-- --

IEEE P1003.0 Draft 13 – September 1991

Section 4: POSIX Open System Environment Services

Responsibility: Fritz Schulz

This section describes the services required in support of the objectives identified in this guide. The services are grouped in major categories defined in Section 3, with more detailed breakdowns within each category as appropriate. These categories are:

Fundamental System Services

- 4.1 Language Services
- 4.2 System Services

Communications Services

- 4.3 Network Services

Information Services

- 4.4 Database Services
- 4.5 Data Interchange Services

Human-Computer Interaction Services

- 4.6 Windowing System Services D
- 4.7 Graphic Services
- 4.8 Character-Based User Interface Services
- 4.9 User Command Interface Services

Domain Services

- 4.10 Transaction Processing Services D
- 4.11 Software Development Environment Services

Criteria used to partition services are outlined in 3.2, and discussed at the beginning of each clause. The discussion for each of the service category subclauses follows the same outline, and is as follows:

4.*n*.1
4.*n*.2 Overview and Rationale, Scope

This text introduces the scope of this service category, and the criteria used to identify the services within it.

4.*n*.3 Reference Model

This subclause builds on the model of clause 3.2 and gives additional detail related to the interfaces and services discussed there. An optional subclause may discuss implementation considerations, similar to the discussion of 3.6.

4.*n*.4 Service Requirements

This text provides the definition of service requirements within the scope described in 4.*n*.2.

4.*n*.5 Standards, Specifications, and Gaps

A table lists the standards and specifications available to meet the service requirements listed in 4.*n*.4. This is followed by a brief discussion of services for which standards are not available.

4.*n*.5.1 Current Standards

The following subclauses cite existing specifications that have been approved as standards by accredited standards bodies, in the order of precedence identified in 3.4.2. When service requirements are satisfied at a higher precedence level, specifications at a lower level are not listed.

4.*n*.5.2 Emerging Standards

The following subclauses list specifications and/or activities that address the functional areas within the 4.*n* section, but which have not yet been completed. Where a group or activity is cited, the charter of the group may address the functionality, but it is possible that a draft may not be available. Only those services not currently addressed by existing standards are to be discussed in this subclause. It is expected that documents will migrate from 4.*n*.5.2 to 4.*n*.5.1 as they complete the consensus process.

4.*n*.5.3 Gaps in Available Standards

This subclause identifies those service requirements that have not been satisfied by existing or emerging standards. If all service requirements in this category have been met by existing or emerging standards, this subclause will be empty. Text in this subclause will be minimal.

4.*n*.5.3.1 Public Specifications

This subclause lists any specification outside of the formal standards community that is available to anyone (e.g., no membership required) for implementation and distribution (including sale) without restriction, including all government and de facto standards.

4.*n*.5.3.2 Unsatisfied Service Requirements

This subclause lists the services for which no specification has been cited in this guide. Products may be cited here to illustrate capabilities that are not addressed by standards.

4.*n*.6 POSIX OSE Cross-Category Services

This subclause contains any discussion of the Cross-Category Services in Section 5 that is specific to subclause 4.*n*.

4.*n*.7 Related Standards

This subclause is optional and may identify interdependencies among standards that should be taken into account when selecting among them.

4.*n*.8 Open Issues

This subclause is optional and may identify issues under discussion in the open systems community.

4.1 Language Services

Responsibility: Don Folland

4.1.1 Overview and Rationale

While a consistent interface to the operating system is essential for applications portability, the application will have been developed using language and system development tools that, in turn, require to be supported by standards to achieve source code portability.

Those responsible for system or software development will wish to write programs in code supported by an international standard and compile the code using a compiler that has a certificate of conformance issued by an accredited test center. Noncompliant extensions must be avoided if applications portability is to be maintained. Compilers should identify nonstandard-compliant code.

The languages that have been identified in this document are those seen to be in most popular use today for software development. The POSIX.2 shell command language is discussed in 4.9. The standards identified are the most widely recognized today, with significant use in the Information Technology industry on a broad range of processors, or where a large installed base of a particular version is known to exist.

4.1.2 Scope

The services described in this clause cover the most widely used third-generation computer languages in use today for the development of applications; i.e., the languages used to write application programs. Fourth-generation languages are not currently addressed in this guide. In order for a program to address an API to the services described in other clauses of this guide, an appropriate language binding to that interface is required. References to those bindings will be found in the clause describing the relevant service.

4.1.3 Reference Model

This subclause identifies the entities and interfaces supporting language services. The reference model based on the reference model in Figure 3-1 is illustrated in Figure 4-1, but because the language services directly support the binding of the applications to the API, there is no EEI. However, the EEI is shown in Figure 4-1 for consistency.

At the simplistic level, the programmer developing an application that requires only basic operating system services will use a compiler that meets both the fundamental language standard (e.g., ISO 1989: 1985 for COBOL, ISO 1359: 1990 for Fortran) and the binding established for the relevant system calls in POSIX.1 {2}.

As identified in 4.10, an application program may also require database services that will be provided by the Database Manager API. The database vendor will