

Estudo de caso

Windows

História do Windows

- MS-DOS
- Consumer Windows (95/98/Me)
- Windows NT
- Windows XP

CP/M

- 1974
- Intel, processador 8080, 8 bits
- Control Program for Microcomputers
- Gary Kildall e Digital Research

IBM-PC



- Início dos anos 80
- Qual sistema utilizar?
- Gary (Digital) não fechou negócio com a IBM

MS-DOS

- QDOS (Quick and Dirty Operating System)?
- Bill Gattes
- Comprado da Seattle Computer Products por 50 mil dólares
- MS-DOS (MicroSoft Disk Operanting System)

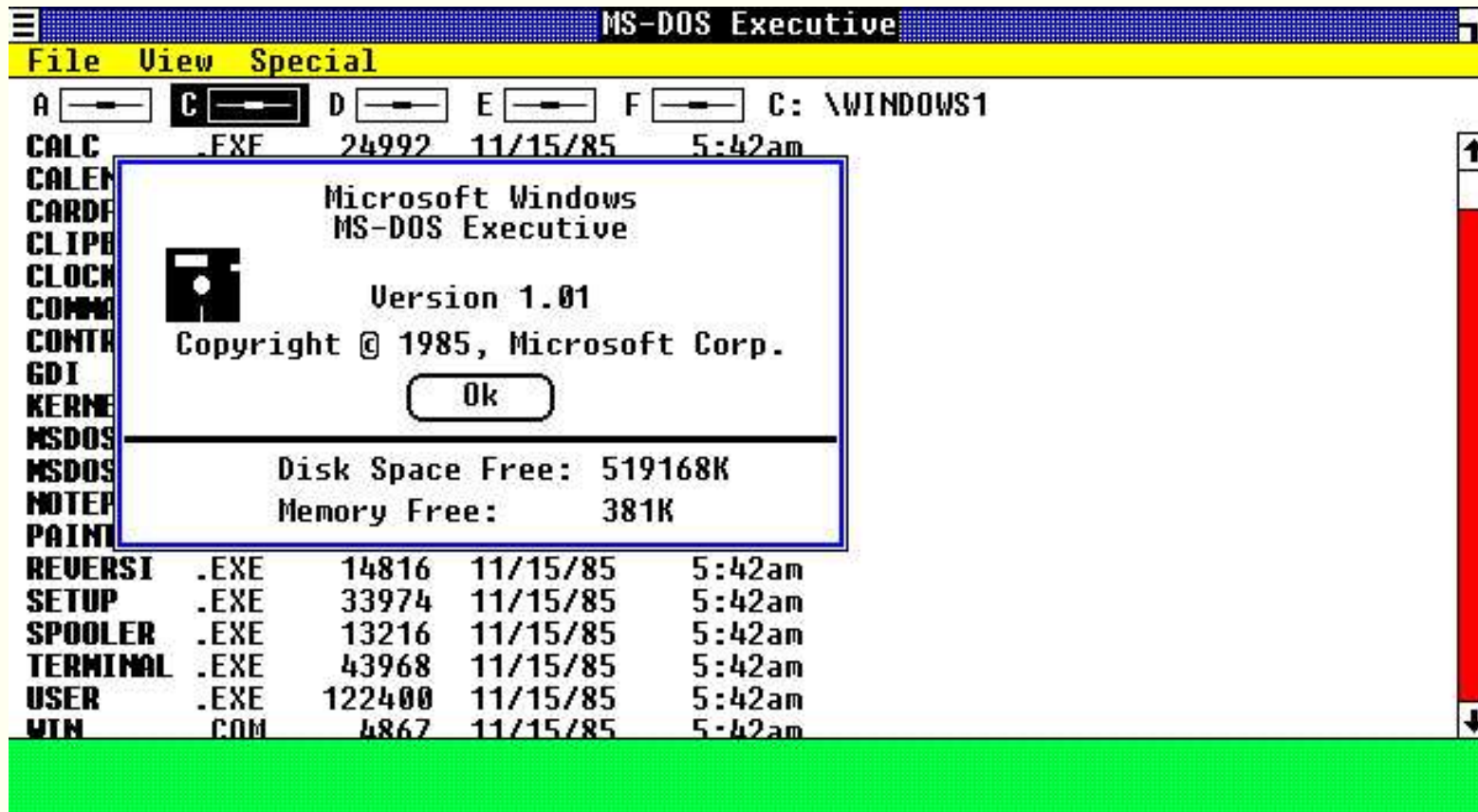


MS-DOS



- linha de comando
- 16 bits
- monousuário

Windows 1.0



- Inspirado no Apple Lisa (Macintosh)

Windows 3.0

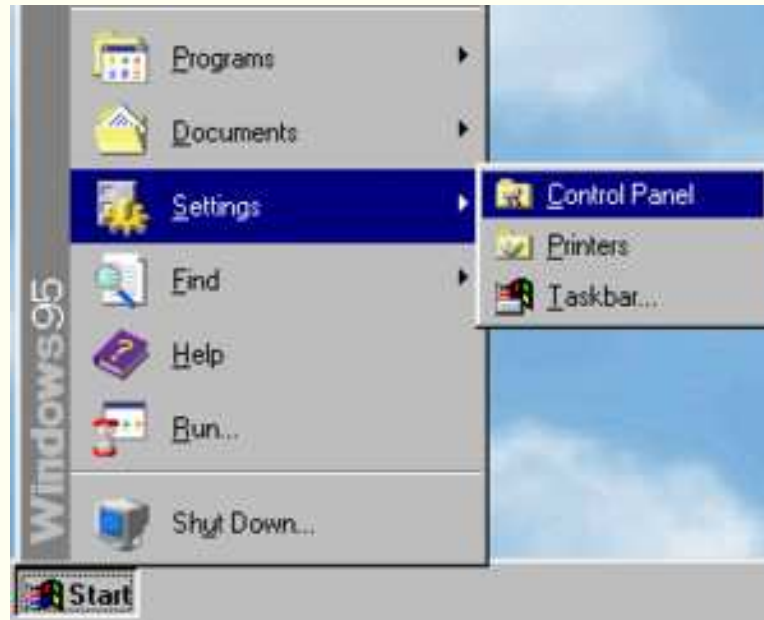


- 1990
- Sucesso comercial

Windows

- Interface gráfica sobre o MS-DOS
- Espaço de endereçamento compartilhado (bugs!)

Windows 95

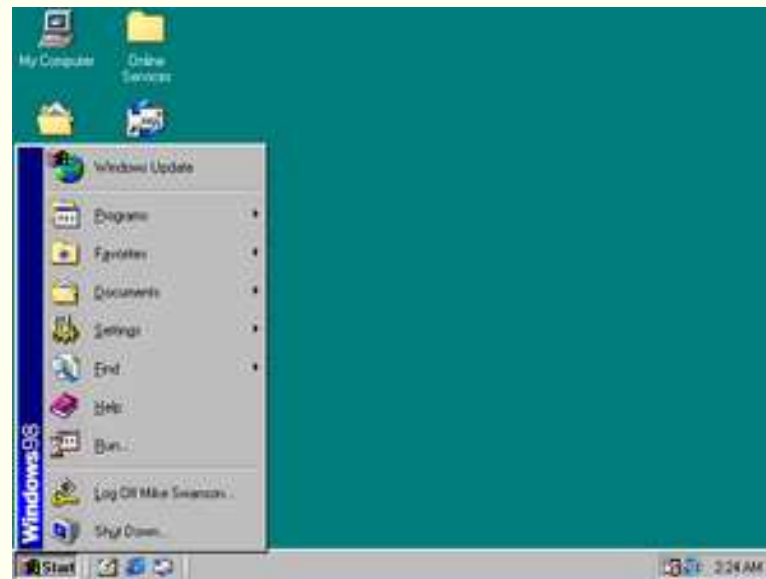


- Não elimina o MS-DOS, mas transfere funcionalidades do MS-DOS para o Windows

Windows 95

- Memória virtual
- Multiprogramação
- 32 bits, mas várias partes de 16 bits
- Nomes de arquivos mais longos (substituindo 8+3)

Windows 98



- Mais funcionalidades são transferidas do MS-DOS para o Windows
- Ainda há partes de código 16-bits

Windows 98

- 4 GB de memória virtual
 - 2 GB privados para um processo
 - 1 GB compartilhado (bugs!)
 - 1 MB para compatibilidade com o MS-DOS
- Kernel não era re-entrante
 - Mutex único trava todo o sistema
 - Um processo pode ter de esperar por outro não relacionado
- Integração entre o desktop e a Internet
 - Acusação de monopólio

Windows Me

Windows Millennium Edition



- Muito semelhante ao Windows 98

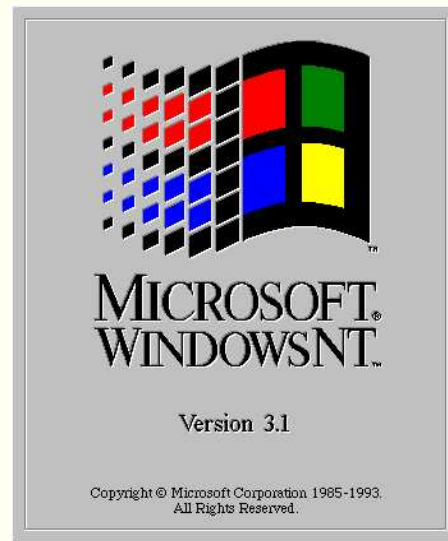
Windows Me

- Facilidades extras:
 - Simplificou compartilhamento de imagens e músicas
 - Aplicações voltadas para a Internet
 - Jogos multiusuário
 - Mecanismo para restaurar configurações anteriores

Windows NT

- Conclusão ao final dos anos 80:
 - Seria difícil construir um sistema 32 bits sobre o MS-DOS 16 bits
- Windows New Technology
- Baseado no VMS da DEC (contratação de David Cutler)
- Visão importante: computadores pessoais seriam usados em aplicações comerciais
- Voltado para aplicações comerciais e de usuários

Windows NT 3.1

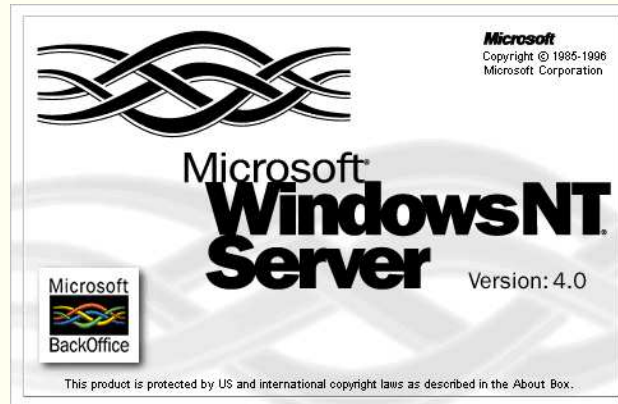


- Primeira versão em 1993. Número escolhido para combinar com o Windows 3.1
- Microsoft esperava um grande sucesso

Windows NT 3.1

- Apesar de muito superior, as pessoas não quiseram trocar o Windows 3.1 pelo Windows NT 3.1
 - NT precisava de máquinas mais poderosas
 - Não havia muitas aplicações disponíveis
- Fracasso do NT fez com que fossem lançados o Windows 95/98/Me

Windows NT 4.0



- Algum espaço no mercado de servidores
- Mesma interface do Windows 95
- Portabilidade
 - 16 milhões de linhas de código (C, C++ e assembly)
 - Pentium, Alpha, MIPS e PowerPC

Windows 95/98 e Windows NT

Item	Windows 95/98	Windows NT
Full 32-bit system?	No	Yes
Security?	No	Yes
Protected file mappings?	No	Yes
Private addr space for each MS-DOS prog?	No	Yes
Unicode?	No	Yes
Runs on	Intel 80x86	80x86, Alpha, MIPS, ...
Multiprocessor support?	No	Yes
Re-entrant code inside OS?	No	Yes
Plug and play?	Yes	No
Power management?	Yes	No
FAT-32 file system?	Yes	Optional
NTFS file system	No	Yes
Win32 API?	Yes	Yes
Run all old MS-DOS programs?	Yes	No
Some critical OS data writable by user?	Yes	No

Windows 2000



- Mesma interface do Windows 98
- Windows NT 5.0 com um nome mais neutro
- Modo usuário e modo supervisor
- Cluster: duas ou mais máquinas parecem uma única para o mundo externo
- Enfim, livres do MS-DOS

Versões do Windows 2000

Version	Max RAM	CPUs	Max clients	Cluster size	Optimized for
Professional	4 GB	2	10	0	Response time
Server	4 GB	4	Unlimited	0	Throughput
Advanced server	8 GB	8	Unlimited	2	Throughput
Datacenter server	64 GB	32	Unlimited	4	Throughput

- Diferenças entre as versões são mínimas
- O sistema verifica se está rodando a versão correta

Tamanho do código de alguns Sistemas Operacionais

Year	AT&T	BSD	MINIX	Linux	Solaris	Win NT
1976	V6 9K					
1979	V7 21K					
1980		4.1 38K				
1982	Sys III 58K					
1984		4.2 98K				
1986		4.3 179K				
1987	SVR3 92K		1.0 13K			
1989	SVR4 280K					
1991				0.01 10K		
1993		Free 1.0 235K			5.3 850K	3.1 6M
1994		4.4 Lite 743K		1.0 165K		3.5 10M
1996				2.0 470K		4.0 16M
1997			2.0 62K		5.6 1.4M	
1999				2.2 1M		
2000		Free 4.0 1.4M			5.8 2.0M	2000 29M

Ciclo de desenvolvimento do Windows

- Centenas de programadores trabalham no Windows durante o dia
- Trechos de código finalizado são submetidos
- Às 18:00hs, o sistema é reconstruído
- O novo sistema operacional é distribuído e submetido a uma série de testes durante a noite
- Na manhã seguinte, verifica-se se o código funciona

Windows XP



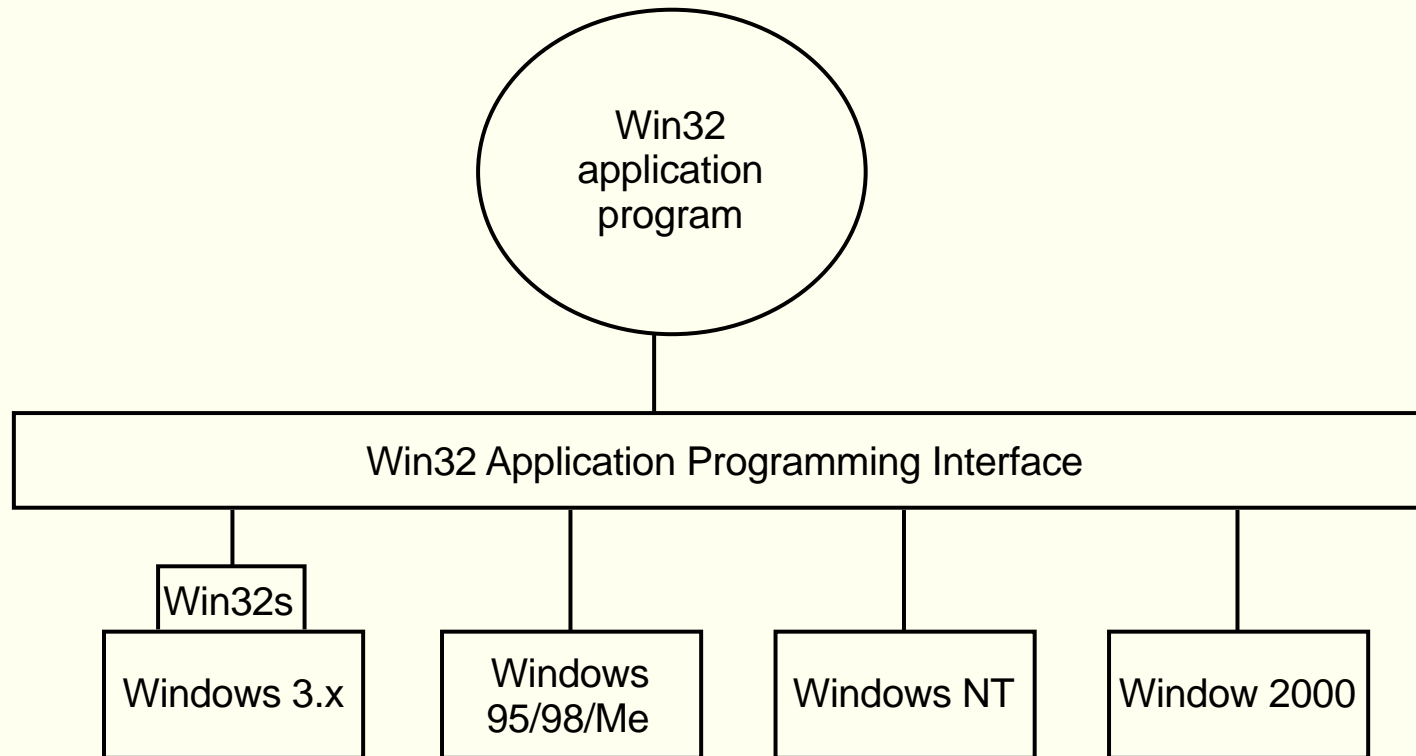
- Sucessor do Windows 2000
- Mais testes para devices drivers
- Eliminou alguns cenários de reboot
- Estruturas críticas do kernel são read-only
-

Programação no Windows 2000

Win32 Application Programming Interface

- Chamadas ao sistema
 - não são públicas
 - mudam de versão para versão
- Win32 API - conjunto de funções bem definidas que fazem as chamadas ao sistema ou operam em espaço do usuário
- Novas funções são acrescentadas frequentemente

Win32 API



- Existem pequenas variações; nem tudo funciona da mesma forma em todas os sistemas ...

Tipos de funções do Win32 API

- Processos, threads
 - Gerência de memória
 - Sincronização
 - I/O
- ⇒ Interface gráfica
- janelas, menus, ícones, barras de controle, ...

Objetos do núcleo

- Arquivos, processos, threads, pipes, ...
- Toda chamada retorna um **handle**, que pode ser usado posteriormente
- Handles possuem permissão de acesso
- Não podem ser simplesmente passados de um processo para outro
- Windows não é **orientado** a objetos
 - Faltam conceitos como herança e poliformismo

Informações de controle

- Windows 3.1: vários arquivos `.ini`
- A partir do Windows 95: **registry**
 - Banco de dados centralizado
 - Semelhante a um sistema de arquivos: diretórios, subdiretórios e entradas (arquivos)

Registry

Key	Description
HKEY_LOCAL_MACHINE HARDWARE SAM SECURITY SOFTWARE SYSTEM	Properties of the hardware and software Hardware description and mapping of hardware to drivers Security and account information for users System-wide security policies Generic information about installed application programs Information for booting the system
HKEY_USERS USER-AST-ID AppEvents Console Control Panel Environment Keyboard Layout Printers Software	Information about the users; one subkey per user User AST's profile Which sound to make when (incoming email/fax, error, etc.) Command prompt settings (colors, fonts, history, etc.) Desktop appearance, screensaver, mouse sensitivity, etc. Environment variables Which keyboard: 102-key US, AZERTY, Dvorak, etc. Information about installed printers User preferences for Microsoft and third party software
HKEY_PERFORMANCE_DATA	Hundreds of counters monitoring system performance
HKEY_CLASSES_ROOT	Link to HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES
HKEY_CURRENT_CONFIG	Link to the current hardware profile
HKEY_CURRENT_USER	Link to the current user profile

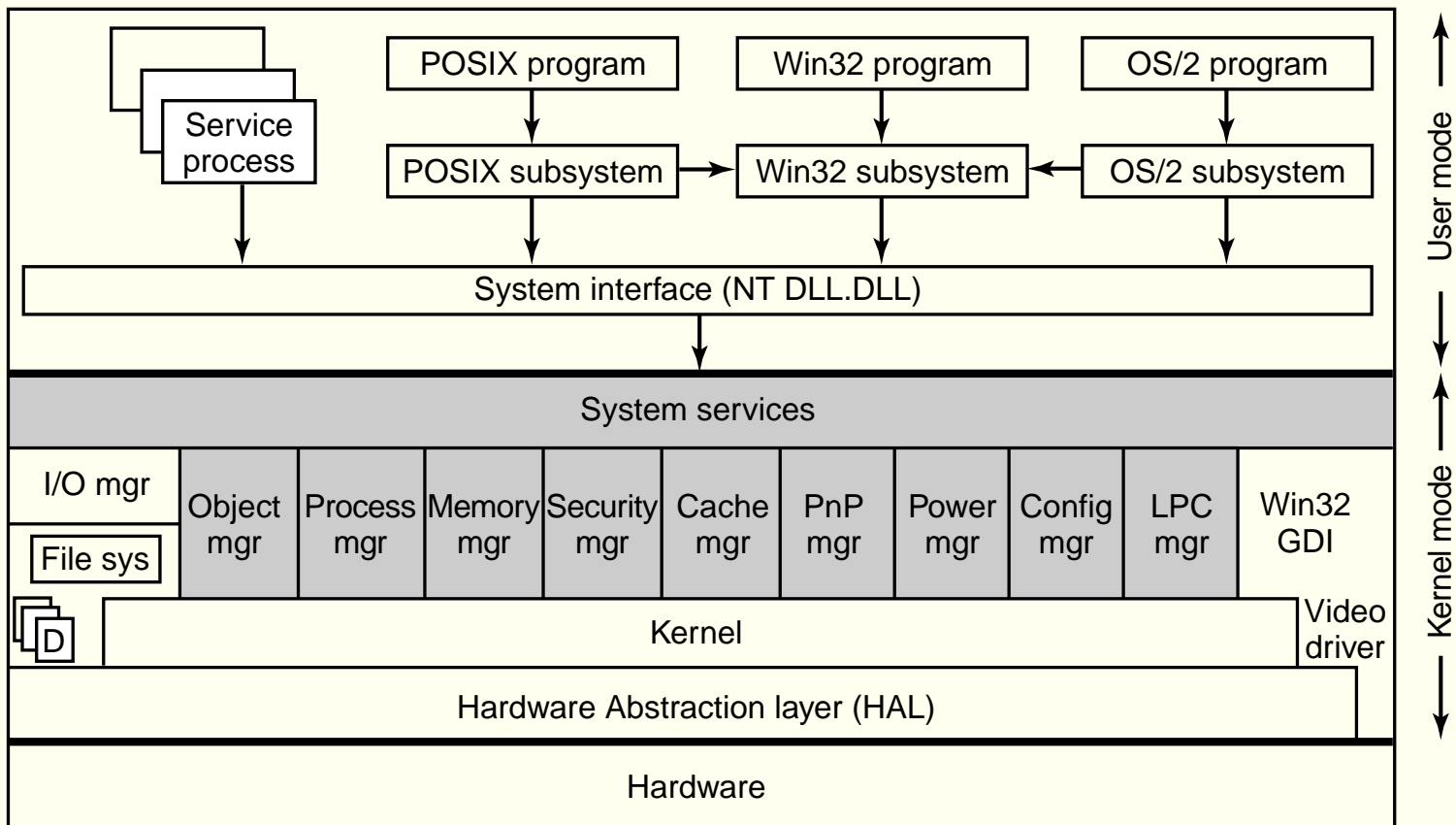
Gerência de entradas no Registry

- Funções disponíveis na API

Win32 API function	Description
RegCreateKeyEx	Create a new registry key
RegDeleteKey	Delete a registry key
RegOpenKeyEx	Open a key to get a handle to it
RegEnumKeyEx	Enumerate the subkeys subordinate to the key of the handle
RegQueryValueEx	Look up the data for a value within a key

- Editores: regedit e regedt32

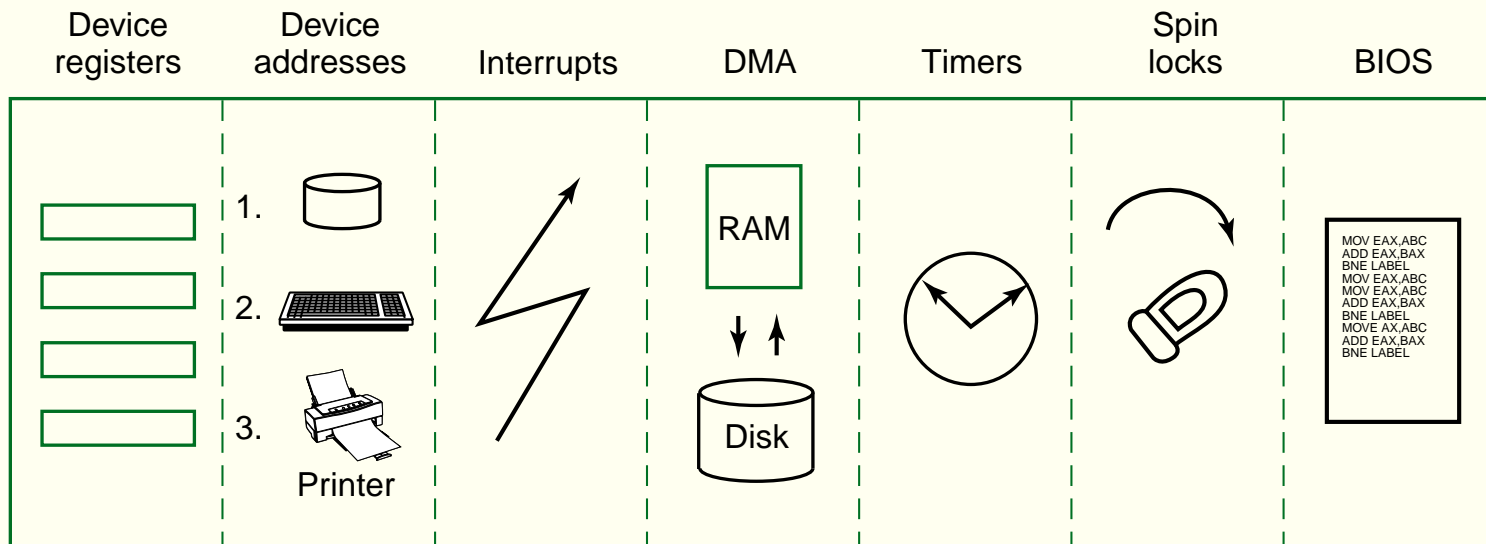
Estrutura do Windows 2000



Hardware Abstraction Layer (HAL)

- Portabilidade
 - fácil nos níveis mais altos
 - difícil para o código que trabalha diretamente no hardware
- HAL é uma camada que abstrai detalhes e esconde diferenças

Hardware Abstraction Layer



Hardware abstraction layer

- Por questões de eficiência, pode ser interessante fazer acesso ao hardware diretamente

A camada do núcleo

- Constrói abstrações de nível mais alto a partir das fornecidas pelo HAL
- A maior parte está escrita em C
- Troca de contexto
- Objetos de controle
 - DPC (Deferred Procedure Call)
 - APC (Asynchronous Procedure Call)
 - dispatcher objects (semáforos, locks, timers)

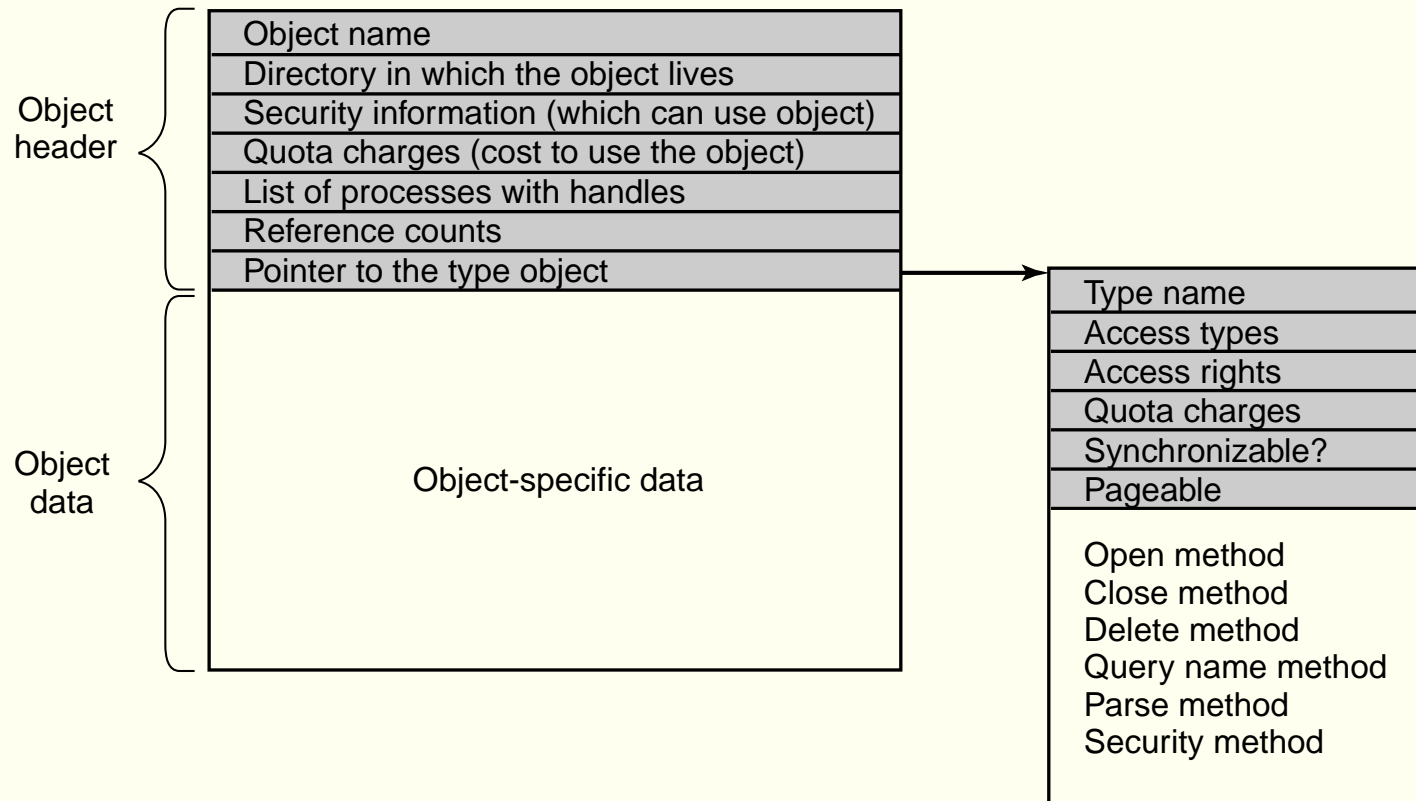
Executive

- Dividido em módulos
 - Object manager
 - I/O manager
 - Process manager
 - Memory manager
 - Security manager
 - Cache manager
 - Plug-and-play manager
 - Power manager
 - Configuration manager
 - Local procedure call manager
- System services

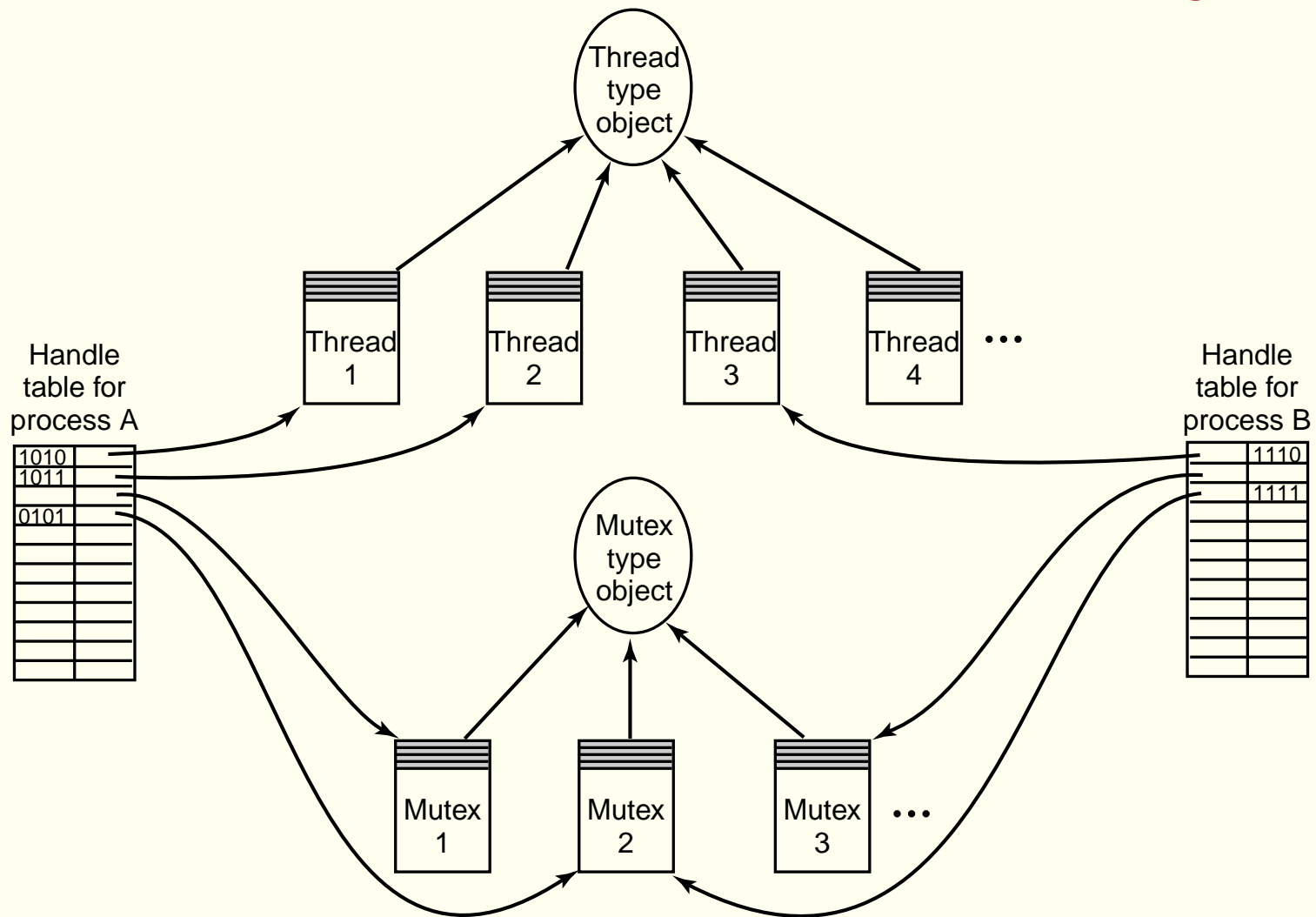
Tipos de objetos

Type	Description
Process	User process
Thread	Thread within a process
Semaphore	Counting semaphore used for interprocess synchronization
Mutex	Binary semaphore used to enter a critical region
Event	Synchronization object with persistent state (signaled/not)
Port	Mechanism for interprocess message passing
Timer	Object allowing a thread to sleep for a fixed time interval
Queue	Object used for completion notification on asynchronous I/O
Open file	Object associated with an open file
Access token	Security descriptor for some object
Profile	Data structure used for profiling CPU usage
Section	Structure used for mapping files onto virtual address space
Key	Registry key
Object directory	Directory for grouping objects within the object manager
Symbolic link	Pointer to another object by name
Device	I/O device object
Device driver	Each loaded device driver has its own object

Estrutura de um objeto



Tabelas de handles e tipos de objetos



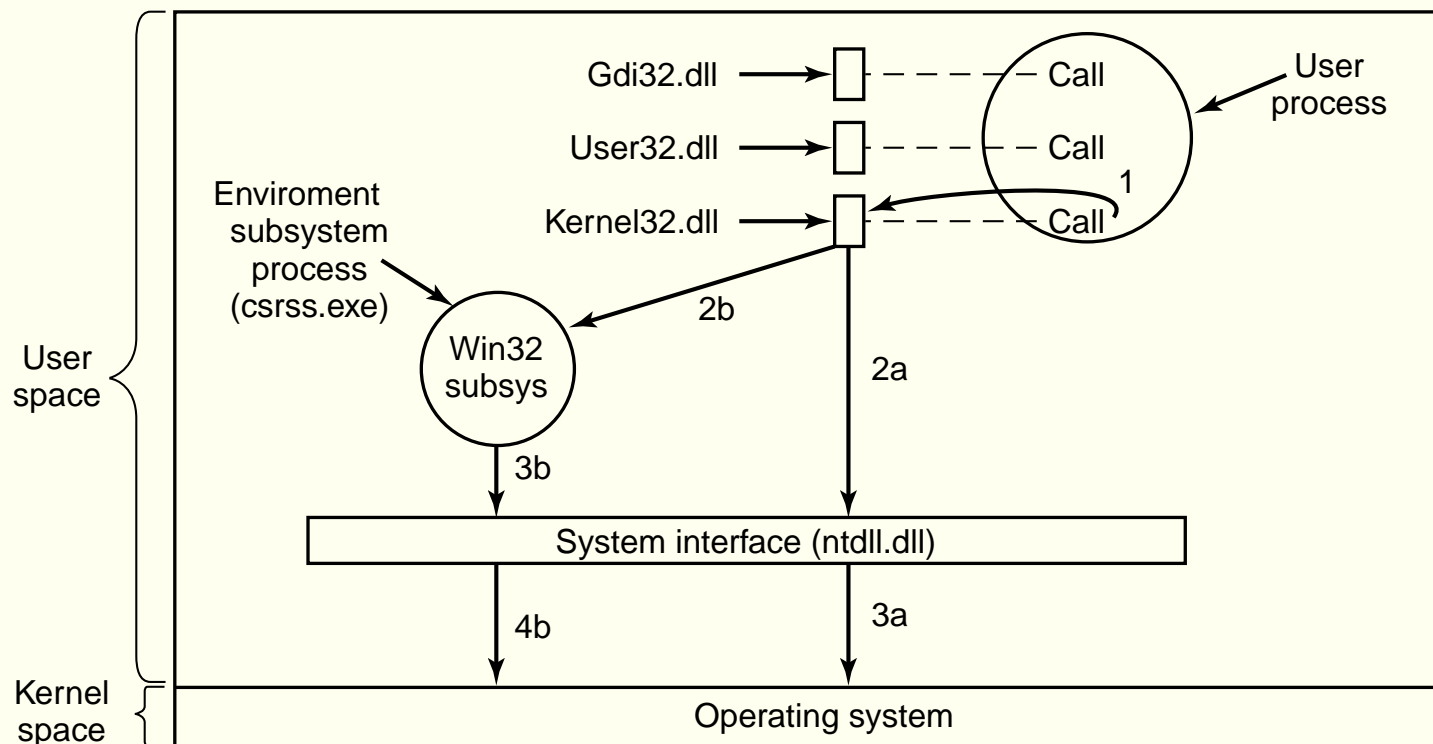
Diretórios no espaço de nomes

Directory	Contents
??	Starting place for looking up MS-DOS devices like C:
Device	All discovered I/O devices
Driver	Objects corresponding to each loaded device driver
ObjectTypes	The type objects shown in Fig. 0-0
Windows	Objects for sending messages to all the windows
BaseNamedObjs	User-created objects such as semaphores, mutexes, etc.
Arcname	Partition names discovered by the boot loader
NLS	National language support objects
FileSystem	File system driver objects and file system recognizer objects
Security	Objects belonging to the security system
KnownDLLs	Key shared libraries that are opened early and held open

Dynamic Link Libraries

- x.dll - arquivo contendo um conjunto de procedimentos relacionados
- Chamadas a procedimentos em DLLs são feitas de maneira indireta por meio de um vetor
- Apenas uma cópia da DLL existe na memória em um dado instante

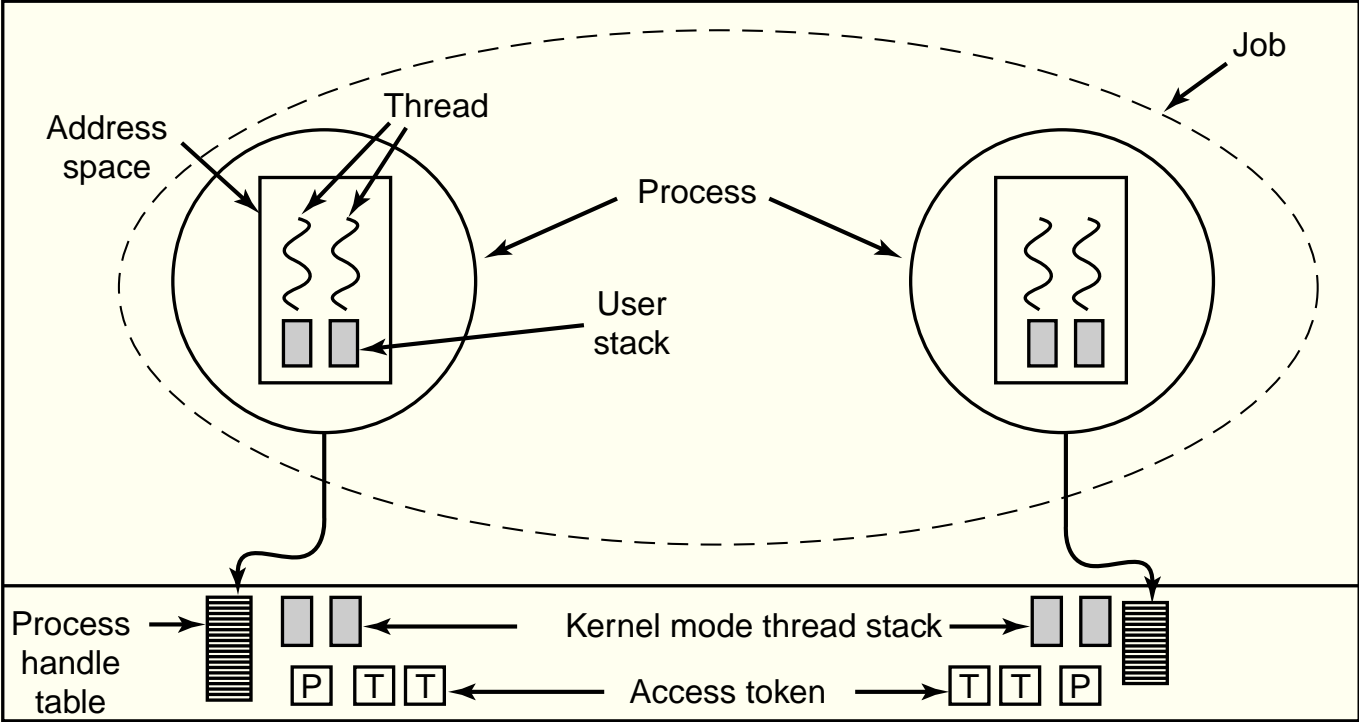
Duas maneiras de implementar as chamadas de funções da API



Conceitos básicos sobre gerência de processos e threads

Name	Description
Job	Collection of processes that share quotas and limits
Process	Container for holding resources
Thread	Entity scheduled by the kernel
Fiber	Lightweight thread managed entirely in user space

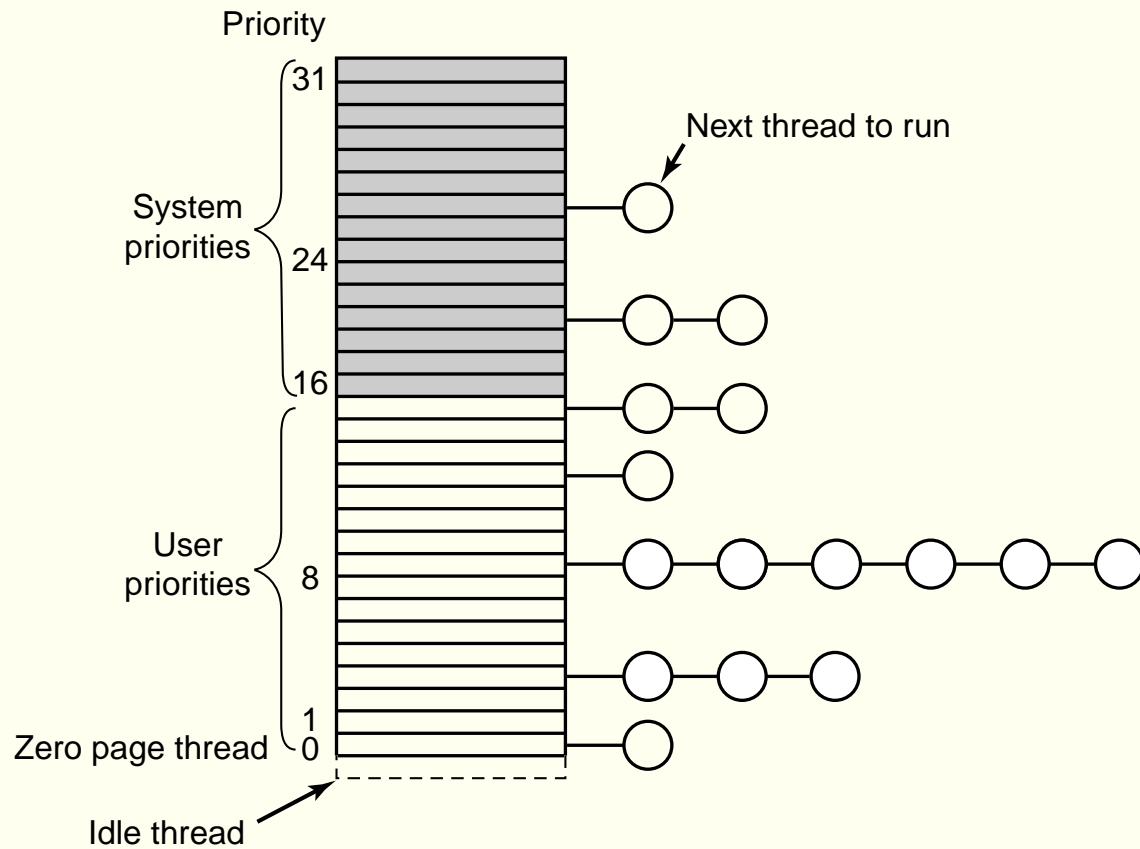
Relacionamento entre jobs, processos e threads



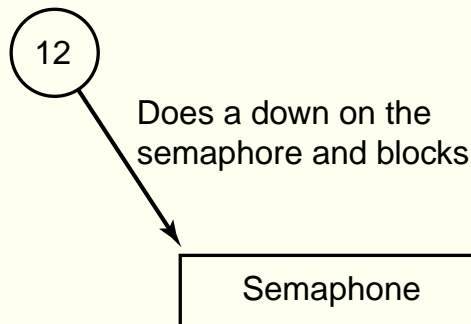
Prioridades para Escalonamento

		Win32 process class priorities					
Win32 thread priorities		Realtime	High	Above Normal	Normal	Below Normal	Idle
	Time critical	31	15	15	15	15	15
	Highest	26	15	12	10	8	6
	Above normal	25	14	11	9	7	5
	Normal	24	13	10	8	6	4
	Below normal	23	12	9	7	5	3
	Lowest	22	11	8	6	4	2
	Idle	16	1	1	1	1	1

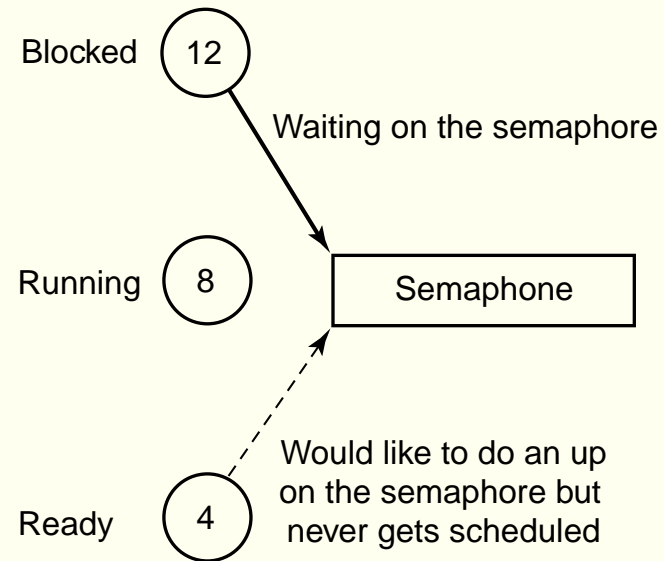
Filas de prioridades



Inversão de prioridades

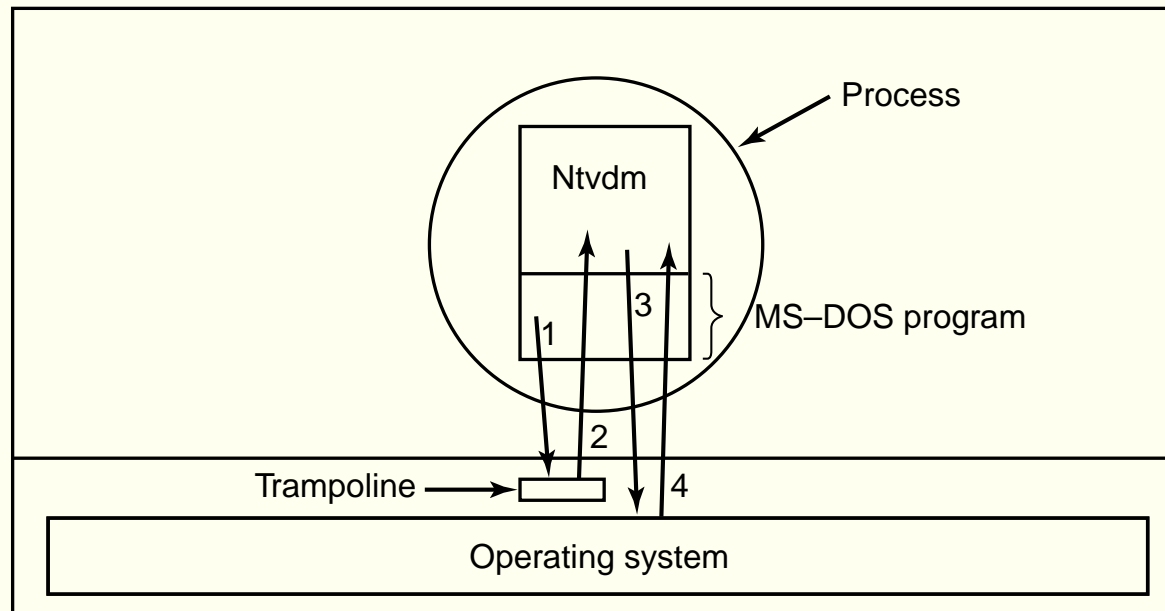


(a)

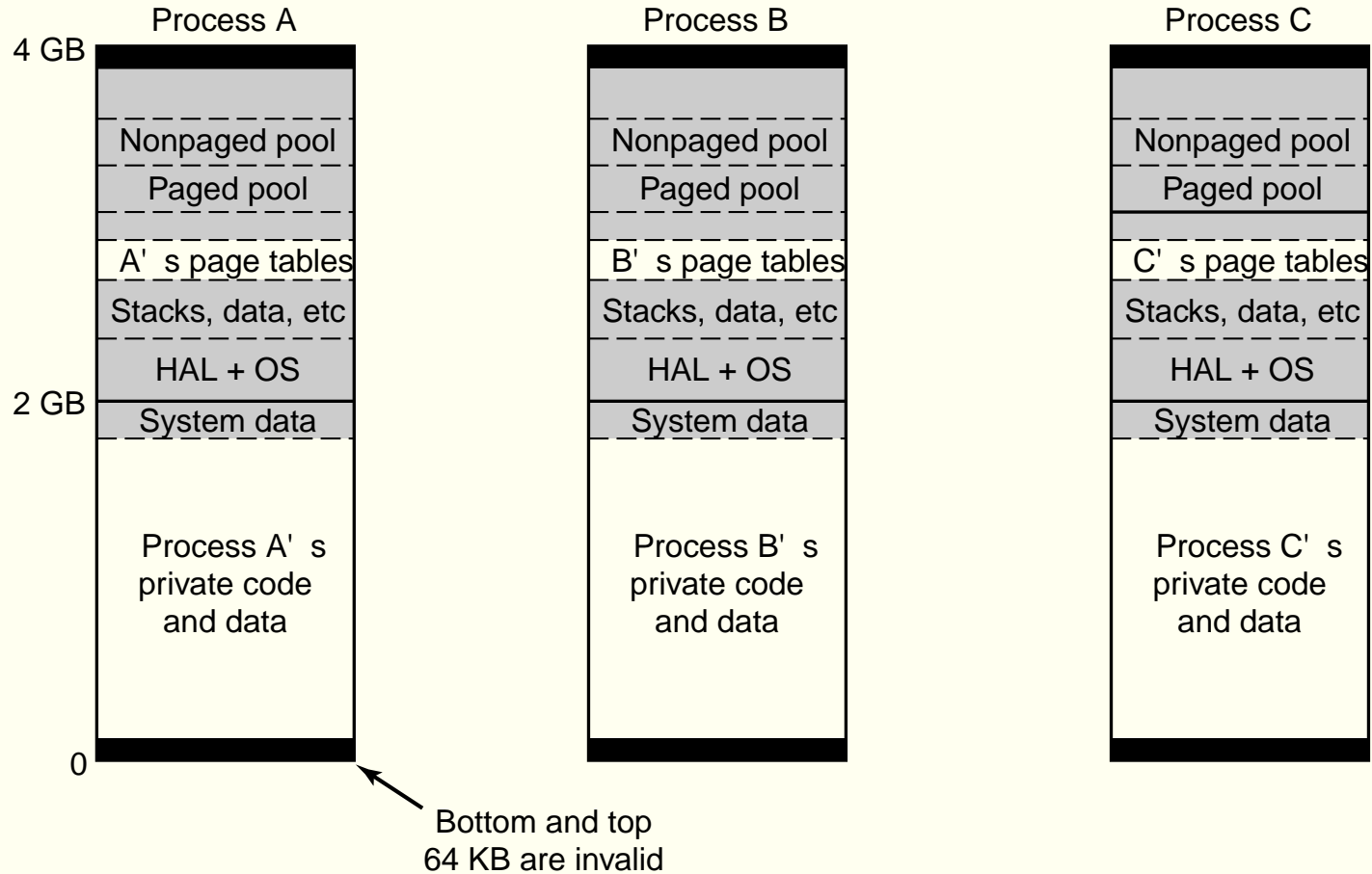


(b)

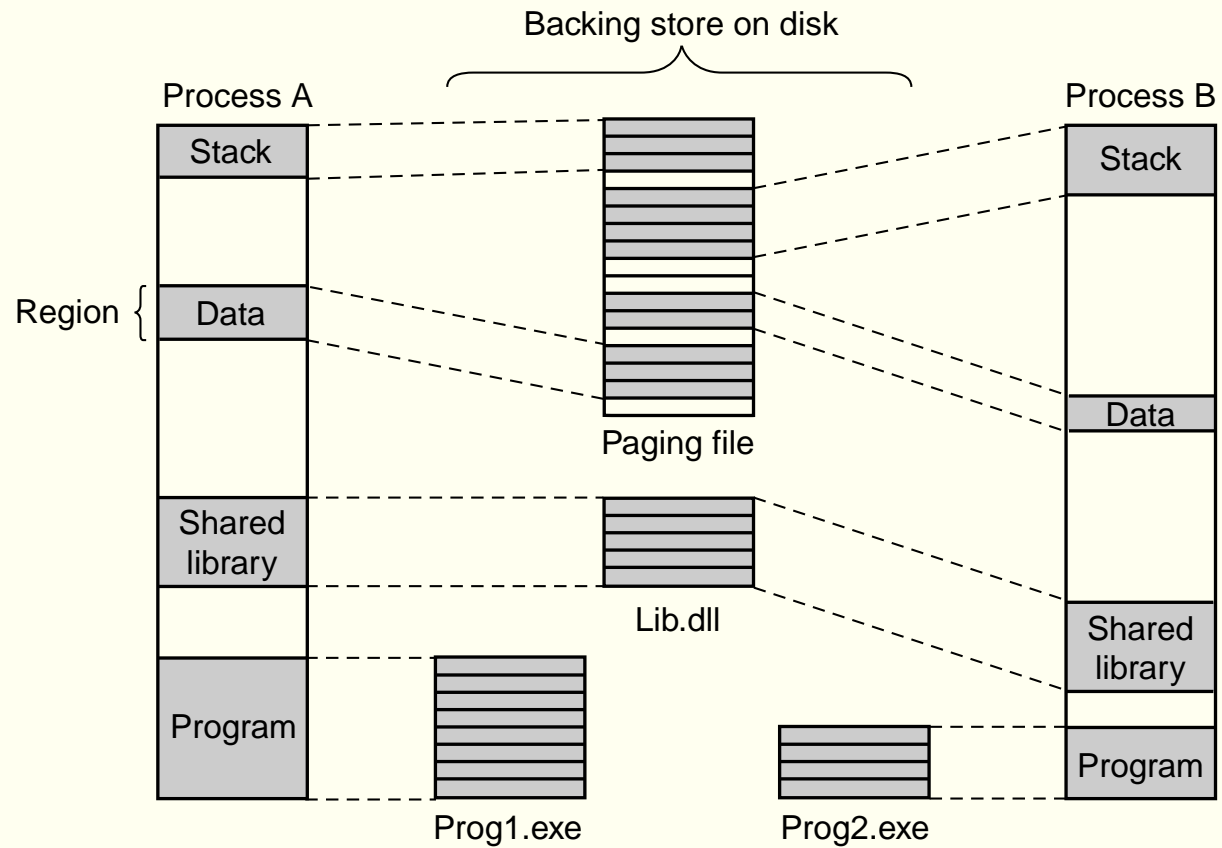
Emulação do MS-DOS



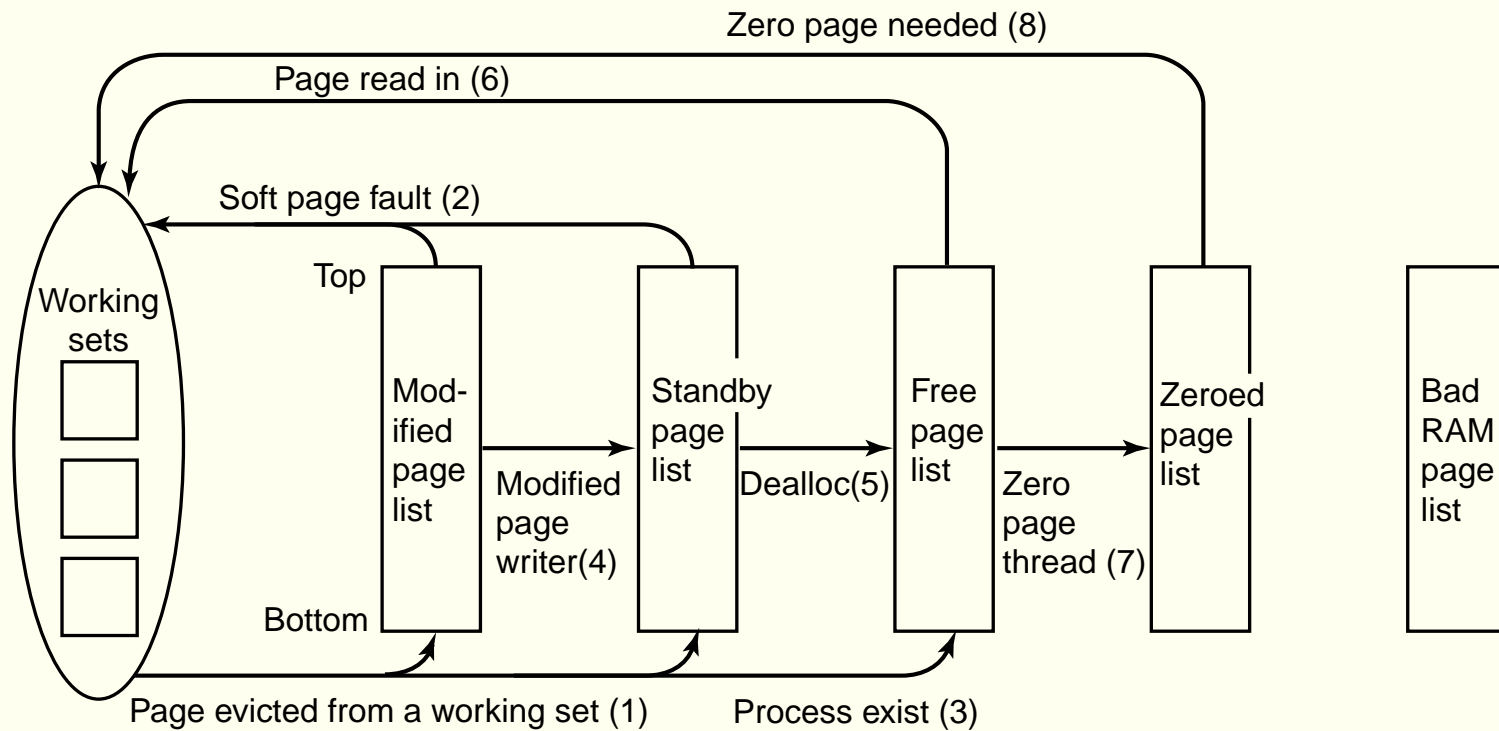
Gerência de memória



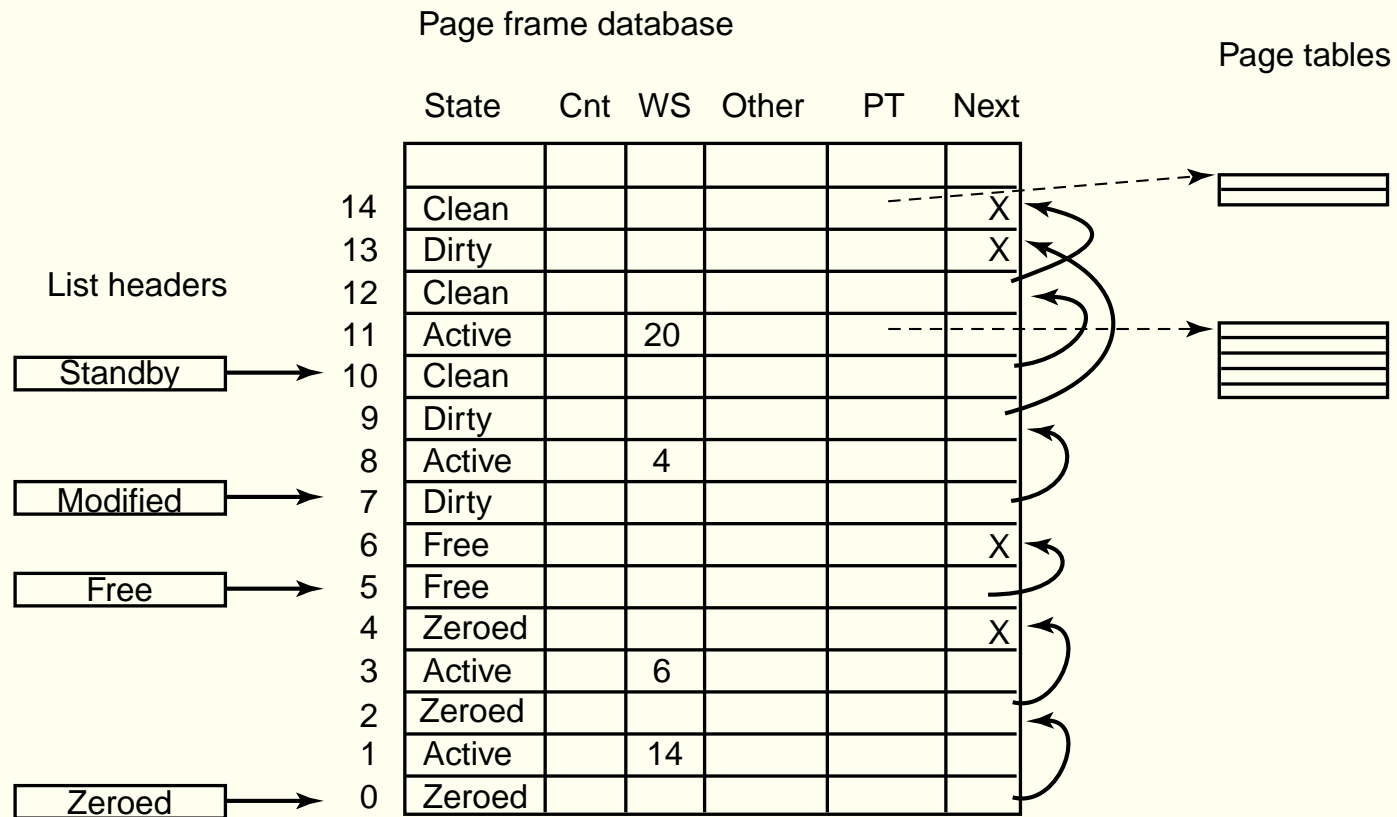
Mapeamento de páginas



Gerência de páginas



Gerência de páginas



Mais informações sobre o Windows

- <http://en.wikipedia.org/wiki/Windows>
- <http://www.microsoft.com/>