

INTERNATIONAL
STANDARD

ISO/IEC
9945-1

ANSI/IEEE
Std 1003.1

Second edition
1996-07-12

**Information technology — Portable
Operating System Interface (POSIX®) —**

Part 1:

**System Application Program Interface (API)
[C Language]**

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/IEC 9945-1:1996

<https://standards.iteh.ai/catalog/standards/sist/27eb2c08-0313-49c2-9d14-4c/iso-iec-9945-1-1996>
Technologies de l'information — Interface pour la portabilité des systèmes
(POSIX®) —

Partie 1: Interface programme de systèmes d'application (API) [Langage C]



Reference number
ISO/IEC 9945-1:1996(E)
ANSI/IEEE
Std 1003.1, 1996 Edition

iTeh STANDARD PREVIEW
This page intentionally left blank
(standards.iteh.ai)

[ISO/IEC 9945-1:1996](https://standards.iteh.ai/catalog/standards/sist/27cb2c08-0313-49c2-9d14-a9b646cf474c/iso-iec-9945-1-1996)

<https://standards.iteh.ai/catalog/standards/sist/27cb2c08-0313-49c2-9d14-a9b646cf474c/iso-iec-9945-1-1996>

International Standard ISO/IEC 9945-1: 1996 (E)
IEEE Std 1003.1, 1996 Edition
(Incorporating ANSI/IEEE Stds 1003.1-1990, 1003.1b-1993,
1003.1c-1995, and 1003.1i-1995)

Information technology— Portable Operating System Interface (POSIX[®])—Part 1: System Application Program Interface (API) [C Language]

Sponsor

iTeh STANDARD PREVIEW
Portable Applications Standards Committee
of the **(standards.iteh.ai)**
IEEE Computer Society

[ISO/IEC 9945-1:1996](https://standards.iteh.ai/catalog/standards/sist/27cb2c08-0313-49c2-9d14-a9b646cf474c/iso-iec-9945-1-1996)

<https://standards.iteh.ai/catalog/standards/sist/27cb2c08-0313-49c2-9d14-a9b646cf474c/iso-iec-9945-1-1996>



Adopted as an International Standard by the
International Organization for Standardization
and by the



International Electrotechnical Commission



Published by
The Institute of Electrical and Electronics Engineers, Inc.



Abstract: This standard is part of the POSIX series of standards for applications and user interfaces to open systems. It defines the applications interface to basic system services for input/output, file system access, and process management. It also defines a format for data interchange. When options specified in the Realtime Extension are included, the standard also defines interfaces appropriate for realtime applications. When options specified in the Threads Extension are included, the standard also defines interfaces appropriate for multithreaded applications. This standard is stated in terms of its C language binding.

Keywords: API, application portability, C (programming language), data processing, information interchange, open systems, operating system, portable application, POSIX, programming language, realtime, system configuration computer interface, threads

POSIX is a registered trademark of the Institute of Electrical and Electronics Engineers, Inc.

iTeh STANDARD PREVIEW (standards.iteh.ai)

Quote in 8.1.2.3 on Returns is taken from ANSI X3.159-1989, developed under the auspices of the American National Standards Accredited Committee X3 Technical Committee X3J11.

<https://standards.iteh.ai/catalog/standards/sist/27c02c08-0513-49c2-9d14-a9b646cf474c/iso-iec-9945-1-1996>

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1996 by the
Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1996.
Printed in the United States of America.

ISBN 1-55937-573-6

*No part of this publication may be reproduced in any form,
in an electronic retrieval system or otherwise,
without the prior written permission of the publisher.*

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Contents

	PAGE
Foreword	xi
Introduction	xii
Organization of This Part of ISO/IEC 9945	xii
Base Documents	xiii
Background	xiii
Audience	xiv
Purpose	xiv
Related Standards Activities	xix
Section 1: General	1
1.1 Scope	1
1.2 Normative References	3
1.3 Conformance	3
Section 2: Terminology and General Requirements	11
2.1 Conventions	11
2.2 Definitions	12
2.3 General Concepts	30
2.4 Error Numbers	33
2.5 Primitive System Data Types	37
2.6 Environment Description	38
2.7 C Language Definitions	40
2.8 Numerical Limits	47
2.9 Symbolic Constants	52
Section 3: Process Primitives	57
3.1 Process Creation and Execution	57
3.1.1 Process Creation	57
3.1.2 Execute a File	59
3.1.3 Register Fork Handlers	64
3.2 Process Termination	65
3.2.1 Wait for Process Termination	65
3.2.2 Terminate a Process	68
3.3 Signals	70
3.3.1 Signal Concepts	70
3.3.2 Send a Signal to a Process	79
3.3.3 Manipulate Signal Sets	80
3.3.4 Examine and Change Signal Action	82
3.3.5 Examine and Change Blocked Signals	84
3.3.6 Examine Pending Signals	86
3.3.7 Wait for a Signal	87
3.3.8 Synchronously Accept a Signal	88

3.3.9	Queue a Signal to a Process	90
3.3.10	Send a Signal to a Thread	92
3.4	Timer Operations	93
3.4.1	Schedule Alarm	93
3.4.2	Suspend Process Execution	94
3.4.3	Delay Process Execution	94
Section 4:	Process Environment	97
4.1	Process Identification	97
4.1.1	Get Process and Parent Process IDs	97
4.2	User Identification	98
4.2.1	Get Real User, Effective User, Real Group, and Effective Group IDs	98
4.2.2	Set User and Group IDs	98
4.2.3	Get Supplementary Group IDs	100
4.2.4	Get User Name	101
4.3	Process Groups	102
4.3.1	Get Process Group ID	102
4.3.2	Create Session and Set Process Group ID	103
4.3.3	Set Process Group ID for Job Control	103
4.4	System Identification	105
4.4.1	Get System Name	105
4.5	Time	106
4.5.1	Get System Time	106
4.5.2	Get Process Times	106
4.6	Environment Variables	108
4.6.1	Environment Access	108
4.7	Terminal Identification	108
4.7.1	Generate Terminal Pathname	108
4.7.2	Determine Terminal Device Name	109
4.8	Configurable System Variables	110
4.8.1	Get Configurable System Variables	110
Section 5:	Files and Directories	113
5.1	Directories	113
5.1.1	Format of Directory Entries	113
5.1.2	Directory Operations	113
5.2	Working Directory	116
5.2.1	Change Current Working Directory	116
5.2.2	Get Working Directory Pathname	117
5.3	General File Creation	118
5.3.1	Open a File	118
5.3.2	Create a New File or Rewrite an Existing One	122
5.3.3	Set File Creation Mask	122
5.3.4	Link to a File	123
5.4	Special File Creation	125
5.4.1	Make a Directory	125
5.4.2	Make a FIFO Special File	126
5.5	File Removal	127
5.5.1	Remove Directory Entries	127

5.5.2	Remove a Directory	129
5.5.3	Rename a File	130
5.6	File Characteristics	132
5.6.1	File Characteristics: Header and Data Structure	132
5.6.2	Get File Status	134
5.6.3	Check File Accessibility	136
5.6.4	Change File Modes	137
5.6.5	Change Owner and Group of a File	139
5.6.6	Set File Access and Modification Times	140
5.6.7	Truncate a File to a Specified Length	142
5.7	Configurable Pathname Variables	143
5.7.1	Get Configurable Pathname Variables	143
Section 6: Input and Output Primitives		147
6.1	Pipes	147
6.1.1	Create an Inter-Process Channel	147
6.2	File Descriptor Manipulation	148
6.2.1	Duplicate an Open File Descriptor	148
6.3	File Descriptor Deassignment	149
6.3.1	Close a File	149
6.4	Input and Output	151
6.4.1	Read from a File	151
6.4.2	Write to a File	153
6.5	Control Operations on Files	156
6.5.1	Data Definitions for File Control Operations	156
6.5.2	File Control	156
6.5.3	Reposition Read/Write File Offset	161
6.6	File Synchronization	163
6.6.1	Synchronize the State of a File	163
6.6.2	Synchronize the Data of a File	164
6.7	Asynchronous Input and Output	165
6.7.1	Data Definitions for Asynchronous Input and Output	165
6.7.2	Asynchronous Read	168
6.7.3	Asynchronous Write	170
6.7.4	List Directed I/O	172
6.7.5	Retrieve Error Status of Asynchronous I/O Operation	175
6.7.6	Retrieve Return Status of Asynchronous I/O Operation	176
6.7.7	Cancel Asynchronous I/O Request	177
6.7.8	Wait for Asynchronous I/O Request	178
6.7.9	Asynchronous File Synchronization	180
Section 7: Device- and Class-Specific Functions		183
7.1	General Terminal Interface	183
7.1.1	Interface Characteristics	183
7.1.1.1	Opening a Terminal Device File	183
7.1.1.2	Process Groups	183
7.1.1.3	The Controlling Terminal	184
7.1.1.4	Terminal Access Control	184
7.1.1.5	Input Processing and Reading Data	185
7.1.1.6	Canonical Mode Input Processing	186

7.1.1.7	Noncanonical Mode Input Processing	186
7.1.1.8	Writing Data and Output Processing	187
7.1.1.9	Special Characters	187
7.1.1.10	Modem Disconnect	189
7.1.1.11	Closing a Terminal Device File	189
7.1.2	Parameters That Can Be Set	189
7.1.2.1	<i>termios</i> Structure	189
7.1.2.2	Input Modes	190
7.1.2.3	Output Modes	191
7.1.2.4	Control Modes	192
7.1.2.5	Local Modes	193
7.1.2.6	Special Control Characters	194
7.1.2.7	Baud Rate Values	195
7.1.3	Baud Rate Functions	195
7.1.3.1	Synopsis	195
7.1.3.2	Description	196
7.1.3.3	Returns	196
7.1.3.4	Errors	196
7.1.3.5	Cross-References	196
7.2	General Terminal Interface Control Functions	197
7.2.1	Get and Set State	197
7.2.2	Line Control Functions	199
7.2.3	Get Foreground Process Group ID	201
7.2.4	Set Foreground Process Group ID	202
(standards.iteh.ai)		
Section 8:	Language-Specific Services for the C Programming Language	205
8.1	Referenced C Language Routines	205
8.1.1	Extensions to Time Functions	206
8.1.2	Extensions to <i>setlocale()</i> Function	208
8.2	C Language Input/Output Functions	209
8.2.1	Map a Stream Pointer to a File Descriptor	210
8.2.2	Open a Stream on a File Descriptor	211
8.2.3	Interactions of Other <i>FILE</i> -Type C Functions	212
8.2.4	Operations on Files — the <i>remove()</i> Function	216
8.2.5	Temporary File Name — the <i>tmpnam()</i> Function	216
8.2.6	Stdio Locking Functions	216
8.2.7	Stdio With Explicit Client Locking	217
8.3	Other C Language Functions	218
8.3.1	Nonlocal Jumps	218
8.3.2	Set Time Zone	219
8.3.3	Find String Token	219
8.3.4	ASCII Time Representation	220
8.3.5	Current Time Representation	221
8.3.6	Coordinated Universal Time	222
8.3.7	Local Time	223
8.3.8	Pseudo-Random Sequence Generation Functions	223
Section 9:	System Databases	225
9.1	System Databases	225
9.2	Database Access	226

9.2.1	Group Database Access	226
9.2.2	User Database Access	227
Section 10:	Data Interchange Format	231
10.1	Archive/Interchange File Format	231
10.1.1	Extended tar Format	231
10.1.2	Extended cpio Format	235
10.1.3	Multiple Volumes	239
Section 11:	Synchronization	241
11.1	Semaphore Characteristics	241
11.2	Semaphore Functions	241
11.2.1	Initialize an Unnamed Semaphore	241
11.2.2	Destroy an Unnamed Semaphore	243
11.2.3	Initialize/Open a Named Semaphore	244
11.2.4	Close a Named Semaphore	246
11.2.5	Remove a Named Semaphore	247
11.2.6	Lock a Semaphore	248
11.2.7	Unlock a Semaphore	250
11.2.8	Get the Value of a Semaphore	251
11.3	Mutexes	252
11.3.1	Mutex Initialization Attributes	253
11.3.2	Initializing and Destroying a Mutex	255
11.3.3	Locking and Unlocking a Mutex	257
11.4	Condition Variables	259
11.4.1	Condition Variable Initialization Attributes	259
11.4.2	Initializing and Destroying Condition Variables	261
11.4.3	Broadcasting and Signaling a Condition	263
11.4.4	Waiting on a Condition	264
Section 12:	Memory Management	267
12.1	Memory Locking Functions	268
12.1.1	Lock/Unlock the Address Space of a Process	268
12.1.2	Lock/Unlock a Range of Process Address Space	270
12.2	Memory Mapping Functions	272
12.2.1	Map Process Addresses to a Memory Object	272
12.2.2	Unmap Previously Mapped Addresses	276
12.2.3	Change Memory Protection	277
12.2.4	Memory Object Synchronization	279
12.3	Shared Memory Functions	281
12.3.1	Open a Shared Memory Object	281
12.3.2	Remove a Shared Memory Object	284
Section 13:	Execution Scheduling	287
13.1	Scheduling Parameters	287
13.2	Scheduling Policies	287
13.2.1	SCHED_FIFO	288
13.2.2	SCHED_RR	289
13.2.3	SCHED_OTHER	289
13.3	Process Scheduling Functions	290

13.3.1	Set Scheduling Parameters	290
13.3.2	Get Scheduling Parameters	291
13.3.3	Set Scheduling Policy and Scheduling Parameters	293
13.3.4	Get Scheduling Policy	294
13.3.5	Yield Processor	295
13.3.6	Get Scheduling Parameter Limits	296
13.4	Thread Scheduling	297
13.4.1	Thread Scheduling Attributes	298
13.4.2	Scheduling Contention Scope	298
13.4.3	Scheduling Allocation Domain	299
13.4.4	Scheduling Documentation	300
13.5	Thread Scheduling Functions	300
13.5.1	Thread Creation Scheduling Attributes	300
13.5.2	Dynamic Thread Scheduling Parameters Access	302
13.6	Synchronization Scheduling	304
13.6.1	Mutex Initialization Scheduling Attributes	304
13.6.2	Change the Priority Ceiling of a Mutex	307
Section 14: Clocks and Timers		309
14.1	Data Definitions for Clocks and Timers	309
14.1.1	Time Value Specification Structures	309
14.1.2	Timer Event Notification Control Block	310
14.1.3	Type Definitions	310
14.1.4	Manifest Constants	310
14.2	Clock and Timer Functions	311
14.2.1	Clocks	311
14.2.2	Create a Per-Process Timer	312
14.2.3	Delete a Per-Process Timer	314
14.2.4	Per-Process Timers	315
14.2.5	High Resolution Sleep	317
Section 15: Message Passing		319
15.1	Data Definitions for Message Queues	319
15.1.1	Data Structures	319
15.2	Message Passing Functions	320
15.2.1	Open a Message Queue	320
15.2.2	Close a Message Queue	323
15.2.3	Remove a Message Queue	324
15.2.4	Send a Message to a Message Queue	325
15.2.5	Receive a Message From a Message Queue	327
15.2.6	Notify Process That a Message is Available on a Queue	328
15.2.7	Set Message Queue Attributes	329
15.2.8	Get Message Queue Attributes	331
Section 16: Thread Management		333
16.1	Threads	333
16.2	Thread Functions	333
16.2.1	Thread Creation Attributes	333
16.2.2	Thread Creation	336
16.2.3	Wait for Thread Termination	337

16.2.4	Detaching a Thread	338
16.2.5	Thread Termination	339
16.2.6	Get Thread ID	340
16.2.7	Compare Thread IDs	341
16.2.8	Dynamic Package Initialization	342
Section 17:	Thread-Specific Data	343
17.1	Thread-Specific Data Functions	343
17.1.1	Thread-Specific Data Key Creation	343
17.1.2	Thread-Specific Data Management	344
17.1.3	Thread-Specific Data Key Deletion	346
Section 18:	Thread Cancellation	347
18.1	Thread Cancellation Overview	347
18.1.1	Cancelability States	347
18.1.2	Cancellation Points	348
18.1.3	Thread Cancellation Cleanup Handlers	349
18.1.4	Async-Cancel Safety	349
18.2	Thread Cancellation Functions	350
18.2.1	Canceling Execution of a Thread	350
18.2.2	Setting Cancelability State	351
18.2.3	Establishing Cancellation Handlers	352
18.3	Language-Independent Cancellation Functionality	353
18.3.1	Requesting Cancellation	353
18.3.2	Associating Cleanup Code With Scopes	353
18.3.3	Controlling Cancellation Within Scopes	354
18.3.4	Defined Cancellation Sequence	354
18.3.5	List of Cancellation Points	354
Annex A (informative)	Bibliography	355
A.1	Related Open Systems Standards	355
A.2	Other Standards	357
A.3	Historical Documentation and Introductory Texts	358
A.4	Other Sources of Information	360
Annex B (informative)	Rationale and Notes	361
B.1	Scope and Normative Cross-References	361
B.2	Definitions and General Requirements	373
B.3	Process Primitives	413
B.4	Process Environment	446
B.5	Files and Directories	453
B.6	Input and Output Primitives	464
B.7	Device- and Class-Specific Functions	481
B.8	Language-Specific Services for the C Programming Language	492
B.9	System Databases	504
B.10	Data Interchange Format	505
B.11	Synchronization	512
B.12	Memory Management	530
B.13	Execution Scheduling	543
B.14	Clocks and Timers	552

B.15	Message Passing	560
B.16	Thread Management	564
B.17	Thread-Specific Data	572
B.18	Thread Cancellation	576
Annex C (informative) Header Contents Samples		587
Annex D (informative) Profiles		607
D.1	Definitions	607
D.2	Options in This Part of ISO/IEC 9945	609
D.3	Related Standards	609
D.4	Related Activities	609
D.5	Relationship to IEEE Std 1003.0-1995	610
Annex E (informative) Sample National Profile		611
E.1	(Example) Profile for Denmark	612
Annex F (informative) Portability Considerations		615
F.1	User Requirements	615
F.2	Portability Capabilities	617
F.3	Profiling Considerations	622
Annex G (informative) Performance Metrics		631
G.1	Performance Measurement Documentation	631
G.2	Signals	632
G.3	Synchronized Input and Output	633
G.4	Asynchronous Input and Output	635
G.5	Semaphores	637
G.6	Mutexes and Condition Variables	638
G.7	Process Memory Locking	639
G.8	Shared Memory	640
G.9	Execution Scheduling	640
G.10	Clocks and Timers	643
G.11	Message Passing	645
G.12	Thread Management	646
G.13	Thread Cancellation	647
Annex H (informative) Realtime Files		649
H.1	Data Definitions for Realtime Files	650
H.2	Realtime File Functions	656
H.3	Realtime Files	668
H.4	Rationale for Realtime Files	670
H.5	Realtime File Functions	673
Identifier Index		679
Alphabetic Topical Index		691

TABLES

Table 2-1	– Primitive System Data Types	38
Table 2-2	– Reserved Header Symbols	42
Table 2-3	– Minimum Values	48
Table 2-4	– Run-Time Inceasable Values	49
Table 2-5	– Run-Time Invariant Values (Possibly Indeterminate)	50
Table 2-6	– Pathname Variable Values	51
Table 2-7	– Invariant Value	52
Table 2-7a	– Maximum Values	52
Table 2-8	– Symbolic Constants for the <i>access()</i> Function	53
Table 2-9	– Symbolic Constants for the <i>lseek()</i> Function	53
Table 2-10	– Compile-Time Symbolic Constants	54
Table 2-11	– Execution-Time Symbolic Constants	55
Table 3-1	– Required Signals	71
Table 3-2	– Job Control Signals	71
Table 3-3	– Memory Protection Signals	72
Table 4-1	– <i>uname()</i> Structure Members	105
Table 4-2	– Configurable System Variables	111
Table 5-1	– <i>stat</i> Structure	132
Table 5-2	– Configurable Pathname Variables	144
Table 6-1	– <i>cmd</i> Values for <i>fcntl()</i>	157
Table 6-2	– File Descriptor Flags Used for <i>fcntl()</i>	157
Table 6-3	– <i>l_type</i> Values for Record Locking With <i>fcntl()</i>	157
Table 6-4	– <i>oflag</i> Values for <i>open()</i>	157
Table 6-5	– File Status Flags Used for <i>open()</i> and <i>fcntl()</i>	158
Table 6-6	– File Access Modes Used for <i>open()</i> and <i>fcntl()</i>	158
Table 6-7	– Mask for Use With File Access Modes	158
Table 6-8	– <i>flock</i> Structure	159
Table 6-9	– <i>fcntl()</i> Return Values	160
Table 7-1	– <i>termios</i> Structure	190
Table 7-2	– <i>termios c_iflag</i> Field	190
Table 7-3	– <i>termios c_cflag</i> Field	192
Table 7-4	– <i>termios c_lflag</i> Field	193
Table 7-5	– <i>termios c_cc</i> Special Control Characters	194
Table 7-6	– <i>termios</i> Baud Rate Values	195
Table 9-1	– <i>group</i> Structure	226
Table 9-2	– <i>passwd</i> Structure	228
Table 10-1	– <i>tar</i> Header Block	232
Table 10-2	– Byte-Oriented <i>cpio</i> Archive Entry	236
Table 10-3	– Values for <i>cpio c_mode</i> Field	237
Table 12-1	– Memory Protection Values	273
Table B-1	– Suggested Feature Test Macros	409
Table F-1	– Portability Capability Summary	618

Foreword

1 ISO (the International Organization for Standardization) and IEC (the International
2 Electrotechnical Commission) form the specialized system for worldwide
3 standardization. National bodies that are members of ISO or IEC participate in
4 the development of International Standards through technical committees established
5 by the respective organization to deal with particular fields of technical
6 activity. ISO and IEC technical committees collaborate in fields of mutual
7 interest. Other international organizations, governmental and nongovernmental,
8 in liaison with ISO and IEC, also take part in the work.

9 In the field of information technology, ISO and IEC have established a joint technical
10 committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint
11 technical committee are circulated to national bodies for voting. Publication as an
12 International Standard requires approval by at least 75% of the national bodies
13 casting a vote.

14 International Standard ISO/IEC 9945-1: 1996 was prepared by Joint Technical
15 Committee ISO/IEC JTC 1, *Information technology*. This edition cancels and
16 replaces the first edition (ISO/IEC 9945-1:1990), which has been technically
17 revised. This edition also consolidates Amendments 1 and 2 and Technical Corri-
18 gendum 1 to ISO/IEC 9945-1:1990.

19 ISO/IEC 9945 consists of the following parts, under the general title *Information*
20 *technology—Portable operating system interface (POSIX)*:

- 21 — *Part 1: System application program interface (API) [C language]*
- 22 — *Part 2: Shell and utilities*
- 23 — *Part 3: System administration* (under development)

24 Annexes A to H of ISO/IEC 9945-1 are provided for information only. |