

RealTime Application Interface para o kernel do Linux

João Pedro Oliveira Silva, *Discente, UFRR*, William Juan da Silva Melo, *Discente, UFRR*

Resumo—O resumo vai aqui

Index Terms—Aqui, vão, as, palavras, chave, do, resumo.

I. INTRODUÇÃO

LINUS Torvalds é o criador do Linux, em 1991, núcleo (kernel) do **sistema operacional GNU/Linux**. É um núcleo monolítico de código aberto para sistemas operacionais tipo UNIX. A API do Linux, na qual programas interagem com o kernel, é projetada para ser extremamente estável e jamais quebrar o espaço do usuário.

O kernel é um software responsável por controlar as interações entre hardware e outros programas da máquina, traduzindo as informações que recebe ao processador e aos demais elementos eletrônicos do computador.

Um sistema em tempo real pode ser definido como um "sistema capaz de garantir os requisitos de tempo dos processos sob seu controle". Deve ser rápido e previsível, ou seja, baixa latência e capaz de determinar o tempo de conclusão da tarefa com certeza.

O RTAI significa Interface de Aplicação em Tempo Real. Estritamente falando, não é um sistema operacional em tempo real, como o VXworks ou o QNX. Ele é baseado no kernel do Linux, oferecendo a capacidade de torná-lo totalmente pré-descartável.

Um módulo do kernel é simplesmente um arquivo objeto contendo rotinas e / ou dados para carregar em um kernel em execução. O RTAI é muito orientado a módulos. Então, para entender e ser capaz de usar o RTAI é necessário conhecer os módulos carregáveis dinamicamente para o Linux.

UFRR

Junho 27, 2019

A. Kernel do Linux

O Kernel é o núcleo do sistema operativo. Cabe ao Kernel fazer a ligação entre o hardware e os programas executados pelo computador. A junção do Kernel mais o software é que tornam o computador usável (drivers, protocolos de comunicação, entre outros), de acordo com a sua aplicação, formam o sistema operativo em si.

1) *Desenvolvimento do Kernel do Linux*: Sempre são lançadas novas versões do kernel do sistema Linux com intuito de trazer melhorias à atualidade em funções específicas. O objetivo é também corrigir falhas e adicionar mais recursos ao kernel, além de compatibilidade com novos hardwares. Cada nova versão vem representada por três números diferentes separados por ponto final. Os dois primeiros números indicam qual série pertencem, já o terceiro aponta qual a versão do kernel dentro daquela série. Quando o segundo número for ímpar, demonstra que a série ainda está em desenvolvimento, logo, representa uma versão instável e em fase de testes e melhorias. Por outro lado, quando o número é par, significa que a série tem estabilidade para funcionar. Por fim, o terceiro número se altera quando algum recurso é modificado ou adicionado.[1]

2) *Funções do Kernel*:

- Abstrair o funcionamento do hardware, fornecendo um meio de desenvolver aplicações que funcionem em diversas configurações de máquinas e melhorando a eficiência de algumas atividades junto ao hardware.
- Garantir a segurança da máquina e dos processos em execução, ao controlar quem e como os recursos do sistema podem ser acessados.
- Realizar uma divisão adequada dos recursos entre os processos, garantindo o isolamento de cada processo.
- Prover ao usuário o acesso aos serviços disponibilizados, através de chamadas de sistema.

B. Divisão do kernel Linux

O sistema operacional é composto de dois níveis, de acordo com a imagem abaixo:

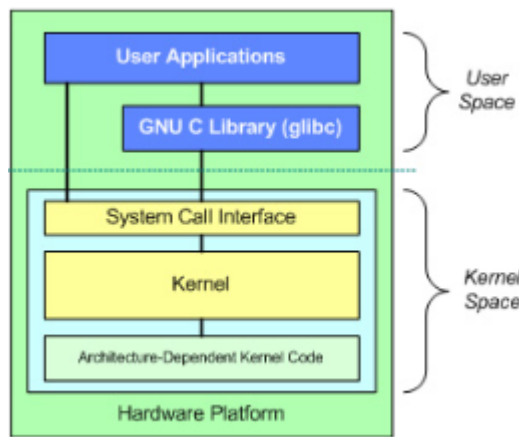


Figura 1. Arquitetura básica do sistema operacional GNU/Linux

O kernel em si, não é um processo, mas um gerenciador de processos, executado nas seguintes situações: em resposta a uma chamada de sistema, exceção ou interrupção ou, mais raramente, em alguma thread do kernel, que seja responsável por executar algumas funções periodicamente.

C. Módulos do Kernel

Um módulo do kernel é simplesmente um arquivo objeto contendo rotinas e / ou dados para carregar em um kernel em execução. Quando carregado, o código do módulo reside no espaço de endereço do kernel e é executado inteiramente dentro do contexto do kernel. Tecnicamente, um módulo pode ser qualquer conjunto de rotinas, com a única restrição de que duas funções, `init-module()` e `cleanup-module()`, devem ser fornecidas. O primeiro é executado quando o módulo é carregado e o segundo, antes de o módulo ser descarregado do kernel.

1) Exemplos:

- adaptador de rede SMC EtherPower 10 PCI
- RTAI

D. Real Time Application Interface

O RTAI é muito orientado a módulos. Então, para entender e ser capaz de usar o RTAI é necessário conhecer os módulos carregáveis dinamicamente para o Linux.

O RTAI oferece os mesmos serviços do kernel do Linux, adicionando os recursos de um sistema operacional industrial em tempo real. Consiste basicamente em um despachante de interrupção: o RTAI captura principalmente as interrupções dos periféricos e, se necessário, redireciona-as para o Linux. Não é uma modificação intrusiva do kernel; Ele usa o conceito de HAL (camada de abstração de hardware) para obter informações do Linux e interceptar

algumas funções fundamentais. Este HAL fornece poucas dependências ao Kernel do Linux. Isso leva a uma adaptação simples no kernel do Linux, uma porta RTAI fácil de versão para versão do Linux e um uso mais fácil de outros sistemas operacionais em vez do RTAI. O RTAI considera o Linux como uma tarefa em segundo plano em execução quando não ocorre atividade em tempo real.

II. CONCLUSÃO

The conclusion goes here.

AGRADECIMENTOS

Os autores querem agradecer a quem?

REFERÊNCIAS

- [1] OLIVEIRA, P. **Kernel do Linux: O que é e para que serve**, 2017. Disponível em: <<https://www.escolalinux.com.br/blog/kernel-do-linux-o-que-e-para-que-serve>>. Acesso em: 23 jun. 2019.
- [2] Red Hat, Inc., **Red Hat Enterprise Linux 3: Guia de Administração do Sistema** North Carolina, USA: LexisNexis, 2003.
- [3] GODOY, Alan S., **Arquitetura do Kernel Linux**, 2009. Disponível em: <<http://www.ic.unicamp.br/~islene/1s2009-mc514/Kernel-Linux.pdf>>. Acesso em: 27 jun. 2019.
- [4] SIMIONI, D. **Como compilar um Kernel Linux passo a passo [TUTORIAL COMPLETO]**, 2017. Disponível em: <<https://www.diolinux.com.br/2017/07/como-compilar-um-kernel-linux-passo-a-passo.html>>. Acesso em: 23 jun. 2019.
- [5] RTAI. **RTAI: a Beginner's Guide**, 2006. Disponível em: <<https://www.rtaio.org/?Documentation>>. Acesso em: 23 jun. 2019.
- [6] The Linux Kernel. **Kernel module signing facility**, 2019. Disponível em: <<https://www.kernel.org/doc/html/latest/admin-guide/module-signing.html>>. Acesso em: 23 jun. 2019.

João Pedro Biography text here.

William Juan Biography text here.