

Andrew S. Tanenbaum

I moderni sistemi operativi

Terza edizione

Edizione italiana a cura di Luciano Baresi



© 2009 Pearson Paravia Bruno Mondadori S.p.A.

*Authorized translation from the English language edition, entitled: **MODERN OPERATING SYSTEMS, 3rd edition**, by Andrew Tanenbaum, published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2008.*

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Italian language edition published by Pearson Paravia Bruno Mondadori S.p.A., Copyright © 2009.

Le informazioni contenute in questo libro sono state verificate e documentate con la massima cura possibile. Nessuna responsabilità derivante dal loro utilizzo potrà venire imputata agli Autori, a Pearson Paravia Bruno Mondadori S.p.A. o a ogni persona e società coinvolta nella creazione, produzione e distribuzione di questo libro.

Per i passi antologici, per le citazioni, per le riproduzioni grafiche, cartografiche e fotografiche appartenenti alla proprietà di terzi, inseriti in quest'opera, l'editore è a disposizione degli aventi diritto non potuti reperire nonché per eventuali non volute omissioni e/o errori di attribuzione nei riferimenti.

I diritti di riproduzione e di memorizzazione elettronica totale e parziale con qualsiasi mezzo, compresi i microfilm e le copie fotostatiche, sono riservati per tutti i paesi.

LA FOTOCOPIATURA DEI LIBRI È UN REATO Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633.

Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate a seguito di specifica autorizzazione rilasciata da AIDRO, corso di Porta Romana n. 108, 20122 Milano, e-mail segreteria@aidro.org e sito web www.aidro.org

Curatore per l'edizione italiana: Luciano Baresi

Traduzione: Marco Aperti e Claudia Baresi

Copy-editing: Donatella Pepe

Impaginazione: Elisabetta Bozzi

Grafica di copertina: Nicolò Cannizzaro

Stampa: Tip.Le.Co. – S. Bonico (PC)

Tutti i marchi citati nel testo sono di proprietà dei loro detentori.

978-88-7192-540-0

Printed in Italy

1^a edizione: giugno 2009

Ristampa

00 01 02 03 04

Anno

09 10 11 12 13

A Suzanne, Barbara, Marvin e in memoria di Bram e Sweetie π

S o m m a r i o

Prefazione all'edizione italiana	xix
Prefazione	xxi
Biografia	xxv
Capitolo 1 Introduzione	1
1.1 Che cos'è un sistema operativo	3
1.1.1 Il sistema operativo come macchina estesa	3
1.1.2 Il sistema operativo come gestore delle risorse	5
1.2 Storia dei sistemi operativi	6
1.2.1 Prima generazione (1945-55): tubi a vuoto	7
1.2.2 Seconda generazione (1955-65): transistor e sistemi batch	7
1.2.3 Terza generazione (1965-80): IC e multiprogrammazione	9
1.2.4 Quarta generazione (dal 1980 a oggi): i personal computer	14
1.3 Analisi dell'hardware	17
1.3.1 Processori	18
1.3.2 Memoria	21
1.3.3 Dischi	24
1.3.4 Nastri	25
1.3.5 Dispositivi di I/O	25
1.3.6 Bus	28
1.3.7 Avvio del computer	30
1.4 Panoramica dei sistemi operativi	31
1.4.1 Sistemi operativi per mainframe	31
1.4.2 Sistemi operativi per server	32
1.4.3 Sistemi operativi multiprocessore	32
1.4.4 Sistemi operativi per personal computer	32
1.4.5 Sistemi operativi per computer palmari	32
1.4.6 Sistemi operativi integrati	33
1.4.7 Sistemi operativi per sensori	33
1.4.8 Sistemi operativi real-time	34
1.4.9 Sistemi operativi per smart card	34

1.5	Concetti di base dei sistemi operativi	35
1.5.1	Processi	35
1.5.2	Spazi degli indirizzi	37
1.5.3	File	37
1.5.4	Input/output	40
1.5.5	Protezione	40
1.5.6	La shell	41
1.5.7	Il concetto di ricapitolazione	42
1.6	Chiamate di sistema	45
1.6.1	Chiamate di sistema per la gestione dei processi	48
1.6.2	Chiamate di sistema per la gestione dei file	51
1.6.3	Chiamate di sistema per la gestione delle directory	52
1.6.4	Altre chiamate di sistema	54
1.6.5	Le API Win32 di Windows	54
1.7	Struttura di un sistema operativo	57
1.7.1	Sistemi monolitici	57
1.7.2	Sistemi a livelli	58
1.7.3	Microkernel	59
1.7.4	Modello client-server	62
1.7.5	Macchine virtuali	62
1.7.6	Exokernel	65
1.8	Il mondo secondo il C	66
1.8.1	Il linguaggio C	66
1.8.2	File d'intestazione	67
1.8.3	Progetti di programmazione di grandi dimensioni	68
1.8.4	Il modello di run-time	68
1.9	Stato della ricerca sui sistemi operativi	69
1.10	Linee generali del resto del volume	70
1.11	Unità metriche	71
	Problemi	72
Capitolo 2	Processi e thread	75
2.1	Processi	75
2.1.1	Il modello di processo	76
2.1.2	Creazione del processo	78
2.1.3	Chiusura di un processo	79
2.1.4	Gerarchie di processi	80
2.1.5	Stati di un processo	81
2.1.6	Implementazione dei processi	83
2.1.7	Modellazione della multiprogrammazione	84
2.2	Thread	86
2.2.1	Uso dei thread	86
2.2.2	Il modello a thread classico	91
2.2.3	Thread POSIX	95
2.2.4	Implementare i thread nello spazio utente	96
2.2.5	Realizzazione di thread nel kernel	99

2.2.6	Implementazioni ibride	100
2.2.7	Scheduler activation	101
2.2.8	Thread pop-up	102
2.2.9	Trasformazione di codice a singolo thread in codice multithread	103
2.3	Comunicazione fra processi	106
2.3.1	Corse critiche	107
2.3.2	Regioni critiche	108
2.3.3	Mutua esclusione con busy waiting	109
2.3.4	Sleep e wakeup	114
2.3.5	Semafori	116
2.3.6	Mutex	118
2.3.7	Monitor	123
2.3.8	Passaggio di messaggi	128
2.3.9	Barriere	131
2.4	Scheduling	132
2.4.1	Introduzione allo scheduling	132
2.4.2	Scheduling nei sistemi batch	138
2.4.3	Scheduling nei sistemi interattivi	140
2.4.4	Scheduling nei sistemi real-time	145
2.4.5	La politica in contrapposizione al meccanismo	146
2.4.6	Scheduling a thread	147
2.5	Classici problemi di IPC	149
2.5.1	Problema dei 5 filosofi	149
2.5.2	Problema dei lettori e scrittori	152
2.6	Stato della ricerca su processi e thread	153
	Problemi	154

Capitolo 3	Gestione della memoria	159
3.1	Nessuna astrazione di memoria	160
3.2	Un'astrazione della memoria: gli spazi degli indirizzi	163
3.2.1	Nozione di spazio degli indirizzi	163
3.2.2	Swapping	166
3.2.3	Gestione della memoria libera	168
3.3	Memoria virtuale	171
3.3.1	Paginazione	172
3.3.2	Tabelle delle pagine	176
3.3.3	Velocizzare la paginazione	177
3.3.4	Tabelle delle pagine per grandi memorie	181
3.4	Algoritmi di sostituzione delle pagine	184
3.4.1	L'algoritmo ottimale di sostituzione delle pagine	185
3.4.2	Not recently used (NRU)	185
3.4.3	First-in, first-out (FIFO)	186
3.4.4	Seconda chance	187
3.4.5	Clock	188
3.4.6	Least recently used (LRU)	188
3.4.7	Simulare l'algoritmo LRU via software	189

3.4.8	Working set	191
3.4.9	WSClock	195
3.4.10	Riepilogo degli algoritmi di sostituzione delle pagine	196
3.5	Problemi di progettazione dei sistemi di paginazione	197
3.5.1	Politiche di allocazione globali e locali a confronto	197
3.5.2	Controllo del carico	200
3.5.3	Dimensione delle pagine	200
3.5.4	Istruzioni separate e spazi dei dati	202
3.5.5	Pagine condivise	202
3.5.6	Librerie condivise	204
3.5.7	File mappati	206
3.5.8	Politica di ripulitura	206
3.5.9	Interfaccia della memoria virtuale	207
3.6	Elementi implementativi	208
3.6.1	Il sistema operativo e la paginazione	208
3.6.2	Gestione degli errori di pagina	209
3.6.3	Backup delle istruzioni	210
3.6.4	Bloccare le pagine in memoria	211
3.6.5	Memoria secondaria	211
3.6.6	Separazione fra politica e meccanismo	213
3.7	Segmentazione	214
3.7.1	Implementazione della segmentazione pura	217
3.7.2	Segmentazione con la paginazione: MULTICS	218
3.7.3	Segmentazione con paginazione: il Pentium Intel	222
3.8	Stato della ricerca sulla gestione della memoria	227
	Problemi	228

Capitolo 4	File system	235
4.1	I file	236
4.1.1	Nomi dei file	236
4.1.2	Struttura dei file	238
4.1.3	Tipi di file	239
4.1.4	Accesso ai file	241
4.1.5	Attributi dei file	242
4.1.6	Operazioni sui file	243
4.1.7	Un esempio di programma che usa le chiamate del file system	244
4.2	Le directory	246
4.2.1	Sistemi di directory a livello singolo	246
4.2.2	Sistemi di directory gerarchici	247
4.2.3	Path name	248
4.2.4	Operazioni sulle directory	250
4.3	Implementazione del file system	251
4.3.1	Layout del file system	251
4.3.2	Implementazione dei file	252
4.3.3	Implementazione delle directory	257
4.3.4	File condivisi	260
4.3.5	File system basati su log strutturati	262

4.3.6	File system journaling	263
4.3.7	File system virtuali	265
4.4	Gestione e ottimizzazione del file system	268
4.4.1	Gestione dello spazio dei dischi	268
4.4.2	Backup del file system	275
4.4.3	Consistenza del file system	280
4.4.4	Prestazioni del file system	282
4.4.5	Deframmentazione dei dischi	287
4.5	Esempi di file system	287
4.5.1	File system dei CD-ROM	287
4.5.2	Il file system MS-DOS	292
4.5.3	Il file system V7 di UNIX	295
4.6	Stato della ricerca sui file system	298
	Problemi	299

Capitolo 5 Input/output 303

5.1	Principi hardware dell'I/O	303
5.1.1	Dispositivi di I/O	303
5.1.2	Controller dei dispositivi	304
5.1.3	Input/output mappato in memoria	306
5.1.4	Direct memory access (DMA)	309
5.1.5	Interrupt rivisitati	312
5.2	Principi del software per l'I/O	316
5.2.1	Obiettivi del software per l'I/O	316
5.2.2	I/O programmato	317
5.2.3	I/O guidato dagli interrupt	319
5.2.4	I/O con uso del DMA	320
5.3	Livelli del software per l'I/O	320
5.3.1	Gestori degli interrupt	320
5.3.2	Driver dei dispositivi	322
5.3.3	Software per l'I/O indipendente dal dispositivo	325
5.3.4	Software per l'I/O nello spazio utente	330
5.4	Dischi	332
5.4.1	Hardware dei dischi	332
5.4.2	Formattazione dei dischi	346
5.4.3	Algoritmi di scheduling del braccio del disco	350
5.4.4	Gestione degli errori	353
5.4.5	Memoria stabile	355
5.5	Clock	358
5.5.1	Hardware del clock	358
5.5.2	Software del clock	360
5.5.3	Timer soft	362
5.6	Interfacce utente: tastiera, mouse e monitor	364
5.6.1	Software per l'input	364
5.6.2	Software per l'output	369
5.7	Thin client	383

5.8	Gestione del risparmio energetico	385
5.8.1	Problemi relativi all'hardware	386
5.8.2	Problemi relativi al sistema operativo	387
5.8.3	Questioni relative ai programmi applicativi	392
5.9	Stato della ricerca sull'input/output	393
	Problemi	394

Capitolo 6 Deadlock 399

6.1	Risorse	400
6.1.1	Risorse preemptable e non-preemptable	400
6.1.2	Acquisizione delle risorse	401
6.2	Introduzione ai deadlock	403
6.2.1	Condizioni per i deadlock delle risorse	403
6.2.2	Modellazione dei deadlock	404
6.3	Algoritmo dello struzzo	406
6.4	Rilevamento e risoluzione dei deadlock	407
6.4.1	Rilevamento dei deadlock con una risorsa per ciascun tipo	407
6.4.2	Rilevamento dei deadlock con più risorse per ciascun tipo	409
6.4.3	Risoluzione di un deadlock	412
6.5	Evitare i deadlock	413
6.5.1	Traiettorie delle risorse	413
6.5.2	Stati sicuri e non sicuri	414
6.5.3	Algoritmo del banchiere per una risorsa singola	416
6.5.4	Algoritmo del banchiere per molteplici risorse	417
6.6	Prevenire i deadlock	418
6.6.1	Attacco alla condizione di mutua esclusione	418
6.6.2	Attacco alla condizione di possesso e attesa	419
6.6.3	Attacco alla condizione di impossibilità di prelazione	419
6.6.4	Attacco alla condizione di attesa circolare	420
6.7	Ulteriori questioni	421
6.7.1	Docking a due fasi	421
6.7.2	Deadlock delle comunicazioni	422
6.7.3	Livelock	423
6.7.4	Starvation	425
6.8	Stato della ricerca sui deadlock	425
	Problemi	426

Capitolo 7 Sistemi operativi multimediali 429

7.1	Introduzione alla multimedialità	430
7.2	File multimediali	434
7.2.1	Codifica video	435
7.2.2	Codifica audio	437
7.3	Compressione video	439
7.3.1	Standard JPEG	440
7.3.2	Standard MPEG	442
7.4	Compressione audio	445

7.5	Scheduling dei processi multimediali	448
7.5.1	Scheduling di processi omogenei	448
7.5.2	Scheduling real-time in generale	448
7.5.3	Scheduling rate monotonic	450
7.5.4	Scheduling earliest deadline first	451
7.6	Paradigmi dei file system multimediali	453
7.6.1	Funzioni VCR di controllo	454
7.6.2	Near video-on-demand	456
7.6.3	Near video-on-demand con le funzioni VCR	457
7.7	Posizionamento dei file	459
7.7.1	Posizionamento di un file su un disco singolo	459
7.7.2	Due strategie alternative nell'organizzazione dei file	460
7.7.3	Posizionamento dei file per il near video-on-demand	464
7.7.4	Posizionamento di più file su un singolo disco	465
7.7.5	Posizionamento dei file su più dischi	467
7.8	Uso della cache	470
7.8.1	Cache dei blocchi	470
7.8.2	Cache dei file	471
7.9	Scheduler del disco per il multimedia	472
7.9.1	Scheduler statico del disco	472
7.9.2	Scheduler dinamico del disco	474
7.10	Stato della ricerca sulla multimedialità	476
	Problemi	476

Capitolo 8	Sistemi a più processori	481
8.1	Multiprocessori	483
8.1.1	Hardware dei multiprocessori	484
8.1.2	Tipi di sistemi operativi multiprocessore	491
8.1.3	Sincronizzazione dei multiprocessori	495
8.1.4	Scheduling dei multiprocessori	499
8.2	Multicomputer	504
8.2.1	Hardware dei multicomputer	505
8.2.2	Software di comunicazione di basso livello	508
8.2.3	Software di comunicazione a livello utente	510
8.2.4	Chiamata di procedura remota	513
8.2.5	Memoria condivisa distribuita	516
8.2.6	Scheduling nei multicomputer	520
8.2.7	Bilanciamento del carico	520
8.3	Virtualizzazione	523
8.3.1	Requisiti per la virtualizzazione	525
8.3.2	Hypervisor di tipo 1	526
8.3.3	Hypervisor di tipo 2	527
8.3.4	Paravirtualizzazione	528
8.3.5	Virtualizzazione della memoria	530
8.3.6	Virtualizzazione dell'I/O	532
8.3.7	Virtual appliances	533
8.3.8	Macchine virtuali sulle CPU multicore	533
8.3.9	Il problema delle licenze	534

8.4	Sistemi distribuiti	534
8.4.1	Hardware di rete	537
8.4.2	Protocolli e servizi di rete	539
8.4.3	Middleware basato sui documenti	543
8.4.4	Middleware basato sul file system	544
8.4.5	Middleware basato sugli oggetti	549
8.4.6	Middleware basato sulla coordinazione	551
8.4.7	Grid	556
8.5	Stato della ricerca sui sistemi a più processori	556
	Problemi	557
Capitolo 9	Sicurezza	563
9.1	Ambiente della sicurezza	565
9.1.1	Minacce	565
9.1.2	Intrusi	566
9.1.3	Perdita accidentale di dati	567
9.2	Basi della crittografia	568
9.2.1	Crittografia a chiave segreta	569
9.2.2	Crittografia a chiave pubblica	569
9.2.3	Funzioni a senso unico	570
9.2.4	Firme digitali	570
9.2.5	Trusted platform module	572
9.3	Meccanismi di protezione	573
9.3.1	Domini di protezione	573
9.3.2	Access control list	575
9.3.3	“Capability”	578
9.3.4	Sistemi affidabili	581
9.3.5	Trusted computing base	582
9.3.6	Modelli formali di sistemi sicuri	583
9.3.7	Sicurezza multivello	584
9.3.8	Canali segreti	587
9.4	Autenticazione	591
9.4.1	Autenticazione tramite password	592
9.4.2	Autenticazione tramite un oggetto fisico	599
9.4.3	Autenticazione attraverso la biometria	602
9.5	Attacchi dall’interno	605
9.5.1	Bombe logiche	605
9.5.2	Trap door	606
9.5.3	Spoofing del login	606
9.6	Sfruttare gli errori del codice	607
9.6.1	Attacchi di tipo “buffer overflow”	608
9.6.2	Attacchi di tipo “format string”	610
9.6.3	Attacchi di tipo “return to libc”	612
9.6.4	Attacchi di tipo “integer overflow”	613
9.6.5	Attacchi di tipo “code injection”	614
9.6.6	Attacchi di tipo “privilege escalation”	615
9.7	Malware	616
9.7.1	Cavalli di Troia	618
9.7.2	Virus	620
9.7.3	Worm	629

9.7.4	Spyware	631
9.7.5	Rootkit	635
9.8	Difese	639
9.8.1	Firewall	639
9.8.2	Antivirus e tecniche antivirus	641
9.8.3	Firma del codice	647
9.8.4	Jailing	649
9.8.5	Rilevamento delle intrusioni basato su un modello	649
9.8.6	Incapsulamento del codice mobile	651
9.8.7	Sicurezza in Java	655
9.9	Stato della ricerca sulla sicurezza	657
	Problemi	658

Capitolo 10 Linux 663

10.1	Storia di UNIX e Linux	664
10.1.1	UNICS	664
10.1.2	UNIX del PDP-11	664
10.1.3	UNIX portabile	665
10.1.4	UNIX Berkeley	666
10.1.5	UNIX standard	667
10.1.6	MINIX	668
10.1.7	Linux	669
10.2	Panoramica di Linux	671
10.2.1	Obiettivi di Linux	671
10.2.2	Interfacce di Linux	672
10.2.3	Shell	674
10.2.4	Programmi di utilità Linux	676
10.2.5	Struttura del kernel	678
10.3	Processi in Linux	681
10.3.1	Concetti fondamentali	681
10.3.2	Chiamate di sistema per la gestione dei processi in Linux	683
10.3.3	Implementazione di processi e thread in Linux	687
10.3.4	Scheduling in Linux	693
10.3.5	Processo d'avvio di Linux	696
10.4	Gestione della memoria in Linux	698
10.4.1	Concetti fondamentali	698
10.4.2	Chiamate di sistema per la gestione della memoria in Linux	702
10.4.3	Implementazione della gestione della memoria in Linux	703
10.4.4	Paginazione in Linux	708
10.5	Input/output in Linux	712
10.5.1	Concetti fondamentali	712
10.5.2	Gestione della rete	713
10.5.3	Chiamate di sistema per l'input/output in Linux	715
10.5.4	Implementazione dell'input/output in Linux	715
10.5.5	Moduli in Linux	719
10.6	File system di Linux	720
10.6.1	Concetti fondamentali	720
10.6.2	Chiamate del file system in Linux	724
10.6.3	Implementazione del file system di Linux	728
10.6.4	NFS: il file system di rete	736

10.7	Sicurezza in Linux	742
10.7.1	Concetti fondamentali	742
10.7.2	Chiamate di sistema per la sicurezza in Linux	744
10.7.3	Implementazione della sicurezza in Linux	745
	Problemi	746

Capitolo 11 Windows Vista 751

11.1	Storia di Windows Vista	751
11.1.1	Gli anni '80: MS-DOS	751
11.1.2	Gli anni '90: Windows basato su MS-DOS	752
11.1.3	Gli anni 2000: Windows basato su NT	753
11.1.4	Windows Vista	756
11.2	Programmazione di Windows Vista	757
11.2.1	Application programming interface NT native	759
11.2.2	Application programming interface Win32	762
11.2.3	Registro di Windows	766
11.3	Struttura del sistema	768
11.3.1	Struttura del sistema operativo	769
11.3.2	Avvio di Windows Vista	782
11.3.3	Implementazione del gestore degli oggetti	783
11.3.4	Sottosistemi, DLL e servizi in modalità utente	793
11.4	Processi e thread in Windows Vista	796
11.4.1	Concetti fondamentali	796
11.4.2	Chiamate API per la gestione di job, processi, thread e fibre	801
11.4.3	Implementazione di processi e di thread	806
11.5	Gestione della memoria	813
11.5.1	Concetti fondamentali	813
11.5.2	Chiamate di sistema per la gestione della memoria	817
11.5.3	Implementazione della gestione della memoria	819
11.6	Uso della cache in Windows Vista	828
11.7	Input/output in Windows Vista	830
11.7.1	Concetti fondamentali	830
11.7.2	API per l'input/output	832
11.7.3	Implementazione dell'I/O	834
11.8	NT file system di Windows	839
11.8.1	Concetti fondamentali	840
11.8.2	Implementazione del file system NT	841
11.9	Sicurezza in Windows Vista	850
11.9.1	Concetti fondamentali	851
11.9.2	Chiamate API per la sicurezza	853
11.9.3	Implementazione della sicurezza	854
	Problemi	856

Capitolo 12 Symbian OS 859

12.1	Storia di Symbian OS	859
12.1.1	Radici di Symbian OS: Psion ed EPOC	860
12.1.2	Versione 6 di Symbian OS	861
12.1.3	Versione 7 di Symbian OS	861
12.1.4	Symbian OS oggi	861

12.2	Panoramica di Symbian OS	862
12.2.1	Progettazione a oggetti	862
12.2.2	Progettazione a microkernel	863
12.2.3	Nanokernel di Symbian OS	864
12.2.4	Accesso client/server alle risorse	864
12.2.5	Funzionalità di un sistema operativo più grande	865
12.2.6	Comunicazione e multimedialità	866
12.3	Processi e thread in Symbian OS	866
12.3.1	Thread e nanothread	866
12.3.2	Processi	868
12.3.3	Oggetti attivi	868
12.3.4	Comunicazione tra processi	869
12.4	Gestione della memoria	869
12.4.1	Sistemi senza memoria virtuale	870
12.4.2	Come Symbian OS indirizza la memoria	871
12.5	Input e output	874
12.5.1	Driver dei dispositivi	874
12.5.2	Estensioni del kernel	874
12.5.3	Direct memory access	875
12.5.4	Un caso speciale: media di memorizzazione	875
12.5.5	I/O bloccante	875
12.5.6	Media rimovibili	876
12.6	Sistemi di memorizzazione	876
12.6.1	File system per dispositivi mobili	876
12.6.2	File system di Symbian OS	877
12.6.3	Sicurezza e protezione del file system	877
12.7	Sicurezza in Symbian OS	878
12.8	Comunicazione in Symbian OS	880
12.8.1	Infrastruttura di base	880
12.8.2	Visione più approfondita dell'infrastruttura	881
	Problemi	884

Capitolo 13 Progettazione di un sistema operativo 885

13.1	Natura del problema della progettazione	885
13.1.1	Obiettivi	885
13.1.2	Perché è difficile progettare un sistema operativo?	887
13.2	Progettazione delle interfacce	888
13.2.1	Principi guida	889
13.2.2	Paradigmi	890
13.2.3	Interfaccia delle chiamate di sistema	893
13.3	Implementazione	896
13.3.1	Struttura dei sistemi	896
13.3.2	Il meccanismo rispetto alla politica	899
13.3.3	Ortogonalità	900
13.3.4	Gestione dei nomi	901
13.3.5	Binding time	902
13.3.6	Strutture statiche rispetto a strutture dinamiche	903
13.3.7	Implementazione top-down e bottom-up	905
13.3.8	Tecniche utili	905

13.4	Prestazioni	910
13.4.1	Perché i sistemi operativi sono lenti?	911
13.4.2	Che cosa dovrebbe essere ottimizzato?	911
13.4.3	Compromessi spazio-tempo	912
13.4.4	Uso della cache	915
13.4.5	Suggerimenti	916
13.4.6	Sfruttare la località	916
13.4.7	Ottimizzazione del caso comune	917
13.5	Gestione dei progetti	917
13.5.1	The Mythical Man Month	918
13.5.2	Struttura del team	919
13.5.3	Ruolo dell'esperienza	921
13.5.4	No Silver Bullet	922
13.6	Tendenze nella progettazione dei sistemi operativi	922
13.6.1	Virtualizzazione	922
13.6.2	Chip multicore	922
13.6.3	Sistemi operativi con spazi degli indirizzi grandi	923
13.6.4	La rete	924
13.6.5	Sistemi paralleli e sistemi distribuiti	924
13.6.6	Multimedia	925
13.6.7	Computer a batterie	925
13.6.8	Sistemi integrati	926
13.6.9	Nodi di sensori	926
	Problemi	926
Capitolo 14 Letture consigliate e bibliografia		929
14.1	Suggerimenti per ulteriori letture	929
14.1.1	Introduzione e lavori generali	929
14.1.2	Processi e thread	930
14.1.3	Gestione della memoria	930
14.1.4	Input/output	931
14.1.5	File system	931
14.1.6	Deadlock	932
14.1.7	Sistemi operativi multimediali	932
14.1.8	Sistemi a più processori	932
14.1.9	Sicurezza	934
14.1.10	Linux	935
14.1.11	Windows Vista	936
14.1.12	Symbian OS	936
14.1.13	Principi di progettazione	937
14.2	Bibliografia	940
Indice analitico		971

Prefazione all'edizione italiana

A mio giudizio, questo è un libro storico, un elemento importante della biblioteca di ogni informatico. “Il Tanenbaum”, come eravamo soliti chiamarlo all’università, è stato ed è ancora uno dei principali libri al mondo sui sistemi operativi. Da studente, ho usato e consumato la prima edizione; ora mi trovo a scrivere la prefazione della traduzione italiana della terza edizione.

Sono passati diversi anni, ma il libro, e il suo autore, rimangono un riferimento assoluto per lo studio dei sistemi operativi. Ogni edizione diventa sempre più corposa, ma la sostanza resta. Il libro rappresenta tuttora uno strumento significativo per gli studenti ma, grazie ai continui esempi di casi reali, anche un ottimo riferimento per il professionista.

Come afferma l’autore nella sua prefazione, questa nuova edizione presenta una riorganizzazione generale dei contenuti, ma anche alcune parti nuove, imposte dall’evoluzione dei sistemi operativi e dall’attualità. Trovo interessanti i nuovi capitoli su Linux e Vista, che sostituiscono i precedenti su UNIX e Windows 2000, e ritengo che il nuovo capitolo su Symbian, e più in generale l’attenzione sui sistemi operativi embedded, siano molto formativi e colmino una lacuna importante. Un telefonino oggi è sempre più un “piccolo” dispositivo di calcolo, dotato di un proprio sistema operativo, e sempre meno un semplice telefono. La diffusione e l’impatto di questi dispositivi sull’informatica moderna (oltre che sulle nostre vite) impongono che i loro sistemi operativi siano realizzati e studiati con la stessa cura che finora abbiamo dedicato ai computer tradizionali.

Questa nuova edizione fa un passo avanti e, di conseguenza, ci fa entrare in una nuova era: oltre alla solita disputa tra UNIX (Linux) e Windows, ora i sistemi operativi embedded si stanno affermando come terzo polo, ma in un futuro non remoto potrebbero contendere (“rubare”) la scena ai loro avversari tradizionali.

Ogni capitolo del libro tende a essere organizzato come entità autonoma. Questo è sicuramente un elemento positivo anche per chi fosse interessato alle sole singole parti. Da un punto di vista didattico, la ripetizione di alcuni argomenti fondamentali e la loro rivisitazione in un contesto particolare aiutano il lettore a capire meglio e a cogliere anche le sfumature.

Per finire, due parole sull’edizione italiana. Questa rispetta e ricalca integralmente l’originale. L’organizzazione del testo è la medesima e tutti i capitoli sono stati tradotti integralmente. L’unica nota riguarda la scelta dei termini. Come sempre tradurre un libro tecnico non è banale. Bisogna evitare l’integralismo linguistico, che ci avrebbe portato a

cercare di rendere in italiano concetti intraducibili, ma anche l'errore opposto, ovvero l'uso eccessivo di termini in inglese. Trovare il giusto compromesso non è stato facile e spero di avere raggiunto lo scopo, offrendo una traduzione comprensibile, non ridicola e omogenea. Sicuramente, in questo, un ringraziamento enorme va a Claudia e Marco, che si sono occupati materialmente della traduzione, e a Donatella, che nel suo ruolo di editor, ha evidenziato problemi e sviste e ci ha dato un aiuto fondamentale con la sua professionalità e simpatia.

Luciano Baresi
Dipartimento di Elettronica e Informazione
Politecnico di Milano

Prefazione

La terza edizione di questo libro differisce dalla precedente in molte parti. Innanzitutto, la disposizione dei capitoli è stata rivista per anticipare gli argomenti più rilevanti. Il Capitolo 1, ampiamente aggiornato, introduce tutti i concetti principali. Il Capitolo 2 presenta il concetto di astrazione della CPU in diversi processi. Il Capitolo 3 riguarda l'astrazione della memoria fisica negli spazi degli indirizzi (memoria virtuale). Il Capitolo 4 propone l'astrazione dei dischi nei file. Processi, spazi virtuali degli indirizzi e file sono tutti elementi fondamentali forniti da un sistema operativo, ecco perché questi capitoli ora si trovano all'inizio del volume.

Il Capitolo 1 è stato modificato e aggiornato in diverse parti. Ad esempio, si propone un'introduzione al linguaggio di programmazione C e al suo modello run-time per i lettori che conoscono solamente Java.

L'analisi dei thread del Capitolo 2 è stata rivista ed espansa per rifletterne i nuovi tratti salienti. Tra le altre cose, è stato aggiunto un paragrafo sui Pthread (standard IEEE).

Il Capitolo 3, dedicato alla gestione della memoria, è stato riorganizzato per porre l'accento sull'idea che la gestione di uno spazio di indirizzi virtuali per ogni processo è oggi una delle funzionalità principali del sistema operativo. Le parti obsolete sulla gestione della memoria in sistemi batch sono state eliminate e la parte relativa all'implementazione della paginazione è stata aggiornata per mettere in evidenza la necessità di gestire spazi di indirizzi molto ampi e anche l'esigenza di velocità.

I Capitoli 4-7 sono stati rivisti rimuovendo le parti più datate e aggiungendone di nuove. I paragrafi relativi allo stato della ricerca in essere sono stati riscritti da zero. Sono stati anche aggiunti diversi nuovi problemi ed esercizi di programmazione.

Il Capitolo 8 è stato aggiornato, aggiungendo del materiale sui sistemi multicore. È stato aggiunto anche un nuovo paragrafo sulle tecnologie per la virtualizzazione, hypervisor e macchine virtuali, usando VMWare come esempio.

Il Capitolo 9 è stato pesantemente rivisto e riorganizzato, con materiale nuovo sullo sfruttamento degli errori nel codice, i malware e le possibili strategie di difesa.

Il Capitolo 10, dedicato a Linux, è una revisione del vecchio Capitolo 10 (su UNIX e Linux). Ora l'enfasi è su Linux con diverse parti nuove.

Il Capitolo 11, dedicato a Windows Vista, è una rivisitazione sostanziale del vecchio Capitolo 11 (su Windows 2000). Ora la presentazione di Windows è completamente aggiornata.

Il Capitolo 12 è nuovo. Ho pensato che i sistemi operativi embedded, quali quelli per i telefoni cellulari e i PDA, non vengono trattati in molti libri di testo, anche se al giorno d'oggi esistono (e usiamo) più cellulari che PC e notebook. Questa edizione colma tale lacuna con una presentazione dettagliata di Symbian OS, impiegato in molti telefoni cellulari.

Il Capitolo 13, sulla progettazione dei sistemi operativi, è sostanzialmente analogo a quello della seconda edizione.

Questo libro è corredato da materiale didattico on-line per docenti e studenti reperibile all'indirizzo

<http://hpe.pearson.it/tanenbaum>

I docenti che adottano il testo potranno poi richiedere la password per accedere al materiale riservato: il manuale delle soluzioni, tutte le immagini contenute nel libro e le slide in PowerPoint (in lingua inglese).

Un certo numero di persone mi ha aiutato nella stesura di questa nuova edizione. In primo luogo, voglio ringraziare il mio editor, Tracy Dunkelberger. Questo è il mio diciottesimo libro e ho una certa pratica di editor. Tracy è andata oltre le mansioni solite, trovando collaboratori, gestendo diverse revisioni, offrendo aiuto con tutto il materiale supplementare, occupandosi dei contratti, interfacciandosi con PH, coordinando una gran quantità di cose in parallelo e, in generale, facendo in modo che fossero fatte in tempo. È stata anche capace di farmi rispettare un piano di lavoro preciso e incalzante per pubblicare questo libro in tempo. Tutto ciò restando allegra e di buon umore nonostante le sue molte occupazioni. Grazie, Tracy. Ho apprezzato molto il tuo aiuto.

Ada Gavriloska di Georgia Tech, un'esperta di Linux, ha rivisto il Capitolo 10 partendo da quello su UNIX (centrato su FreeBSD) per produrne uno più focalizzato su Linux, anche se la maggior parte del capitolo è ancora generica e vale per tutti i sistemi UNIX. Linux è più diffuso tra gli studenti rispetto a FreeBSD e quindi questo è un cambiamento importante.

Dave Probert di Microsoft ha rivisto il Capitolo 11, passando da un capitolo su Windows 2000 a uno su Windows Vista. Pur essendo simili, i due sistemi operativi presentano anche differenze significative. Dave ha una conoscenza approfondita di Windows ed è sufficientemente lungimirante e intuitivo per dire dove Microsoft ha fatto o meno la scelta giusta. Il libro è stato migliorato in modo significativo grazie al suo lavoro.

Mike Jipping di Hope College ha scritto il capitolo su Symbian OS. L'assenza di qualsiasi riferimento ai sistemi operativi embedded costituiva una grave lacuna per questo libro e, grazie a Mike, è stata colmata. I sistemi operativi embedded stanno diventando sempre più importanti nel mondo e questo capitolo fornisce un'eccellente introduzione all'argomento.

A differenza di Ada, Dave e Mike, focalizzati su un singolo capitolo, Shivakant Mishra dell'Università del Colorado a Boulder ha lavorato in modo simile a un sistema distribuito, leggendo e commentando diversi capitoli e fornendo anche un buon numero di nuovi esercizi e problemi di programmazione per l'intero libro.

Anche Hugh Lauer merita una menzione speciale. Quando gli avevamo chiesto idee per aggiornare la seconda edizione, non ci aspettavamo un rapporto di 23 pagine, ma così è stato. Molti dei cambiamenti, come l'enfasi sulle astrazioni dei processi, spazi degli indirizzi e file sono dovuti ai suoi commenti.

Vorrei anche ringraziare altre persone che mi hanno aiutato in vari modi, ad esempio, suggerendo nuovi elementi, leggendo il manoscritto con cura, facendo aggiunte e fornendo esercizi. Tra questi, Steve Armstrong, Jeffrey Chastine, John Connelly, Mischa Geldermans,

Paul Gray, James Griffioen, Jorrit, Herder, Michael Howard, Suraj Kothari, Roger Kraft, Trudy Levine, John Masiyowski, Shivakant Mishra, Rudy Pait, Xiao Qin, Mark Russinovich, Krishna Sivalingam, Leendert van Doorn e Ken Wong.

Il personale di Prentice Hall si è dimostrato disponibile e di grande aiuto, come sempre, in modo particolare Irwin Zucher e Scott Disanno per la produzione e David Alick, ReeAnne Davies e Melinda Haggerty per la parte editoriale.

Infine, *last but not least*, Barbara e Marvin sono stati meravigliosi, come sempre, ognuno in un modo unico e speciale. E ovviamente, vorrei ringraziare Suzanne per il suo amore e la sua pazienza, per non parlare dei *druiven* e *kersen*, che recentemente hanno sostituito il *sinaasappelsap*.

Andrew S. Tanenbaum

Biografia

Andrew S. Tanenbaum ha conseguito la laurea presso il MIT e il dottorato di ricerca presso l'Università della California a Berkeley. Attualmente è professore di Informatica all'Università Vrije di Amsterdam (Olanda), dove dirige il gruppo di ricerca sui sistemi informatici. È stato preside della Advanced School for Computing and Imaging, un'iniziativa interuniversitaria che si occupa di ricerca su sistemi di calcolo paralleli, distribuiti e per la gestione di immagini. Attualmente è Academy Professor della Royal Netherlands Academy of Arts and Sciences, carica che l'ha salvato dal diventare un burocrate.

La sua attività di ricerca si è incentrata su compilatori, sistemi operativi, reti di calcolatori, sistemi distribuiti su scala locale e globale con miliardi di utenti. Il suo interesse principale riguarda attualmente i sistemi operativi affidabili e sicuri. Questi progetti di ricerca hanno generato più di 140 articoli pubblicati su riviste e presentati a conferenze. Il professor Tanenbaum è stato anche autore o co-autore di cinque libri, apparsi ormai in 18 edizioni. Questi libri sono stati tradotti in 21 lingue, dal basco al thailandese e adottati nelle università di tutto il mondo. In tutto, ne esistono 130 versioni (combinazioni di lingua + edizioni).

Il professor Tanenbaum ha anche prodotto una quantità considerevole di software. È stato il responsabile dell'Amsterdam Compiler Kit, un toolkit molto usato per scrivere compilatori portabili. È stato anche uno dei progettisti principali di Amoeba, uno dei primi sistemi distribuiti su un insieme di workstation collegate da una LAN, e di Globe, un sistema distribuito su scala geografica.

È anche l'autore di MINIX, un piccolo clone di UNIX originariamente pensato come ausilio per i laboratori didattici degli studenti. È stato l'ispiratore diretto di Linux e della piattaforma su cui Linux è stato sviluppato originariamente. La versione attuale di MINIX, chiamata MINIX 3, è concepita per essere un sistema operativo estremamente affidabile e sicuro; il professor Tanenbaum considererà il suo lavoro finito quando nessun computer avrà più il tasto di reset. MINIX 3 è ora un progetto open-source al quale tutti sono invitati a contribuire.

www.minix3.org

è il sito di riferimento per scaricare una copia gratuita e avere informazioni sul progetto.

Gli studenti di dottorato del professor Tanenbaum hanno ottenuto diversi successi dopo il conseguimento del titolo. Lui è molto orgoglioso di loro e in questo dimostra un notevole istinto "da chioccia".

Tanenbaum è membro di ACM, di IEEE e della Royal Netherlands Academy of Arts and Sciences. Ha anche avuto diversi riconoscimenti scientifici, tra cui:

- 1994: ACM Karl V. Karlstrom Outstanding Educator Award
- 1997: ACM/SIGCSE Award for Outstanding Contributions to Computer
- 2002: TAA Texty Award for Computer Science and Engineering
- 2003: TAA McGuffey for Computer Science and Engineering
- 2007: IEEE James H. Mulligan, Jr. Education Medal

Compare nel *Who's Who in the World*. La sua home page si trova al seguente URL:

<http://www.cs.vu.nl/~ast/>