

团体标准

T/CESA XXXX—202X

物联网操作系统平台运行规范 第4部分：通用操作系统功能

Operational specifications for IoT operating system platform

Part 4: General-purpose operating system features

征求意见稿

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

已授权的专利证明材料为专利证书复印件或扉页，已公开但尚未授权的专利申请证明材料为专利公开通知书复印件或扉页，未公开的专利申请的证明材料为专利申请号和申请日期。

202X-XX- XX 发布

202X-XX- XX 实施

中国电子工业标准化技术协会 发布



版权保护文件

版权所有归属于该标准的发布机构，除非有其他规定，否则未经许可，此发行物及其章节不得以其他形式或任何手段进行复制、再版或使用，包括电子版，影印件，或发布在互联网及内部网络等。使用许可可于发布机构获取。

目 次

目 次..... II

前 言..... III

引 言..... IV

物联网操作系统平台运行规范 第 4 部分：通用操作系统功能..... 1

1 范围..... 1

2 规范性引用文件..... 1

3 术语和定义..... 1

4 总体要求..... 1

4.1 总体目标..... 1

5 功能要求..... 2

5.1 操作系统核心功能..... 2

5.2 程序运行时..... 2

附 录 A..... 3



前 言

本文件按照 GB/T 1.1-2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》相关规则起草。

本文件是《物联网操作系统平台运行规范》系列标准的第4部分。

本文件由中国电子工业标准化技术协会归口。

本文件起草单位：。

本文件主要起草人：。



引 言

近年来，我国物联网产业蓬勃发展，其在制造、交通、医疗、民生等领域加速应用，推动各行各业数字化、智能化、绿色化、融合化发展。作为物联网产业的核心基础设施之一，物联网操作系统平台正在加快与产业发展的深度融合。

作为平台运行的基础，物联网操作系统平台运行环境标准化对于推动平台兼容适配和稳定高效运行至关重要。CESA-2024-044 至 CESA-2024-048 标准旨在规范物联网操作系统平台的运行环境，拟由下列五个部分组成：

——第 1 部分：总体要求。目的在于确立物联网操作系统平台运行规范标准化工作的总体框架。

——第 2 部分：数据处理。目的在于确立物联网操作系统平台运行环境应具备的数据处理相关能力与应满足的要求。

——第 3 部分：网络通信。目的在于确立物联网操作系统运行环境应具备的网络通信相关能力与应满足的要求。

——第 4 部分：通用操作系统功能。目的在于确立物联网操作系统平台运行环境应具备的通用操作系统核心功能与程序运行时。

——第 5 部分：IaaS 服务技术。目的在于规范 IaaS 层服务应为物联网操作系统平台提供的功能，以及 IaaS 服务应满足的性能指标与测试要求。

物联网操作系统平台运行规范 第4部分：通用操作系统功能

1 范围

本文件规定了物联网操作系统平台（以下简称“平台”）运行环境应具备的通用操作系统功能，包括操作系统核心功能和程序运行时两方面要求。

本文件适用于平台的开发、评估、应用和维护。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，其随后所有的修改单（不包括刊物的内容）或修订版均不适用于本文件。凡是不注日期的引用文件，其最新版本适用于本文件。

ISO/IEC 23360-1:2021 Linux 标准基础（LSB）核心规范 5.0 版本 第1部分：总规范

GB/T 32394-2015 信息技术 中文 Linux 操作系统运行环境扩充要求

GB/T 32395-2015 信息技术 中文 Linux 操作系统应用编程接口（API）扩充要求

GB/T 40778.1-2021 物联网 面向 Web 开放服务的系统实现 第1部分

3 术语和定义

下列术语和定义适用于本文件。

3.1

动态加载 dynamic loading

在程序运行时加载并执行模块或库的机制，允许程序在需要时加载和卸载代码，从而提供更灵活的扩展性和资源管理。

3.2

实时调度 real-time scheduling

是一种操作系统调度策略，确保任务在特定的时间限制内完成，以满足实时性的需求。

3.3

事件循环 event loop

是一种编程模型，通常用于异步编程。在事件循环中，程序通过监听事件的发生并对其进行响应来执行任务。

3.4

协程 coroutine

是一种轻量级的线程，允许在同一线程内的代码段之间进行协作式的多任务处理。协程机制使得在单一线程内可以处理大量的并发任务，提高程序的性能和资源利用率。

4 总体要求

4.1 总体目标

本文件为平台规定通用操作系统功能，涵盖操作系统核心功能和程序运行时两方面要求，旨在为平台运行环境提供通用操作系统的核心功能及机制，以保障平台高效稳定运行。

5 功能要求

5.1 操作系统核心功能

- a) 应提供通用操作系统的基础功能，包括文件系统管理、内存管理、进程管理和设备管理等。
 - 1) 文件系统管理应支持多种文件格式和存储介质，执行对文件的创建、删除、读写、加密等操作；
 - 2) 内存管理应支持对物理内存和虚拟内存的分配、回收、保护和优化，提高内存的利用率和性能；
 - 3) 进程管理应支持对进程的创建、调度、同步、通信和终止，保障进程的并发性和协作性。
 - 4) 设备管理应支持对各种外部设备和内部设备的识别、初始化、配置、控制和监测，提高设备的兼容性和可扩展性。
- b) 应提供实时调度机制，使得系统能提供对软实时任务的调度算法支撑，满足实时性要求较高的操作和处理需求。
 - 1) 支持对中断的快速处理和优先级分配，提供多种中断服务和中断处理机制；
 - 2) 支持对任务的非抢占和抢占式调度，提供多种任务调度算法和策略；
 - 3) 支持对资源的动态分配和互斥访问，提供多种资源管理方法和同步机制。
- c) 宜保障操作系统的稳定性和高可用性。此功能应能够提高系统的容错能力，减少系统的故障率，保障系统在长期运行中的可靠性和安全性。
 - 1) 支持对系统异常的检测和恢复，提供多种异常处理方法和恢复策略；
 - 2) 支持对系统更新和维护的支持，提供多种更新方式和维护模式。
- d) 宜提供操作系统网络通信管理、安全防护管理、用户界面管理等功能。

5.2 程序运行时

- a) 动态加载机制
 - 1) 应提供动态加载机制，允许应用程序在运行时加载外部模块和库。该功能应能够支持模块，实现对应用程序的动态扩展和更新，提高应用程序的灵活性和可维护性。
 - 2) 应提供对动态加载机制的监控和管理功能，防止出现冲突或异常，保证应用程序的稳定性和安全性。
- b) C 函数调用接口的封装
 - 1) 应提供对 C 函数调用接口的封装，以支持动态的 C 函数调用和参数传递。
 - 2) 应提供对 C 函数调用接口的扩展功能，以支持更多的数据类型和结构体，实现对复杂数据的处理和传递。
- c) 多线程支持
 - 1) 应支持线程的创建、销毁、同步、通信等基本操作，实现对线程的有效调度和管理，提高应用程序的并发性和效率。
 - 2) 应提供对多线程程序的优化。应提供对事件循环和协程的支持。
- d) 应提供对多种编程语言的支持，包括但不限于 C++、Java、python 等语言。应提供相应语言的编译器或解释器。应实现对不同编程语言的兼容性和互操作性，保障不同编程语言之间的数据交换和函数调用。

附 录 A
(规范性附录)
系统环境

A.1 Linux技术环境

A.1.1 libc

系统应提供 Libc 库，其共享对象名应为 libc.so.6。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 Libc。

A.1.2 libm

系统应提供 libm 库，其共享对象名应为 libm.so.6。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 libm。

A.1.3 libpthread

系统应提供 libpthread 库，其共享对象名应为 libpthread.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 libpthread。

A.1.4 libgcc_s

系统应提供 Libgcc_s 库，其共享对象名应为 libgcc_s.so.1。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 Libgcc_s。

A.1.5 libdl

系统应提供 libdl 库，其共享对象名应为 libdl.so.2。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 libdl。

A.1.6 librt

系统应提供 librt 库，其共享对象名应为 librt.so.1。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。详见 GB/T 25656-2010 信息技术 中文 Linux 应用编程界面(API)规范第 4 节所述的基本库 librt。

A.1.7 libQtCore

系统应提供 libQtCore 库，其共享对象名应为 libQtCore.so.4。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libQtCore 是 Qt 的核心模块，用于提供非图形模块所需的功能：

- a) 提供多线程编程的功能，包括线程创建、管理和同步。
- b) 实现定时器功能，允许应用程序安排和执行定时任务。
- c) 提供事件循环和事件处理机制，用于响应用户交互和系统事件。

A.1.8 libglib-2.0

系统应提供 libglib-2.0 库，其共享对象名应为 libglib-2.0.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libglib-2.0 提供通用的 C 语言函数库，用于在应用程序中进行底层开发和实现基本数据结构。

- a) 提供常见的数据结构，如链表、哈希表和数组。
- b) 实现内存分配和释放机制，包括智能指针和垃圾回收。
- c) 提供文件操作、字符串处理和日期时间功能等。

A.1.9 libgobject-2.0

系统应提供 libgobject-2.0 库，其共享对象名应为 libgobject-2.0.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libgobject-2.0 提供对象系统的功能，用于在应用程序中进行面向对象编程。

- a) 支持对象的创建、继承和多态等面向对象编程概念。
- b) 实现对象间的通信和事件处理。

A.1.10 libgthread-2.0

系统应提供 libgthread-2.0 库，其共享对象名应为 libgthread-2.0.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libgthread-2.0 提供多线程编程的功能，用于在应用程序中实现多线程编程，同时提供多线程创建和管理功能，包括线程同步和互斥操作。

A.1.11 libxcb

系统应提供 libxcb 库，其共享对象名应为 libxcb.so.1。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libxcb 提供 XCB 协议的实现，用于在应用程序中通过 C 语言访问 X Window 系统的功能。应对 X Window 系统的底层访问功能，包括窗口管理和图形绘制。

A.1.12 libstdc++

系统应提供 libstdc++ 库，其共享对象名应为 libstdc++.so.6。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libstdc++ 提供 C++ 标准库的实现，用于让应用程序使用 C++ 编程语言的标准库功能。应实现 C++ 标准库中的容器、算法和输入输出等功能。

A.1.13 libfontconfig

系统应提供 libfontconfig 库，其共享对象名应为 libfontconfig.so.1。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libfontconfig 提供字体配置功能，用于在应用程序中进行字体的查找、选择和配置，以确保文本显示的一致性和可读性。

A.1.14 libfreetype

系统应提供 libfreetype 库，其共享对象名应为 libfreetype.so.6。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libfreetype 提供字体渲染功能，用于在应用程序中进行字体的解析和渲染。

- a) 实现字体文件的解析，包括 TrueType 和 OpenType 字体格式。
- b) 支持字符和文本的渲染，以便在应用程序中显示文本内容。

A. 2 Apache环境支撑

A. 2.1 libapr-1

系统应提供 libapr-1 库，其共享对象名应为 libapr-1.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libapr-1 是 Apache Portable Runtime，提供跨平台运行时库。

- a) 为底层特定于平台的实现提供统一的 API，使应用程序在不同操作系统上实现跨平台功能。
- b) 提供运行时环境，包括文件操作、内存管理和多线程支持等。

A. 2.2 libaprutil-1

系统应提供 libaprutil-1 库，其共享对象名应为 libaprutil-1.so.0。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：系统应提供 libaprutil-1 库，该库是 apr-1 的补充库，用于扩展 apr-1 库的功能。

A. 3 Java环境支撑

A. 3.1 libjli

系统应提供 libjli 库，其共享对象名应为 libjli.so。库函数的存放位置是实现定义的，但应在动态连接器的搜索目录中。

说明：libjli 提供 Java Launcher Infrastructure 功能的实现，用于启动和管理 Java 应用程序。

- a) 支持 Java 应用程序的启动和初始化。
- b) 实现 Java 类的加载和动态链接。

A. 4第三方动态库环境

操作系统需提供以下第三方动态库，保障物联网操作系统平台与第三方软硬件的可交互性，见表 2。

表 2 第三方动态库环境

libanalyzedata.so	libdsl.so	libdslalien.so	libhpr.so
libhcnetsdk.so	libxmnetsdk.so	libzlnetsdk.so	libdhconfigsdk.so
libdhnetsdk.so	libcomerr.so	libffi.so	libpcre.so
libpcre2-8.so			