



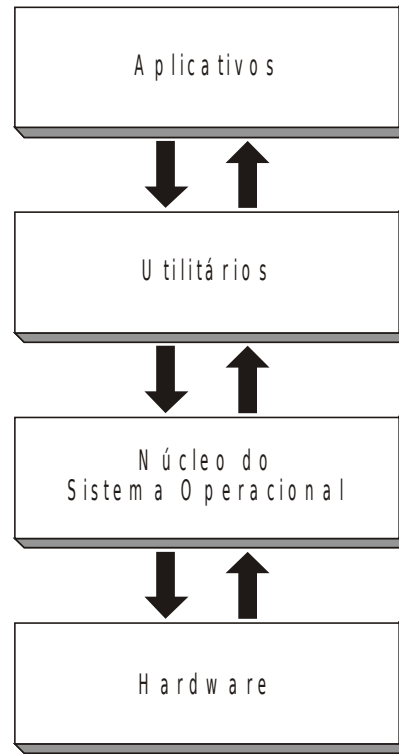
# Fundamentos de Sistemas Operacionais

## **Estrutura, Processos e Threads**

**Prof. Edwar Saliba Júnior**  
**Março de 2007**



## Sistema computacional



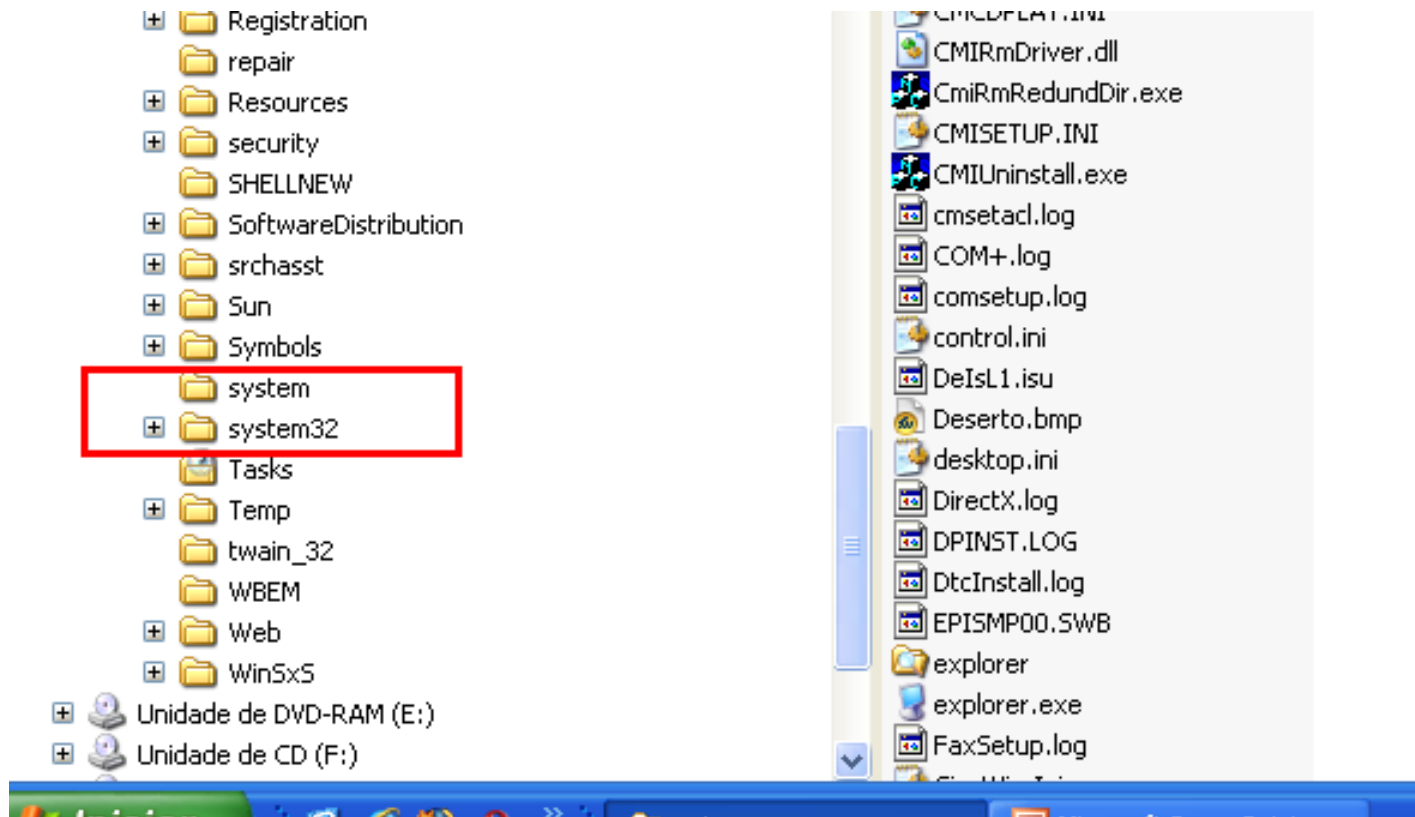


## Introdução

- **Kernel: conjunto de rotinas que oferece serviços aos usuários, suas aplicações e ao próprio sistema operacional;**
- **Principais tarefas do Kernel:**
  - Gerência de processos em execução (criação, sincronização, escalonamento)
  - Gerência de memória
  - Gerência do sistema de arquivos
  - Gerência de dispositivos de I/O
  - Segurança e Log do Sistema (no windows, o menu “Gerenciamento do computador” permite a visualização desses logs).



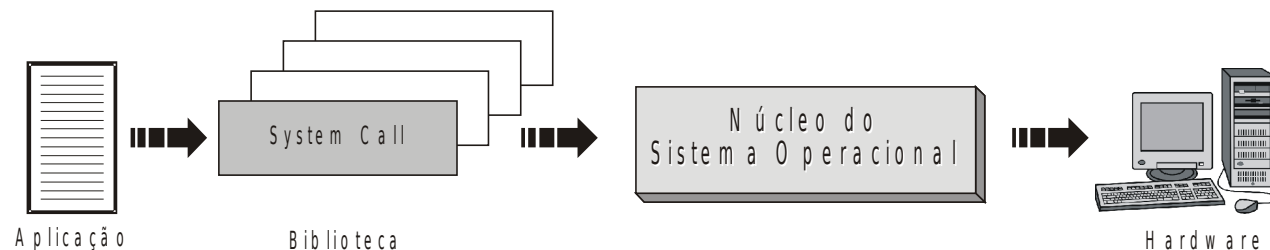
## Principais Bibliotecas do Kernel do Windows





## System Calls

- Interface do sistema operacional para que usuários e programas possam utilizar suas rotinas;
- Função: garantir a integridade do sistema;
- No windows, são as APIs;
- Exemplos:
  - API windows: chamada da calculadora do Windows em um sistema Delphi, .Net, Java e etc.;
  - Comandos de leitura de arquivos;
  - Garbage collector em Java -> gera um comando que, quando executado na máquina virtual, irá utilizar uma System Call do sistema em que estiver sendo executado. Isso permite que sites de bancos possam ser executados em diferentes Sistemas Operacionais.





## Uma DLL em Delphi

```
• library Util;  
•  
• uses  
•   SysUtils,  
•   Classes;  
•  
• {$R *.RES}  
• function Adicao (A, B : Extended) : Extended;  
•  
• begin  
•   Result := A + B;  
• end;  
•  
• function Subtracao (A, B : Extended) : Extended;  
• begin  
•   Result := A - B;  
• end;  
•  
• function Multiplicacao (A, B : Extended) : Extended;  
• begin  
•   Result := A * B;  
• end;  
•  
• function Divisao (A, B : Extended) : Extended;  
• begin  
•   Result := A / B;  
• end;  
•  
• exports  
•   Adicao index 1,  
•   Subtracao index 2,  
•   Multiplicacao index 3,  
•   Divisao index 4,  
•  
• begin  
• end
```

Para o usuário da DLL só é disponibilizado o cabeçalho das funções.

O usuário não tem acesso ao corpo das funções, ou seja, ele não sabe de que maneira a operação é feita, contudo, ele tem como retorno, o resultado esperado.



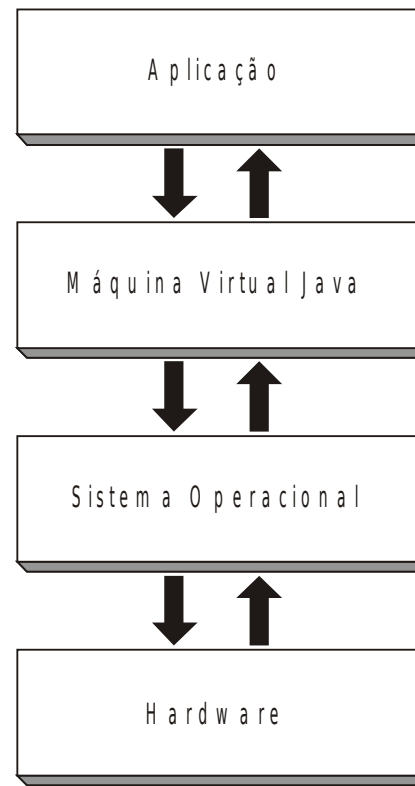
## Visão da Pasta System32...

Endereço D:\WINDOWS\system32

Pastas	Nome	Tamanho	Tip
Resources	c_28605.nls	65 KB	Arq
security	cabinet.dll	59 KB	Dyn
SHELLNEW	cabview.dll	83 KB	Dyn
SoftwareDistribution	cacls.exe	19 KB	Apli
srchasst	calc.exe	113 KB	Apli
Sun	camocx.dll	50 KB	Dyn
Symbols	capesnpr.dll	142 KB	Dyn
system	cards.dll	352 KB	Dyn
system32	catsrv.dll	221 KB	Dyn
3com_dmi	catsrvps.dll	84 KB	Dyn
1025	catsrvut.dll	611 KB	Dyn
1028	CC3260MT.DLL	1.465 KB	Dyn
1031	ccfgnt.dll	27 KB	Dyn
1033	cdfview.dll	148 KB	Dyn
1037	cdm.dll	74 KB	Dyn
1041	cdmodem.dll	16 KB	Dyn
1042	cdosys.dll	2.020 KB	Dyn
1046	cdplayer.exe.manifest	1 KB	Arq
1054	cds60.bpl	32 KB	Borl
2052	Cdxdbe.dll	137 KB	Dyn



## Máquina Virtual Java (JVM)







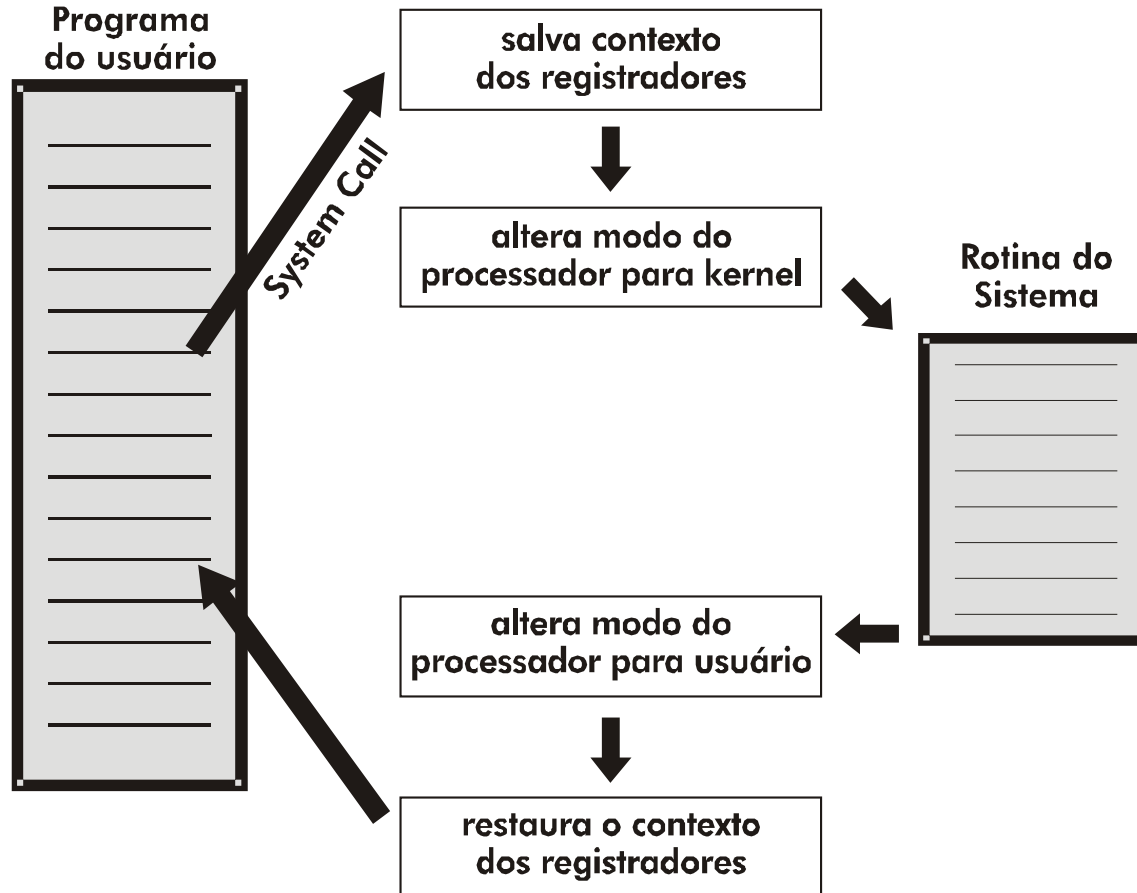
## Modos de Acesso (do processador)

- **Modo Usuário:** instruções não privilegiadas;
- **Modo Kernel:** instruções privilegiadas (acesso a todas as rotinas do sistema);
  - Apenas o S.O. tem acesso a elas;
- **Ex.:** escrita no disco rígido na área onde fica o sistema operacional só pode ser realizada em modo kernel.



# Fundamentos de Sistemas Operacionais

## Exemplo - Atualização de arquivos em disco





## Vamos pensar...

- **Quais as principais funções do Kernel de um S.O.?**
- **Conceituemos *System Call* e sua importância para o sistema;**
- **Qual a relação existente entre os modos de acesso e as *System Calls*?**



## Arquiteturas de S.O.

- **Maneira como os componentes estão interconectados;**
- **Objetivo: avaliar qual a melhor arquitetura de S.O. a ser implementada para um determinado ambiente:**
  - **Monolítica;**
  - **Camadas;**
  - **Máquina Virtual;**
  - **MicroKernel;**

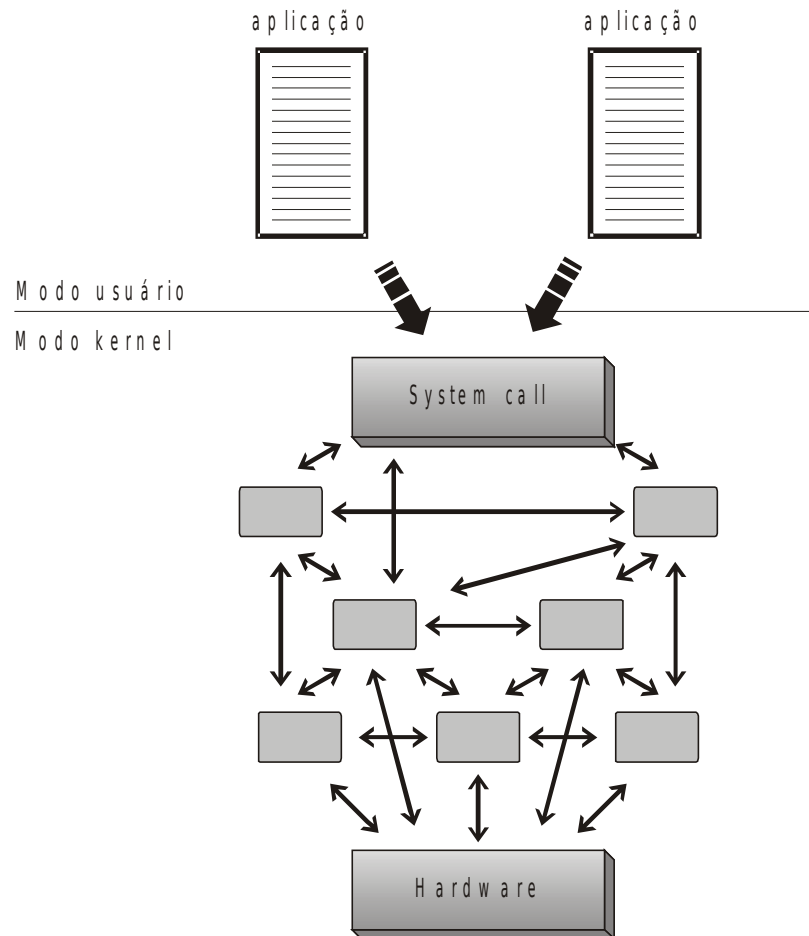


## Arquitetura Monolítica

- **Sistemas pequenos, simples e limitados (o *hardware* também era limitado);**
- **Único executável com procedimentos interagindo livremente: Qualquer procedimento pode chamar qualquer um dos demais;**
- **“A grande confusão” (segundo Tanenbaum);**
- **Utilizado para operações simples e repetitivas;**
- **Ex.: MS-DOS.**



## Arquitetura Monolítica





## Arquitetura em Camadas

- **Divisão em níveis;**
- **Cada nível oferece um conjunto de funções que podem ser utilizadas somente pelas camadas superiores;**
- **Os níveis mais próximos do *hardware* geralmente são executados em modo kernel e os mais distantes em modo usuário;**
- **Vantagem:**
  - **Modularidade (possibilidade de depurar cada uma das camadas, partindo da de mais baixo nível para a de mais alto nível).**



## Arquitetura em Camadas

- **Desvantagens:**
  - Complexidade de planejamento (delimitar o que vai estar em cada camada);
  - Desempenho (uma *System Call* passa por várias camadas até chegar ao *hardware*);
- **2 Desvantagens e 1 vantagem:**
  - Resultado: os sistemas estão sendo implementados em menor quantidade de camadas;
- **Exemplo**
  - Primeiras versões do Windows NT (várias camadas);
  - Windows 2000 (uma camada em modo usuário e uma em modo kernel).



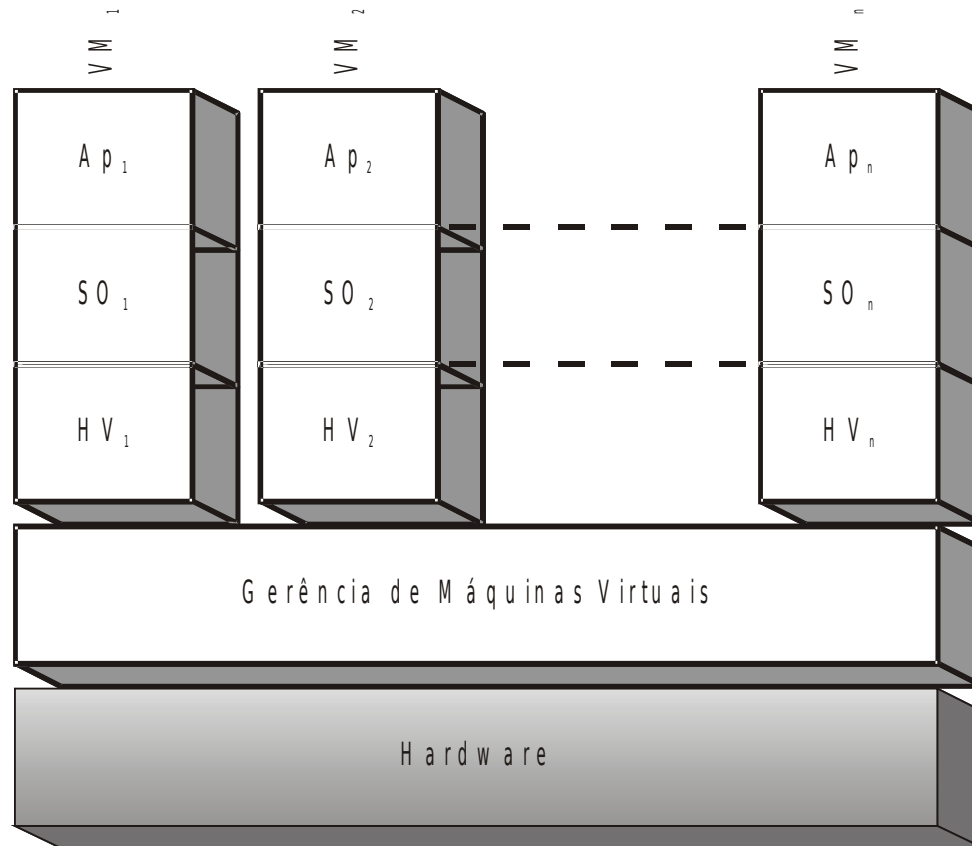


## Máquinas Virtuais

- **A máquina virtual divide o computador em dois ou mais computadores virtuais completamente isoladas;**
- **Utiliza técnicas de escalonamento de CPU e memória virtual para criar a ilusão de que o processo tem seus próprios recursos;**
- **Possibilita a utilização de mais de um S.O. em uma mesma máquina;**
- **Desvantagem: complexidade (necessidade de compartilhar o mesmo *hardware* entre diferentes máquinas virtuais).**



## Máquinas Virtuais



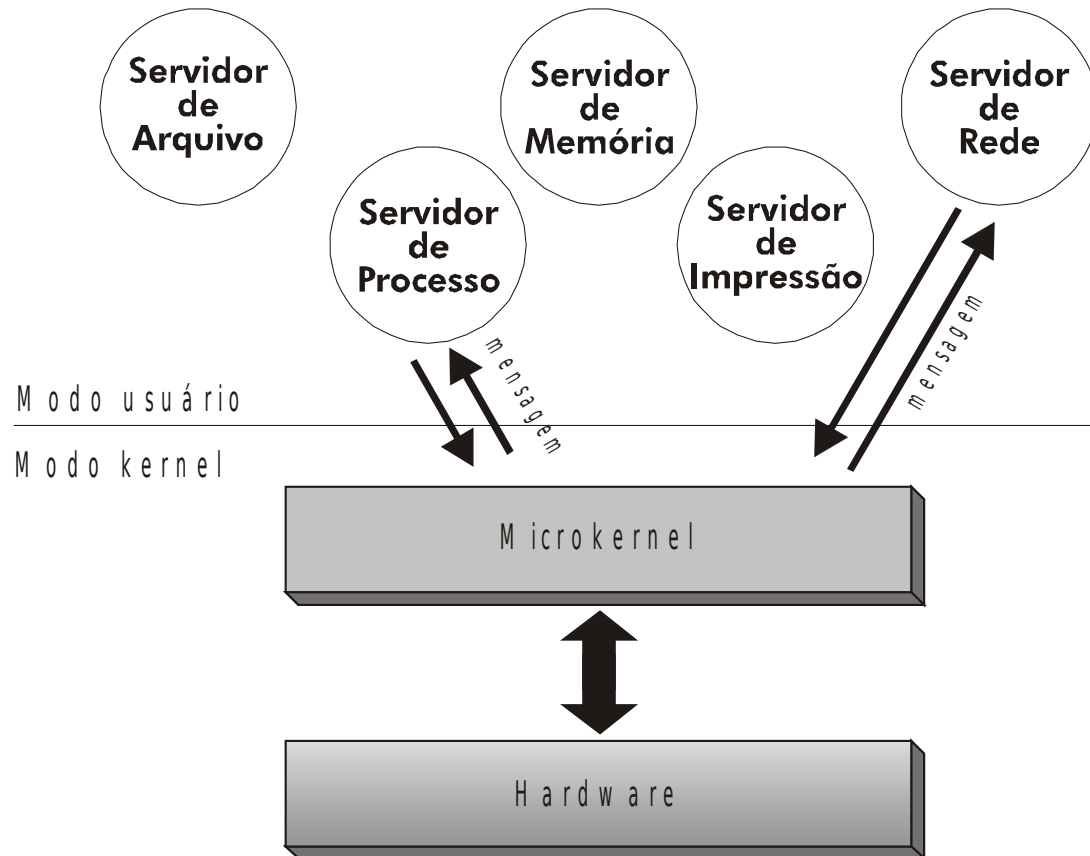


## Microkernel

- **Kernel modularizado de acordo com as funcionalidades (programas em nível de usuário):**
  - **Exemplos:**
    - módulo para gerência de arquivos
    - módulo para gerência de processos
    - módulo para gerência de memória e etc.;
- **Cliente (aplicação) / Servidor (módulo)**
- **Kernel reduzido (somente controle de mensagens).**



## Microkernel





## Microkernel

- **Em sistemas multicomputadores, cada computador pode disponibilizar alguns módulos;**
- **Dificuldade de ser implementado;**
- **Baixo desempenho (excesso de troca de mensagens);**
- **Ainda está restrito a pesquisas.**



## Bibliografia

- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**, 3<sup>a</sup> Ed., Rio de Janeiro: LTC Editora, 2002.
- SILVA, Guilherme Baião S. *Slides da disciplina de Sistemas Operacionais de Arquitetura Fechada*. Faculdade INED, 2005.