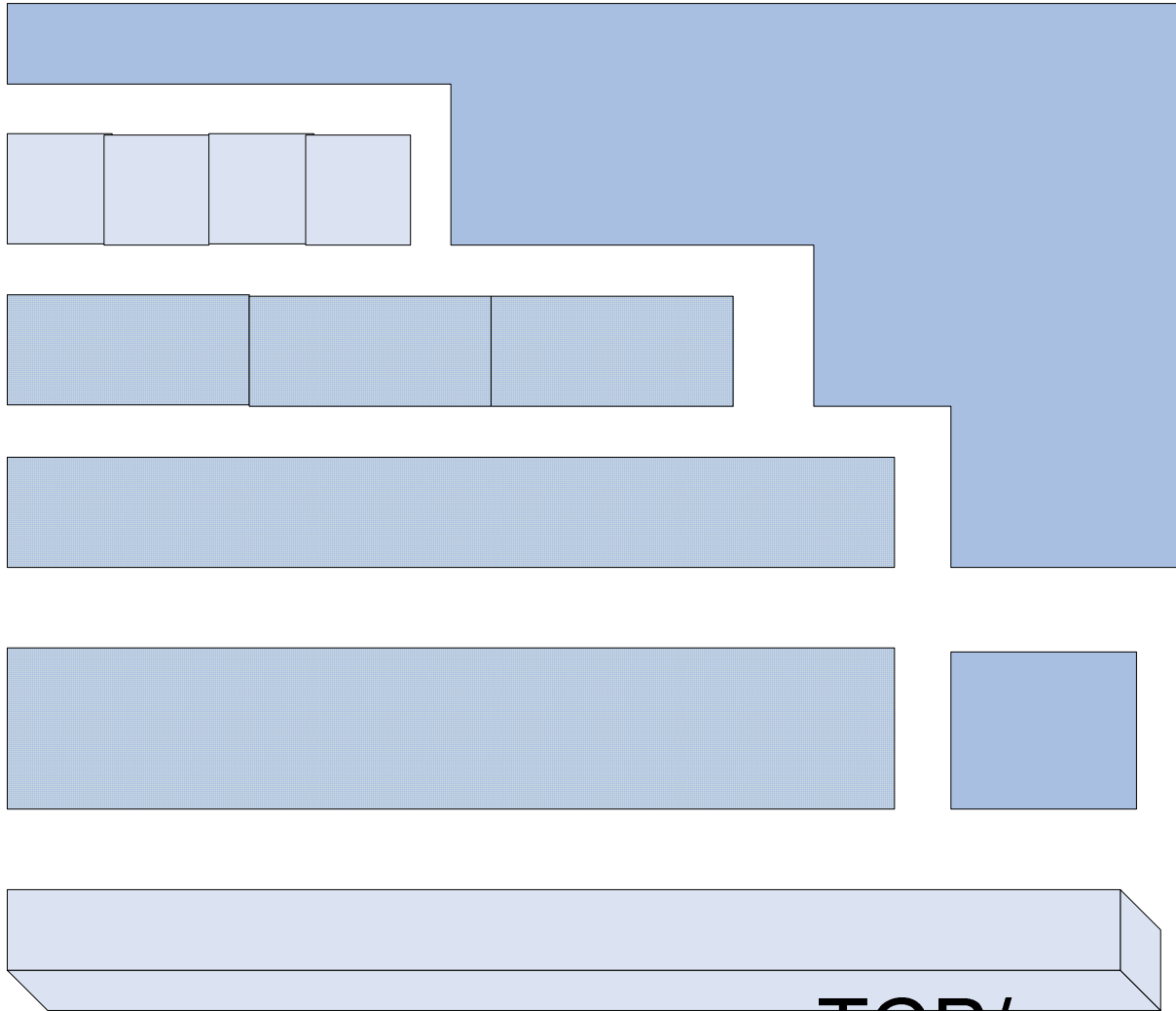
Wait	solicitação de acesso a recurso compartilhado
Signal	liberação do recurso
Check	leitura da disponibilidade de recursos

Módulo	Tamanho do código (bytes)
xl	<i>8492 (incluindo semaphore e message queues)</i>
arm7tdmi	2544
shell	6340
trace	1280

kernel	Tempo de chaveamento de contexto (@clock)	Tempo normalizado para 66 MHz
XL-ARM7TDMI	5.5 us @ 66 MHz	5.5 us
AMX	38 us @ 20 MHz	11.5 us



Versão 2.0



TCP/
IP

USB

Conclusão

- Núcleo Operacional
 - Desenvolvido no Brasil
 - Comercial (no mercado desde 2004)
 - RISC 32-bits (ARM)
 - Bom desempenho comparado com mercado