

NICHOLAS M.
STOUGHTON
AND ANDREW JOSEY

USENIX standards activities



Nick is the USENIX Standards Liaison and represents the Association in the POSIX, ISO C, and LSB working groups. He is the ISO organizational representative to the Austin Group, a member of INCITS committees J11 and CT22, and the Specification Authority subgroup leader for the LSB.

nick@usenix.org



Andrew Josey is the director of Certification within The Open Group and chairs the Austin Group, the working group responsible for development of the joint revision to POSIX and the Single UNIX Specification.

a.josey@opengroup.org

An Update on Standards: Diary of a Standard Geek

NICK STOUGHTON

Have you ever wondered just what happens at a standards committee, what it is like to attend an international meeting to determine the future of a standard that affects millions of your fellow workers? Here are some notes from a trip to Berlin in April 2006 for the ISO-C committee (officially known as ISO/IEC JTC 1 SC 22/WG 14, it's charged with c99, a.k.a. ISO-C, the language accepted by most modern compilers).

Saturday: I get on a plane to go to Germany for an ISO-C meeting (a three-leg flight from Oakland, through Dallas and Zurich), arriving in Berlin sometime tomorrow. I have 14 hours to reread all of the documents submitted for the meeting (11 papers and 2 draft documents). Well, that took up one hour . . . just another 13 to go.

Monday: The meeting starts at 9:30 at the DIN headquarters. Is this the place they invented DIN plugs? Well, yes. But standards shouldn't be about invention (though in the case of electrical connectors, they often are). We start gathering at about 8:45—there's plenty of coffee and pastries on offer in the meeting room (thanks to SAP). There are 26 of us, representing five national bodies. All participation in ISO meetings is by national body (otherwise known as a "country"). In the case of ISO-C, it is also jointly developed by ANSI (the U.S. national body), and so the U.S. contingent is 20 of the 26. (The astute mathematicians will figure out that the delegations from the other countries are modest in comparison.)

Every working group has its own methods for achieving consensus; in ISO-C we try as hard as possi-

ble to avoid formal votes. We regularly will stop proceedings to hold straw votes. These are taken to get a sense of the room on an issue. They are nonbinding, but they often stop us from going down ratholes where it is clear that there's only one or two people who believe in a given direction.

However, meetings are formal. They have a definite pattern, with an agenda to work through and, often, time limits. Low-level working groups, such as ISO-C or the Austin Group (where POSIX gets written and maintained) are driven by technical matters. We will spend an hour arguing over whether or not an optimizer is allowed to reorder certain memory accesses for performance, especially if the program happens to be multi-threaded (and this particular question won't go away any time soon). But this meeting, as are most all standards meetings, is more for direction setting. The "real" work of such a working group happens in the papers that are developed between meetings.

Monday morning is spent working through administrivia, liaison reports (including two from me, one from POSIX, and one from the FSG on LSB status), and potential defect reports. One of the other style issues that differs from working group to working group is how defect reports are handled. In the ISO-C case, defect reports can be raised either through a national body (e.g., the UK can submit a defect report directly) or by individuals who submit them to the chair, and they then get considered during this "potential defect" agenda slot. At this point, we simply have to agree whether or not they are defects, and if so, they get added to the list of defects we will work on later in the week. This time, I get to submit a small handful

on behalf of the Austin Group, where a conflict between C and POSIX appears to be present.

Monday lunch has me off to a local hotel for lunch provided by SAP. This is a rare treat . . . we usually don't get lunch provided for us!

Monday afternoon: Two papers are to be considered, one on "Managed Strings" and one (coming from the C++ committee) to add, to the floating point handling, macros to handle a maximum number of significant decimal digits. Although the concepts in the "Managed Strings" paper were interesting, they were all invention, with little existing practice, and overlapped with a paper I'm developing on I/O functions that use dynamic memory. The decimal digits work could be passed off to a subgroup who are writing a Technical Report on Decimal Floating Point (to align with the new revision of IEEE 754).

After coffee, it is time to talk about the "Security" TR (Technical Report). You may recall I have written about this in previous columns. At least it's now called "the bounds checking TR." It is looking like it may be ready for its next ballot at this point, and it appears to me now to be mostly harmless! However, members of the Austin Group had had serious concerns over it, and I presented a paper from them on these concerns. The author agreed to write a response to this paper. This may need a few discussions in the bar tonight!

Tuesday: It's time to talk about the decimal floating point document. This meeting is pleasant, because there is little political controversy, and for the majority of the meeting I can simply concentrate on the technical aspects of what we are doing. Now, if this had been SC 22, the parent

committee in ISO, the tone would be very different, and the discussions in the bar would have a very different feel to them!

Floating point is not one of my favorite subjects, so I spend much of the time while this is being discussed preparing for the Defect Report work we will be starting this afternoon.

In fact, it seems that most of us in the room feel like this . . . there are only three people talking, and everyone else is frantically clicking the keyboards on their laptops. Maybe they are reading their email. But you can't stop listening to the subject matter. Maybe there will be something that matters to POSIX, or to the LSB. No . . . that was too much to hope for!

Defect Reports take up much of the week. We start on Tuesday and will finish sometime Thursday. We work through the log of defects (which has just gotten longer because of the potential defects we turned into actual defects yesterday), usually starting from the top and going to the end. Those old, old defects at the top of the list have been looked at many times; we just can't find the right answer. Sometimes we will assign a small group to go off and consider a response (if there is anyone prepared to serve on such a breakout group). Or we'll give homework to one individual to write a response. If you want to see what a defect report looks like, look at <http://www.open-std.org/jtc1/sc22/wg14/www/docs/summary.htm>.

Wednesday: More defect reports are discussed. You'd think a language as well established as C wouldn't have a lot of defects in the specification. But it is hard to write a specification, and one that is heavily used will always have many issues in it. Not all of them are bugs in the standard.

Often it is a misunderstanding on the part of the submitter, or a question that is outside the scope of the standard, or any of a host of other issues.

"If an incomplete array type has elements of unknown size, should the incomplete array type be a VLA type?"

"Must bit fields of type char nevertheless have the same signedness as ordinary objects of type char?"

"What if `asctime()` is called with a `tm` structure whose `tm_year` field results in a year > 9999?"

"The first sentence of 6.7.5.2p2 seems to suggest that any ordinary identifier can have both block scope and function prototype scope and no linkage has a variably modified type. This is clearly wrong."

And so on . . .

Thursday: Wow! It looks like we might actually finish the agenda early for the week! We are done with defects. We just have the final reports from the defect review and the closing business to get through! Where's the next meeting? Portland, Oregon. What about the one after that? We have an invitation from the U.K. Then there's the action item review: who has to do what? by when? I have to help write the response to the Austin Group's concerns on the bounds checking TR (or at least ensure that the response is delivered). And I have to write the dynamic memory I/O functions report. My work is cut out for the next meeting. I can finally enjoy a couple of hours in Berlin before my flight home this evening.

Why Get Involved in POSIX?

ANDREW JOSEY

One of the key factors leading to success for Linux and the UNIX system has been the adoption of popular, open standards such as the X Window System, TCP/IP, and the POSIX standards. Today we see a rapid evolution of IT systems and applications brought about by the adoption of the Internet and the changes that has brought to the way we work. But are the standards evolving fast enough to keep pace with the changes? This article gives a high-level look at the current POSIX standardization activity, how it works, and how you can contribute to helping it keep pace.

What is POSIX? POSIX, a registered trademark of the IEEE, is an acronym for “Portable Operating System Interfaces.” The name POSIX was suggested by Richard Stallman in 1986. The most well known POSIX standard is IEEE Std. 1003.1 (or ISO Std. 9945, which is the same document), known for short as “POSIX.1.” It specifies application programming interfaces (APIs) at the source level and is about source code portability. It is neither a code implementation nor an operating system, but a standard definition of a programming interface that those systems supporting the specification guarantee to provide to the application programmer. Both Operating System Vendors (OSVs) and Independent Software Vendors (ISVs) have implemented software that conforms to this standard.

The major sections of POSIX.1 are definitions, utilities (such as `awk`, `grep`, `ps`, and `vi`), headers (such as `unistd.h`, `sys/select.h`, and other C headers), threads, networking, real time, interna-

tionalization, and math functions. In total, the standard describes over 1,350 interfaces.

If POSIX.1 is mentioned as a requirement for your software project, that does not tell you much. POSIX.1 is large (3,600 pages) and no project needs everything (even OSVs rarely implement every optional interface). The POSIX.1 standard is structured into modules known as option groups. A minimal set of interfaces and functionality is required for all POSIX systems. The majority of all functionality is optional. For a good description of options and their status in Linux and glibc, see <http://people.redhat.com/~drepper/posix-option-groups.html>.

There are several common misconceptions about POSIX. Since its development started in the mid 1980s, one common misconception is that it has not changed for some time; it is outdated and irrelevant. The latest version of the POSIX.1 standard was produced in 2001 and updated in 2004: it is known as IEEE Std. 1003.1, 2004 Edition. Work is now underway to revise the standard to produce a new revision in 2008. Although the new versions of the standard are in general upwardly compatible with the original, many hundreds of interfaces that have been added since then. Your participation is needed to help keep it up to date and to keep pace with the developments in the industry.

Another common misconception is that you need to be an IEEE member to participate. The latest edition was developed by the Austin Group, an open working group found at <http://www.opengroup.org/austin/>. Participation is free and open to all interested parties (you just need to join the mailing list). Decisions are made by consensus;

sometimes consensus is reached easily, and sometimes only after heated discussion! The more people involved in such discussions, the more likely it is that when consensus is reached, the right decision has been made. That's why your participation is so important. Readers should note, however, that the mailing lists are not a technical support forum. All the major UNIX players and open source distributions are represented in the Austin Group.

Today the approach to the POSIX standard development is one of “write once, adopt everywhere,” with a single open technical working group and the resulting documents being adopted by IEEE as the POSIX standard, adopted by The Open Group as the Base Specifications of the Single UNIX Specification, and by the International Standards Organization as an international standard (which in turn means that it may be a national standard in your country; for example, the British Standards Institute has adopted ISO 9945 as a BS).

So does this mean that the POSIX.1 standard is complete and perfect? No. Like any large product, it has bugs, and there is an ongoing bug reporting and fixing process to refine the document as implementation experience grows. Although the standard cannot change overnight, there is a mechanism both to make regular technical corrections and also to collect items for future directions. To report a bug or suggest a change, please use the defect report form at <http://www.opengroup.org/austin/defectform.html>.

Is POSIX still relevant? Yes. Standard interfaces mean easier porting of applications. The POSIX interfaces are widely implemented and referenced in other standardization efforts, includ-

ing the Single UNIX Specification and the Linux Standard Base.

Why should you get involved? Feeding back issues with the standard based on implementation experience allows the standard to be improved and extended with new functionality, which in turn can “raise the bar

of commonality” among systems. There is often much more to be gained by having key functionality share a common interface and/or behave in exactly the same way, than for it to be different.

More information on POSIX.1 and the Austin Group, including how to join and participate, is

available from its Web site at <http://www.opengroup.org/austin/>.

The html version of the standard is freely available from The Open Group’s Single UNIX Specification Web site at <http://www.unix.org/version3/>.

Renew Your USENIX and SAGE Memberships Online Today!

Renewing your USENIX and SAGE memberships has never been easier. Visit <http://www.usenix.org/membership> and click on the appropriate links.

In addition to your subscription to *login:*, your USENIX benefits include:

- Online access to all Conference Proceedings from 1993 to the present
- Substantial discounts on technical sessions registration fees for all USENIX-sponsored events
- Jobs Board and Resume posting
- And more . . .

Are you a SAGE member too? SAGE membership benefits include:

- Short Topics booklets in both online and paper versions
- Discount on registration for LISA, the annual Large Installation System Administration conference
- Jobs Board and Resume posting
- Access to sage-members mailing list and the archives
- And more . . .

Don’t miss out on the benefits that help you keep up with the latest technologies, network with your peers, and find out about new job opportunities.

Renew both memberships today! <http://www.usenix.org/membership>