

N95-14165

59-62

27239

P-31

1995 107751

MISSION
SAFETY CRITICAL
SYSTEMS

RESEARCH & APPLICATIONS

ricis

SYMPOSIUM '92

POSIX REAL-TIME EXTENSIONS

Henry H. Robbins

ABSTRACT

POSIX is an evolving set of operating system interface standards, whose parts are in varying stages of production in a number of standards working groups. This presentation separates the real-time POSIX standards work in progress from the rest and provides an overview of the purpose, status, dependencies, content, and projected schedule of each.

BIOGRAPHY

Mr. Robbins is an Advisory Systems Engineer with the IBM Federal Systems Company. His recent work includes software engineering support of NASA JSC's Space Station Control Center, which requires the use of POSIX operating systems and the Ada language. He is also IBM's representative to the IEEE TCOS 1003.5 (Ada Bindings) Working Group, which is currently addressing Ada bindings to real-time POSIX.

POSIX Real-Time Extensions

October 29, 1992

Henry H. Robbins

IBM Federal Systems Company
3700 Bay Area Blvd.
Houston, Texas 77058

(713) 335-3297

robbinsr@houvmssc.vnet.ibm.com



—

—

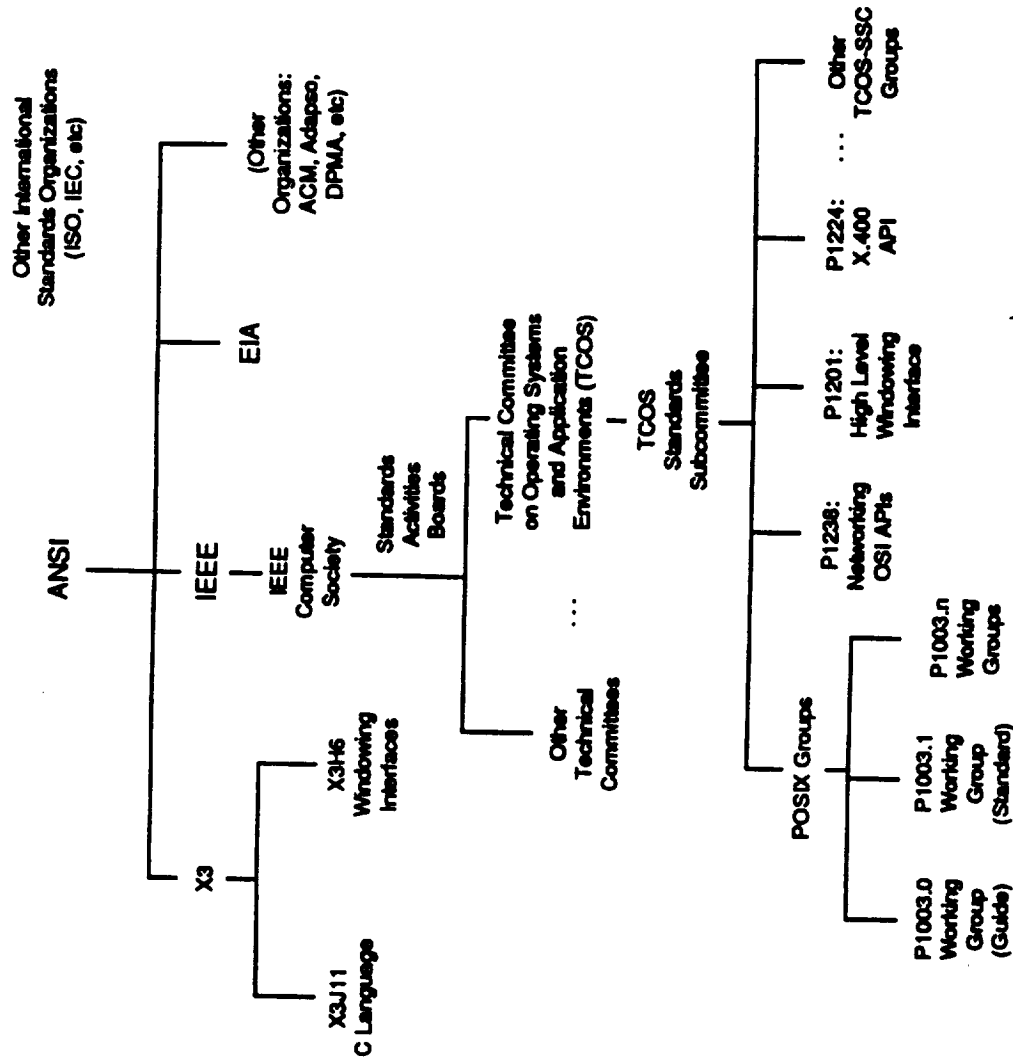
8

—

1. What is POSIX?

- POSIX stands for **P**ortable **O**perating **S**ystem Interface for **(X)** Computer Environments
- POSIX defines a standard operating system *interface* and environment to support application portability at the source code level.
 - The interface is UNIX-like, but does not require an underlying UNIX operating system
 - The POSIX Open Systems Environment has the goals of:
 - Source Code **Portability** through system/application interface standards
 - **Interoperability** of applications resident on dissimilar computer systems
 - **User Portability** from system to system and between applications on the same system with a minimum of retraining
- In the U.S., POSIX standards are written by working groups of the IEEE Computer Society's Technical Committee on Operating Systems

2. IEEE Standards Diagram



3. IEEE TCOS POSIX Standards Process

- Working Group submits a Program Authorization Request (PAR) to IEEE TCOS
- Draft standard written
- Mock Ballot on draft standard
- Modified draft standard
- Balloting group formed
- Formal ballot on draft standard
 - 75% approval required (plus other rules)
 - Otherwise iterate
- Submit to IEEE Standards Board for Approval



October 29, 1992

4. POSIX Standards and Drafts

Standard	Description	Status
P1003.0	Guide to Open Systems Environment	1st Ballot Complete
P1003.1	System API (C Binding)	Approved (POSIX .1-1990)
P1003.1-LI	Language Independent System API	Currently in 1st Ballot
P1003.1a	System API Extensions	1st Ballot - 1993
P1003.2	Shells and Utilities	Approved - Sept. 1992
P1003.2b	Shells and Utilities - Amendment 2	Initial Work
P1003.3	Test Methods	Approved (POSIX .3-1991)
P1003.3.1	Test Methods for 1003.1	Final Ballot Complete
P1003.3.2	Test Methods for 1003.2	1st Ballot - 4Q92
P1003.4	Real Time and Related System API	Final (?) Ballot - 4Q92
P1003.4a	Threads API	Next Ballot - 1/93
P1003.4b	System API Extensions	Next Ballot - 1/93
P1003.4c	Language Independent Spec	Initial Work - 1/93
P1003.5	Ada Bindings (System API)	Approved - June 1992
P1003.6	Security	Next Ballot - 4Q92
P1003.7.1	Print Administration	1st Ballot - 1/93
P1003.7.2	Software Administration	Mock Ballot - 1Q93
P1003.7.3	User Administration	Initial Work (no PAR)
P1003.8	Transparent File Access	Next Ballot - 10/92

5. POSIX Standards and Drafts (Continued)

Standard	Description	Status
P1003.9	FORTAN 77 Bindings	Approved - June 1992
P1003.10	Supercomputing Appl. Environment Profile	1st Ballot - 10/92
P1003.11	Transaction Proc. Appl. Environment Profile	In Ballot Resolution
P1003.12	Protocol Independent Interface	Mock Ballot 4Q92
P1003.13	Real Time Appl. Environment Profile	1st Ballot - 10/92
P1003.14	Multiprocessing Appl. Environment Profile	1st Ballot - 4Q92
P1003.15	Batch Environment Amendments	1st Ballot - 4Q92
P1003.16	C Language Bindings for 1003.1-LI	In Ballot
P1003.17	Directory Server API	In Ballot
P1003.18	Platform Environment Profile	1st Ballot soon
P1003.19	FORTAN 90 Binding to 1003.1-LIS	Initial Work
P1003.20	Ada Bindings (Real Time)	Mock Ballot 1/93
P1003.20a (?)	Ada Bindings (Threads API)	Initial Work (PAR in progress)
P1201.1	Window Intf. for User and Appl. Portability	1st Ballot - 1993
P1201.2	General User Interfaces - Drivability	In Ballot Resolution
P1224	ASN.1 Object Management API	In Ballot Resolution
P1224.1	X.400 API	In Ballot Resolution
P1238.0	Connection Management API	1st Ballot - 2Q93
P1238.1	FTAM API	Initial Work



October 29, 1992

6. Real Time POSIX PARs

Draft Standard	Description
1003.4 Working Group	
P1003.4	Real Time and Related System API
P1003.4a	Threads API
P1003.4b	System API Extensions
P1003.4c	Language Independent Spec
P1003.13	Real Time Appl. Environment Profile
1003.5 Working Group	
P1003.20	Ada Bindings (Real Time)
P1003.20a (?)	Ada Bindings (Threads API)

7. Terminology - Application Environment Profile

- Most POSIX standards are in one of two flavors:

Application Programming Interface (API) The interface between the application software and the application platform, across which all services are provided. P1003.4, P1003.4a, and P1003.4b are all APIs. APIs specify interface standards, but operating systems do not have to implement all of their specified services.

Application Environment Profile (AEP) A profile, specifying a completed and coherent specification of the POSIX Open Systems Environment, in which the standards, options and parameters chosen are necessary to support a class of applications. P1003.13 is an AEP.

- The intent is for operating systems to claim compliance to an AEP, instead of APIs.

8. Terminology - Language Independent Service Specification (LIS)

- An API specification that facilitates the management and development of consistent language binding standards
- Originally, all POSIX API specifications were to be in the C language, with other languages, like Ada, binding to C.
- Several years ago, the rules were changed to require language-independent API specifications with a separate C-language binding to it. Other languages, like Ada, would bind to the LIS.
- P1003.1 (System API (C Binding)), P1003.4 (Real Time and Related System API), and P1003.5 (Ada Bindings (System API)) were "grandfathered" and follow the old rules.
- **Note: P1003.20 (Ada Bindings (Real Time) - not "grandfathered") cannot bind to P1003.4, which is a "grandfathered" C API specification.**

9. P1003.4 Real Time and Related System API

- P1003.4 is the *process-level* API specification for Realtime services. It does not support concurrency within a POSIX process.
- It enhances and adds to the services specified by P1003.1.
- Current Draft is Draft 12.
- Ballots on draft 12 presently stand at 70% approval with many abstentions for lack of time. Most of the objections were on *seminit()*.
- The chairman of the working group expects to be able to get over 75% approval during ballot resolution.
- The working group currently expects to have a circulation of 10-15 pages of updates (only) for ballot in October 1992.
- The December 1992 date for adoption of .4 is optimistic. A more likely date would be March 1993.

10. P1003.4 Real Time and Related System API (Contents)

Revisions to Process Primitives: Contains the realtime signal enhancement to assure signal delivery by queuing signals. P1003.1 signals are not queued and may be lost if another signal follows.

Revisions to Process Environment: Adds to the list of P1003.1 configurable system variables.

Revisions to Files and Directories: Augments I/O primitives defined in P1003.1 and P1003.1a.

Revisions to Input and Output Primitives: Augments P1003.1 Synchronous I/O to guarantee I/O completion to the physical medium. Also supplies Asynchronous I/O, permitting the calling process to continue as soon as the I/O is queued; completion notification is via a POSIX signal.

11. P1003.4 Real Time and Related System API (Contents Continued)

Synchronization:

Provides *counting* semaphores. This changed from binary semaphores in Draft 12. This area still needs some work.

Memory Locking:

Standardizes modern UNIX practice providing the ability to lock physical memory from paging and swapping.

Memory Mapped Files and Shared Memory: A shared memory facility similar to the *mmap* interface in AT&T's System V, plus an interface for generalized file mapping.

Execution Scheduling:

Provides FIFO and round-robin priority scheduling, as well as an interface to set the scheduling method.



October 29, 1992

12. P1003.4 Real Time and Related System API (Contents Continued)

- Clocks and Timers:** A time service loosely based on Berkeley interval timers, allowing for completely time-driven processing.
- Interprocess Message Passing:** Provides priority Message Queues with signal notification.
- Realtime Files:** Provides a mechanism to manipulate realtime attributes of files, such as the preallocation of disk blocks and direct unbuffered I/O to a disk file.



October 29, 1992

13. P1003.4 Real Time and Related System API (Schedule)

Milestone	Projected Date
Expect Next Draft	October 1992
Next Ballot	October 1992
Standard Approved	December 1992

14. P1003.4a Threads API

- P1003.4a is the *thread-level* API specification for Realtime services. It does support concurrency within a POSIX process.
- It enhances and adds to the services specified by P1003.1 and P1003.4.
- It was split off from P1003.4 because of the difficulty in reaching consensus due to varying goals within the working group, especially between Realtime and Multiprocessor advocates.
- Current Draft is Draft 6.
- Ballot objections against draft 6 are at 5000, up from 3000 against the previous draft.
- These objections are being resolved and the working group expects the next draft around the end of 1992.
- P1003.4a must still be converted to a LIS.

15. P1003.4a Threads API (Contents)

Definitions and General Requirements: Extensions to definitions, error numbers, data types, numerical limits, and symbolic constants to support threads

Thread Management: Creation and termination of threads, plus thread ID operations

Synchronization Primitives: Operations on Mutexes and condition variables

Thread-Specific Data: Key/value mechanism to associate an opaque key with each per-thread datum. These keys play the role of identifiers for per-thread data.

Thread Priority Scheduling: Operations on the attributes of thread scheduling. This includes the setting of priority ceilings for Mutexes.

16. P1003.4a Threads API (Contents Continued)

Process Control:

Operations to support *fork()* and *exec()* on processes in a threads environment.

Signals:

Extensions to realtime signals in a threads environment, including per-thread signal masks, per-process signal vectors, and single delivery of each signal

Thread Cancellation:

Operations that allow the cancellation of another thread

Reentrant Functions:

Extensions to other P1003.1 and P1003.4 routines to allow them to work in a multi-threaded environment

17. P1003.4a Threads API (Schedule)

Milestone	Projected Date
Expect Next Draft	December 1992
Next Ballot	January 1993
Standard Approved	December 1993

18. P1003.4b System API Extensions

- P1003.4b provides additional realtime services
- These services are considered to be at a lower priority than those in P1003.4 and P1003.4a.
- Current Draft is Draft 4.
- Recent developments:
 - The proposed chapter on asynchronous services was dropped because nobody was interested in the function.
 - The working group created one new chapter (Device Control), made changes to all four existing chapters, and created corresponding LIS chapters.
- Standard approval is several years away

19. P1003.4b System API Extensions (Contents)

Process Primitives:

spawn(): *fork()* followed by *exec()* in one
syscall

Timeouts for blocking services: Additional interfaces for blocking real time services are provided. These interfaces allow the specification of a time out value. The service is not allowed to block longer than the specified amount of time.

Execution Time Monitoring:

Accumulate CPU time used by individual threads and processes

Sporadic server:

Introduces a scheduling class for aperiodic events

Device control:

Defines a standard way for controlling non-standard devices, similar to *ioctl()* but less controversial



October 29, 1992

20. P1003.4b System API Extensions (Schedule)

Milestone	Projected Date
Expect Next Draft	September 1992 (A)
Next Ballot	January 1993
Standard Approved	?

21. P1003.4c Language Independent Spec

- P1003.4c will be the Language Independent Services Specification corresponding to P1003.4 (Real Time and Related System API).
- The 1003.5 (Ada Bindings) working group is the group primarily interested in the LIS for .4 since, by the requirements of the P1003.20 PAR, it cannot bind to the existing P1003.4.
- In the July meeting, Offer Pazy became the book manager.
- A P1003.4c draft is needed by next spring to meet the P1003.20 schedule.

22. P1003.13 Real Time Appl. Environment Profile

- P1003.13 consists of four profiles ranging from a minimal system for unattended control to a large multi-purpose system.
- Current Draft is Draft 5.
- The first ballot (draft 5) produced over 75% approval. BUT:
 - The group has to wait until .4 and .4a, the underlying standards, have been approved by the IEEE.
 - The group has to split the profiles into four separate documents, one for each profile.
- There was an interesting flap in the July meeting over a proposed TCOS requirement that all application environment profiles include all of P1003.1 and P1003.2. Most realtime folks fail to see the utility of a user shell in the head of a missile.

23. P1003.13 Real Time Appl. Environment Profile (Contents)

Minimal Realtime System Profile: Embedded system dedicated to unattended control of one or more special I/O devices. No user interaction or file system required. Supports a single P1003.1 process with one or more POSIX.4a threads and perhaps an IPC interface for communication with like systems.

Realtime Controller System Profile: Like minimal-profile systems, with the addition of a file system interface, character by character serial I/O interfaces, asynchronous (non-blocking) I/O interfaces, and P1003.1 signals

Dedicated Realtime System Profile: An extension of the minimal-profile system, with support for multiple processes and most P1003.4 functionality except Realtime Files. There is a common interface for device drivers and files, but no hierarchical file system.



October 29, 1992

24. P1003.13 Real Time Appl. Environment Profile (Contents Continued)

Multi-Purpose Realtime System Profile: Provides comprehensive functionality and runs a mix of differing realtime and non-realtime tasks. Includes P1003.1, P1003.2 (user shell), and P1003.4. P1003.4a threads support is optional. Additional functionality is provided by options for networking, windowing, and programming languages.

25. P1003.13 Real Time Appl. Environment Profile (Schedule)

Milestone	Projected Date
Expect Next Draft	October 1992
Next Ballot	October 1992
Standard Approved	?

26. P1003.20 Ada Bindings (Real Time)

- Currently a thin, direct binding to P1003.4, based on Dr. Ted Baker's work for the Army. Contents are the same as for P1003.4.
- A mock ballot is scheduled for January.
- Before P1003.20 can enter the formal balloting process, it will have to become a binding to P1003.4C, the Realtime Language Independent Services Specification.
- Draft 0.5 of P1003.20 held up very well in the July meeting under very close scrutiny, and very few changes were recommended. The group discussed many potential issues, but only recommended a few minor changes in the areas of clocks and timers, plus three issues to present to the 1003.4 working group in clocks and timers and in message queues.
- P1003.20 will be a binding to Ada 83, perhaps with some extensions to take advantage of Ada 9x capabilities.

27. P1003.20 Ada Bindings (Real Time) Schedule

Milestone	Projected Date
Mock ballot mailing (binding to P1003.4)	Oct. 1992
Mock ballot	Jan. 1993 - Feb. 1993
Mock ballot resolution	Mar. 1993 - July 1993
1st ballot (binding to .4 LIS)	Oct. 1993 - Dec. 1993
1st ballot resolution	Jan. 1994 - July 1994
1st recirculation ballot	Aug. 1994 - Oct. 1994
1st recirculation ballot resolution	Jan. 1995 - April 1995
2nd recirculation ballot	May 1995 - June 1995
Approved draft standard to IEEE	Sept. 1995

28. P1003.20a (?) Ada Bindings (Threads API)

- P1003.20a will be an Ada binding to the LIS equivalent of P1003.4a (Threads API). Schedule tbd; name may change when PAR is approved.
- The July decision to move the work of binding to P1003.4a to a separate PAR was made to accommodate the projected schedule for P1003.4a and the resulting LIS, which is estimated to be at least two years behind the P1003.4 schedule. The 1003.5 group intends to split its time between P1003.20 and P1003.20a.
- The working group will concentrate on a threads model in which all threads created by Ada are created by the RTE as Ada tasks. This eliminates the need to bind to *pthread_create()*, *pthread_join()*, *pthread_detach()*, *pthread_cancel()*, and *pthread_exit()*.
- The working group also decided that the problems of Ada thread inter-operation with C threads were too complex for the standard: for example, if a bound C subroutine creates a thread, can Ada rendezvous with it? Any such facilities will be left up to the implementers.

29. Suggested Reading

- In the October 1992 issue of Embedded Systems Programming (pp 28-40):
 - "Real-Time POSIX", by Bill O. Gallmeister (vice-chair of the 1003.4 working group)
- For POSIX.1 basics:
 - Lewine, Donald, *POSIX Programmers Guide*, Sebastapol, Calif.: O'Reilly and Associates, 1991

1

1

1

1

1