10
Windows 2000

O livro *Sistemas Operacionais* da Série didática do Instituto de Informática da UFRGS cobre os tópicos tipicamente abordados em uma disciplina de graduação de Sistemas Operacionais, além de usar as implementações do Linux e do Windows 2000 para ilustrar a teoria apresentada. O livro segue as Diretrizes Curriculares estabelecidas pelo MEC para cursos da área de Computação e Informática. O texto é de fácil leitura, possui cerca de 100 figuras e inclui exercícios no final de cada capítulo, podendo ser adotado como bibliografia básica da disciplina. O livro conta ainda com uma página *web* (http://www.inf.ufrgs.br/~asc/livro) onde o leitor encontrará a sua disposição o material didático de apoio a este livro na forma de transparências.

O capítulo 10, introduzido na segunda edição, apresenta o sistema operacional **Windows 2000** (NT 5.0). Nesse capítulo, são discutimos detalhes da arquitetura do Windows 2000 seguindo, na medida do possível, a mesma ordem de apresentação do capítulo sobre Linux (capítulo 9) para que o leitor possa ter uma visão comparativa entre as soluções adotadas por um e por outro sistema na implementação de seus mecanismos básicos.

O Windows NT é um sistema operacional proprietário, desenvolvido pela Microsoft, que surgiu com o objetivo ser um sistema operacional da década de 90, atento aos novos avanços tecnológicos e às exigências do mercado. Desde o seu lançamento, em 1993, com a versão 3.1, o Windows NT teve a preocupação de fornecer suporte a ambientes de rede e distribuídos. Sua evolução, até chegar ao Windows NT 5.0, comercialmente conhecido como Windows 2000, foi orientada à interoperabilidade com outros sistemas operacionais. A estrutura básica da arquitetura Windows NT manteve-se praticamente inalterada desde o seu lançamento até as versões mais recentes. Neste capítulo, nós apresentaremos detalhes da arquitetura do Windows 2000 seguindo, na medida do possível, a mesma ordem de apresentação do capítulo sobre Linux para que o leitor possa ter uma visão comparativa entre esses dois sistemas. Nós concluíremos este capítulo com uma breve apresentação do sucessor do Windows 2000: Windows XP.

10.1 Introdução: um pouco de história

A história do Windows NT iniciou com o desejo da Microsoft de criar um sistema operacional que explorasse as inovações tecnológicas apresentadas pelos processadores no final da década de 80, início da década de 90. O objetivo era desenvolver um sistema operacional multitarefa para ser utilizado tanto em ambientes monousuário como multiusuário. O nome *Windows* é originário de um sistema de janelas (*Windows 3.x for Workgroup*) projetado para competir com a interface usuário dos computadores Macintosh (Apple). Esse ambiente de janelas emprestou a sua "aparência" para a primeira versão do Windows NT. A sigla NT vem de *New Technology*, e foi criada para caracterizar a nova filosofia que orientou a sua concepção.

A primeira versão do Windows NT (versão 3.1) foi lançada em 1993 e constituiu o primeiro sistema operacional de 32 bits da Microsoft. Esse sistema operacional caracterizava-se por fornecer uma compatibilidade com o sistema operacional MS-DOS, com aplicações desenvolvidas para o "velho" sistema de janelas (*Windows 3.x for Workgroup*) e com o sistema operacional OS/2. Na realidade, o sistema OS/2 foi, em conjunto com a IBM, o primeiro esforço da Microsoft em desenvolver um sistema operacional multitarefa em 32 bits. O OS/2, apesar de possuir em sua concepção uma série de inovações e de soluções inteligentes, jamais vingou como sistema operacional por ser demasiado "pesado" para ser executado nas máquinas existentes na época. A experiência de desenvolvimento do OS/2 influenciou fortemente a concepção do Windows NT.

Após sucessivas versões do Windows NT 3.x, nasce o Windows NT 4.0. Em relação à arquitetura interna de sistema operacional, o NT 4.0 mantém essencialmente a mesma de seu predecessor (Windows NT 3.x). As principais modificações em relação ao Windows NT 3.x estão na interface gráfica, que agora se assemelha à do Windows 98, e na migração de vários serviços, também relacionados com a parte gráfica, do subsistema Win32 para o núcleo do Windows NT 4.0.

Em 1999, a Microsoft lançou uma nova versão do Windows NT, a 5.0, que comercialmente recebeu o nome de Windows 2000. A estrutura básica do sistema operacional é a mesma do NT 4.0. A principal diferença está na inclusão de serviços orientados a ambientes distribuídos e de rede. Na realidade, dependendo das funcionalidades adicionadas ao Windows 2000, existem 4 diferentes versões desse sistema operacional:

- Windows 2000 Professional, o qual substitui o NT workstation, isto é, as máquinas empregadas como ponto de trabalho (máquina cliente).
- Windows 2000 Server, equivalente ao NT Server. Essa configuração apresenta alguns serviços orientados ao compartilhamento de recursos e destina-se, como o próprio nome induz, a máquinas servidoras em uma rede NT.
- Windows 2000 Advanced Server, que fornece uma série de facilidades para ambientes de rede e distribuídos, incluindo o conceito de *clustering* (seção 10.9) e suporte ao balanceamento de carga.
- Windows 2000 Datacenter Server, que agrega todas as funcionalidades disponíveis no Windows 2000 e suporta o endereçamento de até 64 GB.

No momento em que preparávamos esta edição, a Microsoft estava prestes a lançar seu novo produto: o Windows *eXPerience*, comercialmente denominado de Windows XP. O Windows XP consiste na geração seguinte da família Windows, e na seção 10.10, nós faremos uma breve apresentação sua.

10.2 Diretrizes de projeto

O desenvolvimento do Windows 2000 foi orientado por cinco objetivos principais que sempre nortearam o projeto de todos os produtos da família Windows NT: confiabilidade e robustez; extensibilidade e facilidade de manutenção; portabilidade; desempenho; e, por último, conformidade com o padrão POSIX e certificação C2.

O objetivo de confiabilidade e robustez traduz-se no fato de que um sistema deve ter a capacidade de se proteger do mau funcionamento e de problemas oriundos do próprio sistema operacional, assim como de fontes externas (ataques). Para atingir essa meta, algumas diretrizes foram traçadas no desenvolvimento do Windows 2000. Inicialmente, o sistema foi primeiro concebido e documentado para somente após começar a ser codificado. Nesse procedimento inicial, elaborou-se uma interface clara e bem definida para os serviços do núcleo com o intuito de evitar o uso de parâmetros "mágicos" e *flags* para a chamada desses serviços. Essa decisão forçou a elaboração de uma documentação clara da interface de serviços, o que facilitou a fase de testes. Componentes importantes do Windows 2000, como, por exemplo, Win32, OS/2, e POSIX, foram isolados em subsistemas, cada um com uma interface de serviços bem definida e com tratamentos de exceção próprios, isolados a esse subsistema. Essa abordagem, na realidade, dividiu a complexidade do sistema em dois grandes grupos: serviços do núcleo e serviços de subsistemas. Um erro em um subsistema afeta apenas o comportamento deste, não do sistema operacional como um todo.

O segundo objetivo de projeto, extensibilidade e facilidade de manutenção, diz respeito à perenidade do sistema. Era necessário que o Windows 2000 soubesse evoluir, adaptando-se facilmente a novas necessidades, tanto de hardware como de software. O suporte a ambientes distribuídos e a filosofia de subsistemas são pontos importantes para o cumprimento dessa meta. A estruturação em subsistemas permite isolar alterações a apenas uma parte do sistema operacional; se, por exemplo, a norma POSIX for alterada, apenas o subsistema POSIX é diretamente afetado. Essa concepção modular facilita a adição de novos subsistemas ao Windows 2000. O "isolamento" do subsistema através de uma interface de serviços reduz o risco de, ao modificar o sistema operacional, serem introduzidos efeitos colaterais.

Os ambientes de computação distribuída caracterizam-se pelo fato de potencialmente empregarem plataformas heterogêneas. A portabilidade de um sistema torna-se então um objetivo importante para um sistema operacional que almeja ser empregado em tais ambientes. Por portabilidade, entende-se a facilidade que um sistema apresenta em ser recompilado e executado em diferentes plataformas de hardware com o mínimo possível de alterações em seu código.

Os objetivos de confiabilidade e robustez, extensibilidade e facilidade de manutenção, e o de portabilidade levam, de uma certa forma, a uma programação de forma modular, fazendo com que diferentes módulos executem diferentes tarefas disponibilizadas na forma de uma interface de programação (API). Esse modelo de concepção de sistemas normalmente se traduz em sobrecustos (*overheads*) na realização de tarefas do sistema. O desempenho surge, então, como outro objetivo desejável. O projeto do Windows 2000 manifesta essa preocupação em dois aspectos. O primeiro traduz-se em redefinir estruturas de dados e algoritmos de forma a otimizá-los o máximo possível. O segundo aspecto implica que a definição de algoritmos e estruturas de dados não vá contra a filosofia modular de subsistemas e do uso de interfaces claras e bem definidas para os serviços. É necessário definir soluções que contemplem, ao mesmo tempo, a filosofia modular de subsistema e o desempenho, encontrando assim uma boa relação custo-benefício.

O quinto objetivo do Windows 2000 é, na realidade, composto por duas partes: conformidade POSIX e certificação C2. A primeira, conformidade com a norma POSIX, permite que aplicações desenvolvidas para executar em ambientes UNIX sejam facilmente portadas para Windows 2000 e vice-versa. Esse objetivo envolve, de certa forma, projetar as interfaces de serviços (chamadas de

sistema) para ter um aspecto UNIX-*like*. Essa decisão tem um impacto importante se considerarmos a série de ferramentas UNIX voltadas a ambientes distribuídos e de rede, além, é claro, de reduzir o número de "versões'' de interfaces de programação disponibilizadas por diferentes sistemas operacionais. Outro ponto não negligenciável é o fato de que, cada vez mais, licitações nos Estados Unidos impõem como condição o fato de o sistema operacional possuir conformidade POSIX. Aqui surge também a segunda parte, a certificação C2. Essa certificação diz respeito a normas de segurança, facilidades de auditoria, deteção de ataques, controle de quotas e de acesso a recursos do sistema. Novamente, todas as licitações americanas estão exigindo essa certificação. Então para atender esse requisito de mercado, cada vez mais comum no mundo inteiro, o projeto do Windows 2000 foi orientado a possuir conformidade POSIX e a oferecer os serviços necessários à obtenção da certificação C2.

10.3 Arquitetura do Windows 2000: visão geral

Um sistema operacional é um *software* extremamente complexo. Assim, vários modelos de arquiteturas foram propostos para melhor organizar os detalhes de sua implementação. Esses modelos vão desde sistemas baseados em *kernel* monolítico até sistemas totalmente moduláveis, baseados em micronúcleo (*microkernel*).

A arquitetura do Windows 2000 é fortemente inspirada no princípio de micronúcleo. Assim, cada funcionalidade do sistema é oferecida e gerenciada por um único componente do sistema operacional. Os demais componentes do sistema operacional e todas as aplicações acessam os serviços providos por um determinado componente através de uma interface bem definida. Teoricamente, cada módulo (componente) pode ser removido, atualizado, ou substituído sem necessitar de alterações nais demais partes do sistema. O Windows 2000 não é puramente orientado à filosofia micronúcleo porque módulos fora do micronúcleo executam operações em modo protegido (modo kernel). A justificativa para essa decisão de projeto está no desempenho. Em uma abordagem orientada a micronúcleo "pura", uma aplicação que necessite executar uma operação privilegiada deve solicitar esse serviço ao micronúcleo. Esse procedimento envolve uma série de trocas de contexto. No Windows 2000, para evitar essa troca de contexto, certos subsistemas (módulos ou componentes) passam de modo usuário para modo protegido e implementam diretamente a função desejada, evitando assim a passsagem pelo núcleo e as trocas de contexto que isso implica.

O Windows 2000 segue também uma organização em camadas. Nessa abordagem, o sistema operacional é dividido em módulos que são dispostos uns sobre os outros em camadas. Cada camada oferece um conjunto de serviços à camada superior e só pode utilizar serviços fornecidos pela camada imediatamente inferior. Outro conceito explorado pelo Windows 2000 é o modelo orientado a objetos. Nesse modelo, recursos do sistema, arquivos, memória e dispositivos físicos, são implementados por objetos e manipulados através de métodos (serviços) associados a esses objetos.

O Windows 2000 foi projetado de forma a permitir a execução de aplicações escritas para outros sistemas operacionais. Essa facilidade é suportada a partir de subsistemas que, implementados como um processo separado, fornecem um ambiente de execução compatível a um determinado sistema operacional. Esse ambiente é composto, além de uma interface gráfica e de um interpretador de comandos, por uma interface de programação (API) compatível com os serviços (chamadas de sistema) do sistema operacional que o subsistema implementa. Isso implica que uma aplicação escrita para um sistema operacional particular pode executar sem alterações no Windows 2000 por "enxergar" as mesmas funções existentes no sistema nativo para o qual foi escrito. O mais importante dos subsistemas do Windows 2000 é o Win32, que possibilita que aplicações escritas

para outros sistemas operacionais Microsoft executem no Windows 2000 sem problemas. Outros subsistemas disponíveis são o subsistema OS/2 e o subsistema POSIX.

A estrutura do Windows 2000 pode ser dividida em duas partes: modo usuário (onde estão localizados os subsistemas protegidos) e modo kernel (o executivo). Os subsistemas protegidos são assim denominados porque residem em processos separados cuja memória é protegida do acesso de outros processos. Os subsistemas interagem entre si através de um mecanismo de troca de mensagens (*Local Procedure Call - LPC*). No modo kernel, rodam os componentes do sistema operacional que necessitam de desempenho e por isso interagem com o hardware e um com o outro sem estarem sujeitos a trocas de contexto e de modo. Todos os componentes estão protegidos das aplicações porque estas não possuem acesso à parte protegida do sistema operacional. Ainda, cada componente está protegido um do outro devido à adoção da orientação a objetos. Todo acesso a um objeto é feito através de um método.

O modo kernel é estruturado em três grandes módulos funcionais: *hardware abstraction layer*, drivers de dispositivos e o executivo. A camada denominada de *hardware abstraction layer* (HAL) é um módulo carregável do núcleo. Esse módulo respeita uma interface padrão de serviços, porém possui uma implementação específica para o hardware no qual o Windows 2000 está executando. Todas as funcionalidades que são dependentes de um determinado hardware, como interfaces de E/S, controladores de dispositivos e de interrupções, ou ainda, o próprio processador, são implementadas dentro desse módulo. Esse tipo de projeto permite que todos os componentes do sistema operacional acima do módulo HAL executem de forma independente do hardware, fornecendo assim o grau de portabilidade necessário a um sistema que visa a operar em ambientes heterogêneos.

Os drivers de dispositivos são outra categoria de módulos carregáveis do núcleo. Esses drivers oferecem, dentro do executivo do Windows 2000, uma interface entre o sistema de E/S e o HAL. O gerenciamento dos drivers de dispositivos foi um dos aspectos em que o Windows 2000 (NT 5.0) apresentou uma evolução significativa em relação a sua versão anterior (Windows NT 4.0) através da integração de suporte a *plug-and-play*.

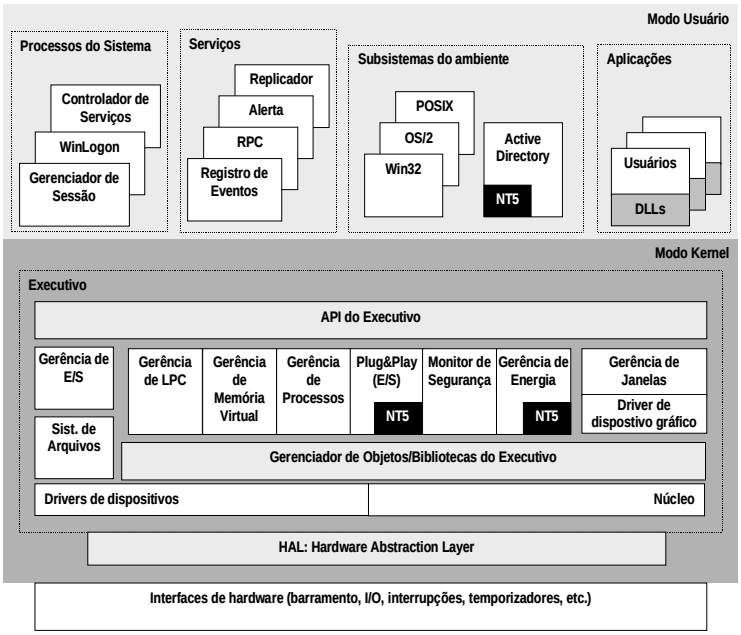

Figura 10.1 – Componentes da estrutura do Windows 2000

O executivo constitui o núcleo do sistema operacional Windows 2000. É ele que implementa os serviços básicos do Windows 2000, exportando funções para serem utilizadas em modo usuário e funções que só são acessíveis por componentes (módulos) pertencentes ao próprio núcleo. Os principais componentes do executivo são (Figura 10.1):

- **Gerência de objetos**: é o componente responsável por criar, gerenciar e excluir objetos do Executivo Windows 2000. Entende-se por objetos do executivo a abstração de todos os tipos de dados utilizados para representar recursos do sistema operacional como processos, *threads*, alocação de memória, mecanismos básicos de sincronização (mutex e semáforos), etc.

- **Gerência de processos e *threads***: responsável por criar, encerrar, suspender e dar prosseguimento à execução de *threads* e processos. Ainda, armazena e recupera informações sobre os processos e *threads* do Windows 2000. A forma de tratamento de *threads* e processos do Windows 2000 será detalhada na seção 10.4.

- **Gerência de memória virtual**: módulo responsável pela implementação do suporte a memória virtual e do gerenciamento de outras atividades relacionadas à gerência de memória como proteção, cache, mapeamentos, etc.

- **Monitor de segurança**: faz cumprir as políticas de segurança no computador local. Verifica acesso aos recursos do sistema operacional, protegendo e auditando os objetos durante sua execução.

- **Módulo de suporte a *Local Procedure Call* (LPC)**: módulo responsável pela comunicação por troca de mensagens entre processos. Esse mecanismo é basicamente uma versão otimizada do conceito de *Remote Procedure Call* (RPC).

- **Gerência de E/S:** compreende um grupo de componentes responsáveis pelo processamento de informações de entrada e por emitir saída para uma grande variedade de dispositivos. A gerência de E/S fornece uma interface padrão para o executivo de forma independente do tipo de dispositivo de E/S. As solicitações de E/S são traduzidas para os dispositivos específicos de *hardware* através da utilização dos drivers de dispositivos. Outros elementos relacionados a esse componente são o sistema de arquivos, o gerenciador de cache e o driver de rede.

Para concluir esta seção, resta comentar que o núcleo do Windows 2000 foi projetado de forma a dar suporte a multiprocessamento simétrico quando executado em máquinas multiprocessadoras. Essa abordagem distingue-se do multiprocessamento assimétrico. No multiprocessamento assimétrico, na presença de *n* processadores, um processador é pré-selecionado e dedicado à execução do sistema operacional, deixando para os processos de usuários os *n-1* processadores restantes. Já no multiprocessamento simétrico, o sistema operacional pode ser executado em qualquer processador que esteja livre, ou ainda em todos os processadores simultaneamente, explorando melhor o potencial dos vários processadores existentes. Essa possibilidade tem impacto importante na concepção dos serviços do sistema operacional e deve ser considerada no momento do projeto do sistema.

10.4 Processos e *threads*

Cada sistema operacional tem a sua própria forma de implementar processos; as variações estão nas estruturas de dados utilizadas para representar fluxos de execução, sua denominação, como são protegidos uns em relação aos outros e na forma de inter-relacionamento. O Windows 2000 implementa o conceito de processo a partir de dois objetos: objeto processo e objeto *thread*. O objeto processo é a entidade que corresponde a recursos do sistema tais como memória, arquivos, etc. O objeto *thread*, por sua vez, constitui uma unidade de trabalho que é executada de forma seqüencial e podendo ser interrompida em qualquer ponto.

A criação de um processo em Windows 2000 corresponde a instanciar (criar) um objeto do tipo processo, o qual é uma espécie de "molde" para novos processos. Nesse momento, uma série de atributos são inicializados para esse novo processo, como, por exemplo, um identificador de processo (*pid*), descritores de proteção, prioridades, quotas, etc. A unidade de escalonamento do Windows 2000 é o conceito de *thread*. A cada processo está associada, no mínimo, uma *thread*. Cada *thread* pode criar outras *threads*. Essa organização permite a execução concorrente dos processos, além de possibilitar uma concorrência entre as *threads* que pertencem a um mesmo processo. Uma *thread* pode estar em um de seis estados (Figura 10.2):

- **Apto** (*Ready*): corresponde ao estado no qual se encontram as *threads* aptas a executar, ou seja, as *threads* que o escalonador considera para selecionar a próxima a ser executada. Uma vez selecionada, a *thread* passa ao estado ativo (*standby*).

- **Ativa** (*Standby*): estado intermediário no qual a *thread* selecionada pelo escalonador espera pelo chaveamento de contexto para entrar efetivamente em execução. No sistema existe, por processador, apenas uma *thread* nesse estado.

- **Em execução** (*running*): estado que assume uma *thread* quando está ocupando o processador. Uma *thread* em *running* executa até que ela seja preemptada por uma *thread* de mais alta prioridade, esgote a sua fatia de tempo, realize uma operação bloqueante, ou termine. Nos dois primeiros casos, o descritor da *thread* é reinserido na lista de aptos (estado *ready*).

- **Espera** (*waiting*): uma *thread* passa a esse estado sempre que (1) for bloqueada pela espera da ocorrência de um evento (e.g. E/S); (2) realizar uma primitiva de sincronização; ou (3) quando um subsistema ordena a suspensão da *thread*. Quando a condição de espera é satisfeita, a *thread* é inserida na lista de aptos.
- **Transição** (*transition*): corresponde ao estado em que uma *thread* está apta a ser executada, porém os recursos de sistema necessários a sua execução (e.g. estar paginada em memória), ainda não estão disponíveis. Quando esses recursos estão disponibilizados, a *thread* passa ao estado apto.
- **Término** (*terminated*): estado que uma *thread* assume quando atinge seu final, ou é terminada por uma outra *thread*, ou ainda quando o processo a que está associada termina.

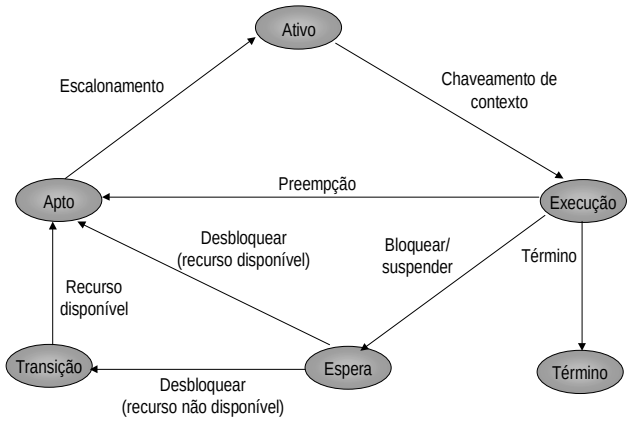

Figura 10.2 – Diagrama de estados de *threads* no Windows 2000

As *threads* de qualquer processo, inclusive as do executivo do Windows 2000, podem, em máquinas multiprocessadoras, ser executadas em qualquer processador. Dessa forma, o escalonador do Windows 2000 atribui uma *thread* pronta a executar (apta) para o próximo processador disponível. Múltiplas *threads* de um mesmo processo podem estar em execução simultaneamente.

O escalonador do Windows 2000 é preemptivo com prioridades. As prioridades são organizadas em duas classes: tempo real e variável. Cada classe possui 16 níveis de prioridades, sendo que as *threads* da classe tempo real têm precedência sobre as *threads* da classe variável, isto é, sempre que não houver processador disponível, uma *thread* de classe variável é preemptada em favor de uma *thread* da classe tempo real. Todas as *threads* prontas para executar são mantidas em estruturas de filas associadas a prioridades em cada uma das classes. Cada fila é atendida por uma política *Round-robin*.

A atribuição de prioridades a *threads* é diferente para cada uma das classes. Enquanto na classe de tempo real, as *threads* possuem prioridade fixa, determinada no momento de sua criação, as *threads* da classe variável têm suas prioridades atribuídas de forma dinâmica (por isso o nome de "variável" para essa classe). Dessa forma, uma *thread* de tempo real, quando criada, recebe uma prioridade e será sempre inserida na fila dessa prioridade, ao passo que uma *thread* da classe variável poderá migrar entre as diferentes filas de prioridades. Em outros termos, o Windows 2000 implementa um esquema de múltiplas filas para as *threads* da classe real e múltiplas filas com realimentação para as *threads* da classe variável. A Figura 10.3 ilustra o sistema de prioridades do Windows 2000.

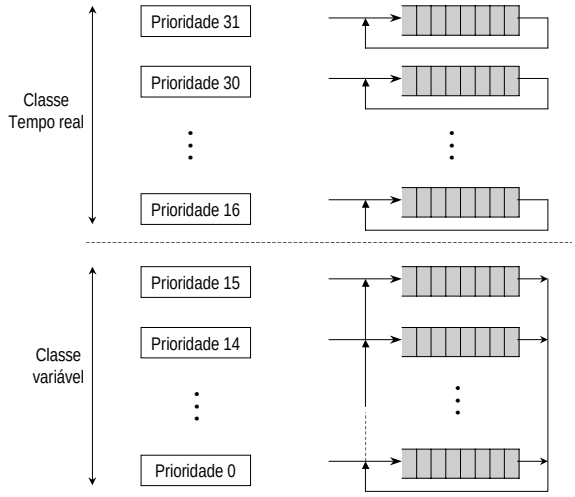

Figura 10.3 – O esquema de prioridade do Windows 2000

Na classe variável, a prioridade de uma *thread* é estabelecida a partir de dois parâmetros, um vínculado à própria *thread* e outro, ao processo a que pertence. Um objeto processo, durante sua criação, recebe um valor, entre zero e 15 inclusive, para sua prioridade de base. Cada *thread* recebe uma prioridade inicial, variando 2 unidades acima ou abaixo da prioridade de base do processo, que indica sua prioridade relativa dentro desse processo. A prioridade de uma *thread* varia durante a sua vida, mas nunca assumirá valores inferiores a sua prioridade base nem superiores a 15. O critério empregado para variar a prioridade de uma *thread* é o tempo de utilização do processador. Se a *thread* for preemptada por ter executado durante todo o quantum de tempo que lhe foi atribuído, o escalonador do Windows 2000 diminui sua prioridade; caso contrário, sua prioridade é aumentada.

Em outros termos, o escalonador do Windows 2000 tende a atribuir prioridades mais elevadas para *threads* do tipo *I/O bound*.

Em máquinas monoprocessadoras, a *thread* de mais alta prioridade está sempre ativa a menos que esteja bloqueada esperando por um evento (E/S ou sincronização). Caso exista mais de uma *thread* com um mesmo nível de prioridade o processador é compartilhado de forma *round-robin* entre essas *threads*. Em um sistema multiprocessador com *n* processadores, as *threads* de mais alta prioridade executam nos *n-1* processadores extras. As *threads* de mais baixa prioridade disputam o processador restante.

Na realidade, existe ainda um fator adicional que influencia fortemente o escalonamento do Windows 2000: a afinidade de uma *thread*. Uma *thread* pode definir sobre qual (ou quais) processador(es) ela deseja executar. Nesse caso, se a *thread* estiver apta a executar, porém o processador não estiver disponível, a *thread* é forçada a esperar, e uma outra *thread* é escalonada em seu lugar. O conceito de afinidade é motivado pela tentativa de reaproveitar os dados armazenados na cache do processador pela execução anterior de uma *thread*. O Windows 2000 possibilita dois tipos de afinidade: *soft* e *hard*. Por *default*, a política de afinidade *soft* é utilizada para escalonar *threads* a processadores. Nesse caso, o *dispatcher* tenta alocar uma *thread* ao mesmo processador que ela executou anteriormente, porém, se isso, não for possível, a *thread* poderá ser alocada a outro processador. Já com a política de afinidade *hard*, uma *thread* (com os devidos privilégios) executa em apenas um determinado processador.

10.5 Gerência de memória

O Windows 2000 implementa um sistema de memória virtual baseado em um espaço de endereçamento linear (plano) de 32 bits, o que fornece até 4 Gbytes de memória virtual. Esse espaço de endereçamento é normalmente dividido em duas partes de igual tamanho (2 Gbytes): uma destinada ao processo usuário (parte inferior) e outra parte destinada ao sistema operacional (parte superior). Em outros termos, uma aplicação (processo usuário) possui, no máximo, um tamanho de 2 Gbytes. Na realidade, o Windows 2000 oferece a opção de modificar essa alocação inicial de forma a atribuir 3 Gbytes ao processo usuário e 1 Gbyte ao sistema operacional. Essa possibilidade permite que certas aplicações, como, por exemplo, banco de dados, armazenem uma grande parcela de dados dentro do espaço de endereçamento da própria aplicação (processo). O Windows 2000 prevê ainda uma extensão, denominada de VLM (*Very Large Memory*) - destinada aos processadores de arquitetura de 64 bits - que permite a um processo usuário alocar até 28 Gbytes de memória virtual suplementar.

A alocação de memória por um processo Windows é realizada em duas fases. Inicialmente, o processo reserva um certo número de páginas da memória virtual sem necessariamente utilizá-las. Em seguida, à medida que o processo necessita de memória, essas páginas pré-alocadas são mapeadas a áreas efetivas de armazenamento no disco (área de *swap*). Essa segunda fase é conhecida, em terminologia Windows, como *commit*. Dessa forma, as páginas relativas ao espaço de endereçamento total de um processo usuário (2 Gbytes) podem estar em um de três estados: livres, reservadas ou dedicadas (*commited*). As páginas livres são as páginas do espaço de endereçamento não utilizadas pelo processo. As páginas reservadas correspondem àquelas pré-alocadas mas ainda não mapeadas a uma área de armazenamento real. Uma vez esse mapeamento efetuado, as páginas pré-alocadas tornam-se páginas dedicadas. A distinção entre páginas reservadas e páginas dedicadas é justificada pela redução do tamanho do arquivo de paginação (*swap*) necessário a um processo. Apenas as páginas dedicadas consomem área de armazenamento. Por questões de desempenho, o Windows permite também que um processo, possuindo os

privilégios necessários, bloqueie páginas em memória, fazendo com que essas páginas nunca sofram um procedimento de *swapping*.

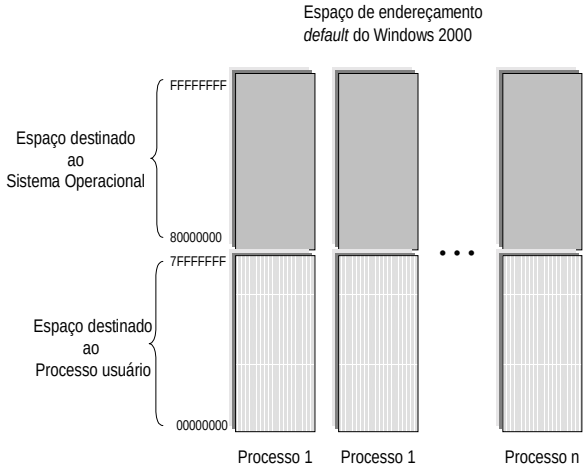

Figura 10.4 – *Layout* do espaço de endereçamento virtual do Windows 2000

Como a concepção do Windows é toda orientada a objetos, a memória alocada por um processo é representada através de um objeto memória. Dois processos podem compartilhar um mesmo espaço de endereçamento referenciando um objeto memória comum. Para o caso de compartilhamento de apenas uma região da memória o Windows oferece a abstração de visão (*view*). Essa abstração consiste em um processo mapear uma porção de seu espaço de endereçamento a um objeto (*section object*) o qual é utilizado por outros processos para acessos compartilhados a essa região. O mecanismo de *view* é bastante flexível. Ele permite que, em caso de *swapping* de uma região compartilhada de memória, as páginas correspondentes a essa região sejam transferidas ou para a área de *swap* ou para um arquivo especial (*mapped file*). É possível ainda fixar um endereço virtual para essa região compartilhada, permitindo assim que ela resida sempre em um mesmo endereço (virtual) em todos os processos que a utilizam. Finalmente, a área de *view* pode ter diferentes tipos de acesso, como, por exemplo, apenas leitura, leitura e escrita, execução, etc.

10.5.1 Tradução de endereço virtual em endereço físico

A gerência de memória do Windows 2000 é baseada em paginação com um tamanho de páginas variando entre 4 Kbytes e 64 Kbytes, dependendo do processador. O mecanismo de tradução de um endereço virtual em um endereço físico é baseado em uma tabela de paginação em dois níveis. Nesse caso, considera-se que o endereço virtual de 32 bits é formado por três componentes: índice de diretório de páginas, índice da tabela de páginas, e índice de byte (deslocamento dentro da página).

A tradução de um endereço virtual em um endereço físico é realizada da seguinte forma. Inicialmente, utilizando a parte mais significativa do endereço virtual (índice de diretório de páginas), o diretório de páginas é acessado para determinar qual tabela de páginas está associada ao endereço virtual que se deseja traduzir. Uma vez definida a tabela de páginas, o índice de tabela de páginas é empregado para determinar a página correspondente a esse endereço virtual. A entrada da tabela de páginas fornece informações de controle e a localização em memória da página a ser

acessada (se presente). Finalmente, o índice de byte é somado ao endereço inicial da página em memória, resultando no endereço físico correspondente ao endereço virtual desejado. A Figura 10.5 apresenta a relação entre esses três valores e a forma pela qual eles são utilizados para mapear um endereço virtual em endereço físico.

Cada processo possui um único diretório de páginas para mapear a localização das tabelas de páginas pertencentes a esse processo. O diretório de páginas possui 1024 entradas, o que limita o número máximo de tabelas de páginas de um processo. As tabelas de páginas são criadas sob demanda; por conseqüência, normalmente, muitas das entradas do diretório de página não são válidas. Cada tabela de páginas possui, por sua vez, também 1024 entradas, o que, de forma análoga, limita a quantidade de páginas por tabela de páginas. O tamanho de cada entrada, tanto do diretório de páginas como da tabela de páginas, é de 4 bytes. Isso implica que cada uma dessas estruturas ocupe 4 Kbytes de memória, ou seja, exatamente uma página (lembre-se de que o tamanho mínimo de páginas é 4 Kbytes). Essas tabelas são armazenadas no espaço de endereçamento virtual do processo, na área destinada ao sistema operacional (2 Gbytes superiores), e sua localização exata é mantida no descritor de processo.

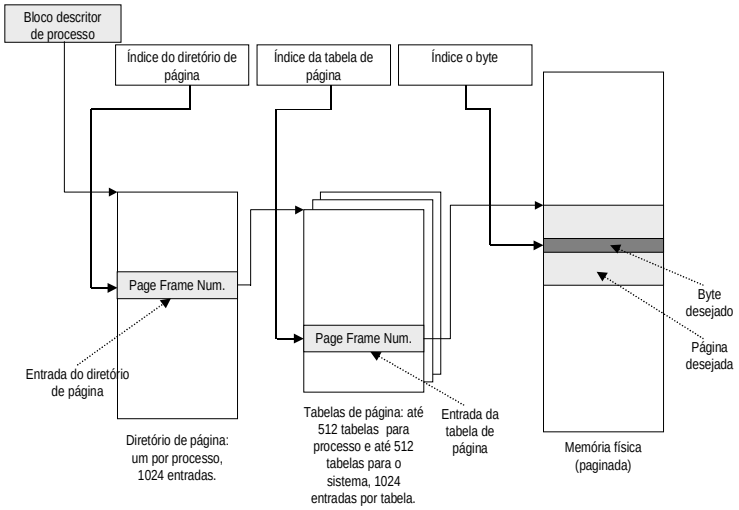

Figura 10.5 – Esquema de tradução de endereço virtual para endereço físico em arquiteturas Intel

10.5.2 Estratégias de paginação

Um sistema de gerência de memória baseado em paginação necessita determinar quando e como buscar páginas do disco (arquivo de paginação ou *swap*) para a memória. Além disso, é preciso definir, em caso de falta de espaço de memória, uma página a ser substituída para satisfazer a necessidade de carga de uma nova página. O algoritmo de paginação do Windows é baseado por demanda com *clustering* (em grupos). Nesse esquema, quando ocorre uma falta de página, o gerenciador de memória carrega na memória a página que faltava e mais um pequeno número de páginas ao seu redor. Essa estratégia tenta minimizar o número de acessos ao disco provocado pela paginação de um processo, explorando o princípio de localidade (um processo tende a executar, a todo instante, uma região limitada de seu espaço de endereçamento). A carga de páginas em avanço reduz o número de leituras individualizadas ao disco e, por conseqüência, o tempo de entrada e

saída. A quantidade de páginas lidas em avanço (*cluster*) difere para páginas de código e páginas de dados e ainda varia conforme o tamanho da memória física.

A política para substituição de páginas na memória empregada pelo Windows depende do tipo do arquitetura da máquina e de seu processador. Para arquiteturas do tipo multiprocessador baseadas em processadores da família Intel, e em todas as máquinas baseadas em processadores da família Alpha, a estratégia utilizada é essencialmente FIFO local. Nessa estratégia, considera-se para fins de seleção da página a ser substituída em memória, apenas as páginas pertencentes ao processo. Nas arquiteturas de monoprocessadores Intel, o algoritmo de seleção de página a ser substituída é LRU, implementado através do algoritmo do relógio (*clock*).

O número de páginas presentes em memória para um processo é mantido através do mecanismo de *working set*. Para cada processo, esse número varia entre um valor mínimo e um valor máximo definidos apenas a partir do tamanho da memória física. Quando ocorre uma falta de página, os limites do tamanho do *working set* e a quantidade de memória livre são examinados. Havendo memória livre disponível, a gerência de memória permite que o processo aumente seu *working set* até atingir o valor máximo (na realidade, o valor máximo pode vir a ser ultrapassado em função da quantidade de memória livre). Se um processo atingir o limite máximo de seu *working set*, e ainda necessitar da carga de páginas em memória, sem que haja memória disponível, a gerência de memória iniciará o mecanismo de substituição, considerando apenas as páginas do *working set* do processo. Caso o número de faltas de página seja elevado, a gerência de memória verifica o *working set* de todos os processos em memória. Os processos que tiverem mais páginas em memória que o valor mínimo terão seus *working sets* reduzidos, e suas áreas de memória liberadas serão disponibilizadas para alocação global, isto é, um processo que necessite de memória poderá requisitar esse espaço.

10.6 Sistemas de arquivos

O Windows 2000 possui um sistema de arquivos próprio, o NTFS (NT *File System*), projetado de forma a oferecer segurança de acesso, garantia da consistência de dados em presença de falhas e suporte a discos de grande capacidade. O Windows 2000 oferece ainda suporte a outros sistemas de arquivos como o FAT (MS-DOS e Windows 3.1), FAT32 (Windows 95, 98, Millenium), o HPFS (OS/2), além de formatos para cdrom (CDFS) e UDF (*Universal Disk Format*) para acesso a dados armazenados em DVDs. O grande diferencial do NTFS em relação aos seus predecessores da linha Microsoft está em cobrir as necessidades de alguns pontos considerados como críticos para aplicações em ambientes corporativos, a saber:

- **Facilidade de recuperação de dados e tolerância a falhas:** para alcançar a confiabilidade, que é requisito nas aplicações corporativas, uma das técnicas utilizada pelo Windows 2000 é o processamento de transações. Uma transação é definida como uma operação de E/S que altera os dados do sistema de arquivo, ou a estrutura de diretório do volume, de forma indivisível, isto é, cada alteração no sistema de arquivo só é considerada efetivada se for completamente realizada. O NTFS emprega esse modelo de transações para implementar seu recurso de recuperação do sistema de arquivos permitindo que este seja reconstruído, ou, ao menos, mantido em um estado consistente, após uma falha no sistema. Além disso, o Windows 2000 oferece suporte para RAID.

- **Segurança:** o NTFS explora o modelo de objetos para oferecer segurança aos arquivos. Um arquivo aberto é implementado como um objeto arquivo o qual atua como um descritor, definindo os diferentes privilégios de acesso e requisitos de segurança a este.

- **Suporte a grandes discos e arquivos:** o NTFS é projetado para suportar de forma eficiente acesso, manipulação e armazenamento em discos de grandes capacidades.
- **Fluxos de dados múltiplos:** um arquivo e seus atributos são vistos como uma seqüência de bytes, denominada fluxo de dados. Dessa forma, no NTFS, um único arquivo pode ter associado vários fluxos de dados. Essa organização oferece uma grande flexibilidade pois permite que o fluxo de dados que compõe o arquivo seja interpretado de acordo com o fluxo de dados de seu atributo.
- **Facilidades de indexação:** o NTFS permite que arquivos sejam acessados de forma indexada através de atributos (chaves de pesquisa) criados para cada arquivo.
- **Suporte para sistema POSIX:** o NTFS implementa os recursos exigidos pelo POSIX como diferenciação de maiúsculas e minúsculas para nomes de arquivos e diretórios e atalhos (*soft links*).

O NTFS é organizado sobre três estruturas básicas: setor, *cluster* e volume. O setor, normalmente composto de 512 bytes, é a menor unidade de alocação física do disco. Setores contíguos podem ser organizados em grupos, formando os *clusters*. Um *cluster* constitui então a unidade básica de alocação do NTFS, isto é, um arquivo ocupa em disco sempre um número de bytes múltiplo do tamanho do *cluster*. Um volume corresponde a uma partição lógica do disco. Um volume é composto por uma série de *clusters* e possui de forma autocontida informações relacionadas a esse disco lógico, isto é, estrutura de diretório, *clusters* livres e ocupados, etc. Atualmente, o número máximo de *clusters* permitido a um arquivo no NTFS é 2^{32}, sendo o tamanho máximo de um *cluster* limitado a 64 kbytes, o que nos leva a arquivos de até, no máximo, 2^{48} bytes.

Um volume (disco lógico ou partição) é organizado em 4 regiões. A primeira região de um volume NTFS corresponde ao setor de boot, que na realidade pode ocupar até 16 setores físico do disco, apesar do nome "setor de boot". O setor de boot possui informações sobre o *layout* do volume, a estrutura do sistema de arquivos e o programa de boot do Windows. Essa região é seguida pela *Master File Table* (MFT) a qual contém as informações sobre todos os arquivos e diretórios (*folders*) desse volume, assim como sobre o espaço livre. O MTF é organizado na forma de um conjunto de registros de tamanho variável, em que cada arquivo ou diretório possui um registro associado. Nesses registros, está incluído o próprio MTF, já que ele também não deixa de ser um tipo de arquivo. Cada registro do MTF mantém informações relacionadas ao atributo do arquivo (leitura, escrita, etc.), datas de criação e modificação, nome do arquivo, descritor de segurança. A terceira região é a dos arquivos de sistema, tipicamente de 1 Mbyte, em que são armazenados: uma cópia parcial da MFT (informações suficientes para recuperar erros físicos acontecidos na MFT); um arquivo de *logs*, referente ao controle de transações do NTFS; um *bit map* que fornece a ocupação dos *clusters* do volume; e uma tabela de atributos que define o tipo de acesso a esse volume (seqüencial, indexado, etc). Finalmente, a quarta e última região corresponde à área disponível para os arquivos.

Como nós mencionamos anteriormente, um dos principais objetivos do NTFS é facilitar a recuperação em caso de falhas. A capacidade de recuperação do NTFS é essencialmente baseada em *logs* de transação. Uma operação que altere o sistema de arquivos é tratada como uma transação, a qual é gravada na região de *logs* associada a cada volume. Apenas após a gravação da transação é que a operação é efetivada. É importante salientar que a capacidade de recuperação do NTFS foi projetada para garantir a coerência e a recuperação de estruturas e dados do sistema operacional e não dados de arquivos de usuário. Dessa forma, um usuário jamais perderá o acesso a um volume (partição) em decorrência de uma falha física, ou do sistema, embora possa perder acesso ao conteúdo de um arquivo seu. Entretanto, para suprir a necessidade de recuperação total, o Windows

disponibiliza uma série de ferramentas para a tolerância a falhas; entre elas, podemos citar o suporte a RAID.

10.7 Gerência de entrada e saída

O sistema de gerência de entrada e saída (E/S) do Windows 2000 é responsável pelos acessos ao sistema de arquivos, pelo gerenciamento da cache de dados do sistema operacional, pelos drivers de dispositivos e pelo driver de rede. O fato de o driver de rede ser visto de forma separada dos demais drivers de dispositivos é conseqüência do objetivo do Windows 2000 oferecer um suporte a ambientes distribuídos. O driver de rede realiza muito mais que o simples envio e recepção de pacotes de dados são de sua responsabilidade todas as facilidades de rede de o Windows 2000, o que justifica esse tipo de "tratamento especial".

As requisições de E/S são todas convertidas pela gerência de E/S do Windows 2000 em um formato padrão denominado de IRP (*I/O Request Packet*). Em seguida, o IRP é direcionado ao driver de dispositivo responsável pela operação a ser realizada. Quando a operação é finalizada, o driver de dispositivo sinaliza a gerência de E/S. Esse mecanismo oferece suporte para operações tanto assíncronas como síncronas.

10.7.1 A interface WDM

Plug-and-play é a capacidade que um sistema operacional possui de reconhecer e adaptar-se de forma dinâmica a alterações na sua configuração inicial de hardware. Através do *plug-and-play* um usuário pode inserir e remover periféricos ao seu sistema sem se preocupar com sua configuração, ou interferência com os demais componentes do sistema. A Microsoft segue a filosofia *plug-and-play* desde o sistema operacional Windows 95, porém esse mecanismo sofreu muitas evoluções. A mais importante evolução do mecanismo de *plug-and-play* foi a definição, e posteriormente a adoção, da especificação ACPI (*Advanced Configuration and Power Interface*) por parte de vários fabricantes de placas-mãe (*motherboard*). Basicamente essa especificação retira da BIOS o controle do *plug-and-play* e a gerência de energia para deixá-los a cargo do sistema operacional. As funcionalidades previstas pela especificação de ACPI são independentes de sistema operacional e de processador e possuem uma interface bem definida. Essa interface é utilizada pelos projetistas de sistemas operacionais na concepção de uma série de serviços para a gerência de entrada e saída.

No caso específico do Windows 2000, as melhorias introduzidas na funcionalidade de *plug-and-play* visam a simplificar o desenvolvimento de drivers de dispositivos e de sua gerência. Na prática, isso se traduz na unificação dos diferentes módulos "assistentes de instalação" (*wizards*) existentes para diferentes tipos de periféricos em um único "assistente de instalação". Essa abordagem baseia-se no emprego de um modelo genérico para implementação e gerência de drivers de dispositivos, o WDM (*Win32 Driver Model*). O modelo WDM é um padrão a ser seguido no desenvolvimeno de drivers de dispositivos de forma a permitir a fácil portabilidade desses de um sistema operacional a outro. Os drivers de dispositivos que seguem o WDM possibilitam uma compatibilidade a nível de código binário entre todas as plataformas baseadas em processadores x86 executando Windows 2000 e Windows 98, e são portáveis a nível do código fonte para qualquer outra arquitetura.

10.7.2 O suporte a RAID

O Windows 2000 oferece suporte a RAID por hardware e por software. O suporte a RAID por hardware na realidade significa que o Windows 2000 oferece drivers de disco com capacidade de gerenciar controladoras RAID. Assim, discos físicos distintos podem ser combinados de diferentes maneiras para compor um ou mais discos lógicos. No RAID por hardware, é a própria controladora

que gerencia a criação e a manutenção da informação de redundância necessária à recuperação de dados.

O suporte a RAID via software, disponível apenas nas versões *server* do Windows 2000, emula, a partir de serviços do próprio sistema operacional, o funcionamento de uma controladora com suporte RAID. Um driver de dispositivo (FTDISK) é responsável por essa tarefa. Esse driver oferece RAID 1 e RAID 5.

10.8 O serviço de Active Directory

Uma das principais novidades introduzidas no Windows 2000 é a inclusão do serviço de diretório denominado de *active directory*. Mas o que é um serviço de diretório? No contexto de uma rede, um diretório é uma estrutura hierárquica que armazena informações a respeito de itens (recursos) de uma rede. Um diretório pode ser composto pelos mais diferentes tipos de itens (objetos), como por exemplo, servidores, discos, impressoras, contas de usuários, arquivos compartilhados; ou ainda domínios de *logon*, aplicativos, políticas de segurança e de acesso, etc. Cada item é um objeto e possui uma série de informações (atributos) associados a ele. Por exemplo, os atributos de um determinado objeto usuário podem ser nome, endereço, nome de usuário (*login name*), senha de acesso e grupos de que faz parte. O conceito de serviço de diretório é o conjunto de funções para criação, armazenamento e recuperação de informações de um diretório. O *active directory* do Windows 2000 tem exatamente esse objetivo, isto é, permitir que objetos sejam criados e manipulados facilmente.

O *active diretory* implementa um espaço de nomes. Um espaço de nomes é uma área em que um determinado nome pode ser resolvido, isto é, transformado em um objeto ou nas informação que representa. Esse processo de transformação é denominado de resolução de nomes. Um exemplo de um espaço de nomes e de resolução é o guia telefônico. Nesse caso, os nomes de assinantes são resolvidos para seus respectivos números de telefones. A estrutura do *active directory* é fortemente baseada em outro exemplo de espaço de nomes bastante comum no dia-a-dia de ambientes de redes: o DNS (*domain name server*). Devido a essa forte relação, nós iremos, de uma forma bastante sucinta, apresentar o princípio de funcionamento do DNS.

O DNS é uma hierarquia de domínios que representa toda a Internet em um único espaço de nomes. Essa hierarquia é organizada na forma de uma árvore. O nível logo abaixo da raiz corresponde aos domínios de mais alto nível (*top level domain - TLD*), os quais são subdivididos em domínios de segundo nível, e estes por sua vez em domínios de terceiro nível e assim sucessivamente. Os domínios TLD correspondem aos domínios associados a grandes categorias, como por exemplo, comercial (*.com*), governamental (*.gov*), educacional (*.edu*), ou ainda a países, Brasil (*.br*), França (*.fr*), etc. Esses domínios são administrados pela InterNIC (*Internet Network Information Center*). A InterNIC delega a responsabilidade de gerência do segundo nível a entidades administrativas oficialmente reconhecidas e registradas. No caso do domínio *.br* (Brasil) essa entidade é a FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo). A FAPESP, por sua vez, administra domínios de segundo nível como *.com.br* (comercial), *.gov.br* (governamental), ou ainda domínios do tipo *.ufrgs.br*. A resolução de nomes de um domínios DNS consiste em, dado o nome de uma máquina em um domínio, obter o endereço IP equivalente. A resolução de nomes do DNS é baseada em sua estrutura hierárquica. Assim, um máquina identificada como, por exemplo, *asterix.inf.ufrgs.br* diz respeito a um computador (*asterix*) que existe no domínio *inf* que pertence ao domínio *ufrgs* no Brasil (*.br*). É responsabilidade do domínio *inf.ufrgs.br* resolver o nome da máquina *asterix*, isto é, retornar qual é seu endereço IP.

O *active directory* utiliza a mesma estrutura de nomes dos domínios DNS. Entretanto, é importante observar que, apesar de empregar nomes idênticos, eles não compõem um único espaço de nomes,

ou seja, em cada espaço de nomes (DNS ou *active directory*), os nomes são resolvidos para informações diferentes. Na realidade, o DNS é um serviço de resolução puro, isto é, um cliente DNS envia uma requisição a um servidor DNS para "traduzir" um nome de máquina em IP. Já o *active directory* é um serviço de diretório. A resolução do nome depende do tipo de objeto que o nome representa. Nesse procedimento, um cliente *active directory* realiza requisições a um servidor *active directory* (também denominado de controlador de domínio) através de um protocolo específico: o LDAP (*Lightweight Directory Access Protocol*). O LDAP é um protocolo desenvolvido dentro da Internet para fornecer acesso a serviços de diretório. O fato de o *Windows* empregá-lo oferece a este um grau adicional de conectividade a ambientes heterogêneos.

Mas, de uma forma pragmática, o que finalmente é, e para que serve o *active directory* ? O objetivo principal do *active directory* é facilitar a adminstração da rede. Versões anteriores do Windows incluíam uma série de serviços e conceitos para auxiliar usuários e administradores de redes a localizar e gerenciar recursos da rede. O *network neighborhood*, o *WINS Manager*, o *Server Manager* são alguns exemplos desses serviços. Os objetos gerenciados por essas ferramentas consistiam no que se denominava de domínio Microsoft Windows NT. Essa abordagem, entretanto, mostrou-se um pouco prática para usuários e administradores – especialmente quando a rede apresentava uma certa complexidade pois forçava a sua subdivisão em vários domínios Windows NT, o que dificultava a localização e o gerenciamento de seus recursos. O *active directory* foi concebido para executar esse mesmo serviço, porém de uma maneira mais eficiente.

Os principais componentes que formam o *active directory* são: objeto, esquema, contêiner. Um objeto é qualquer usuário, sistema, recurso ou serviço existente dentro do *active directory*. Os objetos são descritos por seus atributos, como por exemplo, nome de uma máquina e seu endereço IP. O conjunto de atributos para qualquer tipo particular de objeto é chamado de esquema. Um contêiner é um tipo especial de objeto utilizado para organizar o *active directory*. Sua idéia é similar de *folder* (pasta) do *Windows*, isto é, se um *folder* contém arquivos e outros *folders*, um contêiner armazena objetos e outros contêiners. Os três tipos possíveis de contêiner são domínios, sites e unidades organizacionais.

Um domínio em Windows 2000 é muito similar ao conceito de domínio do Windows NT 4.0. Um domínio é um grupo de usuários e computadores que formam uma unidade administrativa isolada. Um site consiste em uma localização geográfica empregada para distinguir localizações remotas de localizações locais. Os sites podem ser comparados a subredes, isto é, sua estrutura pode ser empregada por aplicativos para localizar um determinado servidor mais próximo a esta subrede, reduzindo assim o tráfego em redes remotas. Uma unidade organizacional é um contêiner utilizado para agrupar objetos com políticas de acesso idênticas, podendo ser criada com base em vários critérios, como função, localização, recursos, etc. As unidades organizacionais existem dentro de um domínio.

Uma característica das grandes organizações é a necessidade de criar vários domínios para controlar de forma mais apropriada os recursos de um determinado setor, departamento, filial, etc. O *active directory* permite que os domínios sejam organizados hierarquicamente na forma de uma árvore. Essa organização cria uma relação de filiação entre os domínios com uma relação de confiança: um pai confia em seu filho, e o filho confia em seu pai. Essa relação de confiança permite que recursos sejam compartilhados entre pais e filhos, e entre filhos de um mesmo pai. Cada árvore possui um espaço de nome próprio. O conjunto de árvores, isto é, de espaços de nomes diferentes define o que se denomina, em terminologia Windows, de uma floresta.

O *active directory* é disponível no Windows 2000 apenas na sua versão server e, além de prover meios para o armazenamento de dados e acesso a serviços de um diretóiro, ele também integra mecanismos de segurança para evitar acessos não autorizados a objetos e mecanismos de replicação para garantir um certo grau de tolerância a falhas.

10.9 O serviço de *cluster*

Um *cluster* é conceituado como sendo uma coleção de sistemas independentes conectados entre si para executar um conjunto de aplicações, fornecendo a "ilusão" de ser um único sistema. A principal motivação de incluir no Windows 2000 capacidade de *clustering* foi o fato de se desejar garantir o funcionamento ininterrupto de serviços em ambientes corporativos. O suporte a *clustering* existe apenas nas versões *Advanced Server* e *Datacenter Server*.

O Windows 2000 oferece duas tecnologias de *clustering* que podem ser utilizadas de forma individual ou combinada. A primeira, serviço de *cluster*, tem como principal objetivo melhorar o tempo de funcionamento do sistema através da recuperação de falhas (mecanismo de *fail-over*). A idéia básica consiste em possibilitar que servidores sirvam de *backup* um do outro. Nessa configuração, quando um servidor principal falha, um segundo servidor assume automaticamente a responsabilidade da realização do serviço. Essa facilidade pode ainda ser explorada para atividades de manutenção agendadas. Nesse caso, é possível que um servidor *backup* continue a prover o serviço enquanto o servidor principal está sendo reparado. A segunda tecnologia de *clustering* disponível no Windows 2000 tem por objetivo melhorar a escalabilidade do sistema mediante o equilíbrio de carga entre múltiplas máquinas. Esse serviço é denominado de NLB (*Network Load Balancing*) e permite que o tráfego IP associado a um tipo de requisição (serviço) seja dividido em até 32 máquinas diferentes, configuradas para realizar um determinado serviço.

A utilização eficiente de *clusters* depende primariamente de três fatores: facilidade de implementação (configuração) do *cluster*, desenvolvimento de programas e gerenciamento. O Windows 2000, nas versões *Advanced Server* e *Datacenter Server*, incluem um "wizard" para auxiliar a criação do *cluster* e para facilitar a sua integração com serviços e aplicações distribuídas baseadas em DCOM. Os serviços de *clustering* no Windows 2000 estão embutidos no próprio sistema operacional para facilitar essas três tarefas. Uma vantagem do suporte a clustering no Windows 2000 é que ele foi concebido de forma a não necessitar de nenhum tipo especial de plataforma ou de conectividade. Um conjunto de máquinas interconectados normalmente em rede podem compor um *cluster*.

10.10 Uma palavrinha sobre o Windows XP

O Windows XP é a geração seguinte da família Windows. A denominação XP vem da palavra *eXPerience*. O Windows XP foi idealizado pela Microsoft com o objetivo de unificar, em torno de um único produto, seu mercado corporativo com seu mercado de usuários domésticos. Na realidade, essa unificação é feita através de duas versões do Windows XP: o *Windows XP Personal Edition*, destinado ao mercado doméstico, que substitui o Windows 95, 98, Millennium, NT (versão workstation); e o *Windows XP Professional Edition*, voltado ao mercado corporativo que substitui o NT nas suas versões *server*.

As principais novidades introduzidas pelo Windows XP estão relacionadas com mecanismos de proteção ao sistema de arquivos e conectividade à Internet. Sob o ponto de vista de proteção do sistema de arquivos, o Windows XP impede que arquivos antigos substituam versões mais recentes. No caso de drivers de dispositivos, é possível restaurar a versão anterior na eventualidade da instalação de um driver mais recente apresentar problemas. O suporte à proteção, na presença de múltiplos usuários, foi estendido em relação aos mecanismos oferecidos pelo Windows 98 e pelo Windows Millenium embutidos em seu próprio núcleo. Em relação à conectividade em rede, quando conectado à Internet, o Windows XP oferece ao usuário algumas funcionalidades típicas de *firewall* embutidas em seu próprio núcleo. Além disso, uma série de ferramentas buscando simplificar o uso do Windows por usuário leigos foram introduzidas, como por exemplo, gravação de cdrom

diretamente a partir do *Windows Explorer*, emprego de temas (*skins*) para tela de fundo, atualizações automáticas, mecanismos para publicação de arquivos de imagens e de texto na Internet, etc.

Entre as novidades do Windows XP, está ainda um mecanismo de proteção contra pirataria denominado de WPA (*Windows Product Activation*). Seu funcionamento é baseado na criação de um código único, válido apenas para o computador no qual o Windows XP é instalado. Esse código é criado no momento da instalação e é obtido através de identificadores próprios únicos a cada computador, tais como número de série da BIOS, do disco rígido, o endereço físico da placa de rede (endereço MAC), etc. Esse código é então informado à Microsoft, que imediatamente gera e reenvia um código de liberação para o uso do Windows XP. Ambos os códigos, o gerado na instalação e o código de instalação, são cadastrados na Microsoft. Sempre que houver uma modificação de hardware da máquina, ou a tentativa de burlar o sistema de ativação do Windows XP, o usuário deverá repetir o procedimento de ativação do Windows XP, ou seja, recontactar a Microsoft e gerar uma nova chave de ativação. Esse procedimento criou, na comunidade de usuários, muita controvérsia devido a rumores de que a Microsoft aproveitava-se dele para obter informações adicionais sobre a configuração da máquina, como por exemplo, os softwares instalados. Essa polêmica levou à análise das transações realizadas pelo mecanismo WPA por consultores independentes que concluíram que apenas o código de ativação é enviado à Microsoft.

O Windows XP foi desenvolvido com a preocupação de manter a compatibilidade com várias aplicações já existentes para a família Windows, principalmente jogos e multimídia. A Microsoft divulga que todos os aplicativos existentes para Windows 98, Millenium, e Windows NT continuarão a funcionar normalmente no Windows XP. A mesma preocupação de compatibilidade existe a nível de hardware; assim sendo, o Windows XP dispõe de uma grande gama de drivers para os mais diversos periféricos. Além disso, é possível instalar em uma máquina Windows XP drivers existentes para outras versões da família Windows.

As diferenças entre as versões Windows XP *Personal Edition* e Windows XP *Professional Edition* estão relacionadas com desempenho e atividades de gerenciamento. A versão *Professional* explora o multiprocessamento real oferecido pelas máquinas multiprocessadoras, ao passo que a versão *Personal* é otimizada para máquinas monoprocessadoras.

Sob o ponto de vista do sistema operacional, o Windows XP é um sistema operacional de 32 bits e herda em muito a arquitetura NT 5.0. As modificações, segundo a Microsoft, estão em otimizações de algoritmos básicos e nas estruturas de dados internas ao núcleo. Essas melhorias fazem com que o Windows XP apresente um desempenho melhor que seus antecessores. O Windows XP, assim como o Windows 2000 (NT 5.0), não fornece a capacidade de realizar *boot* em modo DOS. A compatibilidade com aplicativos que rodam sob DOS (como o Clipper) é feita exatamente da mesma forma que no Windows 2000, ou seja, através de um emulador DOS.

10.11 Exercícios

1. Compare o sistema de paginação do Windows com o do Linux, considerando os aspectos: políticas de substituição de páginas, tradução de endereço lógico a endereço físico, e estratégia de alocação de páginas em memória.

2. O Linux utiliza um modelo de estados de processo diferente do Windows. Faça uma correspondência entre os estados utilizados em cada um desses sistemas operacionais.

3. Pesquise sobre a capacidade do Linux de atribuir prioridades para *threads* no mesmo estilo da classe de tempo real oferecida pelo Windows. Caso exista algum mecanismo, faça uma análise comparativa.

4. Cite prós e contras do sistema de *swapping* utilizado pelo Windows quando confrontado com o do Linux.

5. Analise como é organizado o sistema de arquivos NTFS. Compare-o com o do Linux.