

操作系统标准之一

计算机环境的
可移植操作系统界面
POSIX.1

(ISO/IEC DIS 9945-1.2 译文)

中软总公司第二开发部 译

周明德 田云解 华审校

电子工业出版社

操作系统标准之一

计算机环境的可移植操作系统界面

POSIX. 1

(ISO/IEC DIS 9945-1.2 译文)

中软总公司第二开发部 译

周明德 田云 解华 审校

电子工业出版社

内 容 提 要

本书是操作系统国际标准之一——POSIX.1的译文。POSIX.1的目的是基于UNIX操作系统文档定义一个标准的操作系统界面和环境,以便支持源码级的应用例程的可移植性。POSIX.1主要侧重于操作系统的C语言界面。

POSIX.1是ISO提出的国际标准,也是我国操作系统国家标准的基础,因此本书对于操作系统的实现者特别广大的应用软件开发者是必不可少的手册。

计算机环境的可移植操作系统界面(POSIX.1)

中软总公司第二开发部 译

周明德 田云 解华 审校

责任编辑:潘海

电子工业出版社出版(北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京科技印刷厂印刷

开本:787×1092毫米 1/16 印张:15.25 字数:390千字

1991年5月第1版 1991年5月第1次印刷

印数:1—4000册 定价:20元

ISBN 7-5053-1408-4/TP·236

译者的话

在 90 年代,随着 386、486 等高档个人计算机和各种工程工作站的迅速发展,基于 UNIX 的多用户,多任务操作系统也必将迅速推广。在美国以至世界的计算机市场上,UNIX 产品的增长速度超出了计算机工业平均增长速度的一倍。美国的最主要的计算机厂商都进入了 UNIX 世界。工作站和 UNIX 系统产品成为计算机市场激烈竞争的又一个焦点。

用户的需求,决定了 90 年代的计算机必然朝开放系统的方向发展,应用软件的可移植性成为软件开发的重要目标。为此,美国 IEEE 建立了 P1003 委员会,制订操作系统以及整个应用环境的标准。随着工作的深入和发展,P1003 委员会建立了十个工作小组分别制订有关的标准。目前正式通过了 1003.1 和 1003.2 两个标准,其他标准正在积极制订中。

ISO(国际标准化组织)和 IEC(国际电子技术委员会)以 IEEE std 1003.1—1988 为基础提出了 ISO/IEC DIS 9945—1.2。随着 IEEE 制订的标准发展,相信 ISO/IEC 也会发布相应的标准。

操作系统是最重要、最基本的系统软件,随着我国高档个人计算机和工程工作站研制、开发、生产的发展,开发一个国产的多用户、多任务操作系统是我国计算机和软件行业进一步发展的一个关键问题,特别在我国的著作权法和软件保护条例即将执行的形势下更为重要,所以已列入国家“八五”重点攻关项目之一,且由机电部下达任务,开发工作已经启动。操作系统的国家标准制订工作也已经在进行。

为了实现我国开发生产的系统及产品与世界上的主流产品相兼容,为了实现应用软件源码级的可移植性,我国制订的操作系统国家标准以及国产的多用户多任务操作系统必须与国际标准相一致。故项目组在开发过程中翻译了 IEEE std 1003.1—1988 和 ISO/IEC DIS 9945—1.2。

在 1990 年 8 月由机电部下达任务,正式成立了操作系统国家标准工作组,准备提出国家标准的征求意见稿。征求意见稿显然必须以国际标准的译文为基础,故工作组责成中软总公司把中软总公司和中科院软件实验室的两个译文综合起来,并经过了工作组的专家认真、细致的审阅,正式通过了国际标准的译文,并由中软总公司以此译文为基础提出国家标准的征求意见稿。

本书就是国际标准的译文稿。我们把译文稿提供给广大读者是基于:一方面国家标准必然是基于国际标准的,所规定的功能和应用界面必然是与国际标准相一致的;另一方面国家标准尚有征求意见、送审等一连串的过程,需要较长时间。而 UNIX 系统的应用已经在国内迅速推广,及早地把国际标准介绍给广大读者,不论是对于大家学习和熟悉 UNIX 系统以及 UNIX 系统的实现者和以 UNIX 系统为基础的应用软件开发者都是十分重要的,是一本不可缺少的法典式的工具书。

本书内容分成正文和附录两部分。正文部分规定了标准中所用的术语和系统最主要的配置参数。正文又分六章规定了一个开放式操作系统核心所应该提供的功能(以 C 语言的函数形式规定),也规定了此标准与 C 语言之间的协约(所引用的 C 语言的函数以及对 C 语言的补充和扩展)。附录部分主要是说明了标准对已存在的 UNIX 系统的功能采纳与不采纳的理由,以及对标准中一些较复杂、较难理解的内容的说明和注释。所以,虽然附录不是标准的正文,但对于深入地理解和掌握标准是必不可少的。

本书是操作系统国家标准工作组审查通过的国际标准的译文。许多同志参加了翻译也经过许多同志的反复校对,推敲。最后由周明德、田云、解华定稿。由于参加翻译、校对、录入、修改的同志很多,这里不能一一列出。谨向所有参加工作的同志致谢。

中软总公司系统软件工程部
一九九一年二月七日

前 言

ISO (国际标准化组织)和 IEC(国际电子技术委员会)一起组成一个系统,形成全世界范围内的整体的标准。ISO 或 IEC 的各成员国的标准化委员会通过该技术委员会参加国际标准的制订,这些技术委员会由处理技术活动的各特殊领域的组织建立。ISO 和 IEC 技术委员会在相互有兴趣的领域里合作。其它政府的和非政府的国际组织也与 ISO 和 IEC 协作参加这一工作。

在信息技术领域里,ISO 和 IEC 已经建立了一个联合技术委员会——ISO/IEC JTC 1,由该联合技术委员会采纳的国际标准草案在它们被接收为国际标准之前要得到各国标准化委员会的批准。依据审批例程,至少需要 75%国家标准化委员会的赞成票才能获得通过。

国际标准 ISO/IEC 9945—1:1990 是由联合技术委员会 ISO/IEC JTC 1——信息技术委员会准备的。

在信息技术这一总标题下,ISO/IEC 9945 由下列部分组成:

- 第一部分:计算机环境的可移植操作系统界面
- 第二部分:计算机操作系统环境的 shell 和实用程序的应用界面(称为 ISO/IEC CD9945—2.1 正在制订)
- 第三部分:计算机操作系统环境的系统管理界面(正在制订)

ISO/IEC 9945—1 的附录 A 到附录 E 仅仅是提供某种信息。

引 言

ISO/IEC 9945 的这部分的目的是基于 UNIX 操作系统文档定义一个标准的操作系统界面和环境,以便支持源码级的应用例程的可移植性。它主要由系统实现者和应用软件开发人员使用。

按它目前的形式看,ISO/IEC 9945 的这部分主要侧重于操作系统的 C 语言界面。

在本份文档中,术语“POSIX. 1”表示 ISO/IEC 9945 的这部分本身。

标准的组织

ISO/IEC 9945 的这部分分成四大部分:

- (1)范围的陈述(第一章)
- (2)定义和全局概念(第二章)
- (3)各种界面设施(第三章到第九章)
- (4)数据交换格式(第十章)

大多数章节单独描述一种服务界面。服务界面的 C 语言协约在各个函数的格式小节内给出。描述小节提供由服务界面执行的操作的说明。可能提供某些实例解释所描述的接口。在大多数情况下,返回和出错小节规定了返回值和可能的出错条件。参阅小节用于指导读者参照其它有关的章节。对 ISO/IEC 9945 的这部分一些章节作补充的其它材料可以在附录 B 依据及注释中找到。该附录提供了 ISO/IEC 9945 的这部分的制订者作出的技术选择的历史上已存在的观点。它亦提供一些信息以强调在 ISO/IEC 9945 的这部分的对应章节所描述的界面的结果。

非正式的附录(在每一个附录的开始都是这么标明的)都不是本标准的组成部分而只是作为信息提供。(还有一类称为“正式的”附录,它们是标准的一个部分并且强调需要,但是在 ISO/IEC 9945 的这部分中当前没有这种正式的附录。)由于它们不象本标准的主体那样作为严格表决的对象,因此不能把它们看成是决定性的陈述意见而依据它们,它们仅提供作为指南和帮助理解。

在出版 ISO/IEC 9945 的这部分时,它的制订者直截了当地提供了一种度量各种操作系统实现一致性的尺度。衡量或估价任何产品,奖励或承认任何产品的厂家与 ISO/IEC 9945 的这部分一致或不一致,或者试图通过这样或那样的手段强调 ISO/IEC 9945 的这部分,这些都不是制订者所想干的。确定与 ISO/IEC 9945 的这部分一致性的程度或不足的责任仍仅仅取决于分别的评价声称与 ISO/IEC 9945 的这部分一致的产品的人。

基本文档

这里描述的各种界面设施是基于 UniForum(其前身是/usr/group)标准化委员会(Santa Clara, California, USA)提出并出版的“1984 /usr/group 标准”、“1984/usr/group 标准”并且 ISO/IEC 9945 的这部分主要基于那些 UNIX V. 7、UNIX System III、UNIX System V、4. 2BSD 和 4. 3BSD 文档,但是,只要可能就维持出自于 UNIX 操作系统或与 UNIX 兼容的系统的系统兼容性。附录 B 依据及注释中的材料部分来自于在 UniForum 赞助下制订的有版权的草案文档,作为该协会正在进行中的支持 POSIX 的一个组成部分。

ISO/IEC JTC 1 非常感谢 AT&T 和 UniForum 允许我们使用它们材料。

背景

POSIX. 1 基于 UNIX 操作系统,并以 UniForum(其前身是/usr/group)标准化委员会的工作为起点。UniForum 是在 UNIX 系统上有兴趣的个人、公司、学术团体组成的协会,它已经在制订独立的工业标准方面作了长期的工作。本标准的制订者代表了 UNIX 系统社团的各个侧面:它由 450 多个成员组成,代表了硬件厂商、操作系统和其它软件开发工具的销售商、软件设计者、顾问、学者、作家、应用系统程序员及其他人。在审议期间,制订者已经查阅了相关的美国标准和国际标准,包括已发表的和正在制订的。

从理论上讲,本标准描述了一组对于有效的构造应用程序必需的基本服务。已经通过用 C 编程语言定义一种界面来提供对这些服务的访问,这种界面建立了标准语义和语法。由于这种界面可以让应用程序的编写者编写可以移植的程序(该界面的开发一直以此为目的),因此它已被称作 POSIX (Portable Operating System Interface for computer environments),可移植操作系统界面的缩写。使用名字 POSIX 是理查德·斯特乐曼的建议。

术语 POSIX 虽然原先用来表示 IEEE Std 1003. 1—1988,但是更确切地说是用来表示相关的标准——IEEE 1003. n 和国际标准化 ISO/IEC 9945 部分的一个族。在较早的版本中术语 POSIX 用作 IEEE Std 1003. 1—1988 的同义词。现在提倡使用更恰当的术语 POSIX. 1。它维持了符号“POSIX”的可读性的优点而不会与 POSIX 标准族发生矛盾。

术语 POSIX 希望象 positive 那样,读成 pahz—icks 而不是 poh—six 或其它音。已经发表了其术语的读音是力图颁布一个表示标准操作系统界面的一种途径。

读者

本标准预期的读者是所有关心基于 UNIX 系统的工业标准操作系统的人,至少包括下列四组人:

- (1) 购买硬件和软件系统的人;
- (2) 管理将来与计算机方向有关的公司的人;
- (3) 尤其是实现操作系统的人;
- (4) 以可移植性为目标的应用程序的开发者。

目的

若干原则制约了本标准的制订。

面向应用

其基本目的是,通过为基于 UNIX 系统文档的可移植操作系统的界面规范制订一种清晰、一致并且无二义性的标准,促进不同 UNIX 系统环境间应用程序的可移植性。这个标准整理了 UNIX 系统的共同且现存的定义。它并不想定义一个新的系统界面。

界面,而不是实现

本标准定义了一种界面而不是实现。在库函数和系统调用之间不作区分:两者都作为函数引用。没有指出任何函数实现的细节(虽然在论证中有时指出了以往的实践)。常量(比如信号和出错编号)用符号而不是用数字指出。

源码兼容性而不是目标码兼容性

编制本标准的目的是使得在一个一致的实现上编写并为执行而转换的程序也可以搬到另一个一致的实现上为执行而转换。即使基础硬件相同,标准也不保证在一个实现上转换的可执行码(目标码或二进制码)可在另一个一致的实现上执行。然而,在二进制的兼容性方面不设置什么障碍,某些要点可在“依据”中找到。参见 B. 1. 2. 1. 1 和 B. 4. 8。

C 语言

本标准用由 C 标准(2)中规定的标准 C 语言的术语编写。参见 B. 1. 2 和 B. 1. 1. 1。

无超级用户,无系统管理

这里没有规定操作系统的所有方面。标准中排除了系统管理设施和功能,只能由超级用户使用的函数也没有包括进来。依据指出了若干种这样的情况。当然也可以在一种标准界面的实现中也可以实现并非本标准的特性。参见 1. 2. 1. 1。本标准也不考虑硬件约束或系统维护。

最小的界面,尽可能少的定义

为了保持 UNIX 系统过去的设计原理,本标准尽可能地小。例如,它通常规定一组函数只实现一种功能。但某些情况,也有一些例外,如长期的习惯以及许多已经存在的应用中包含了一些函数如 creat()。此时,就象整个标准中所支持的避免冗余的定义:把 creat() 定义成 open() 的一个特例。排除了传统中较少使用的冗余函数或实现。

可广泛实现

本标准的制订者已经尽力使所有规定的函数可在广泛存在的及和潜在的系统上实现。这些系统包括:

- (1)来自原始 AT&T^① 系统(V7 或其后的版本)代码的所有主要流行的主要系统;
- (2)虽不是来自于原始 AT&T 系统但却与之兼容的系统;
- (3)基于完全不同原始系统上的模拟系统;
- (4)网络式系统;
- (5)分布式系统;
- (6)运行在广泛硬件上的系统。

对于这个目标的直接引用并未出现在本标准中,但是某些结果却在论证中提到了。

对以往的实现修改尽可能少

尚不知是否有已存在的实现不必要在某些方面作修改就与本标准是一致的。在少量的领域本标准并不精确匹配任何现存的系统界面(作为例子,参见 B. 6 中 O-NONBLOCK 的讨论)。不过,有一些函数、类型、定义以及概念组成了以往的大多数实现共有的界面。本标准规定了那个公共界面并且在以往没有一致意见的那些方面对它进行了扩展,宁可

- (1)通过标准化一个象以往实现中已有的那样的界面,例如,目录,或
- (2)通过规定一个按现存实现的形式可实现的并且兼容的界面,例如,10. 1. 1 中扩展的 tar 格式,或
- (3)通过规定一个加到以往的实现中时不与之矛盾的界面,象 B. 6 那样。

对以往的实现所需的修改尽可能地少,但是总有一些要修改,依据中罗列了其中的一些。本标准并不是一个特定厂家的产品说明书。它象 UNIX 系统,但并不等同于它。由于这个原因,词 UNIX

^① UNIX 是 AT&T 的在美国和其他国家的注册商标

不能用于标准本身,因为它是一个特定厂家的商标。应该注意到,实现将有不同种类的扩展。某些扩展将反映“以往的用法”,并为已经存在的应用程序的执行而保留。对于新的应用程序来说,应该用标准规定的函数而不要用这些函数。某些扩展将表示 POSIX.1 范围以外的函数,这些函数要小心管理以便适应于将来的 POSIX.1 的扩展,并/或适应以不同方式提供这些服务的实现。

对现存的应用程序修改尽量少

本标准的一个目的是使应用程序开发者尽可能减少额外的工作。然而,因为每一种以往的实现至少要作少量修改才能与标准一致,所以某些应用程序要作修改。依据部分指出本标准中隐含这种修改的一些主要地方。

本标准的修订和扩展

正在进行一系列活动来扩展 ISO/IEC 9945 的这部分以满足其他一些需求,估计类似的工作可以在不久的将来进行。这些扩展如何组合在一起,下面是一个轮廓。

下列方面是现在正在考虑实施的工作,或者期望在不久的将来开始实施。

- (1) shell 和实用程序功能
- (2) 验证测试
- (3) 实时功能
- (4) Ada 语言协约
- (5) 安全/可信的系统考虑
- (6) 与语言独立的服务描述(这是 ISO/IEC 9945 的这部分将来修订的对象)
- (7) FORTRAN 语言协约
- (8) 网络界面功能
- (9) 系统管理
- (10) 基于 POSIX 或相关的开放系统标准的全局指南

如果你有兴趣参加上面所列的工作组的工作,请把你的名字,地址及电话号码寄至:

Secretary, IEEE Standard Board

Institute of Electrical and Electronics Engineers, Inc.

P. O. Box 1331

445 Hoes Lane

Prscalaway, NJ 08855-1331

并请转告相应 TCOS 工作组的主席。

计算机环境的可移植操作系统界面(POSIX.1)

目 录

引言	(1)
1 范围和引用标准	(1)
1.1 范围	(1)
1.2 一致性	(2)
1.3 引用标准	(4)
2 定义和通用要求	(6)
2.1 约定	(6)
2.2 定义	(6)
2.3 一般概念	(12)
2.4 出错号	(13)
2.5 基本系统数据类型	(16)
2.6 环境描述	(16)
2.7 C语言定义	(18)
2.8 数字限定值	(22)
2.9 符号常量	(24)
3 进程原语	(26)
3.1 进程的创建和执行	(26)
3.1.1 进程的创建	(26)
3.1.2 执行文件	(27)
3.2 进程终止	(29)
3.2.1 等待进程终止	(30)
3.2.2 终止进程	(32)
3.3 信号	(32)
3.3.1 信号的概念	(32)
3.3.2 给进程发信号	(36)
3.3.3 信号集操作	(37)
3.3.4 测定和改变信号动作	(38)
3.3.5 测定和改变被阻滞的信号	(39)
3.3.6 测定挂起的信号	(40)
3.3.7 等待一个信号	(41)
3.4 报时操作	(41)
3.4.1 报时安排	(42)
3.4.2 挂起进程执行	(42)
3.4.3 延迟进程执行	(42)
4 进程环境	(44)

4.1	进程标识	(44)
4.1.1	取进程 ID 和父进程 ID	(44)
4.2	用户标识	(44)
4.2.1	取实际用户 ID、有效用户 ID、实际组 ID、有效组 ID	(44)
4.2.2	置用户 ID 和组 ID	(45)
4.2.3	取附加组 ID	(46)
4.2.4	取用户名	(46)
4.3	进程组	(47)
4.3.1	取进程组 ID	(47)
4.3.2	创建会晤期并置进程组 ID	(47)
4.3.3	为作业控制设置进程组 ID	(48)
4.4	系统标识	(48)
4.4.1	系统名	(48)
4.5	时间	(49)
4.5.1	取系统时间	(49)
4.5.2	进程时间	(49)
4.6	环境变量	(50)
4.6.1	环境访问	(50)
4.7	终端标识	(51)
4.7.1	生成终端路径名	(51)
4.7.2	确定终端设备名	(51)
4.8	可配置的系统变量	(52)
4.8.1	取可配置的系统变量	(52)
5	文件与目录	(54)
5.1	目录	(54)
5.1.1	目录项格式	(54)
5.1.2	目录操作	(54)
5.2	工作目录	(56)
5.2.1	改变当前工作目录	(56)
5.2.2	工作目录路径名	(56)
5.3	创建普通文件	(57)
5.3.1	打开一个文件	(57)
5.3.2	创建一个新文件或改写一个已存在的文件	(59)
5.3.3	设置文件创建屏蔽	(59)
5.3.4	链接一个文件	(60)
5.4	创建特别文件	(61)
5.4.1	创建一个目录	(61)
5.4.2	创建一个 FIFO 特别文件	(62)
5.5	文件的删除	(62)
5.5.1	删除目录项	(62)
5.5.2	删除一个目录	(63)
5.5.3	文件改名	(64)
5.6	文件特性	(65)
5.6.1	文件特性: 前导文件和数据结构	(65)
5.6.2	取文件状态	(67)

5.6.3	文件访问权	(68)
5.6.4	改变文件方式	(69)
5.6.5	改变文件属主或文件组	(69)
5.6.6	设置文件的访问及修改时间	(70)
5.7	可配置的路径名变量	(71)
5.7.1	获得可配置的路径名变量	(71)
6	输入和输出原语	(74)
6.1	管道	(74)
6.1.1	创建一个进程之间的通道	(74)
6.2	文件描述字操作	(74)
6.2.1	复制一个打开的文件描述字	(74)
6.3	文件描述字的回收	(75)
6.3.1	关闭一个文件	(75)
6.4	输入和输出	(76)
6.4.1	读文件	(76)
6.4.2	写文件	(78)
6.5	文件控制操作	(80)
6.5.1	文件控制操作的数据定义	(80)
6.5.2	文件控制	(81)
6.5.3	重定位读/写文件位移量	(84)
7	设备和设备类的专用函数	(86)
7.1	通用终端界面	(86)
7.1.1	界面特性	(86)
7.1.2	可设置的参数	(90)
7.2	通用终端界面控制函数	(96)
7.2.1	取和置状态	(97)
7.2.2	行控制函数	(98)
7.2.3	取前台进程组 ID	(99)
7.2.4	置前台进程组 ID	(100)
8	关于 C 程序设计的专用的服务	(101)
8.1	引用的 C 语言例程	(101)
8.1.1	对时间函数的扩展	(102)
8.1.2	对 setlocale 函数的扩展	(103)
8.2	C 语言的输入/输出函数	(104)
8.2.1	数据流指针到文件描述字的映射	(104)
8.2.2	按文件描述字打开一个流	(105)
8.2.3	其它 FILE 型的 C 函数的相互作用	(106)
8.2.4	对文件的操作——remove() 函数	(108)
8.3	其它的 C 语言函数	(108)
8.3.1	非局部跳转	(108)
8.3.2	置时区	(109)
9	系统数据基集	(110)
9.1	系统数据基集	(110)
9.2	对数据基集的访问	(110)
9.2.1	对组数据基集的访问	(110)

9.2.2 对用户数据基集的访问	(111)
10 数据交换格式	(112)
10.1 归档文件格式及交换文件格式	(112)
10.1.1 扩展的 tar 格式	(112)
10.1.2 扩展的 cpio 格式	(116)
10.1.3 多卷删文件	(118)
附录 A 参考文献	(119)
A.1 相关的开放系统标准	(119)
A.2 其它标准	(121)
A.3 以往的标准	(121)
A.4 以往的实现	(121)
A.5 C 语言参考资料	(123)
A.6 以往的应用系统程序设计辅导资料	(123)
附录 B 依据和注释	(126)
B.1 范围和引用标准	(126)
B.2 定义和通用需求	(134)
B.3 进程原语	(150)
B.4 进程环境	(170)
B.5 文件和目录	(175)
B.6 输入和输出原语	(182)
B.7 设备和设备类专用的函数	(189)
B.8 程序设计 C 语言专用的服务	(197)
B.9 系统数据基集	(204)
B.10 数据交换格式	(205)
附录 C 前导文件内容的样本	(211)
附录 D 特征文件	(222)
D.1 定义	(222)
D.2 标准中的可选项	(223)
D.3 相关的标准	(223)
D.4 相关的活动	(224)
D.5 与 IEEE 1003.0 的关系	(224)
附录 E 国家特征文件	(225)
E.1 (例子)POZ 地区的特征文件	(225)
E.2 POZ/Yellow 环境特征文件	(226)
E.3 地点定义	(227)
附录 F 函数索引	(228)

1 范围和引用标准

1.1 范围

ISO/IEC 9945 的这一部分(以后简称为本标准译者注)定义一个标准的操作系统界面及环境以支持源码级应用程序的可移植性。它可供应用开发人员及系统实现人员使用。

在初期,本标准的中心是借助 C 语言界面提供标准化的服务。希望未来的版本除了包含 C 语言以外还包含对其它编程语言的协约。这一目标通过把标准分为两个部分来完成。其中一部分是定义了独立于编程语言的核心要求,另一部分由编程语言的协约组成。

核心的要求部分定义了一组各种编程语言共同需要的服务,这些服务将形成本标准对语言的协约。所有的服务只提出功能要求而不定义依赖于语言的编程界面。对语言的协约包含两个部分:一部分是编程语言的标准化界面,以访问在标准的核心要求节中定义的独立于编程语言的核心服务;另一部分是语言专用服务的标准界面。任何声明与本标准一致的具有语言协约的实现必须遵循上述对语言协约的两个部分的要求。

本标准由四个主要部分组成:

- ① 用于决定结构,前导文件,环境变量及有关要求的术语,概念,定义和说明。
- ② 系统服务界面及各种子例程的定义。
- ③ C 编程语言专用的系统服务。
- ④ 界面问题,包括可移植性,出错处理,及出错恢复。

以下领域超出本标准的范围:

- ① 用户界面(shell)及其命令。
- ② 网络协议及其系统调用界面。
- ③ 图形界面。
- ④ 数据基集管理系统界面。
- ⑤ 具有记录的 I/O 考虑。
- ⑥ 目标码或二进制码的可移植性。
- ⑦ 系统配置及资源。
- ⑧ 单进程内支持并发系统的系统服务行为。

本标准描述开发人员关心的外部特性及设施,而不是描述为达到这些功能采用的内部结构技术。着重点放在各种各样的商品化应用所需要的那些功能及设施。

本标准已在源码级定义,其目标是与 POSIX.1 严格一致的应用源程序可在与 POSIX.1 一致的实现上编译和执行。

1.2 一致性

1.2.1 实现一致性

1.2.1.1 要求

一致的实现必须满足如下准则:

① 系统必须支持所有本标准要求的且在本标准中定义的界面,这些界面必须支持在标准中描述的功能特性。

② 系统可以提供本标准所未要求的附加功能或设施。非标准的扩展应该在系统文档中说明。当使用非标准扩展时可以改变在本标准中定义的函数或设施的行为。在这种情况下,系统文档必须定义一个环境,在这个环境中应用程序可以以标准所规定的行为运行。这样的环境对于严格地与 POSIX.1 一致的应用程序不要求做任何的修改。

1.2.1.2 文档编制

声明与本标准一致的实现,应提供含有如下信息的文档。该文档有和本标准同样的结构,及适当编号的章节,小节中提供的信息。一致性文档不得包括本标准范围以外有关扩展的设施或能力方面的信息。

一致性文档必须包含一个声明指出所用标准的全名,编号,标准提供日期。一致性文档也可以列出与 POSIX.1 一致的应用可利用的,ISO/IEC 或 ISO/IEC 成员体批准的软件标准。也可以包括由这些标准之一或 ISO 政府成员所批准的标准所要求的可用的特性。

一致性文档必须描述从前导文件<limits.h>及<unistd.h>中找到的限定值,状态值,和某些变量可能改变值的条件以及如果条件满足这些改变的限定值。

一致性文档需描述所有在本标准中定义为由实现定义的特性,它们在实现时的行为。这样的要求必须通过列出这些特性或提供一个对系统文档的特定的参考文件或提供这些特性的全部语法和语意来满足。一致性文档可以规定本标准中指明为实现时可改变的或指明为未定义或未规定的特性在实现时的行为。

除了在本子节(1.2.1.2)中描述的外,没有其他的规格说明应在一致性文档中出现。

除非系统明确地提到,否则本标准中所用的词“须编制成文档”意味着这些特性的文档应象如上所述的在一致性文档中出现。

系统文档也应包含在一致性文档中能找到的信息。

1.2.1.3 一致的实现选择

在指定的子节中描述的下列符号常量反映了实现对于标准的选择,这些选择能保证与 POSIX.1 一致的应用的要求或在一致的系统的规格说明中的要求或两者都能保证。

{NGROUPS_MAX} 多组选择(在 2.8.3)
{POSIX_JOB_CONTROL} 作业控制选择(在 2.9.3)
{POSIX_CHOWN_RESTRICTED} 管理/安全选择(在 2.9.4)

其余的在 2.9.3 和 2.9.4 中的符号常量用于测试,并作为需要容纳的行为类型的应用指南。它们并不能反映保证与 POSIX.1 一致的应用的要求之间的功能差别或与 POSIX.1 一致的实现之间的区别。

在省略了可选择项会使本标准所定义的函数成为未被定义的情况下,实现必须提供一个可用

本标准定义的语法进行调用的函数。即使此函数只返回错误而不做任何别的操作。

1.2.2 应用一致性

所有声明与本标准一致的应用必须只使用在 1.2.3 中描述的 C 语言的依赖于语言的服务且必须属于如下类别之一：

1.2.2.1 严格与 POSIX.1 一致的应用

严格与 POSIX.1 标准一致的应用是只使用本标准所描述的设施及可用的语言标准的应用。这种应用必须接受本标准描述为未规定的或实现定义的各种行为，对于符号常数应接受本标准所允许范围内的各种值。这种应用还允许由在 2.8 和 2.9 中的常数指示的设施的可利用性。

1.2.2.2 与 POSIX.1 一致的应用

1.2.2.2.1 与 ISO/IEC POSIX.1 一致的应用

一个与 ISO/IEC POSIX.1 一致的应用是一种只使用本标准所描述的设施及经批准的任何 ISO 或 IEC 标准的语言协约的应用。这种应用必须包括一致性的说明，它说明了所有任选项及限定值关系以及所使用的其它 ISO 或 IEC 标准。

1.2.2.2.2 与〈国家组织〉POSIX.1 一致的应用

这种应用与 ISO/IEC POSIX.1 一致的应用的不同之处在于它可以使用称为〈国家组织〉的单个 ISO/IEC 成员的特定标准。这种应用必须包括一致性说明，它说明所有任选项及限定值关系以及所使用的其它〈国家组织〉标准。

1.2.2.3 使用扩展的与 POSIX.1 一致的应用

这种应用同与 POSIX.1 一致的应用的区别仅在于它使用与本标准一致的非标准设施。除了与 POSIX.1 一致的应用所需的文档外，这种应用还必须充分说明其扩展设施的要求。使用扩展的与 POSIX.1 一致的应用必须是使用扩展的与 POSIX.1 的 ISO 一致的应用或者是使用扩展的与 POSIX.1 的〈国家组织〉一致的应用(1.2.2.2.1 及 1.2.2.2.2)。

1.2.3 对于 C 编程语言的依赖于语言的服务

ISO/IEC 9899:…{2} (在下文中表示成“C 标准{2}”)被引用来描述也由本标准提到的要求。用于描述本标准的要求的 C 标准{2}的某些节在第八章中规定。第八章中规定了 C 标准{2}相应章节的补充与扩展。任何声明与具有 C 语言协约的本标准一致的实现，须提供在第八章提到的设施以及第八章所要求的补充和扩展。

虽然本标准引用了 C 标准{2}的有关部分以描述一些要求，但与本标准一致并不要求与 C 标准{2}一致。任何能提供第八章中所规定的设施的 C 语言都可以声明与本标准一致，然而，它必须清楚地说明其 C 语言并不与 C 标准{2}一致。

1.2.3.1 一致性类型

具有 C 语言协约的实现声明与本标准一致必须申请两种一致性类型之一：与 POSIX.1 C 语言协约(依赖于 C 标准语言系统支持)一致，或与 POSIX.1 C 语言协约(依赖于通用 C 语言系统支持)一致。

1.2.3.2 依赖于 C 标准{2}语言系统支持

实现者必须以 C 标准{2}为基准满足第八章的有关要求，并清楚地记载实现第八章要求所使用的 C 标准{2}的版本。

企图用 C 标准{2}声明一致性的实现须声明与 C 语言协约(依赖于 C 标准{2}语言系统支持)