

S.Ha.R.K OS

(Soft Hard Real-time Kernel)



Tiago Varum 35577

Ruben Gonçalves 35736

Introdução



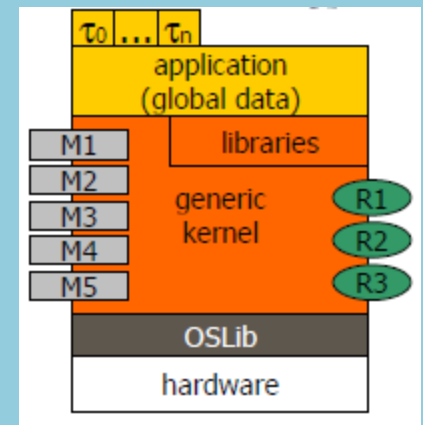
- **S.Ha.R.K** é uma arquitectura de kernel configurada dinamicamente, projectada para suportar aplicações de Tempo Real críticas, não críticas e de não Tempo Real, com algoritmos de escalonamento intercambiáveis.
- É um kernel totalmente modular em termos de políticas de escalonamento e controle de protocolos correntes contrariamente aos SO's tradicionais que geralmente não são modulares.
- A Modularidade é conseguida dividindo as actividades do sistema entre um kernel genérico e um conjunto de módulos que podem ser registados na inicialização, para configurar o kernel de acordo com as exigências específicas da aplicação.

Vantagem



- A principal Vantagem desta arquitectura é que uma aplicação pode ser desenvolvida de forma independente da configuração de um sistema específico, de maneira que novos módulos podem ser adicionados ou substituídos na mesma aplicação, para avaliar os efeitos da política de escalonamento específica.
- O S.Ha.R.K é compatível com quase todas as especificações POSIX 1003.13, com vista simplificar a portabilidade do código da aplicação desenvolvida para outros kernels compatíveis com POSIX.

- Foi desenvolvido por RETISLab, um laboratorio de pesquisa da *Sant'Anna School of Advanced Studies* (Pisa, Italy), e na Universidade de Pavia.
- O S.Ha.R.K é a evolução do Hartik Kernel e é baseado no OSLib Project.
- O S.Ha.R.K contem um conjunto de livrarias , adicionadas as livrarias OSLib's, cada uma delas cobre uma parte especifica da implementação do sistema





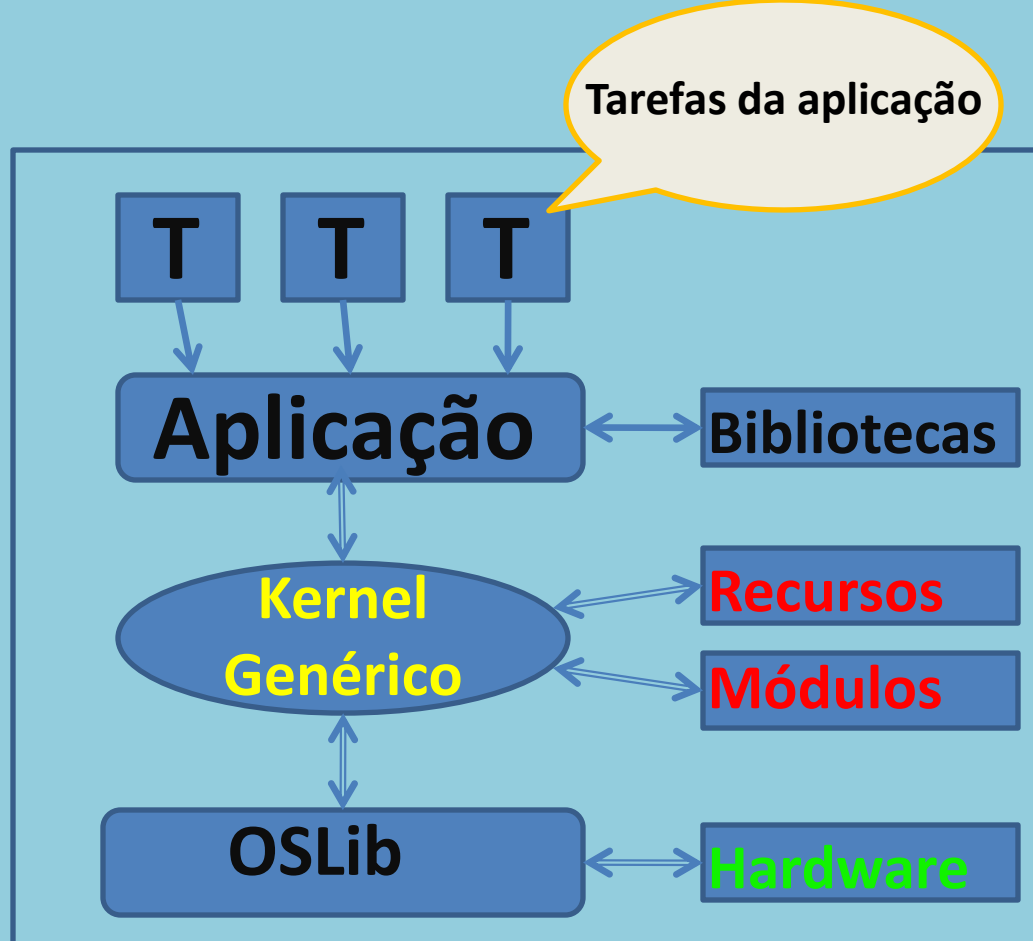
Why S.Ha.R.K?

- Actualmente , a computação em tempo real é necessária em muitos domínios de aplicação, que vão desde o controlo de processos embutidos ate aos sistemas multimédia.
- Cada aplicação tem características peculiares, e á custa disso têm sido propostos vários algoritmos de escalonamento e protocolos de alocação de recursos para obedecer às exigências das aplicações específicas.



Arquitectura

- O objectivo do S.Ha.R.K é construir um kernel de pesquisa concebido para ajudar a implementação e teste dos novos algoritmos de escalonamento.
 - Para isso têm que ser cumpridas certas especificações:
 - Alcançar a independência entre o kernel e as políticas de escalonamento das tarefas e gestão de recursos.
 - Configurar o sistema em tempo de execução, especificando os algoritmos a ser utilizados para o escalonamento das tarefas e acesso a recursos.
 - Alcançar a independência entre as aplicações e os algoritmos de escalonamento.
- Portanto, de maneira geral, **o S.Ha.R.K é um kernel genérico com a finalidade de obter a independência entre a aplicação e os algoritmos de escalonamento.**



Arquitectura do Kernel S.Ha.R.K

- A comunicação entre o kernel e o hardware é fornecida pelo OSLib.

- Cada módulo comunica com o kernel genérico.

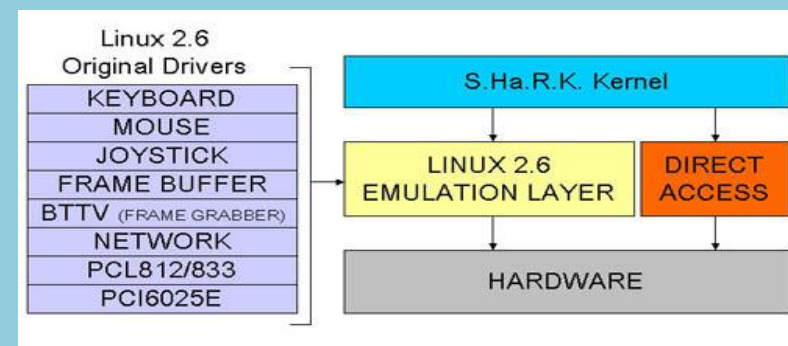
- A aplicação é completamente independente dos módulos de escalonamento.

Modulos



- Módulos de escalonamento são usados pelo kernel genérico para escalonar tarefas, e estão organizados em níveis.
- Os modulos são necessários para fazer a independência entre o kernel genérico e os algoritmos de escalonamento implementados.
- Existem dois tipos de Módulos no S.Ha.R.K
 - Scheduling Modules – que implementam os algoritmos de scheduling e servidores não periódicos. (ex: EDF, RM, DM).
 - Resource Modules – que implementam os protocolos de acesso a recursos compartilhados (ex: semáforos & mutex).

Drivers



- Os drivers representam a parte do hardware que depende do código gerado pelos dispositivos, implementando as rotinas necessárias para o acesso baixo-nível aos dispositivos.
- O código do driver pode ser herdado de outros sistemas operacionais livres, que pode ser compilado no ambiente S.Ha.R.K usando código extra, ficando o sistema com outras *'system calls'* que o kernel genérico.
- Esta solução permite suportar todos o hardwares suportados pelo sistema operacional livre que o driver é herdado (pode herdar código do Linux que suporta a maior parte do PC actual).



- Alguns projectos com S.Ha.R.K :
 - Servo control interface
 - Inverted Pendulum
 - CHIMERA
 - Etc...

Mais informações :

<http://shark.sssup.it/>

http://en.wikipedia.org/wiki/Soft_Hard_Real-Time_Kernel

Perguntas?

