鉴定文件 标识：PT-R2499-TO-1.00

公开

敏感器主控软件软件

鉴定测评大纲

册号：1/1 总页数：76

**中国科学院卫星软件评测中心**

2024年06月27日

敏感器主控软件软件

鉴定测评大纲

**拟制： 陈俊亦 日期：20240625**

**校对： 陈俊亦 日期：20240626**

**审核： 施敏华 日期：20240627**

**批准： 常 亮 日期：20240627**

|  |  |
| --- | --- |
| **有 效 性 声 明** | |
| 1．本中心严格按照安全有关保密要求开展测评工作，承担安全保密与知识产权保护有关的法律责任；  2．本测评大纲无授权签字人批准签字，大纲封面无中心印章无效；  3．本测评大纲未经中国科学院卫星软件评测中心书面批准，不得复制，任何未经允许的复制本，任何形式的残损本，不具备大纲原件的效力；  4．本大纲只适用于被测件介质相同版本的软件；  5．测评大纲版权归委托方和中国科学院卫星软件评测中心共有，其他任何单位和个人未经许可均无权使用本测评大纲，否则将保留对其追究法律责任的权利。 | |
| 测试机构名称 | 中国科学院卫星软件评测中心 |
| 测试机构地址 | 上海市浦东新区海科路99号 |

文件状态表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 日期 | 更改摘要 | 修改章节 | 备注 |
| V0.10 | 20240625 | 创建文档 | / | 内审版本 |
| V1.00 | 20240627 | 1、修改调整软硬件环境 | 1、5.2章节 | 内部评审修改 |
|  |  |  |  |  |
|  |  |  |  |  |

敏感器主控软件软件测评大纲

# 测评依据

## 测评任务来源

受上海微小卫星工程中心委托，由中国科学院卫星软件评测中心承担敏感器主控软件软件测评任务。

## 依据文件

# 测评性质

本次测评任务性质为鉴定测评。

# 测评目的

本次测评主要目的为发现并纠正软件问题缺陷，考核敏感器主控软件软件功能性能指标是否满足规定的需求，对软件的性能指标进行验证，验证软件是否符合作战使用要求，为装备状态鉴定提供依据。

本大纲文档主要包括测评依据、测试内容与方法、测试环境、测试安排、测试结束条件、配置管理、质量保证及风险分析等内容。

# 测评时间和地点

# 测评对象及环境

## 测评对象

### 被测软件功能

### 被测软件接口

### 被测软件性能

### 主要战技指标

### 被测软件基本信息

## 软硬件环境

# 测评内容、方法及要求

## 测试总体要求

## 测试项及方法

## 测试内容充分性及测试方法有效性分析

## 软件评价内容与方法

### 软件需求符合性

从以下几方面评价软件需求符合性：

1. 软件是否实现了研制总要求、系统规格说明中规定的全部软件需求；
2. 软件是否实现了任务书中规定的全部需求；
3. 软件是否实现了软件需求规格说明中规定的全部需求；

### 软件指标符合性

从以下方面评价软件指标符合性：

1. 软件是否实现了任务书和需求中规定的全部软件相关技术指标；
2. 软件性能是否满足相关要求；
3. 软件软件是否符合装备使用要求；
4. 软件在实装环境下的适应性；

### 软件质量评价

从以下几个方面评价软件质量：

1. 软件编码质量是否满足要求，用代码的质量度量元统计度量编码质量，包括：
2. 软件总注释率不小于20%；
3. 模块的平均规模不大于200行；
4. 模块的平均圈复杂度不大于10；
5. 模块的最大圈复杂度不大于80；
6. 模块圈复杂度大于20的比例不大于20%；
7. 模块的平均扇出数不大于7；
8. 统计软件首轮测试缺陷率及每轮回归测试缺陷率。
9. 软件文档种类是否齐套、内容是否完整、描述是否准确、格式是否规范及文档是否文文一致、文实相符。
10. 测试中发现问题的修改处理情况，对遗留问题的影响分析；

## 软件问题类别及等级

### 问题类别

测试过程中发现的问题统称为软件问题，类别分为：

表6‑5 问题类别表

| 问题类别 | 解释说明 |
| --- | --- |
| 需求问题 | 用户需求、系统需求或软件需求问题 |
| 设计问题 | 系统设计或软件设计问题 |
| 文档问题 | 文档描述问题 |
| 编码问题 | 代码实现问题 |
| 数据问题 | 数据规格、数量、语义等问题 |
| 其他问题 | 上述问题之外的问题 |

### 问题等级

软件问题等级分为重大、严重和一般三个等级。测试过程中发现的其他不方便使用或对软件功能有轻微影响的，可作为改进建议提出。

1. 重大问题：软件问题导致程序无法继续运行、丧失主要功能或造成重大损失的，视为重大问题：
2. 导致系统死机、崩溃或异常退出；
3. 主要功能未实现或实现错误；
4. 造成人员、环境等重大损失；
5. 重要数据丢失，且很难恢复。
6. 严重问题：软件问题对主要功能性能有较大影响或造成严重损失，视为严重问题：
7. 没有完整实现软件需求，对软件主要功能性能等有较大影响；
8. 没有正确实现软件需求，对软件主要功能性能等有较大影响；
9. 造成人员、环境等严重损失；
10. 重要数据丢失，但能以某种方式恢复；
11. 软件文档对主要功能、性能描述缺失或错误。
12. 一般问题：软件问题对功能性能有较小影响或造成一般损失，视为一般问题：
13. 没有完整实现软件需求，对软件主要功能性能影响较小，或对一般功能性能造成影响；
14. 没有正确实现软件需求，对软件主要功能性能影响较小，或对一般功能性能造成影响；
15. 软件操作与软件使用说明不符；
16. 软件文档存在准确性、一致性、错别字等影响较小的问题。

## 测评通过准则

软件测评通过准则如下：

1. 评测中心按照要求对被测软件按照软件测评大纲中规定的内容完成了测试；
2. 软件文档齐套，内容完整，描述准确，文文一致、文实一致，满足相关标准的要求；
3. 软件实现了软件研制任务书、软件需求规格说明所规定软件功能、性能、接口、安全性等需求；
4. 承研单位对测评机构报告的软件问题均进行了正确有效的整改，或对遗留问题进行了合理的解释并通过测评审查；

## 测评数据采信

无。

# 测评度量数据及采集要求

无。

# 测评暂停、恢复与中止

出现下列情况之一，测评机构应及时提出暂停鉴定测评工作的申请：

1. 测评过程中存在重大安全保密隐患；
2. 被测软件存在重大技术问题，影响测评工作继续开展；
3. 研制单位不能在规定时限内解决软件问题（缺陷），或大部分问题（缺陷）无法达成共识。

经任务委托方批准后，测评工作可暂停。当导致暂停的问题已解决，并经软件测评总体单位（如有）、军事代表机构（如有）、项目研制总体等相关方确认，且获得批准后可重新启动测评工作；如问题无法得到解决，获得批准后可中止测评工作。

# 测评组织及任务分工

# 测评保障

## 测评条件保障

## 配置管理

在测试项目的整个生存周期内应对配置项进行有效管理，配置项通常包括测评依据（软件测评工作计划、合同、任务书及其他等效文件）、被测软件（各测试轮次的软件源/可执行程序、软件文档、基础数据等）、测评工作产品（测评文档、测试数据、自研测试程序或工具、评审记录等）和软件测评所需的各种软硬件资源。配置管理要求如下：

1. 编制并评审配置管理计划，按照配置管理计划开展配置管理工作；
2. 建立配置管理库，对配置项进行标识，并控制其更改和发布；
3. 规定受控的配置项，并规定其出入配置管理库要求；
4. 建立基线，基线的状态和内容应通知到各相关人员；
5. 应进行配置审核并将结果通知到相关人员。

## 质量保证

中国科学院卫星软件评测中心为该项目配备专职质量保证人员，制定项目的质量保证计划，明确需开展的评审、需保证的工作产品和项目采用的标准规范等内容。本项目需开展的评审包括测评大纲、测试说明、测评报告评审，以及测试就绪和测试总结评审等阶段管理评审。在项目开展过程中，由项目质量保证人员按照质量保证计划，对项目正在运行的过程和正在形成的工作产品开展质量保证活动，以验证它们符合相应的程序和标准。所有外发的测试工作产品都要经过质量保证人员审核。质量保证要求如下：

1. 制定并评审质量保证计划，按照质量保证计划开展质量保证工作；
2. 质量保证人员应客观地验证工作产品及其活动遵循所用标准、规程和需求的情况；
3. 质量保证人员应将质量保证活动的时机、结果通知到相关人员；
4. 领导层应负责处理在测评项目组内无法解决的问题。

## 测评风险分析

测试中可能出现的风险及相应对策见下表。

表10‑1 风险分析表

| 序号 | 风险名称 | 风险描述及风险分析 | 风险概率 | 风险后果 | 风险处理对策 |
| --- | --- | --- | --- | --- | --- |
| 1 | 测试范围风险 | 需求规格说明书不够清晰、明确 | 中 | 严重 | 协调客户提供更为详细的需求说明书，参考原型图，沟通确认需求，再进行书面书写确认 |
| 2 | 技术风险 | 测试异常数据准备不够全面，可能会影响测试的充分性。 | 中 | 一般 | 与开发方沟通协调需提供测试所需数据，如确实无法提供，与委托方确认后，在报告中评价。 |
| 在性能测试、容错性测试等测试环节可能出现异常情况（如：系统缺陷、误操作）导致系统出现中断或系统崩溃 | 低 | 严重 | 在测试过程中，尽量保证系统开发人员能够及时提供必要的技术支持。 |
| 3 | 资源风险 | 相关性能测试环境资源不能够及时到位，包括实施场地、所需设备等。 | 高 | 严重 | 及时与委托方协调相关性能测试环境资源。 |
| 4 | 测试进度风险 | 客户要求测试时间缩短 | 中 | 一般 | 与客户协商，说明测试时间缩短的质量风险；增加测试人员，加班测试 |
| 提供的测试样品无法测试，导致进度风险 | 高 | 严重 | 与客户协商，尽快提供正确的样本进行测试；与客户协商，测试延期 |
| 5 | 测试环境风险 | 公网测试，服务器不在实验室，存在不受控的风险 | 中 | 一般 | 与客户协商，确认客户方的服务器各项基本信息；必要的时候，去客户方进行确认测试 |
| 已有测试设备不符合测试要求 | 中 | 一般 | 测试执行前，对环境的准备和确认，降低环境风险；申请采购设备进行测试 |
| 5 | 测试人力资源风险 | 测试人员中途离开 | 中 | 一般 | 测试人员加班；抽调测试人员 |
| 人员技能生疏 | 中 | 严重 | 通过培训来保证人员技能满足岗位要求 |
| 6 | 测试工具风险 | 无 |  |  |  |
| 7 | 测试质量风险 | 测试不能穷举，存在不充分性 | 中 | 严重 | 需求分析采用场景、流程等方法尽可能细致；测试用例设计采用边界值、等价类等方法进行设计；测试执行按照测试用例执行，并举一反三 |
| 测试策划和需求分析阶段未开展风险策划和分析 | 低 | 一般 | 进行测试需求分析和确认评审，确认100%覆盖了软件需求和隐含需求；将风险管理计划是否纳入测试计划中；对测试中可能存在的全部风险进行分析并做出相应的对策；按照《测试过程评审程序》对策划中的风险分析进行评审。 |
| 测试设计和实现阶段，考虑不充分 | 中 | 严重 | 依据《测试过程评审程序》对测试说明进行评审，以减低测试用例设计充分性的风险；建立软件测试用例和测试需求的追踪关系，100%覆盖了测试需求。 |
| 测试执行过程测试数据遗漏或发生错误记录 | 中 | 一般 | 通过对比测试的方式来保证测试结果的有效性；加强人员监督。 |
| 测试总结阶段出具的报告不能完整反映被测系统的真实质量情况 | 低 | 一般 | 依据《测试过程评审程序》对测试报告做分析评审。 |
| 8 | 沟通风险 | 与客户方沟通不充分 | 中 | 一般 | 每天告知客户当日进度；每周告知客户周进度；阶段结束及时与客户联系说明 |
| 与测试人员沟通不充分 | 低 | 一般 | 每天项目负责人和测试团队成员进行沟通当日情况；阶段结束，及时开评审会议 |

## 测评分包

本次项目无分包。

# 测评安全与保密

本测评项目的密级为公开，在整个测评工作过程中，严格遵守相关保密规定。对国家和军队秘密以及委托方的知识产权进行保密。涉密文件，程序等相关资料，由专人负责保存，无关人员不得接触。严格按照本中心程序文件《保密性和可信性程序》中有关规定执行。

为确保软件测评成功，需制定安全与保密相关措施。针对测评中可能出现的特殊情况，组织制定特情处置预案，当特情发生后严格按特情处置预案进行处置，并及时上报，严谨瞒报、漏报等现象出现。

# 有关问题的说明

无。

# 附件

## 术语与缩略语

## 测试项与研制总要求对照表

## 需求规格说明与测试项对照表

## 测试项与软件需求对照表

## 文档审查单

软件文档齐套性检查单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 审查内容 | 期望结果 | 审查结果 | 备注 |
| 1 | 是否具有研制要求 | Y |  |  |
| 2 | 是否具有软件系统需求规格说明 | Y |  |  |
| 3 | 是否具有软件系统设计说明 | Y |  |  |
| 4 | 是否具有接口需求规格说明 | Y |  |  |
| 5 | 是否具有接口设计说明 | Y |  |  |
| 6 | 是否具有软件研制任务书 | Y |  |  |
| 7 | 是否具有软件配置项需求规格说明 | Y |  |  |
| 8 | 是否具有软件配置项设计说明 | Y |  |  |
| 9 | 是否具有用户使用文档 | Y |  |  |
| 10 | 是否具有配置管理计划 | Y |  |  |
| 11 | 是否具有配置管理报告 | Y |  |  |
| 12 | 是否具有测试计划 | Y |  |  |
| 13 | 是否具有测试报告 | Y |  |  |
| 审查结果说明：Y—是； N—否；NA—不适用。 | | | |  |

需求类文档审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文档名称 | |  | | | | |
| 序号 | 审查项 | 审查内容 | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
| 1 | 完整性 | 标识描述本文档所适用系统和软件的完整标识，包括其标识号、名称、缩略名、版本号和发布号。 |  |  |  |  |
| 2 | 系统概述概述本文档适用的系统和软件的用途，描述系统与软件的一般特性 |  |  |  |  |
| 3 | 文档概述概述了本文档的用途和内容，并描述与它的使用有关的保密性方面的要求。 |  |  |  |  |
| 4 | 引用文档应列出文档的编号、标题、编写单位、修订版及日期 |  |  |  |  |
| 5 | CSCI能力需求标识出所要求的每一个CICI能力，并列出与该能力有关的需求，包括所需的CSCI行为 |  |  |  |  |
| 6 | CSIC的外部接口需求应使用唯一的标识符标识CSCI每一个外部接口，并描述为实现该接口提出的该CSCI的需求。 |  |  |  |  |
| 7 | CSCI内部接口需求应指明施加于CSCI内部接口的需求 |  |  |  |  |
| 8 | 保密性需求应指明与维护保密性有关的CSCI需求 |  |  |  |  |
| 9 | 安全性需求应指明关于防止或尽可能降低对人员、财产和物理管径产生以外危险的CSCI安全性需求 |  |  |  |  |
| 10 | 环境适应性需求应指明CSCI的运行环境需求 |  |  |  |  |
| 11 | 计算机资源需求应指明CSCI必须使用的计算机硬件的需求、计算机硬件资源的使用需求、必须使用或必须被纳入本CSCI的计算机软件需求、必须使用的计算机通信方面的需求 |  |  |  |  |
| 12 | 设计和实现约束应指明约束CSCI的设计和实现的需求 |  |  |  |  |
| 13 | 人员相关需求描述与使用或保障本CSCI的人员有关的CSCI需求，包括人员的数量、职能等级、工作周期、必须的培训以及其他信息 |  |  |  |  |
| 14 | 训练相关需求应指明与训练相关的CSCI需求 |  |  |  |  |
| 15 | 软件保障需求应指明与软件保障考虑有关的CSCI需求 |  |  |  |  |
| 16 | 包装需求应指明为了交付而对CSCI进行包装、标记和处理的需求 |  |  |  |  |
| 17 | 需求的优先顺序和关键性应指明本规格说明中各需求的优先次序、关键性或表示其相对重要性的权重 |  |  |  |  |
| 18 | 合格性规定应定义一组合格性检验方法，针对每个需求制定确定需求得到满足所使用的方法 |  |  |  |  |
| 19 | 需求可追踪性应描述从本规格说明中的每一个CSCI需求所涉及的系统/子系统需求的可追踪性，以及从分配给本CSCI的每一个系统/子系统需求，到所涉及的CSCI需求的可追踪性 |  |  |  |  |
| 20 | 准确性 | 对各项需求项目的描述准确无歧义 |  |  |  |  |
| 21 | 列举的各项需求均具备可测试性 |  |  |  |  |
| 22 | 在已知的条件限制下，所有的需求都应为可实现的需求 |  |  |  |  |
| 23 | 一致性 | 各项需求之间无相互矛盾或冲突 |  |  |  |  |
| 24 | 需求规格说明与其他软件文档之间无相互矛盾或冲突 |  |  |  |  |

设计类文档审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文档名称 | |  | | | | |
| 序号 | 审查项 | 审查内容 | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
| 1 | 完整性 | 标识描述本文档所适用系统和软件的完整标识，包括其标识号、名称、缩略名、版本号和发布号。 |  |  |  |  |
| 2 | 系统概述概述本文档适用的系统和软件的用途，描述系统与软件的一般特性 |  |  |  |  |
| 3 | 文档概述概述了本文档的用途和内容，并描述与它的使用有关的保密性方面的要求。 |  |  |  |  |
| 4 | 引用文档应列出文档的编号、标题、编写单位、修订版及日期 |  |  |  |  |
| 5 | CSCI级设计决策应根据需要分条给出CSCI行为设计的决策和其他影响组成该CSCI的软件单元的选择与设计的决策。 |  |  |  |  |
| 6 | CSCI部件描述构成该CSCI的所有软件单元、软件单元的静态关系、每个软件单元的用途、每个软件单元的开发状态/类型、CSCI计划使用的计算机硬件资源、标识实现每个软件单元的软件放置在哪个程序库中。 |  |  |  |  |
| 7 | 执行方案应说明软件单元间的执行方案，说明软件单元间的动态关系 |  |  |  |  |
| 8 | 接口标识和接口图应标识每一个接口，包含项目唯一的标识符，并声明哪些实体具有固定的接口特性，说明哪些实体正在开发或修改之中 |  |  |  |  |
| 9 | 通过唯一的标识符来标识接口，并标识接口实体，分条描述单方或双方接口实体的特性 |  |  |  |  |
| 10 | CSCI详细设计应通过项目唯一的标识符来标识软件单元，并对该单元进行说明。  a. 单元设计决策；  b. 该软件单元设计中的任何约束、限定或非常规特征；  c. 如果使用的编程语言不同于该CSCI所指定的语言，则应指出并说明使用它的理由；  d. 如果该软件单元包含过程性命令或由过程性命令组成，应列出这些过程性命令，并引用解释它们的用户手册或其他文档；  e. 如果该软件单元包含、接收或输出数据，应对它的输入、输出及其他数据元素和数据元素组合体进行说明。  f. 如果该软件单元包含逻辑，则给出该软件单元所用到的逻  辑。 |  |  |  |  |
| 11 | 需求可追踪性包含从本文档所标识的每个软件单元，到分配给它的CSCI需求的可追踪性，以及从每个CSCI需求，到被分配这些需求的软件单元的可追踪性。 |  |  |  |  |
| 12 | 准确性 | 程序流程图、时序图等符合标准规范 |  |  |  |  |
| 13 | 所有的变量说明和模块说明均清晰无歧义 |  |  |  |  |
| 14 | 所有的文字和图形应描述清晰准确 |  |  |  |  |
| 15 | 无二义性的定义、术语或内容 |  |  |  |  |
| 16 | 一致性 | 设计文档上下文一致无冲突 |  |  |  |  |
| 17 | 设计文档中图文一致无冲突 |  |  |  |  |
| 18 | 设计文档与需求文档一致，无冲突 |  |  |  |  |

手册类文档审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 文档名称 | |  | | | | |
| 序号 | 审查项 | 审查内容 | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
| 1 | 完整性 | 标识描述本文档所适用系统和软件的完整标识，包括其标识号、名称、缩略名、版本号和发布号。 |  |  |  |  |
| 2 | 系统概述概述本文档适用的系统和软件的用途，描述系统与软件的一般特性 |  |  |  |  |
| 3 | 文档概述概述了本文档的用途和内容，并描述与它的使用有关的保密性方面的要求。 |  |  |  |  |
| 4 | 引用文档应列出文档的编号、标题、编写单位、修订版及日期 |  |  |  |  |
| 5 | 软件应用应加药说明软件预期的用途 |  |  |  |  |
| 6 | 软件清单应标识使软件运行而必须安装的所有软件文件 |  |  |  |  |
| 7 | 软件环境应描述用户安装并运行所需的硬件、软件、手工操作和其他资源 |  |  |  |  |
| 8 | 软件组织和操作概述应从用户角度简要描述软件的组织与操作 |  |  |  |  |
| 9 | 意外事故及运行的备用状态和方式应说明在紧急时刻以及在不同运行状态和方式下用户处理软件的差异 |  |  |  |  |
| 10 | 保密性应概述与本软件先关的保密性考虑 |  |  |  |  |
| 11 | 帮助和问题报告应标识联系方式、获得帮助和报告软件使用中遇到问题所遵循的规则 |  |  |  |  |
| 12 | 软件入门应描述软件的首次用户熟悉设备、访问控制、安装配置软件的操作和规程，软件启动的规程，软件停止和挂起的方法 |  |  |  |  |
| 13 | 能力应简述事务、菜单、功能或其他的处理之间的关系 |  |  |  |  |
| 14 | 约定应描述软件使用的任何约定 |  |  |  |  |
| 15 | 处理应描述软件所有功能、菜单、事务或其他的过程 |  |  |  |  |
| 16 | 有关的处理应标识并描述全部关于未被用户直接调用，也未描述的由软件执行的批处理、脱机处理或后台处理 |  |  |  |  |
| 17 | 数据备份应描述创建和保留备份数据的规程 |  |  |  |  |
| 18 | 错误、故障和紧急情况下的恢复应给出从发生错误或故障中国重启和恢复的详细规程 |  |  |  |  |
| 19 | 消息应列出完成用户功能时可能发生的所有错误消息、诊断消息、提示消息等 |  |  |  |  |
| 20 | 快速参考指南应为使用该软件提供或引用快速参考卡或页 |  |  |  |  |
| 21 | 准确性 | 明确说明软件各功能运行时的输入数据类型和范围、精度 |  |  |  |  |
| 22 | 明确说明软件输出数据的类型和范围，若输出为固定目录下的固定文件，应明确文件路径 |  |  |  |  |
| 23 | 软件运行若存在等待时间，应给出等待周期范围，并说明超过范围时应采取的操作 |  |  |  |  |
| 24 | 所有的文字和图形应描述清晰准确 |  |  |  |  |
| 25 | 一致性 | 与需求、设计等其他文档保持一致、无冲突 |  |  |  |  |
| 26 | 文档上下文无冲突 |  |  |  |  |
| 27 | 定义的操作流程与实际操作流程一致 |  |  |  |  |
| 28 | 文档中图文一致无冲突 |  |  |  |  |

## 代码审查单

C语言代码审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 审查内容 | | 审查结果（填√） | | | 备注 |
| 通过 | 未通过 | 不适用 |
|  | 存储类 | 未使用内存的内容是否影响系统安全  其处理是否得当 | √ |  |  |  |
|  | 动态申请内存是否判断成功后使用 |  |  |  |  |
|  | 动态分配的内存是否及时释放 |  |  |  |  |
|  | 防止内存操作越界 |  |  |  |  |
|  | 声明定义 | 变量是否正确定义 |  |  |  |  |
|  | 不同标识符的书写是否清晰可辨 |  |  |  |  |
|  | 数组声明时，其大小是否定义好 |  |  |  |  |
|  | 共用体、结构体、枚举的所有成员变量是否进行了初始化 |  |  |  |  |
|  | 指针使用 | 是否存在指针越界（指针指向的地址不超过了有效地址范围） |  |  |  |  |
|  | 不应将局部变量的地址赋值给外部指针 |  |  |  |  |
|  | 是否对参数指针进行赋值 |  |  |  |  |
|  | 参数变量的传递是否使用了无类型指针 |  |  |  |  |
|  | 是否使用了野指针 |  |  |  |  |
|  | 类型转换 | 数据类型转换是否正确 |  |  |  |  |
|  | 隐式的整型转换是否改变了基础类型的符号 |  |  |  |  |
|  | 整型或浮点的隐式转换是否缩小了基础类型的大小 |  |  |  |  |
|  | 指针变量赋予非指针变量时是否进行了强制转换 |  |  |  |  |
|  | 是否使用无实质作用的类型转换 |  |  |  |  |
|  | 比较判断 | 是否进行了浮点相等比较（不能进行浮点相等比较，也不能与0值进行相等比较，包括不等于的比较，只能进行大于小于的比较） |  |  |  |  |
|  | 是否对无符号数进行大于或等于零或小于零的比较 |  |  |  |  |
|  | 变量有效性判断时范围是否与文档一致 |  |  |  |  |
|  | 是否使用指针进行大于或小于的逻辑比较 |  |  |  |  |
|  | 数学运算 | 选择的运算符合理吗？运算符的优先结合次序正确吗？ |  |  |  |  |
|  | 应对运算中可能存在的数据溢出（如除数为零、大数相加减乘除、数据连续加、数据连续减、根号下为负数等）情况进行处理 |  |  |  |  |
|  | 是否将bool类型用做内置运算符的操作数？（除了&&、||、!、==、!=、&和条件运算符） |  |  |  |  |
|  | 是否将enum类型用作内置运算符的操作数？（除了[]、=、==、!=、&、<、<-、>、->运算符） |  |  |  |  |
|  | 运算符!、&&或||的每个操作数是否都是bool类型？ |  |  |  |  |
|  | 对变量进行移位运算是否超出变量长度 |  |  |  |  |
|  | 是否将越界整数赋给整型变量 |  |  |  |  |
|  | 数组下标是否是大于等于零的整型数 |  |  |  |  |
|  | 非枚举类型变量是否使用了枚举类型的值 |  |  |  |  |
|  | 是否存在一个很大的数与一个很小的数进行+，-运算 |  |  |  |  |
|  | 是否给无符号类型变量赋负值 |  |  |  |  |
|  | 数据精度是否满足运算要求 |  |  |  |  |
|  | 需要时，应对运算进行正确的限幅（输出的结果是否有上下限要求） |  |  |  |  |
|  | 变量的定义类型和使用类型是否一致 |  |  |  |  |
|  | 函数使用 | 函数是否返回了定义在函数内部的局部变量 |  |  |  |  |
|  | 有返回值函数的返回语句是否带有返回值 |  |  |  |  |
|  | 全部所需的参数是否已传送给每一个被调用的模块（参数传的全不全） |  |  |  |  |
|  | 被传送的参数值是否正确的设置 |  |  |  |  |
|  | 函数定义与声明一致 |  |  |  |  |
|  | 函数的错误返回值都必须进行处理 |  |  |  |  |
|  | 实参和形参所使用的变量类型和个数是否一致 |  |  |  |  |
|  | 程序语言的使用 | 文件、数据库和注册表等打开后，在对其进行操作之后是否进行了关闭 |  |  |  |  |
|  | 用于表示字符串的数组是否以‘\0’结束 |  |  |  |  |
|  | 函数宏参数的定义是否使用括号 |  |  |  |  |
|  | for循环中控制变量是否为局部变量 |  |  |  |  |
|  | for循环中控制变量是否为整型变量 |  |  |  |  |
|  | for循环中是否对循环控制变量进行修改 |  |  |  |  |
|  | 寄存器使用 | 寄存器使用是否按照芯片手册的操作要求 |  |  |  |  |
|  | 宏扩展或子程序调用是否使用了已使用着的寄存器而未保存数据 |  |  |  |  |
|  | 默认使用的寄存器的值是否正确（注意每个寄存器的初始默认值） |  |  |  |  |
|  | 可维护性 | 嵌套的IF、ELSE、WHILE等语句是否已正确地缩进、是否少括号，是否忘写了括号 |  |  |  |  |
|  | 注释准确并且有意义 |  |  |  |  |
|  | 注释语句应不少于20 |  |  |  |  |
|  | 是否存在嵌套注释 |  |  |  |  |
|  | 标号和子程序名符合代码的逻辑含义吗 |  |  |  |  |
|  | 避免使用不易理解的数字，用有意义的标识来替代，涉及物理状态或者含有物理意义的常量，不应直接使用数字，必须用有意义的枚举或宏来代替 |  |  |  |  |
|  | 注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级 |  |  |  |  |
|  | 逻辑 | 代码是否做了设计规定的内容 |  |  |  |  |
|  | 全部设计是否均已实现（和详细设计的一致性） |  |  |  |  |
|  | 每个循环是否执行正确的次数 |  |  |  |  |
|  | 是否存在死循环的风险（例如当等待某变量为一特定值才退出时存在风险；循环语句中必须有超时保护） |  |  |  |  |
|  | 不能使用未赋值的变量 |  |  |  |  |
|  | Switch语句必须有default分支 |  |  |  |  |
|  | 必须处理程序所能遇到的各种出错情况 |  |  |  |  |
|  | 软件多余物 | 是否有不可能执行到的代码 |  |  |  |  |
| 是否有即使不执行也不影响程序功能的指令 |  |  |  |  |
| 是否有未引用的变量、标号和常量 |  |  |  |  |
|  | 有无未引用的宏 |  |  |  |  |
|  | 其他 | 应根据看门狗触发周期正确设置其清除周期（不能超过狗咬时间） |  |  |  |  |
|  | 应根据需要监视的流程，正确设置清除看门狗的位置（在该清除的地方要清除，比如初始化阶段、各个进程等；在不该清除的地方不能清除，比如在需求中要求等待狗咬的位置等） |  |  |  |  |
|  | 对于全局变量的读写要合理 |  |  |  |  |
|  | 代码审查覆盖项 | 此处填写功能、接口等测试项中由代码审查覆盖的测试项 |  |  |  |  |

C++语言代码审查单

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 审查内容 | | 审查结果（填√） | | | 备注 |
| 通过 | 不通过 | 不适用 |
|  | 存储类 | 未使用内存的内容是否影响系统安全  其处理是否得当 |  |  |  |  |
|  | 动态申请内存是否判断成功后使用 |  |  |  |  |
|  | 动态分配的内存是否及时释放 |  |  |  |  |
|  | 防止内存操作越界 |  |  |  |  |
|  | 声明定义 | 变量是否正确定义 |  |  |  |  |
|  | 不同标识符的书写是否清晰可辨 |  |  |  |  |
|  | 数组声明时，其大小是否定义好 |  |  |  |  |
|  | 共用体、结构体、枚举的所有成员变量是否进行了初始化 |  |  |  |  |
|  | 共用体、结构体、枚举的所有成员变量是否进行了初始化数组声明时，其大小是否定义好 |  |  |  |  |
|  | 对象是否赋值给重叠对象 |  |  |  |  |
|  | 指针使用 | 是否存在指针越界（指针指向的地址不超过了有效地址范围） |  |  |  |  |
|  | 不应将局部变量的地址赋值给外部指针 |  |  |  |  |
|  | 是否对参数指针进行赋值 |  |  |  |  |
|  | 参数变量的传递是否使用了无类型指针 |  |  |  |  |
|  | 是否使用了野指针 |  |  |  |  |
|  | 类型转换 | 数据类型转换是否正确 |  |  |  |  |
|  | 隐式的整型转换是否改变了基础类型的符号 |  |  |  |  |
|  | 整型或浮点的隐式转换是否缩小了基础类型的大小 |  |  |  |  |
|  | 指针变量赋予非指针变量时是否进行了强制转换 |  |  |  |  |
|  | 是否使用无实质作用的类型转换 |  |  |  |  |
|  | 强制转换是否移除指针或引用类型中的const或volatile限定符 |  |  |  |  |
|  | 不相关的对象指针类型是否使用reinterpret\_cast强制转换 |  |  |  |  |
|  | 比较判断 | 是否进行了浮点相等比较（不能进行浮点相等比较，也不能与0值进行相等比较，包括不等于的比较，只能进行大于小于的比较） |  |  |  |  |
|  | 是否对无符号数进行大于或等于零或小于零的比较 |  |  |  |  |
|  | 变量有效性判断时范围是否与文档一致 |  |  |  |  |
|  | 是否使用指针进行大于或小于的逻辑比较 |  |  |  |  |
|  | 数学运算 | 选择的运算符合理吗？运算符的优先结合次序正确吗？ |  |  |  |  |
|  | 应对运算中可能存在的数据溢出（如除数为零、大数相加减乘除、数据连续加、数据连续减、根号下为负数等）情况进行处理 |  |  |  |  |
|  | 是否将bool类型用做内置运算符的操作数？（除了&&、||、!、==、!=、&和条件运算符） |  |  |  |  |
|  | 是否将enum类型用作内置运算符的操作数？（除了[]、=、==、!=、&、<、<-、>、->运算符） |  |  |  |  |
|  | 运算符!、&&或||的每个操作数是否都是bool类型？ |  |  |  |  |
|  | 对变量进行移位运算是否超出变量长度 |  |  |  |  |
|  | 是否将越界整数赋给整型变量 |  |  |  |  |
|  | 数组下标是否是大于等于零的整型数 |  |  |  |  |
|  | 非枚举类型变量是否使用了枚举类型的值 |  |  |  |  |
|  | 是否存在一个很大的数与一个很小的数进行+，-运算 |  |  |  |  |
|  | 是否给无符号类型变量赋负值 |  |  |  |  |
|  | 数据精度是否满足运算要求 |  |  |  |  |
|  | 变量的定义类型和使用类型是否一致 |  |  |  |  |
|  | 函数使用 | 函数是否返回了定义在函数内部的局部变量 |  |  |  |  |
|  | 有返回值函数的返回语句是否带有返回值 |  |  |  |  |
|  | 全部所需的参数是否已传送给每一个被调用的模块（参数传的全不全） |  |  |  |  |
|  | 被传送的参数值是否正确的设置 |  |  |  |  |
|  | 函数定义与声明一致 |  |  |  |  |
|  | 函数的错误返回值都必须进行处理 |  |  |  |  |
|  | 实参和形参所使用的变量类型和个数是否一致 |  |  |  |  |
|  | 函数是否返回了通过引用或const引用传递的参数 |  |  |  |  |
|  | 程序语言的使用 | 文件、数据库和注册表等打开后，在对其进行操作之后是否进行了关闭 |  |  |  |  |
|  | 用于表示字符串的数组是否以‘\0’结束 |  |  |  |  |
|  | 函数宏参数的定义是否使用括号 |  |  |  |  |
|  | for循环中控制变量是否为局部变量 |  |  |  |  |
|  | for循环中控制变量是否为整型变量 |  |  |  |  |
|  | for循环中是否对循环控制变量进行修改 |  |  |  |  |
|  | 有多个线程的程序中，资源分配是否合理，会不会造成死锁 |  |  |  |  |
|  | 如果有运算符重载，则检查运算符重载是否正确 |  |  |  |  |
|  | 类与继承 | 基类虚拟函数的参数缺省值在派生类重写函数中是否被改变 |  |  |  |  |
|  | 纯虚函数被虚拟函数重写后，是否再重新声明为纯虚函数 |  |  |  |  |
|  | 在继承层次的每个路径中每个虚函数的定义是否多于1个 |  |  |  |  |
|  | 构造函数 | 类的所有构造函数是否显示调用所有中间基类和所有虚基类的构造函数 |  |  |  |  |
|  | 拷贝构造函数是否初始化基类以及类中的静态成员 |  |  |  |  |
|  | 构造函数是否使用了全局变量 |  |  |  |  |
|  | 异常处理 | 是否使用goto或switch语句将控制流传递到try或catch块中 |  |  |  |  |
|  | throw语句中的表达式是否引发新的异常 |  |  |  |  |
|  | NULL是否被作为指针异常对象抛出 |  |  |  |  |
|  | 函数抛出的异常类型是否与函数声明时指定的异常类型一致 |  |  |  |  |
|  | 异常是否是在程序开始之后且程序结束之前被抛出 |  |  |  |  |
|  | 析构函数在退出时是否含有异常 |  |  |  |  |
|  | 模板使用 | 在具有基类的类模板中，任何可以在基类中找到的标识符是否使用类标识或this->来引用 |  |  |  |  |
|  | 模板参数的显示实例化是否是程序不规范 |  |  |  |  |
|  | 可维护性 | 嵌套的IF、ELSE、WHILE等语句是否已正确地缩进、是否少括号，是否忘写了括号 |  |  |  |  |
|  | 注释准确并且有意义 |  |  |  |  |
|  | 注释语句应不少于20% |  |  |  |  |
|  | 是否存在嵌套注释 |  |  |  |  |
|  | 标号和子程序名符合代码的逻辑含义吗 |  |  |  |  |
|  | 避免使用不易理解的数字，用有意义的标识来替代，涉及物理状态或者含有物理意义的常量，不应直接使用数字，必须用有意义的枚举或宏来代替 |  |  |  |  |
|  | 注意运算符的优先级，并用括号明确表达式的操作顺序，避免使用默认优先级 |  |  |  |  |
|  | 逻辑 | 代码是否做了设计规定的内容 |  |  |  |  |
|  | 全部设计是否均已实现（和详细设计的一致性） |  |  |  |  |
|  | 每个循环是否执行正确的次数 |  |  |  |  |
|  | 是否存在死循环的风险（例如当等待某变量为一特定值才退出时存在风险；循环语句中必须有超时保护） |  |  |  |  |
|  | 不能使用未赋值的变量 |  |  |  |  |
|  | Switch语句必须有default分支 |  |  |  |  |
|  | 必须处理程序所能遇到的各种出错情况 |  |  |  |  |
|  | 软件多余物 | 是否有不可能执行到的代码 |  |  |  |  |
|  | 是否有即使不执行也不影响程序功能的指令 |  |  |  |  |
|  | 是否有未引用的变量、标号和常量 |  |  |  |  |
|  | 有无未引用的宏 |  |  |  |  |
|  | 其他 | 应根据看门狗触发周期正确设置其清除周期（不能超过狗咬时间） |  |  |  |  |
|  | 应根据需要监视的流程，正确设置清除看门狗的位置（在该清除的地方要清除，比如初始化阶段、各个进程等；在不该清除的地方不能清除，比如在需求中要求等待狗咬的位置等） |  |  |  |  |
|  | 对于全局变量的读写要合理 |  |  |  |  |
|  | 代码审查覆盖项 | 此处填写功能、接口等测试项中由代码审查覆盖的测试项 |  |  |  |  |

汇编语言代码审查单

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 审查内容 | 审查结果（填√） | | | 备注 |
| 通过 | 不通过 | 不适用 |
|  | 禁止子程序名被重用 |  |  |  |  |
|  | 禁止标号名被重用 |  |  |  |  |
|  | 未用中断的中断向量要有处理 |  |  |  |  |
|  | 保留字、标号、子程序名必须统一大小写 |  |  |  |  |
|  | 不允许以大小写敏感来区分变量 |  |  |  |  |
|  | 必要使用难以区分的相近字符作变量、标号及子程序名 |  |  |  |  |
|  | 变量、标号及子程序名不能超过31个字符或少于2个字符 |  |  |  |  |
|  | 编程时应有CAST标识的显示模块化 |  |  |  |  |
|  | 每个子程序一定要做到唯一入口、唯一出口 |  |  |  |  |
|  | 中断处理程序中的压栈、出栈指令要匹配 |  |  |  |  |
|  | 应显示对T1或者R1进行清零操作 |  |  |  |  |
|  | 切换工作寄存器时，应使用SETB、CLR指令操作PSW寄存器 |  |  |  |  |
|  | 对于需求文档中明确规定的数值应使用常数定义而不直接使用立即数 |  |  |  |  |
|  | 应使用寄存器名对SFR（特殊功能寄存器）进行寻址 |  |  |  |  |
|  | 程序复位要进行两次中断返回 |  |  |  |  |
|  | 禁止把特殊功能寄存器（SFR）作通用寄存器使用 |  |  |  |  |
|  | 禁止程序中的垃圾代码 |  |  |  |  |
|  | 注释比例不少于20% |  |  |  |  |

代码审查项（C语言）

|  |  |  |
| --- | --- | --- |
| 序号 | 审查项 | 审查内容 |
|  | Use of Uninitialized Data  ---使用未初始化的数据 | Uninitialized Variable – possible  ---变量可能未初始化 |
|  | Uninitialized Variable  ---变量未初始化 |
|  | Uninitialized Array – possible  ---队列元素可能未初始化 |
|  | Uninitialized Array  ---队列元素未初始化 |
|  | Partialy Uninitialized Array  ---队列元素部分初始化 |
|  | Uninitialized Variable in Constructor – possible  ---构造函数中变量可能未初始化 |
|  | Uninitialized Variable in Constructor  ---构造函数中变量未初始化 |
|  | Uninitialized Heap Use – possible  --使用的堆栈未初始化 |
|  | Concurrency  --并发处理 | Missing unlock for variable  ---变量缺少解锁 |
|  | Deadlock  ---程序死锁 |
|  | Memory Leaks  ---内存泄露 | Inconsistent Freeing of Memory  ---释放内存不一致 |
|  | Memory Leak – possible  ---内存可能泄漏 |
|  | Memory Leak  ---内存泄漏 |
|  | Memory Leak - in destructor  ---在析构函数中，内存泄露 |
|  | Memory Leak - possible in destructor  --在析构函数中，可能内存泄露 |
|  | Memory Leak - in operator=  --赋值操作符=，导致内存泄露 |
|  |
|  | Attempt to Use Memory after Free  ---使用已经释放  的内存 | Use of free memory (access) – possible  ---使用(读写)可能已经释放的内存 |
|  | Use of Freed Memory by Pointer  ---通过指针调用已经被释放的内存 |
|  | Use of free memory (double free) – possible  ---使用可能已经被释放的内存(重复释放) |
|  | Freeing Freed Memory  ---释放已经被释放的内存 |
|  | Use of freed memory (return) – possible  ---使用(返回)可能已经释放的内存 |
|  | Use of Freed Memory on Return  ---使用已经释放的返回内存 |
|  | Use of free memory – possible  ---使用可能已经被释放的内存 |
|  | Use of Freed Memory  ---使用已经被释放的内存 |
|  | Use of free memory (double free) - no copy constructor  ---释放已经被释放的内存(重复释放)---拷贝  构造函数没有定义 |
|  | Use of free memory (double free) – no operator=  ---释放已经被释放的内存(重复释放)---赋值操作符=没有定义 |
|  | Function returns address of local variable  ---函数返回局部变量(指针)地址 |
|  | Function returns address of local variable  ---函数返回局部变量(内存)地址 |
|  | Function returns address of local variable  ---函数返回局部变量(全局变量)地址 |
|  | Improper Memory Deallocation  ---不合理的内存释放 | Freeing Mismatched Memory - in destructor  ---在析构函数中，释放不匹配的内存 |
|  | Freeing Non-Heap Memory – possible  ---可能释放非堆栈内存 |
|  | Freeing Non-Heap Memory  --释放非堆栈内存 |
|  | Freeing Unallocated Memory – possible  ---释放可能未开辟的内存 |
|  | Freeing Unallocated Memory  ---释放未开辟的内存 |
|  | Freeing Mismatched Memory – possible  ---释放可能不匹配的内存 |
|  | Freeing Mismatched Memory  ---释放不匹配的内存 |
|  | Null Pointer Dereference  ---空指针引用取消 | Null pointer may be dereferenced  ---对空指针引用可能会被取消 |
|  | Null pointer will be dereferenced  ---对空指针引用将取消 |
|  | Null pointer may be passed to function that may dereference it  ---空指针可能会传递给函数，可能会取消对它的引用 |
|  | Null pointer will be passed to function that may dereference it  ---空指针将传递给函数，可能会取消对它的引用 |
|  | Result of function that can return NULL may be dereferenced  ---函数返回NULL 结果，可能会被取消引用 |
|  | Result of function that may return NULL will be dereferenced  ---函数可能返回NULL 结果，将被取消引用 |
|  | Pointer may be dereferenced after it was positively checked for NULL  ---经过对指针是否为空的检查后，对指针的引用可能会被取消 |
|  | Pointer will be dereferenced after it was positively checked for NULL  ---经过对指针是否为空的检查后，对指针的引用将会被取消 |
|  | NULL is dereferenced  ---对NULL 的引用被取消 |
|  | Result of function that may return NULL may be passed to another function that may dereference it  ---函数可能返回NULL 结果，并可能被传递到另一个函数，这可能会取消对它的引用 |
|  | Result of function that may return NULL will be passed to another function that may dereference it  ---函数可能返回NULL 结果，并将被传递到另一个函数，可能会取消对它的引用 |
|  | Pointer may be passed to function that can dereference it after it was positively checked for NULL  ---经过对指针是否为空的检查后，指针可能会被传递给函数，对指针的引用将被取消 |
|  | Pointer will be passed to function that may dereference it after it was positively checked for NULL  ---经过对指针是否为空的检查后，指针将被传递给函数，对指针的引用可能会取消 |
|  | NULL is passed to function that can dereference it  ---将NULL 传递给函数，这种引用将被取消 |
|  | Suspicious dereference of pointer before NULL check  ---在检查指针是否为空之前，指针的引用不可信 |
|  | Suspicious dereference of pointer in function call before NULL check  ---在检查指针是否为空之前，函数调用中指针的引用不可信 |
|  | Inappropriate Iterator Usage  ---不兼容的迭代使用 | Use of invalid iterator  ---使用无效的迭代器 |
|  | Dereference of 'end' iterator  ---取消迭代器调用end()成员函数 |
|  | Use of iterator with inappropriate container object  ---使用迭代器的不适当容器对象 |
|  | Unreachable Code  ---不可达代码 | Unreachable code  ---不可达代码 |
|  | Mismatched Return Types  ---不匹配的返回类型 | Operator= return a reference to \*this  赋值操作符=，返回内容引用了\*this |
|  | Non-void function does not return value  ---非Void 函数没有返回值 |
|  | Non-void function implicitly returning int does not return value  ---非Void 函数明确返回int 值，实际没有返回值 |
|  | Void function returns value  ---Void 函数有返回值 |
|  | Non-void function returns void value  ---非Void 函数返回void 结果 |
|  | Implicitly int function returns void value  ---明确的int 函数返回void 结果 |
|  | Buffer Overflow  ---缓冲区溢出 | Buffer Overflow - Array Index Out of Bounds  ---队列索引出超出边界，导致缓冲区溢出 |
|  | Buffer Overflow - Local Array Index Out of Bounds  ---局部队列索引超出边界，导致缓冲区溢出 |
|  | Buffer Overflow from Unvalidated Input  ---无效的输入导致缓冲区溢出 |
|  | Unvalidated User Input Causing Buffer Overflow - Non-Null Terminated String  ---无效的用户输入(非空终断字符串)导致缓冲区溢出 |
|  | Buffer Overflow - Non-null Terminated String  ---非空终断字符串导致缓冲区溢出 |
|  | Buffer Overflow - Non-null Terminated String  ---非空终断字符串可能导致缓冲区溢出 |
|  | Buffer Overflow - Array Index Out of Bounds  ---队列索引超出边界导致缓冲区溢出 |
|  | Mapping function failed  ---映射函数失败 |
|  | Buffer overflow in mapping character function  ---字符映射函数导致缓冲区溢出，使用MultiBytetoWideChar 和WidecharToMultiByte函数容易导致缓冲区溢出 |
|  | Buffer Overflow in Bound String Copy  --使用邦定的字符串拷贝，导致缓冲区溢出 |
|  | Buffer Overflow in Bound sprintf  ---使用邦定的函数sprintf，导致缓冲区溢出 |
|  | Usage of 'gets'  --使用gets 函数，该函数不检查内存边界，容易导致内存溢出 |
|  | Buffer Overflow in Unbound String Copy  ---使用未邦定的字符串拷贝，导致缓冲区溢出 |
|  | Buffer Overflow in Unbound sprintf  ---使用未邦定的函数sprintf，导致缓冲区溢出 |
|  | Buffer Overflow - Array Index may be out of Bounds  --局部队列索引可能超出边界，导致缓冲区溢出 |
|  | Input format specifier error  ---输入格式符错误 |
|  | Print functions format  ---打印函数格式问题 | Incompatible type of a print function parameter  ---打印函数中使用不匹配的参数类型 |
|  | Too few arguments in a print function call  ---调用打印函数的参数太少 |
|  | Unknown format specifier in a print function call  ---在打印函数中使用未知格式符 |
|  | Scan functions format  --扫描函数格式问题 | Incompatible type of a scan function parameter ---扫描函数参数使用不匹配类型 |
|  | Too few arguments in a scan function call  ---调用扫描函数的参数太少 |
|  | Unknown format specifier in a scan function call  ---调用扫描函数中，使用未知格式的修饰符 |
|  | Unvalidated User Input  ---不可用的用户输入 | Use of Unvalidated Integer in Memory Allocation  ---内存开辟过程中使用无效的整数 |
|  | Use of Unvalidated Data in a Format String.  ---在固定格式的字符串中，使用无效数据 |
|  | Use of Unvalidated Integer as Array Index  ---使用无效的整数作为队列索引 |
|  | Use of Unvalidated Integer in Loop Condition  ---在迭代条件中使用无效的整数 |
|  | Command Injection  ---命令行注入 |
|  | Loading File without Use of Absolute Path  ---加载文件没有使用绝对路径 |
|  | Use of Unvalidated String Data  ---使用无效字符串数据 |
|  | Weak Encryption  ---弱加密 | Use of Poor Encryption  ---使用弱加密 |
|  | Use of Poor Encryption  ---使用弱加密 |
|  | Insecure (Constant) Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致不安全的固定文件名 |
|  | Modification of Temporary File Name before Call to CreateFile  ---调用函数CreateFile 将改变临时文件名 |
|  | Missing Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致丢失临时文件名 |
|  | Insecure Temporary File Name in Call to CreateFile  ---调用函数CreateFile 导致不安全的临时文件名 |
|  | Missing Secure Temporary File Names in Call to CreateFile  ---调用函数CreateFile 将丢失安全的临时文件名 |
|  | Registry Manipulation  ---注册表篡改 | HKEY\_LOCAL\_MACHINE Used as 'hkey' Parameter for Registry Manipulation Function  --注册时使用宏HKEY\_LOCAL\_MACHINE作为参数，将导致非管理员获得权限 |
|  | DNS Spoofing  ---DNS 欺骗 | Use of INADDR\_ANY in sin\_addr.s\_addr field of struct sockaddr\_in Structure Used for Call to bind Function  ---调用函数bind 时，结构体sockaddr\_in 中sin\_addr.s\_addr 被设置为INADDR\_ANY |
|  | Suspicious Code Practices  ---不可信的代码操作 | Command Injection into Shell Execution  ---通过命令行注入Shell 再执行，无法限制输入命令内容和长度 |
|  | Format String Vulnerability  ---使用固定格式的字符串，但是没有定义字符串长度可能导致内存溢出 |
|  | Use of Privilege Elevation  ---使用提升权限 |
|  | Use of Dangerous Process Creation  ---危险进程创建，容易通过环境变量运行任意命令 |
|  | Missed conversion from network to host byte order  ---接收时，没有将网络字节顺序转换为主机字节顺序 |
|  | Missed conversion from host to network byte order  ---发送时，没有将主机字节顺序转换为网络字节顺序 |
|  | Missed conversion from network to host byte order  --读取时，没有将网络字节顺序转换为主机字节顺序 |
|  | Missed conversion from host to network byte order  ---写入时，没有将主机字节顺序转换为网络字节顺序 |
|  | Suspiciously placed semicolon  ---可疑的分号放置 |
|  | Assignment in condition (call)  ---调用的条件语句中有赋值操作 |
|  | Assignment in condition |
|  | ---在条件语句中赋值操作 |

## 代码质量度量分析表单

编码规则要求

|  |  |  |
| --- | --- | --- |
| 规则编号 | 规则描述 | 违反规则代码 |
| R-1-1-1 | 禁止通过宏定义改变关键字和基本类型含义 |  |
| R-1-1-2 | 禁止将其他标识宏定义为关键字和基本类型 |  |
| R-1-1-3 | 用typedef自定义的类型禁止被重新定义 |  |
| R-1-1-4 | 禁止重新定义C或C++的关键字 |  |
| R-1-1-5 | 禁止#define被重复定义 |  |
| R-1-1-6 | 函数中的#define和#undef必须配对使用 |  |
| R-1-1-7 | 以函数形式定义的宏，参数和结果必须用括号括起来 |  |
| R-1-1-8 | 结构、联合、枚举的定义中必须定义标识名 |  |
| R-1-1-9 | 结构体定义中禁止含有无名结构体 |  |
| R-1-1-10 | 位定义的有符号整型变量位长必须大于1 |  |
| R-1-1-11 | 位定义的整数型变量必须明确定义是有符号还是无符号的 |  |
| R-1-1-12 | 位定义的变量必须是同长度的类型且定义位禁止跨越类型的长度 |  |
| R-1-1-13 | 函数声明中必须对参数类型进行声明，并带有变量名 |  |
| R-1-1-14 | 函数声明必须与函数原型一致 |  |
| R-1-1-15 | 函数中的参数必须使用类型声明 |  |
| R-1-1-16 | 外部声明的变量，类型必须与定义一致 |  |
| R-1-1-17 | 禁止在函数体内使用外部声明 |  |
| R-1-1-18 | 数组定义禁止没有显示的边界限定 |  |
| R-1-1-19 | 禁止使用extern声明对变量初始化 |  |
| R-1-1-20 | 用于数值计算的字符型变量必须明确定义是有符号还是无符号 |  |
| R-1-1-21 | 禁止在#include语句中使用绝对路径 |  |
| R-1-1-22 | 禁止头文件重复包含 |  |
| R-1-1-23 | 函数参数表为空时，必须使用void明确说明 |  |
| R-1-2-1 | 循环体必须用大括号括起来 |  |
| R-1-2-2 | If、else if、else必须用大括号括起来 |  |
| R-1-2-3 | 禁止在头文件前有可执行代码 |  |
| R-1-2-4 | 引起二义性理解的逻辑表达式，必须使用括号显式说明优先级顺序 |  |
| R-1-2-5 | 逻辑判别表达式中的运算项必须要使用括号 |  |
| R-1-2-6 | 禁止嵌套注释 |  |
| R-1-3-1 | 禁止指针的指针超过两级 |  |
| R-1-3-2 | 函数指针的使用必须加以&明确说明 |  |
| R-1-3-3 | 禁止对参数指针进行赋值 |  |
| R-1-3-4 | 禁止将局部变量地址做为函数返回值返回 |  |
| R-1-3-5 | 禁止使用或释放未分配空间或已释放的指针 |  |
| R-1-3-6 | 指针变量被释放后必须置为NULL |  |
| R-1-3-7 | 动态分配的指针变量定义时如未被分配空间必须初始化为NULL |  |
| R-1-3-8 | 动态分配的指针变量第一次使用前必须进行是否为NULL的判别 |  |
| R-1-3-9 | 空指针必须使用NULL，禁止使用整型数0 |  |
| R-1-3-10 | 禁止文件指针在退出时没有关闭文件 |  |
| R-1-4-1 | 在if-else if语句中必须使用else分支 |  |
| R-1-4-2 | 条件判定分支如果为空，必须以单独一行的分号加注释进行明确说明 |  |
| R-1-4-3 | 禁止使用空switch语句 |  |
| R-1-4-4 | 禁止对bool量使用switch语句 |  |
| R-1-4-5 | 禁止switch语句中只包含default语句 |  |
| R-1-4-6 | 除枚举类型列举完全外，switch语句必须要有default |  |
| R-1-4-7 | Switch中的case和default必须以break或return终止，共用case必须加以明确注释 |  |
| R-1-4-8 | Switch语句的所有分支必须具有相同的层次范围 |  |
| R-1-5-1 | 禁止从复合语句外goto到复合语句内，或由下向上goto |  |
| R-1-5-2 | 禁止使用setjmp/longjmp |  |
| R-1-6-1 | 禁止将浮点常数赋值给整型变量 |  |
| R-1-6-2 | 禁止将越界整数赋值给整型变量 |  |
| R-1-6-3 | 禁止在逻辑表达式中使用赋值语句 |  |
| R-1-6-4 | 禁止对逻辑表达式进行位运算 |  |
| R-1-6-5 | 禁止在运算表达式中或函数调用参数中使用++或—操作符 |  |
| R-1-6-6 | 对变量进行移位运算禁止超出变量长度 |  |
| R-1-6-7 | 禁止移位操作中的移位数为负数 |  |
| R-1-6-8 | 数组禁止越界使用 |  |
| R-1-6-9 | 数组下标必须是大于等于零的整型数 |  |
| R-1-6-10 | 禁止对常数值做逻辑非的运算 |  |
| R-1-6-11 | 禁止非枚举类型变量使用枚举类型的值 |  |
| R-1-6-12 | 除法运算中禁止被零除 |  |
| R-1-6-13 | 禁止在sizeof中使用赋值 |  |
| R-1-6-14 | 缓存区读取操作禁止越界 |  |
| R-1-6-15 | 缓存区写入操作禁止越界 |  |
| R-1-6-16 | 禁止使用已被释放了的内存空间 |  |
| R-1-6-17 | 被free的指针必须指向最初malloc、calloc分配的地址 |  |
| R-1-6-18 | 禁止使用gets函数，应使用fgets函数替代 |  |
| R-1-6-19 | 使用字符串赋值、拷贝、追加等函数时，禁止目标字符串存储空间越界 |  |
| R-1-7-1 | 禁止覆盖标准函数库的函数 |  |
| R-1-7-2 | 禁止函数的实参和形参类型不一致 |  |
| R-1-7-3 | 实参和形参的个数必须一致 |  |
| R-1-7-4 | 禁止使用旧形式的函数参数表定义形式 |  |
| R-1-7-5 | 函数声明和函数定义中的参数类型必须一致 |  |
| R-1-7-6 | 函数声明和函数定义中的返回类型必须一致 |  |
| R-1-7-7 | 有返回值的函数必须通过返回语句返回 |  |
| R-1-7-8 | 禁止无返回值函数的返回语句带有返回值 |  |
| R-1-7-9 | 有返回值的函数的返回语句必须带有返回值 |  |
| R-1-7-10 | 函数返回值的类型必须与定义一致 |  |
| R-1-7-11 | 具有返回值的函数，其返回值如果不被使用，调用时应有（void）说明 |  |
| R-1-7-12 | 无返回值的函数，调用时禁止再用（void）重复说明 |  |
| R-1-7-13 | 静态函数必须被使用 |  |
| R-1-7-14 | 禁止同一个表达式中调用多个顺序相关函数 |  |
| R-1-7-15 | 禁止在函数参数表中使用省略号 |  |
| R-1-7-16 | 禁止使用直接或间接自调用函数 |  |
| R-1-8-1 | 禁止不可达语句 |  |
| R-1-8-2 | 禁止不可达分支 |  |
| R-1-8-3 | 禁止使用无效语句 |  |
| R-1-8-4 | 使用八进制数必须明确注释 |  |
| R-1-8-5 | 数字类型后缀必须使用大写字母 |  |
| R-1-9-1 | For循环控制变量必须使用局部变量 |  |
| R-1-9-2 | For循环控制变量必须使用整数型变量 |  |
| R-1-9-3 | 禁止在for循环体内部修改循环控制变量 |  |
| R-1-9-4 | 无限循环必须使用while(1)语句，禁止使用for(;;)等其他形式的语句 |  |
| R-1-10-1 | 浮点数变量赋值给整型变量必须强制转换 |  |
| R-1-10-2 | 长整数变量赋值给短整数变量必须强制转换 |  |
| R-1-10-3 | Double型变量赋给float型变量必须强制转换 |  |
| R-1-10-4 | 指针变量的赋值类型必须与指针变量类型一致 |  |
| R-1-10-5 | 将指针量赋予非指针变量或非指针量赋予指针变量，必须使用强制转换 |  |
| R-1-10-6 | 禁止使用无实质作用的类型转换 |  |
| R-1-11-1 | 变量禁止未赋值就使用 |  |
| R-1-11-2 | 变量初始化禁止隐含依赖于系统的缺省值 |  |
| R-1-11-3 | 结构体初始化的嵌套结构必须与定义一致 |  |
| R-1-11-4 | 枚举元素定义中的初始化必须完整 |  |
| R-1-12-1 | 禁止对逻辑量进行大于或小于的逻辑比较 |  |
| R-1-12-2 | 禁止对指针进行大于或小于的逻辑比较 |  |
| R-1-12-3 | 禁止对浮点数进行是否相等的比较 |  |
| R-1-12-4 | 禁止对无符号数进行大于等于零或小于零的比较 |  |
| R-1-12-5 | 禁止无符号数与有符号数之间的直接比较 |  |
| R-1-13-1 | 禁止局部变量与全局变量同名 |  |
| R-1-13-2 | 禁止函数形参与全局变量同名 |  |
| R-1-13-3 | 禁止变量名与函数名同名 |  |
| R-1-13-4 | 禁止变量名与标识名同名 |  |
| R-1-13-5 | 禁止变量名与枚举元素同名 |  |
| R-1-13-6 | 禁止变量名与typedef自定义的类型名同名 |  |
| R-1-13-7 | 禁止在内部块中重定义已有的变量名 |  |
| R-1-13-8 | 禁止仅依赖大小写区分的变量 |  |
| R-1-13-9 | 禁止仅依赖小写字母“l”与数字“1”区分的变量 |  |
| R-1-13-10 | 禁止仅依赖小写字母“O”与数字“0”区分的变量 |  |
| R-1-13-11 | 禁止单独使用小写字母“l”或大写字母“O”作为变量名 |  |
| R-1-13-12 | 程序外部可改写的变量，必须使用volatile类型说明 |  |
| R-1-13-13 | 禁止在表达式中出现多个同一volatile类型变量的运算 |  |
| R-1-13-14 | 禁止将NULL做为整型数0使用 |  |
| R-1-13-15 | 禁止给无符号类型变量赋负值 |  |
| R-1-13-16 | 用于表示字符串的数组必须以‘\0’结束 |  |
| R-2-1-1 | 含有动态分配成员的类，必须编写拷贝构造函数，并重载赋值操作符 |  |
| R-2-1-2 | 虚拟基类指针转换为派生类指针必须使用dynamic cast转换 |  |
| R-2-1-3 | 菱形层次结构的派生设计，对基类派生必须使用virtual说明 |  |
| R-2-1-4 | 抽象类中的复制操作符重载必须是保护的或私有的 |  |
| R-2-2-1 | 构造函数中禁止使用全局变量 |  |
| R-2-2-2 | 类中必须明确定义缺省构造函数 |  |
| R-2-2-3 | 单参数构造函数必须使用explicit声明 |  |
| R-2-2-4 | 类中所有成员变量必须在构造函数中初始化 |  |
| R-2-2-5 | 派生类构造函数必须在初始化列表中说明直接基类构造函数 |  |
| R-2-3-1 | 具有虚拟成员函数的类，析构函数必须是虚拟的 |  |
| R-2-3-2 | 析构函数中禁止存在不是由自身捕获处理的异常 |  |
| R-2-4-1 | 基类虚拟函数的参数缺省值在派生类重写函数中禁止被改变 |  |
| R-2-4-2 | 派生类对基类虚拟函数重写的声明必须使用virtual显示说明 |  |
| R-2-4-3 | 禁止非纯虚函数被纯虚函数重写 |  |
| R-2-5-1 | 禁止将不相关的指针类型强制转换为对象指针类型 |  |
| R-2-5-2 | 指针或引用的类型转换中禁止移除const或volatile属性 |  |
| R-2-6-1 | 使用new分配的内存空间，用完后必须使用delete释放 |  |
| R-2-6-2 | 必须使用delete[]释放new[]分配的内存空间 |  |
| R-2-6-3 | 被delete的指针必须指向最初new分配的地址 |  |
| R-2-7-1 | 函数中固定长度数组变量的传递必须使用引用方式 |  |
| R-2-7-2 | 定义为const的成员函数禁止返回非const的指针或引用 |  |
| R-2-7-3 | 禁止可导致非资源性对象数据被外部修改的成员函数返回 |  |
| R-2-8-1 | 捕获的顺序必须按由派生类到基类的次序排序 |  |
| R-2-8-2 | 每个指定的抛出必须由与之匹配的捕获 |  |
| R-2-8-3 | 异常抛出的对象必须使用引用方式捕获 |  |
| R-2-8-4 | 缺省捕获必须放在所有指定捕获之后 |  |
| R-2-8-5 | 禁止显式直接抛出NULL |  |
| R-2-9-1 | 模板的声明、定义与实现必须在同一个文件之中 |  |

## 测评大纲评审问题及处理意见

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 问题 | 提出人 | 采纳情况 |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

————————