

# Sobre os Autores

**Andrew S. Tanenbaum** é bacharel em Ciências pelo M.I.T. e Ph.D. pela California University, em Berkeley. Atualmente, é professor de Ciência da Computação na Vrije Universiteit em Amsterdã, Holanda, onde lidera o Grupo de Sistemas de Computação. O autor também é diretor da Faculdade Avançada de Computação e Imagens Digitais, uma instituição interuniversitária que faz pesquisa sobre sistemas avançados paralelos, distribuídos e de imagem. Contudo, está tentando evitar a todo custo transformar-se em um burocrata.

No passado, fez pesquisa sobre compiladores, sistemas operacionais, redes remotas e sistemas distribuídos em rede local. Suas pesquisas atuais concentram-se principalmente no projeto de sistemas distribuídos de rede remota que atingem a escala de milhões de usuários. Esses projetos de pesquisa levaram a mais de 70 artigos em periódicos e conferências, além de cinco livros.

O professor Tanenbaum também produziu um considerável volume de software. Ele foi o principal arquiteto do Amsterdam Compiler Kit, um *kit* de ferramentas largamente utilizado para escrever compiladores portáteis, assim como do MINIX. Junto com seus programadores e alunos Ph.D., ele ajudou a projetar o sistema operacional distribuído Amoeba, um sistema operacional distribuído de alto desempenho baseado em *microkernel*. O MINIX e o Amoeba estão agora disponíveis gratuitamente para fins educacionais e de pesquisa via Internet.

O professor Tanenbaum é *Fellow* da ACM, membro sênior do IEEE, membro da Real Academia de Artes e Ciências da Holanda, vencedor do Prêmio Karl V. Karlstrom de Destaque na Educação de 1994 da ACM e do Prêmio ACM/SIGCSE para Contribuições Destacadas para Educação da Ciência da Computação de 1997. Seu nome também consta no *Who's Who in the World*. Sua *home page* na World Wide Web pode ser localizada na URL <http://www.cs.vu.nl/~ast/>

**Albert S. Woodhull** é bacharel em Ciências pelo M.I.T. e Ph.D. pela Washington University. Ele ingressou no M.I.T. com o objetivo de tornar-se engenheiro elétrico, mas saiu de lá biólogo. É professor associado da Escola de Ciência Natural do Hampshire College em Amherst, Massachusetts, desde 1973. Como biólogo, utilizando instrumentação eletrônica, começou a trabalhar com microcomputadores logo que estes apareceram. Seus cursos de instrumentação para alunos de ciências evoluíram para cursos sobre interfaces de computador e de programação em tempo real.

O Dr. Woodhull sempre teve forte interesse em ensinar e no papel da ciência e da tecnologia no desenvolvimento. Antes de entrar na faculdade, lecionou Ciência no nível secundário por dois anos na Nigéria. Mais recentemente passou vários fins-de-semana ensinando Ciência da Computação na Universidad Nacional de Ingenieria da Nicarágua e na Universidad Nacional Autónoma de Nicarágua.

Ele é interessado em computadores e em sistemas eletrônicos, bem como nas interações de computadores com outros sistemas eletrônicos. Gosta de ensinar nas áreas de arquitetura de computadores, programação em linguagem *assembly*, sistemas operacionais e comunicações de computador. Woodhull também trabalhou como consultor no desenvolvimento de instrumentação eletrônica e de software relacionado.

O Dr. Woodhull também tem muitos interesses não-acadêmicos, incluindo esportes ao ar livre, radioamadorismo e leitura. Ele gosta de viajar e tentar fazer-se entender em outros idiomas que não o seu inglês nativo. Sua *home page* na World Wide Web está localizada em um sistema executando MINIX, na URL <http://minix1.hampshire.edu/asw/>.

# Prefácio

A maioria dos livros sobre sistemas operacionais é rica na parte teórica e inconsistente, na prática. Este, contudo, objetiva proporcionar um melhor equilíbrio entre ambas. Ele abrange todos os princípios fundamentais pormenorizadamente, incluindo processos, comunicação interprocessos, semáforos, monitores, passagem de mensagens, algoritmos de agendamento, entrada/saída, impasses, *drivers* de dispositivos, gerenciamento de memória, algoritmos de paginação, projeto de sistema de arquivos, segurança e mecanismos de proteção, mas, também, discute em detalhes um sistema particular — o MINIX, sistema operacional compatível com o Unix — além de oferecer uma listagem completa do código-fonte para estudo. Esse arranjo permite que o leitor não só aprenda os princípios, mas, também, veja como são aplicados em um sistema operacional real.

Quando a primeira edição deste livro foi publicada, em 1987, causou uma pequena revolução na maneira como os cursos de sistemas operacionais eram ministrados. Até então, a maioria dos cursos somente abordava a teoria. Com o aparecimento do MINIX, muitas escolas começaram a oferecer aulas de laboratório, onde os alunos examinavam um sistema operacional real para ver como ele funcionava por dentro. Consideramos essa tendência muito bem-vinda e esperamos que esta segunda edição a fortaleça.

Nos seus primeiros 10 anos, o MINIX sofreu muitas mudanças. O código original foi projetado para um IBM PC baseado no 8088 de 256K com duas unidades de disquete e nenhum disco rígido. Além disso, foi baseado na Versão 7 do UNIX. Com o tempo, o MINIX desenvolveu-se de várias maneiras. Por exemplo, a versão atual agora rodará em qualquer computador situado entre o PC original (em modo real de 16 bits) até poderosos Pentium com grandes discos rígidos (no modo protegido de 32 bits). Ele também mudou da antiga base na Versão 7, para basear-se no padrão internacional POSIX (IEEE 1003.1 e ISO 9945-1). Por fim, muitos recursos foram acrescentados, demais em nosso pon-

to de vista, mas poucos, na visão de outras pessoas, que levaram à criação do LINUX. Além disso, o MINIX foi portado para muitas outras plataformas, para incluir o Macintosh, o Amiga, o Atari e as estações SPARC. Este livro abrange apenas o MINIX 2.0, que até agora só roda em computadores com uma CPU 80x86, em sistemas que podem emular uma CPU desse tipo ou em uma estação SPARC.

Esta segunda edição apresenta mudanças em todo o livro. Quase todo o material sobre princípios foi revisado, e considerável material novo foi acrescentado. Entretanto, a mudança principal é a discussão do novo MINIX baseado em POSIX, além da inclusão do novo código neste livro. Também nova é a inclusão de um CD-ROM, para oferecer o código-fonte completo do MINIX, com instruções para instalar o MINIX em um PC (veja o arquivo LEIAME.TXT no diretório principal do CD-ROM).

Instalar o MINIX em um PC 80x86, seja para uso individual seja para um laboratório, é simples e direto. Uma partição de disco de pelo menos 30MB deve ser feita para ele, que então pode ser instalado, bastando que as instruções do arquivo LEIAME.TXT do CD-ROM sejam seguidas. Para imprimir o arquivo LEIAME.TXT em um PC, primeiro inicie o MS-DOS, se ele ainda não estiver rodando (a partir do WINDOWS, clique no ícone MS-DOS). Então digite:

**copy leiame.txt prn**

para imprimir. O arquivo também pode ser examinado no Edit do MS-DOS, no WordPad, no Bloco de Notas do Windows, ou em qualquer outro editor de texto que manipule texto puro em ASCII.

Para escolas (ou indivíduos) que não dispõem de PCs, duas outras opções estão agora disponíveis. Foram incluídos dois simuladores no CD-ROM. Um, escrito por Paul Ashton, roda em SPARCs. Ele executa o MINIX como um programa de usuário sobre o Solaris. Como consequência, o MINIX é compilado em um binário SPARC e roda à toda velocidade. Deste modo, o MINIX não é mais um sistema

operacional, mas um programa de usuário. Portanto, foram necessárias algumas mudanças no código de baixo nível.

O outro simulador foi escrito por Kevin P. Lawton da Bochs Software Company. Esse simulador interpreta o conjunto de instruções Intel 80386 e o suficiente do mecanismo de E/S para o MINIX poder rodar no simulador. Naturalmente, rodar em um interpretador custa algum desempenho, mas facilita muito a depuração para o aluno. Esse simulador tem a vantagem de executar em qualquer computador que suporte o M.I.T. X Window System. Para mais informações sobre esses dois simuladores, veja o CD-ROM.

O desenvolvimento do MINIX é uma proposta em progresso. O conteúdo deste livro e seu CD-ROM são meramente uma amostra do sistema na época da sua publicação. Para o estado atual das coisas, visite a *home page* do MINIX na World Wide Web, <http://www.cs.vu.nl/~ast/minix.html>. Além disso, o MINIX tem seu próprio grupo de discussão na USENET: *comp.os.minix*, no qual os leitores podem inscrever-se para saber o que está acontecendo no mundo do MINIX. Para aqueles com correio eletrônico, mas sem acesso a grupos de discussão, também há uma lista de discussão via *e-mail*. Escreva para [listserv@listserv.nodak.edu](mailto:listserv@listserv.nodak.edu) com "subscribe minix-l <seu nome completo>" como a primeira e única linha do corpo da mensagem. Você receberá mais informações pelo correio eletrônico.

Para uso em sala de aula, a Prentice-Hall oferece um manual de soluções de problemas, mas apenas para professores. Os arquivos PostScript com todas as figuras no livro, conveniente como material didático de apoio, podem ser encontradas seguindo o link "*Software and supplementary material*" em <http://www.cs.vu.nl/~ast/>.

Fomos felizes em contar com a ajuda de muitas pessoas durante o curso deste projeto. Antes de tudo, gostaríamos de agradecer a Kees Bot por fazer a parte do leão, no trabalho, adequando o MINIX ao padrão e gerenciando a distribuição. Sem sua grande ajuda nunca teríamos feito tudo isso. Ele escreveu pedaços grandes do código (p. ex., a E/S de terminal POSIX), limpou outras seções e corrigiu numerosos *bugs* que surgiram no decorrer dos anos.

Bruce Evans, Philip Homburg, Will Rose e Michael Temari contribuíram para o desenvolvimento do MINIX durante anos. Centenas de outras pessoas contribuíram para o MINIX por meio dos grupos de discussão. Há tantos deles e suas contribuições são tão variadas que não podemos nem começar a listar todos aqui. Então, o melhor que podemos fazer é dar um muito obrigado a todos eles.

Várias pessoas leram partes dos originais e apresentaram sugestões. Gostaríamos de dedicar nosso especial agradecimento a John Casey, Dale Grit e Frans Kaashoek.

Na Vrije Universiteit, diversos alunos testaram a versão beta do CD-ROM, destacando-se: Ahmed Batou, Goran Dokic, Peter Gijzel, Thomer Gil, Dennis Grimbergen, Roderick Groesbeek, Wouter Haring, Guido Kollerie, Mark Lassche, Raymond Ris, Frans ter Borg, Alex van Ballegooy, Ries van der Velden, Alexander Wels e Thomas Zeeman. Agradecemos pelo trabalho cuidadoso e pelos relatórios detalhados.

ASW também gostaria de agradecer a vários dos seus ex-alunos, particularmente Peter W. Young, do Hampshire College, e Maria Isabel Sanchez e William Puddy Vargas, da Universidad Nacional Autónoma de Nicaragua, cujo interesse no MINIX sustentou seus esforços.

Por fim, queremos agradecer às nossas famílias. Suzanne entrou nessa pela décima vez. Barbara pela nona vez. Marvin pela oitava. E até o pequeno Bram já está na sua quarta vez. A coisa parece estar tornando-se rotina, mas o amor e o apoio ainda muito são apreciados. (AST)

Quanto à Barbara, esta é a sua primeira vez. Nada teria sido possível sem seu apoio, paciência e bom humor. Foi sorte do Gordon ter estado longe, na faculdade, durante a maior parte disso tudo. Mas é maravilhoso ter um filho que entende e preocupa-se com as mesmas coisas que me fascinam. (ASW)

Andrew S. Tanenbaum  
Albert S. Woodhull

# Sumário

<b>1</b>	<b>ASPECTOS GERAIS</b>	<b>17</b>
1.1	O QUE É UM SISTEMA OPERACIONAL	18
1.1.1	O sistema operacional como uma máquina estendida	18
1.1.2	O sistema operacional como um gerenciador de recursos	19
1.2	HISTÓRIA DOS SISTEMAS OPERACIONAIS	19
1.2.1	A primeira geração (1945-55): válvulas e painéis de conectores	20
1.2.2	A segunda geração (1955-65): transístores e sistemas de lote	20
1.2.3	A terceira geração (1965-1980): CIs e multiprogramação	21
1.2.4	A quarta geração (1980-hoje): computadores pessoais	24
1.2.5	A história do MINIX	24
1.3	CONCEITOS DE SISTEMA OPERACIONAL	25
1.3.1	Processos	26
1.3.2	Arquivos	27
1.3.3	O <i>Shell</i>	29
1.4	CHAMADAS DE SISTEMA	30
1.4.1	Chamadas de sistema para gerenciamento de processos	30
1.4.2	Chamadas de sistema para sinalização	33
1.4.3	Chamadas de sistema para gerenciamento de arquivos	34
1.4.4	Chamadas de sistema para gerenciamento de diretórios	37
1.4.5	Chamadas de sistema para proteção	38
1.4.6	Chamadas de sistema para gerenciamento de tempo	39
1.5	A ESTRUTURA DO SISTEMA OPERACIONAL	39
1.5.1	Sistemas monolíticos	39
1.5.2	Sistemas em camadas	40
1.5.3	Máquinas virtuais	42
1.5.4	Modelo cliente-servidor	43
1.6	VISÃO GERAL DO RESTANTE DESTA LIVRO	44
1.7	RESUMO	44