



RTOS - Real Time Operating Systems

Samuel Pereira - 882
Vitor Rodrigues Di Toro - 983

Daniel Sader P. Neves - 713
Luiz Fernando da Silva - 746

5/6/2017

Seminário de [EC 009] - Sistemas Operacionais Instituto
Nacional de Telecomunicações

1

RTOS - Definição

- **Sistema Operacional de Tempo Real**, ou **RTOS** (*Real Time Operating System*);
- **Definição:**
 - RTOS é um sistema onde cada operação tem um tempo limite, *deadline*, para acontecer;
- **Classificação:**
 - *Hard Real Time*;
 - *Soft Real Time*;



Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

2

→ Tempo real não tem nada haver com o tempo de execução de uma tarefa, executada imediatamente; (como costuma-se pensar)
→ Mas sim com o tempo de resposta pré-definido à um evento; Sistema Determinístico;

RTOS - Classificações

- **Hard Real Time:**

- Também conhecido como Tempo Real Rígido;
- Caso uma *deadline* não seja respeitada, gera uma falha catastrófica;

- **Soft Real Time:**

- Também conhecido como Tempo Real Flexível;
- Caso uma *deadline* não seja respeitada, gera uma falha não catastrófica;
- Uma falha costuma afetar:
 - Qualidade do Serviço;
 - Experiência do Usuário;

HARD

- Sistemas de segurança, como o Airbag de carros e Ejeção de Jatos;
- Sistemas de controle de aeronave;
- Sistema de controle do fluxo de combustível de foguetes e jatos;

SOFT

- Tocador de Música de um Carro;
- Driver de CD do PC;
- Alguns sistemas de controle Fuzzy;

RTOS - Exemplos



- Começou como um sistema acadêmico;
- Um dos RTOSs mais largamente utilizados;



- Utilizado por diversos fabricantes de carros;
- Exemonia inabalada até mesmo pela Google e Apple;



- Real-time scheduler;
- Prevents deadlocks;

μLipeRTOS

- 100% brasileiro;
- OpenSource;



HeartOS

- A POSIX-based Hard Real-time Operating System
- Amplamente utilizado na indústria aeronáutica;



- Utilizado pela NASA, Boeing, exército americano;
- Utilizado na Sonda Curiosity;

FreeRTOS

- Começou como um sistema acadêmico e totalmente gratuito;
- Hoje em dia possui licença para usos comerciais;
- Largamente utilizado pela indústria de produtos e bens de consumo;
- Utilizado desde equipamentos de segurança e monitoramento, à robos, passando por diversos eletrônicos e eletroeletrônicos (Geladeira e SmartCoisas);

QNX

- Comprado pela BlackBerry em 2010;
- Utilizado por marcas como: Ford, BMW, Mercedes-Benz, Honda, Audi, Ferrari, Volvo, Jeep, Jaguar, Lexus, Dodge, entre outras.

uC/OS

- Scheduler de Tempo Real;
- Utiliza Round-robin;
- Prevents deadlocks (Proceço);
- Prevents memory fragmentation;

uLipeRTOS

- 100% Nacional;
- OpenSource;
- Disponível no GitHub;

HeartOS

- Amplamente utilizado na industria aeronautica;
- Como em sistemas de controle de fluxo de combustível;
- Programação em C, C++ e ADA.

VxWorks

- Utilizado pela NASA, Exército Americano e Boeing;
- Utilizado na Sonda Curiosity;

RTOS - SIRTOS

- **Criado por:** Vasileios Kouliaridis;
Vasileios Vlachos;
Ilias Savvas;
Iosif Androulidakis.
- **Universidades:** "TEI of Thessaly" e "University of Ioannina";
- **Propósito:** Educacional, visando auxiliar no estudo de RTOS;
- **Arquitetura:** x86, multitarefa, com interface gráfica. Visando a facilidade de uso dos alunos;
- **OpenSource:** "Disponível" no GitHub, porém não é compilável por faltar parte dos códigos;
- **Tecnologias:** C, Assembly, Makefile e C++;

Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

5

TEI → Technological Educational Institute

Sobre os Autores

- O 3 primeiros são da 1ª universidade
- O último é da 2ª universidade

OpenSource:

Apenas de disponível no GitHub, parte do código encontra-se ausente.

Como a MAIN.C;

Segundo os autores, seria compilável em Windows, Linux e MAC;

C - 90.8%

Assembly - 4.7%

Makefile - 3.1%

C++ - 1.4%-

(Fonte: GitHub) COM ADAPTAÇÕES!!!!!!

SIRTOS - Interrupção

- **Premissas:**

- Manter a maior simplicidade possível;
- Atender as requisições o mais rápido;
- Evitar problemas de concorrência;

- **Concepção:**

- Interrupções com maior prioridade que tarefas do sistema;
- Não existe encadeamento de interrupções;
- Não existe interrupção enquanto uma tarefa utiliza alguma estrutura do Kernel;

Crucial em qualquer Sistema Operacional

Indica que algum **SW** ou **HW** precisa de **atenção Imediata**;

SIRTOS - Gerenciamento de Memória

- **Premissas:**

- Aproveitar o poder computacional do sistema;
- Permitir a implementação de um sistema multitarefa com interface gráfica;

- **Concepção:**

- Gerenciamento dinâmico de memória;
- Suporte a paginação e segmentação;
- Permite a implementação de Swapping e Memória Virtual;

OBS: A implementação desta forma pode causar fragmentação, mas o problema foi deixado de lado por questões educacionais;

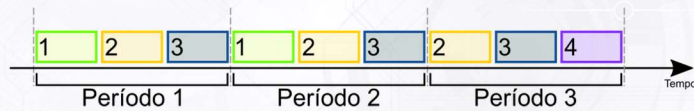
SIRTOS - Multitasking

○ Premissas:

- Criar a ilusão de paralelismo;
- Permitir a implementação de um sistema com GUI;
- Garantir que cada tarefa "respeite" sua *deadline*;

○ Concepção:

- Uso de "*Time Periods*", com duração definida em TICKS;
- Prioridade de execução definida com base nas *deadlines*;
- Nenhuma tarefa é executada duas vezes no mesmo *Time Period*; a menos que ela seja a única tarefa restante;



Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

8

- Parte mais difícil de ser implementada no SIRTOS;
- Por isso há muito espaço para melhoras.

Exemplo de tarefas com deadlines curtas:

- GUI (manter a sensação de estabilidade);

Time Periods

- Garante também que tarefas com prioridade menor sejam executadas

SIRTOS - Multitasking

○ Problemas:

- O tempo das *deadlines* são múltiplos de *Time Periods*;
- A ocorrência de interrupções pode fazer com que uma tarefa não finalize dentro de sua *deadline*;

○ Exemplo:



- No exemplo, toda tarefa tem uma *deadline* de 3 *Time Periods*;
- A tarefa 3 não conseguirá ser executada em tempo certo devido a interrupção;

Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

9

* ISR = Interrupt Service Routine;

* Scheduling de tarefas é extremamente complexo; isto causa problemas como os acima;

* Demonstra a importância de um sistema de scheduling eficiente;

--> Pois varrer listas pode ser muito lento; importante deixar as tarefas em ordem)

SIRTOS - Modo Real e Modo Protegido

- **Modo Real:**

- Modo de Compatibilidade com programas 16-bits;
- Acesso a somente 1MB de memória;
- Acesso direto a funções da BIOS;
- Sem suporte a paginação ou gerenciamento de memória;

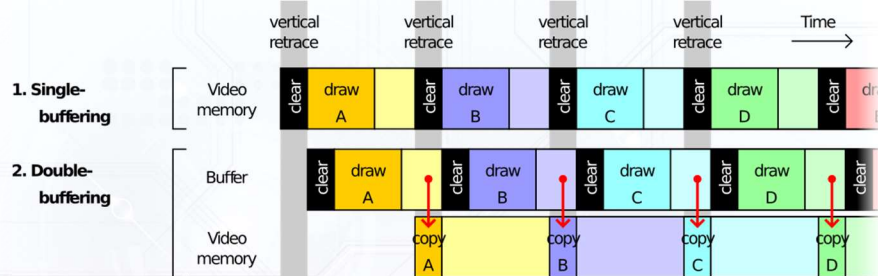
- **Modo Protegido:**

- Acesso somente a funções do SO (*System Calls*);
- Acesso completo a memória;

SIRTOS - Grafico (Implementação)

- **VBE (*VESA Bios Extensions*):**
 - Compatível com qualquer BIOS ou placa gráfica, mas com recursos limitados;
 - Exige um context switch para o modo real para ser executado;
- **Drivers específicos:**
 - Permite acessar todos os recursos da placa gráfica, mas necessita de um driver específico para cada placa
 - Necessita da implementação de um Double Buffer, para evitar cintilação da tela;

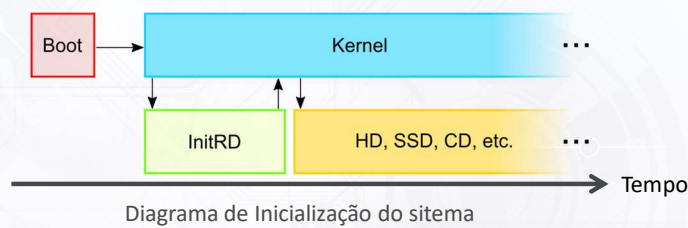
SIRTOS - Grafico (Implementação)



Demonstração da técnica de double buffering

SIRTOS - Sistemas de Arquivos

- **Bootloader:**
 - **GRUB** (Grand Unified Bootloader) - Bootloader padrão na maioria dos sistemas operacionais;
- **InitRD (InitRD - Initial Ram Disk):**
 - Um pequeno Sistema de Arquivos é carregado na RAM, com os drivers necessários para o sistema;



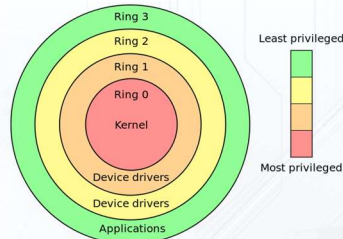
Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

13

// Linux usa um sistema parecido (initramfs.gz)

SIRTOS - Segurança

- Os recursos são divididos em "anéis" de proteção:



- Usuários diferentes tem acessos a anéis diferentes;
- Senhas são criptografadas utilizando um hash MD5;
- Ao fazer login, a senha digitada é criptografada e comparada com a senha em disco;



Seminário de [EC 009] - Sistemas Operacionais
Instituto Nacional de Telecomunicações

15

Exemplo Prático





Obrigado!

ну почему ты такой тупой Андрюха?

Seminário de [EC 009] - Sistemas Operacionais Instituto
Nacional de Telecomunicações

17