



# RTOS - Real Time Operating Systems

Samuel Pereira - 882  
Vitor Rodrigues Di Toro - 983

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

# 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*;





# 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;

# RTOS - Exemplos



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



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



- Real-time scheduler;
- Prevents deadlocks;

## μLipeRTOS

- Sistema 100% brasileiro;
- OpenSource;



μLipeRTOS



- A POSIX-based Hard Real-time Operating System
- Amplamente utilizado na industria aeronautica;



- Utilizado pela NASA, Boeing, exército americano;
- 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++;

# 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;

# 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;



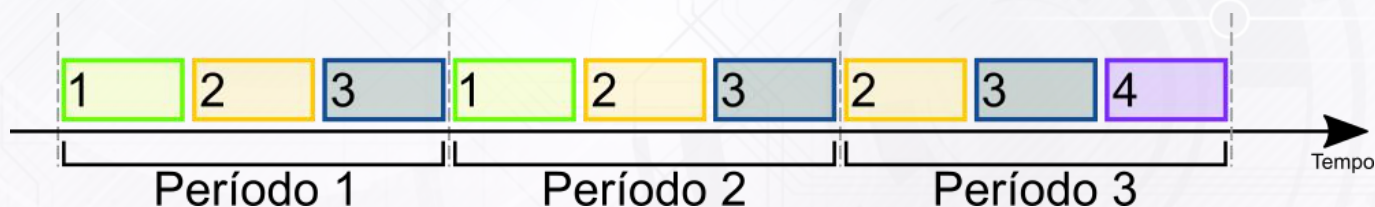
# SIRTOS - Multitasking

## 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;





# 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;

# 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)

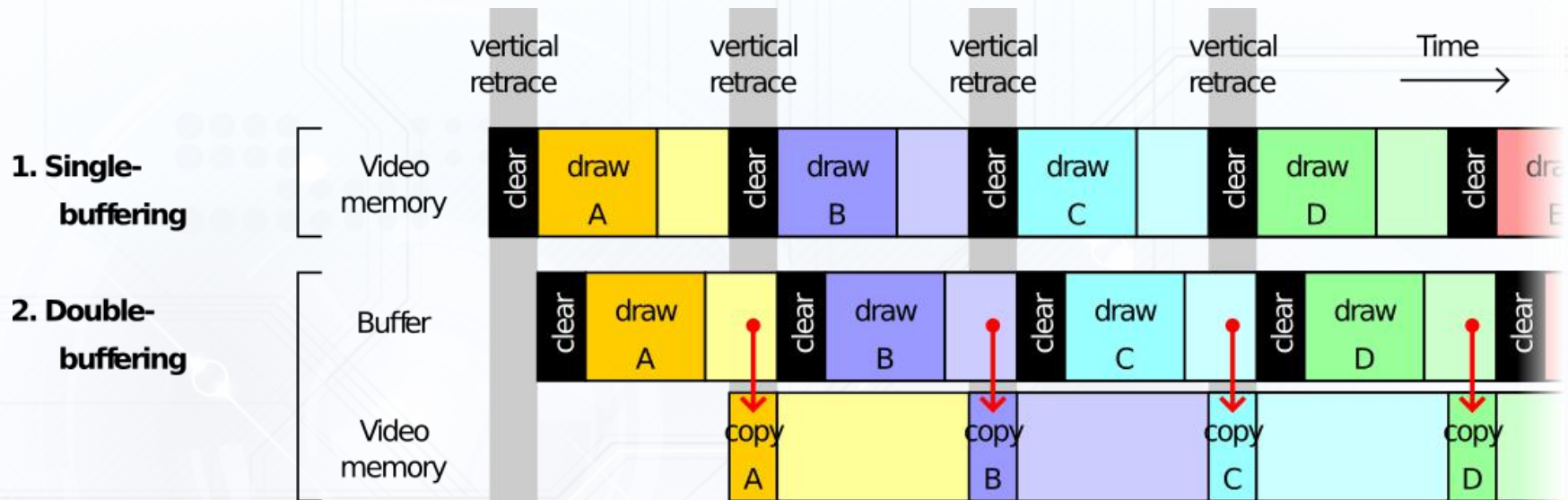
## **VBE (*VESA Bios Extensions*):**

- 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;

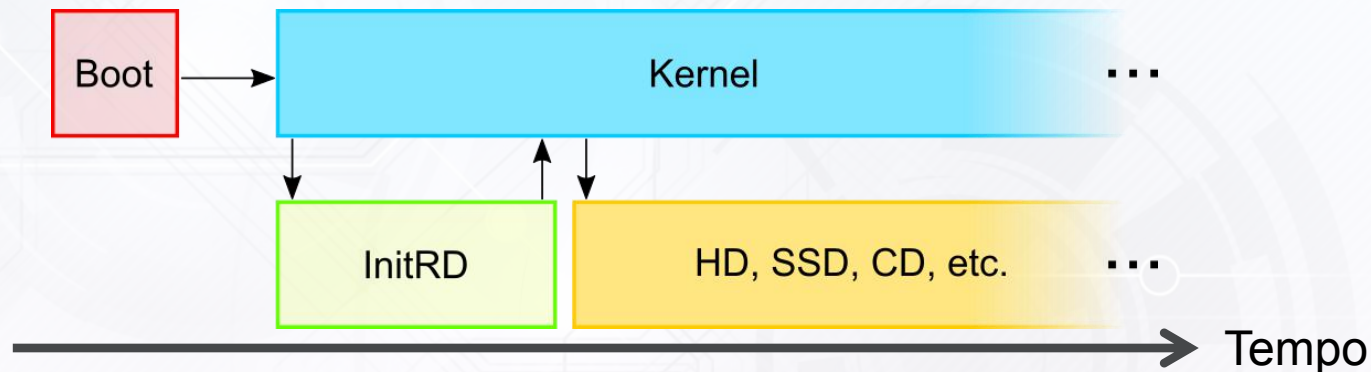
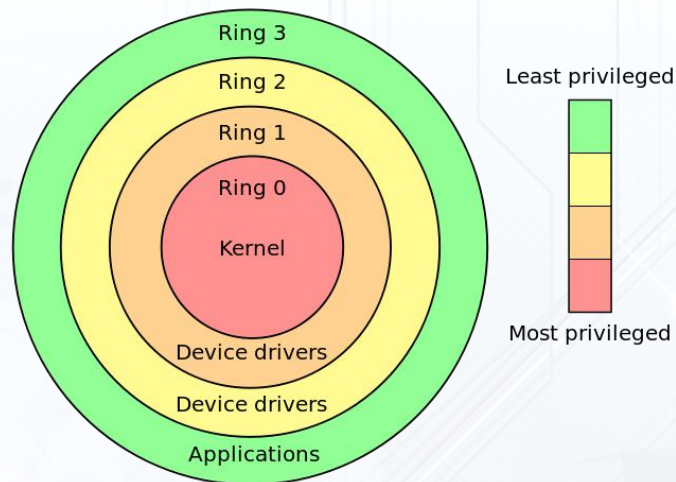


Diagrama de Inicialização do sistema

# 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;



**MD5?**



**TOTALLY INSECURE**

# Obrigado!

ну почему ты такой тупой Андрюха?