


© Volnys Bernal 1998-2010 1

Introdução aos Sistemas Operacionais

Volnys Borges Bernal
volnys@lsi.usp.br
<http://www.lsi.usp.br/~volnys>

Laboratório de Sistemas Integráveis
<http://www.lsi.usp.br/>



© Volnys Bernal 1998-2010 2

Agenda

- ❑ **Objetivos de um Sistema Operacional**
- ❑ **Componentes de um Sistema Operacional**
- ❑ **Máquina de Níveis**
- ❑ **Classificação dos sistemas operacionais**
- ❑ **Variedades de sistema operacional**
- ❑ **Chamadas ao sistema**
- ❑ **Exemplo de arquiteturas**
 - ❖ Arquitetura UNIX
 - ❖ Arquiteturas Windows
 - ❖ Arquitetura WindowsNT


© Volnys Bernal 1998-2010 3

Sobre esta apresentação

- ❑ **Esta apresentação ...**
 - ❖ Não apresenta todos os detalhes sobre este tópico.
 - ❖ É um resumo para auxiliar a apresentação do tópico em sala de aula.
- ❑ **Para estudo, deve ser utilizada uma das seguintes referências:**
 - ❖ Capítulos 1 e 2 do livro:
 - ANDREW S. TANENBAUM; Sistemas Operacionais Modernos. Prentice-Hall
 - ❖ Capítulos 1 e 2 do livro:
 - ANDREW S. TANENBAUM; Sistemas Operacionais. Prentice-Hall.

© Volnys Bernal 1998-2010 4

Objetivos de um Sistema Operacional



© Volnys Bernal 1998-2010 5

Objetivos de um Sistema Operacional

- ❑ **Principais objetivos de um Sistema Operacional:**
 1. Fornecer uma interface de alto nível
 - Para usuários
 - Para desenvolvimento de software
 2. Gerenciar os recursos do sistema
 - Gerenciar o compartilhamento dos recursos
 - Gerenciamento de conflitos de acesso aos recursos
 - Segurança no acesso aos recurso

© Volnys Bernal 1998-2010 6

Objetivos do Sistema Operacional

- 1) **Fornecer uma interface de alto nível**
 - ❖ Facilitar de acesso aos recursos do sistema
 - ❖ Facilidade de uso sem preocupação com detalhes de baixo nível
 - Exemplo:
 - leitura de disco: acionar a cabeça da leitura, posicionar na trilha e setor, transferir os dados para memória
 - ❖ Máquina Virtual
 - ❖ Interface entre:
 - Recursos do sistema
 - Usuário / programas

© Volnys Bernal 1998-2010 7

Objetivos do Sistema Operacional

2) Gerenciar os recursos do sistema

- ❖ Problema
 - Gerenciar o compartilhamento
 - Otimização de uso
 - Resolução de conflitos
 - Proteção / segurança
- ❖ Recursos
 - Processador (tempo de CPU)
 - Memória Primária
 - Memória Secundária (Disco)
 - Memória Terciária (Fita)
 - Impressoras
 - etc
- ❖ Objetivo
 - Maior taxa de utilização dos recursos
 - Diminuição de custos

© Volnys Bernal 1998-2010 8

Objetivos do Sistema Operacional

❑ Interface entre processos/usuários e recursos do sistema

© Volnys Bernal 1998-2010 9

Componentes de um Sistema Operacional

© Volnys Bernal 1998-2010 10

Componentes de um Sistema Operacional

❑ Estrutura geral de um sistema de computação

© Volnys Bernal 1998-2010 11

Componentes de um Sistema Operacional

❑ É composto por:


- ❖ Núcleo (kernel) do sistema operacional
 - Permanece carregado em memória
- ❖ Processos de gerência
- ❖ Arquivos de configuração
- ❖ Utilitários do sistema
 - Programas básicos necessários para operação do sistema
 - Ex: DOS: format, dir, edit, copy, type, ...
 - Ex: UNIX: mkfs, ls, ps, vi, ...

© Volnys Bernal 1998-2010 12

Componentes de um Sistema Operacional

© Volnys Bernal 1998-2010 13

Máquina de Níveis




© Volnys Bernal 1998-2010 14

Máquina de níveis

Software	Aplicativos
	Utilitários e Processos de gerência
	Núcleo do Sistema Operacional
	Monitor
Hardware	Linguagem de Máquina
	Microprogramação
	Dispositivos Físicos

© Volnys Bernal 1998-2010 15

Classificação dos sistemas operacionais



© Volnys Bernal 1998-2010 16

Monoprogramado x multiprogramado

❑ **Classificação quanto à quantidade de aplicações simultâneas:**

- ❖ Sistema operacional monoprogramado
 - Suporta somente a execução de um processo (aplicação) por vez
 - Exemplo:
 - DOS
 - CPM
- ❖ Sistema operacional multiprogramado
 - Suporta a execução simultânea de várias aplicações (processos) por vez
 - Exemplos (sistemas operacionais modernos):
 - Unix, Windows 95, WindowsNT

© Volnys Bernal 1998-2010 17


Classificação dos sistemas operacionais

❑ **Classificação quando ao suporte a ambiente multiprocessadores**

- ❖ Monoprocessador
 - Pode ser executado somente em hardwares monoprocessadores (1 único processador)
- ❖ Multiprocessador
 - Pode ser executado em hardwares multiprocessadores (vários processadores)

© Volnys Bernal 1998-2010 18

Variedades de Sistemas Computacionais




© Volnys Bernal 1998-2010 19

Variedades de Sistemas Computacionais

- ❑ Atualmente existe uma grande variedade de sistemas computacionais, cada qual necessitando de características específicas do sistema operacional.
- ❑ Exemplo de sistemas computacionais:
 - ❖ Computadores de grande porte
 - ❖ Servidores
 - ❖ Sistemas multiprocessadores
 - ❖ Computador para uso corporativo
 - ❖ Computadores pessoais
 - ❖ Sistemas de tempo real
 - ❖ Sistemas embarcados
 - ❖ Cartões inteligentes

© Volnys Bernal 1998-2010 20

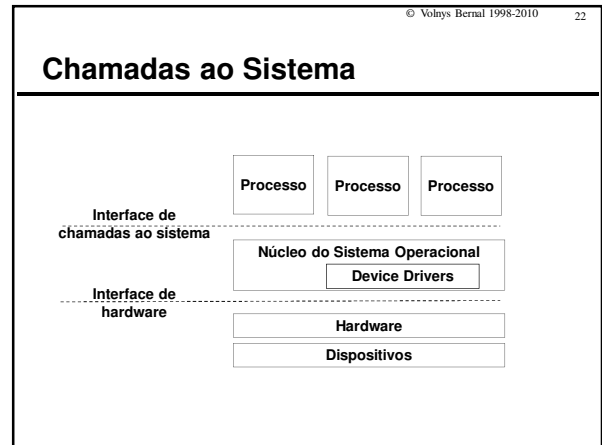
Chamadas ao Sistema



© Volnys Bernal 1998-2010 21

Chamadas ao Sistema

- ❑ O que é?
 - ❖ Conjunto de funções que o núcleo do sistema operacional disponibiliza aos processos
 - ❖ Em inglês: "System Calls"
- ❑ Cada sistema operacional possui uma interface de chamadas ao sistema específica



© Volnys Bernal 1998-2010 23

Chamadas ao Sistema

- ❑ Principais Chamadas UNIX
 - ❖ Manipulação de processos

Chamada	Descrição
fork	Duplica um processo
waitpid	Aguarda um processo terminar
execve	Troca a imagem de memória do processo
exit	Termina a execução do processo

© Volnys Bernal 1998-2010 24

Chamadas ao Sistema

- ❑ Principais Chamadas UNIX
 - ❖ Ações sobre arquivos

Chamada	Descrição
open	Abre um arquivo
close	Fecha um arquivo aberto
read	Lê dados de um arquivo
write	Escreve dados em um arquivo
ioctl	Funções de controle para arquivos especiais (dispositivos)
lseek	Posiciona o ponteiro de deslocamento do arquivo
stat	Obtém informações de controle do arquivo (dono, proteção, ...)

© Volnys Bernal 1998-2010 25

Chamadas ao Sistema

❑ Principais Chamadas UNIX

❖ Manipulação de arquivos e diretórios

Chamada	Descrição
mkdir	Cria um novo diretório
rmdir	Remove um diretório vazio
link	Cria um hard link
unlink	Remove uma entrada do diretório
mount	Monta um sistema de arquivos
umount	Desmonta um sistema de arquivos

© Volnys Bernal 1998-2010 26

Chamadas ao Sistema

❑ Principais Chamadas UNIX

❖ Outras chamadas

Chamada	Descrição
chdir	Muda o diretório de trabalho
chmod	Altera o modo de permissão do arquivo
kill	Envia um sinal para um processo
time	Obtém o data/hora corrente

© Volnys Bernal 1998-2010 27

Chamadas ao Sistema

❑ Principais Chamadas Win32

❖ Manipulação de processos

Chamada	Descrição
CreateProcess	Cria um processo
WaitForSingleObject	Aguarda um processo terminar
ExitProcess	Termina a execução do processo

© Volnys Bernal 1998-2010 28

Chamadas ao Sistema

❑ Principais Chamadas Win32

❖ Ações sobre arquivos

Chamada	Descrição
CreateFile	Cria um arquivo ou abre um arquivo existente
CloseHandle	Fecha um arquivo aberto
ReadFile	Lê dados de um arquivo
WriteFile	Escreve dados em um arquivo
SetFilePointer	Posiciona o ponteiro de deslocamento do arquivo
GetFileAttributeEx	Obtém informações de controle do arquivo

© Volnys Bernal 1998-2010 29

Chamadas ao Sistema

❑ Principais Chamadas Win32

❖ Manipulação de arquivos e diretórios

Chamada	Descrição
CreateDirectory	Cria um novo diretório
RemoveDirectory	Remove um diretório vazio
DeleteFile	Remove uma entrada do diretório

© Volnys Bernal 1998-2010 30

Chamadas ao Sistema


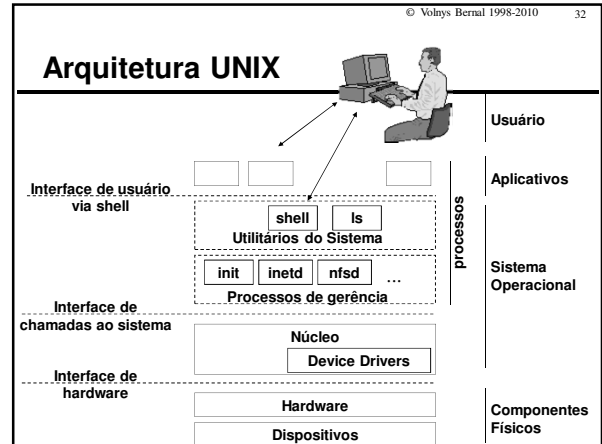
❑ Principais Chamadas Win32

❖ Outras chamadas

Chamada	Descrição
SetCurrentDirectory	Muda o diretório de trabalho
GetLocalTime	Obtém o data/hora corrente

© Volnys Bernal 1998-2010 31

Arquitetura UNIX


© Volnys Bernal 1998-2010 33

Arquitetura UNIX

- **Núcleo do Sistema Operacional**
 - ❖ Existem diferentes implementações de UNIX
 - ❖ A maior partes dos sistemas UNIX possui um núcleo do tipo Monolítico
 - ❖ Observação: Software monolítico:
 - Software que é estruturado como um único programa com coerção de chamadas às funções.

© Volnys Bernal 1998-2010 34


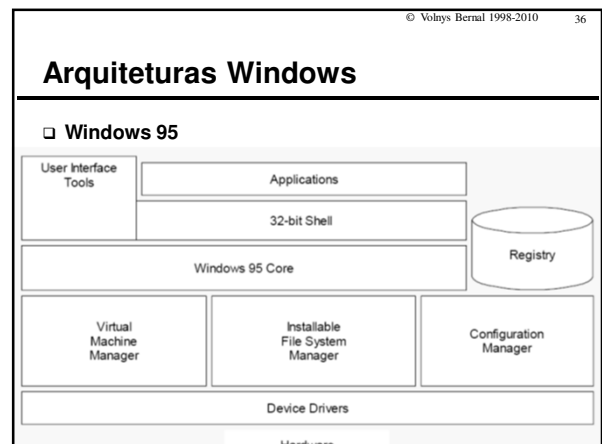
Arquiteturas Windows



© Volnys Bernal 1998-2010 35


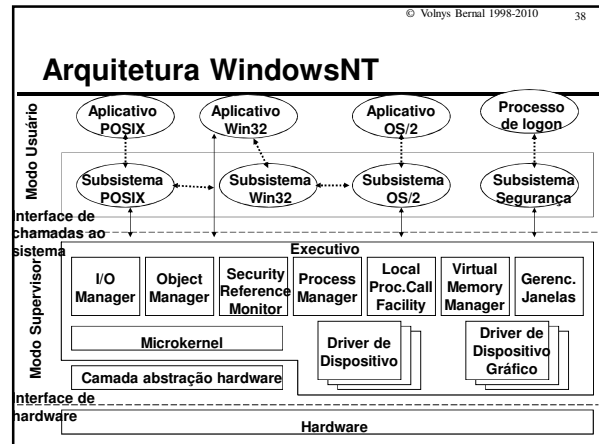
Arquiteturas Windows

- **Família Windows 3.x**
 - ❖ Voltado a processadores de 16 bits
 - ❖ Suporte a multiprogramação cooperativa
 - ❖ Memória virtual primitiva
- **Windows 95**
 - ❖ Voltado a processadores de 32 bits
 - ❖ Suporte a multiprogramação preemptiva
 - ❖ Memória virtual
- **Família Windows NT (Windows NT, Windows 2000, Windows 2003)**
 - ❖ Voltado a processadores de 32 bits
 - ❖ Multiprogramado
 - Multiprogramação preemptiva
 - ❖ Memória virtual
 - ❖ Núcleo multi-threaded

© Volnys Bernal 1998-2010 37

Arquitetura WindowsNT

© Volnys Bernal 1998-2010 39

Arquitetura WindowsNT

- **Interface de chamadas ao sistema**
 - ❖ Chamada também de Interface NT nativa
 - ❖ Define o conjunto de serviços que o sistema operacional fornece aos processos: ~250 funções
 - ❖ Chamada ao sistema é implementada através de TRAP
 - TRAP = interrupção de software
 - Permite garantir que somente o "kernel do NT" será executado em modo supervisor
 - As interrupções são atendidas em modo supervisor
 - O vetor de interrupções é controlado pelo *microkernel*

© Volnys Bernal 1998-2010 40

Arquitetura WindowsNT

- **Executivo**
 - ❖ Características
 - Monolítico
 - Imagem ntoskrnl.exe contém todo código dos serviços executivos (exceto o Microkernel)
 - Multi-threaded
 - Reentrante

© Volnys Bernal 1998-2010 41

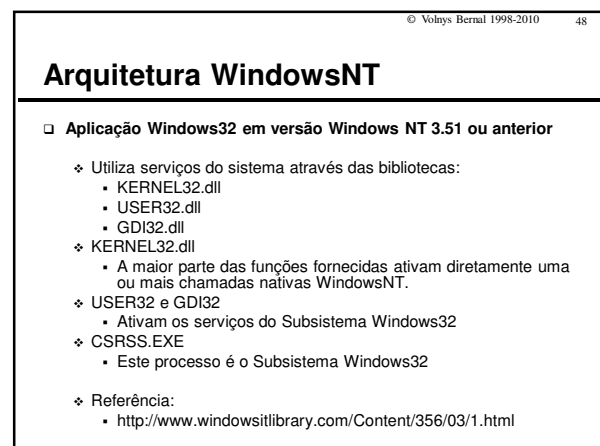
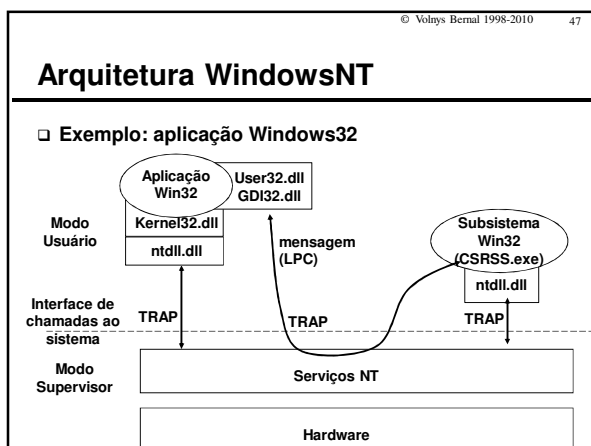
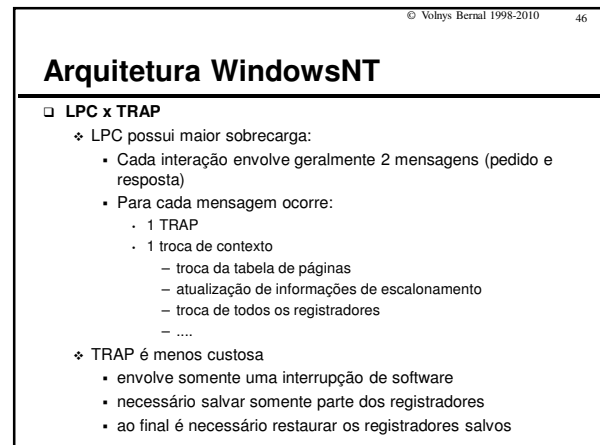
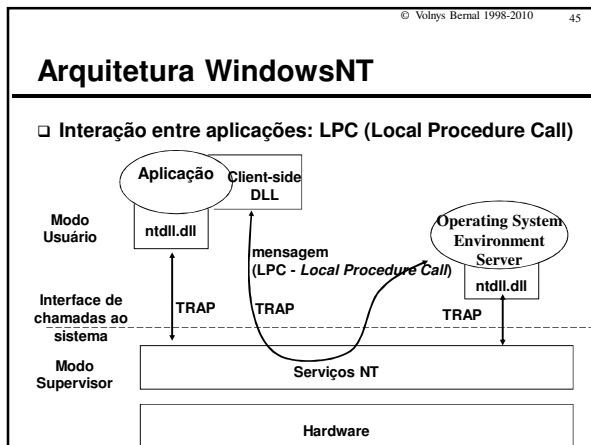
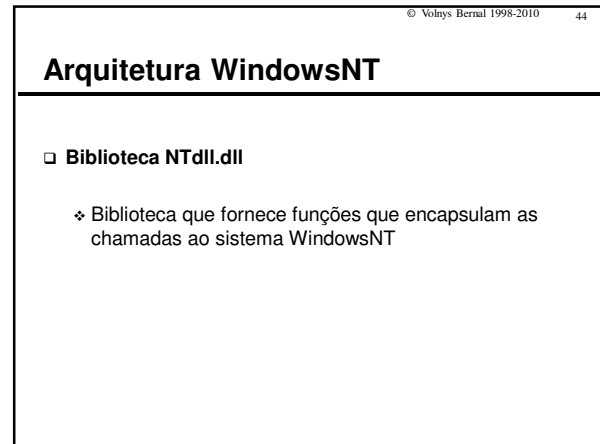
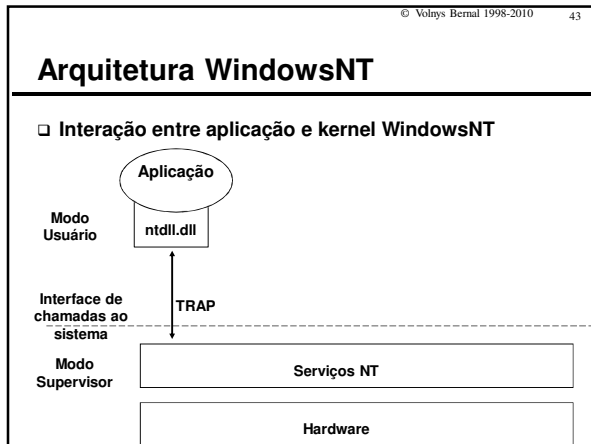
Arquitetura WindowsNT

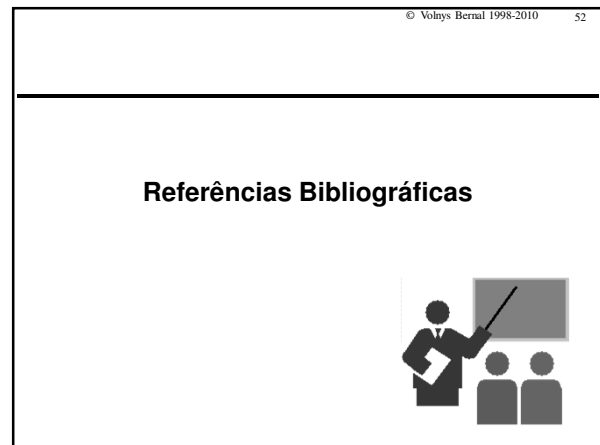
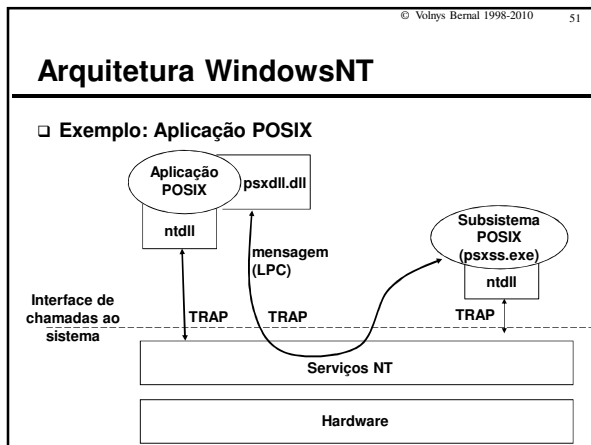
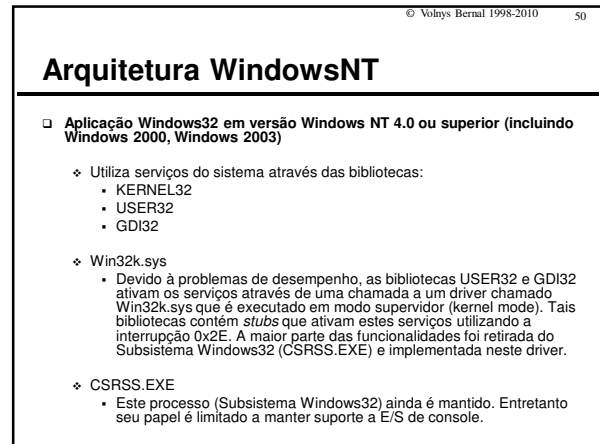
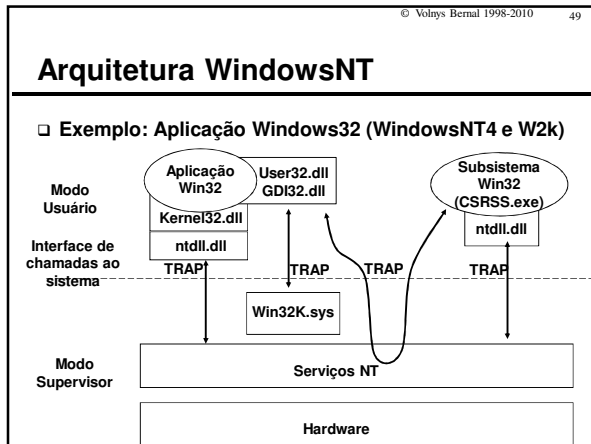
- **Microkernel (também chamado de NT kernel)**
 - ❖ Responsável por
 - Escalonamento de *threads (dispatcher)*
 - Sincronização
 - Manipulação dos vetores de interrupção
 - ❖ Implementa 32 níveis de prioridade
 - ❖ Implementa escalonamento preemptivo
 - ❖ Implementa sincronização baseada em:
 - *mutex*
 - semáforos
 - eventos
 - *spinlocks*

© Volnys Bernal 1998-2010 42

Arquitetura WindowsNT

- **HAL**
 - ❖ *Hardware Abstraction Layer*
 - (camada de abstração de hardware)
 - ❖ Camada utilizada para esconder dependências de arquitetura
 - ❖ Exemplo:
 - Sistemas monoprocessadores x multiprocessadores
 - Para cada versão NT são fornecidas 3 versões
 - monoprocessador
 - multiprocessador
 - para depuração (p/ desenvolvimento e device drivers)





© Volnys Bernal 1998-2010 53

Referências Bibliográficas

- ❑ ANDREW S. TANENBAUM; **Sistemas Operacionais Modernos.** Prentice-Hall.
- ❑ ANDREW S. TANENBAUM; **Sistemas Operacionais.** Prentice-Hall.
- ❑ Windows 2000 Magazine Online
 - <http://www.winntmag.com/Articles>
- ❑ <http://www.windowstlibrary.com/Content/356/03/1.html>