虚拟化系统测试报告

测试团队介绍：（包括报告编写人，分工情况等）

目 录

[目 录 1](#_Toc197108650)

[1 报告概述 3](#_Toc197108651)

[1.1 测试目的 3](#_Toc197108652)

[（说明测试的目标，如功能验证、性能评估、安全性测试等） 3](#_Toc197108653)

[1.2 测试范围 3](#_Toc197108654)

[（描述测试的虚拟化系统范围及测试场景） 3](#_Toc197108655)

[1.3 术语定义 3](#_Toc197108656)

[2 测试环境 3](#_Toc197108657)

[2.1 硬件配置 3](#_Toc197108658)

[2.2 软件配置 3](#_Toc197108659)

[2.3 网络配置 3](#_Toc197108660)

[3 测试内容 3](#_Toc197108661)

[3.1 功能测试 3](#_Toc197108662)

[3.1.1 基础功能 3](#_Toc197108663)

[3.1.2 任务抢占机制测试 3](#_Toc197108664)

[3.1.3 中断处理机制测试 4](#_Toc197108665)

[3.1.4 事件管理功能测试 4](#_Toc197108666)

[3.1.5 消息队列通信功能测试 4](#_Toc197108667)

[3.2 性能测试 4](#_Toc197108668)

[3.2.1 CPU性能测试 4](#_Toc197108669)

[3.2.2 内存分配性能测试 4](#_Toc197108670)

[3.2.3 进程通信性能测试 4](#_Toc197108671)

[3.2.4 死锁检测与恢复机制测试 4](#_Toc197108672)

[3.3 安全性测试 4](#_Toc197108673)

[3.3.1 隔离性测试 4](#_Toc197108674)

[3.3.2 异常处理健壮性测试 5](#_Toc197108675)

[4 测试方法 5](#_Toc197108676)

[4.1 功能测试 5](#_Toc197108677)

[4.2 性能测试 5](#_Toc197108678)

[4.3 安全性测试 5](#_Toc197108679)

[5 测试结果与分析 5](#_Toc197108680)

[6 结论与建议 5](#_Toc197108681)

[7 附录 5](#_Toc197108682)

# 报告概述

## 测试目的

## （说明测试的目标，如功能验证、性能评估、安全性测试等）

## 测试范围

## （描述测试的虚拟化系统范围及测试场景）

## 术语定义

（关键术语解释，如VCPU：虚拟化CPU）

# 测试环境

## 硬件配置

（开发板描述等）

## 软件配置

（虚拟化平台版本，宿主机和客户机操作系统，管理工具等）

## 网络配置

（宿主机，客户机，多个虚拟化系统，之间的网络拓扑关系）

# 测试内容

## 功能测试

### 基础功能

例如：

虚拟化系统整体的搭建和启动

虚拟化资源的修改（VCPU数量，内存分配）

### 任务抢占机制测试

参考指标

抢占响应时间（低优先级任务被高优先级任务打断的延迟）

调度顺序正确性

优先级反转发生次数

### 中断处理机制测试

参考指标

中断响应延迟、中断嵌套深度设计（允许更高优先级中断打断当前中断处理的层数统计）、中断使能/屏蔽正确率

### 事件管理功能测试

参考指标

事件传递成功率、同步/异步事件延迟、并发事件处理准确性

### 消息队列通信功能测试

参考指标

消息投递/接收成功率

阻塞/非阻塞发送的行为是否一致

消息优先级是否正确执行

## 性能测试

### CPU性能测试

参考指标

上下文延迟、上下文切换次数、CPU使用率

### 内存分配性能测试

参考指标

内存分配/释放时间、最大可用内存块大小、内存碎片率

### 进程通信性能测试

参考指标

单消息传输延迟、批量消息吞吐率、通信错误率（丢包等）

### 死锁检测与恢复机制测试

参考指标

死锁检测执行时间、恢复成功率、假死锁误报率

## 安全性测试

### 隔离性测试

**测试目的**

验证在 Linux VM 上运行高 cpu、内存开销应用时，不

影响 reworks VM 上任务的执行时间（CPU的隔离性），不影响内存的读写时延（内存的隔离性）以及内存访问冲突率。

### 异常处理健壮性测试

**测试目的**

测试本系统对于异常处理的健壮性，分为异常捕获率、异常响应时间、异常处理函数执行正确性。

# 测试方法

**3.3.1测试方法**

1. **CPU隔离性测试：**编写高CPU程序，同时记录运行前后的时间，在 Linux VM 空闲和忙碌的时候进行统计，各运行获取数据，比较两次执行的时间消耗，如果结果相似则表面有良好的隔离性。预期结果为时间消耗相似。
2. **内存隔离性测试**：编写具有内存开销的程序，同时记录运行前后的内存访问时延，在 Linux VM 空闲和忙碌的时候进行统计，各运行获取数据，比较两次执行的时间消耗，如果结果相似则表面有良好的隔离性。预期结果为时间消耗相似。
3. **内存访问冲突率：**编写程序。通过测量内存访问延迟来间接评估内存访问冲突率，思路是通过反复访问同一缓存行的不同位置来制造内存竞争，从而观察延迟变化。预期结果为两次时延相近。

3.3.2测试方法

1. **异常捕获率测试**：本实验使用空指针引用作为目标异常，程序中注册处理异常处理函数以记录每次异常触发后正常处理情况。最终如果正常处理次数与触发异常次数相近，则异常处理的健壮性良好。预期应当良好。
2. **异常响应时间：**本实验使用空指针引用作为目标异常，程序中在触发异常前记录当前时间戳，注册处理异常处理函数以记录每次异常触发时的时间戳，利用两者数据计算出一次异常处理前后的时间间隔。如果最终时间间隔足够短，则异常处理的健壮性良好。预期应当良好。
3. **异常处理函数执行正确性：**本实验使用空指针引用作为目标异常，程序中注册处理异常处理函数以记录每次异常触发后正常处理情况。最终如果能有效调用异常处理函数，则异常处理的健壮性良好。预期应当良好。

## 功能测试

### 基础功能

## 性能测试

## 安全性测试

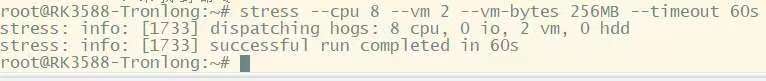
# 测试结果与分析

3.3.1结果

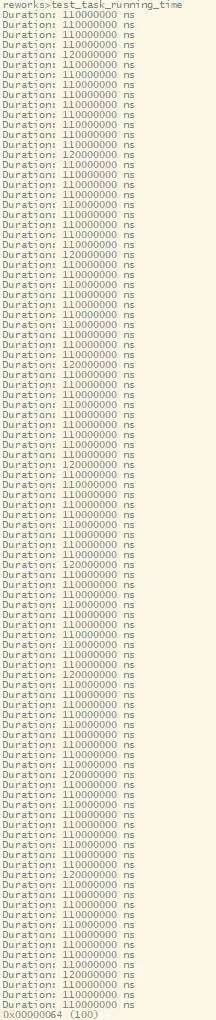
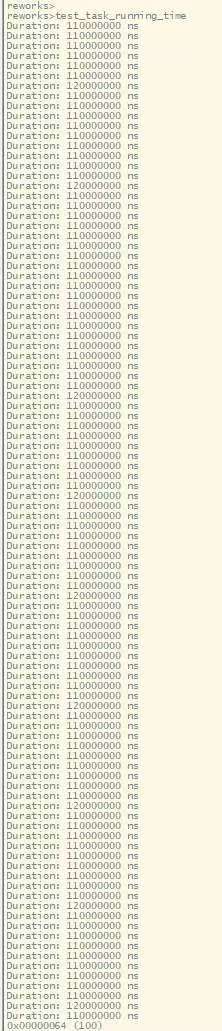
**实际结果：**

1. **CPU隔离性测试结果**

运行在Linux VM的高CPU、内存开销的程序为

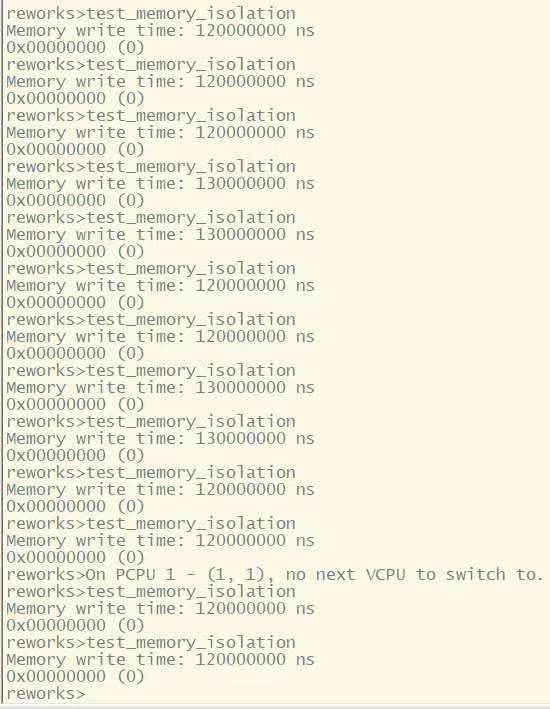
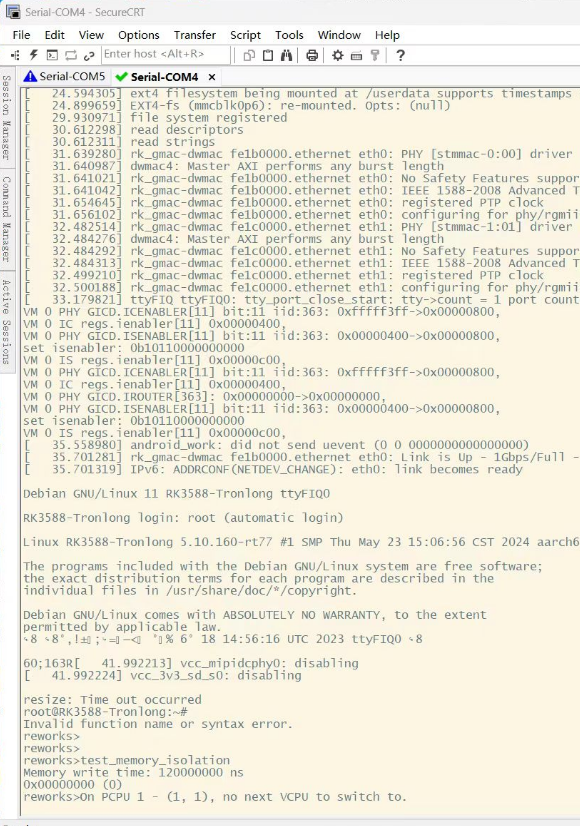
****

左图为Linux VM 空闲时结果，右图为Linux VM负载时的结果，可以看出数据相似，说明CPU隔离性良好。

****

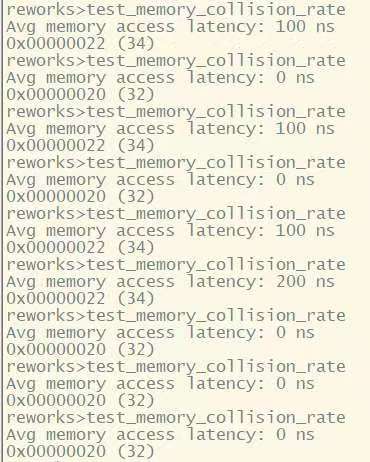
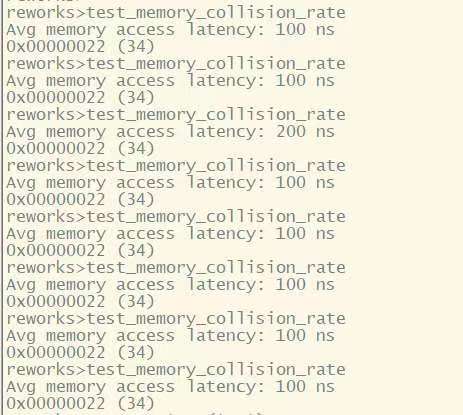
1. **内存隔离性测试结果**

左图为Linux VM 空闲时结果，右图为Linux VM负载时的结果，可以看出内存读写的时延数据相似，说明内存隔离性良好。



1. **内存访问冲突率**

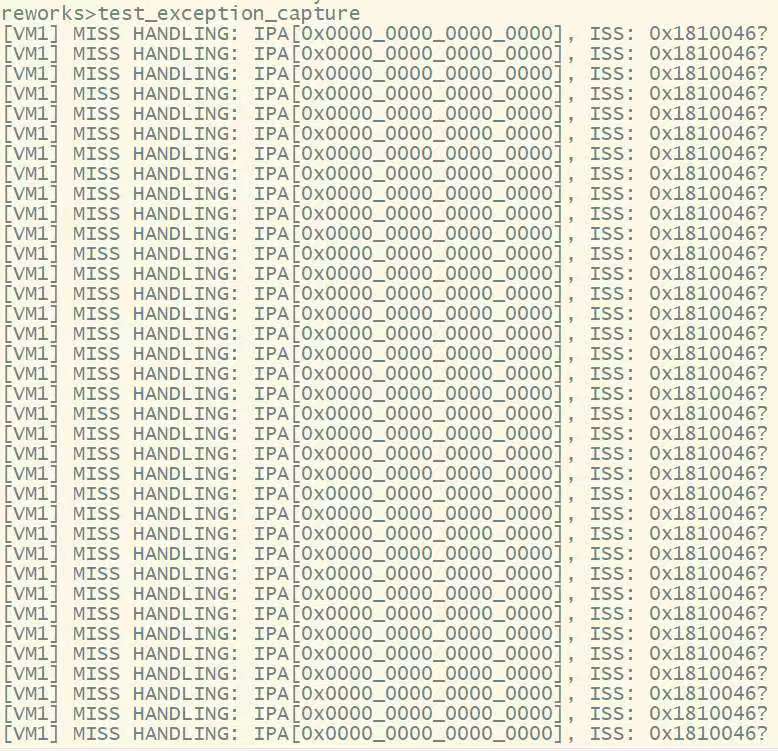
左图为Linux VM 空闲时结果，右图为Linux VM负载时的结果，可以看出内存读写的内存访问冲突率数据相差较大。

****

**3.3.2测试结果**

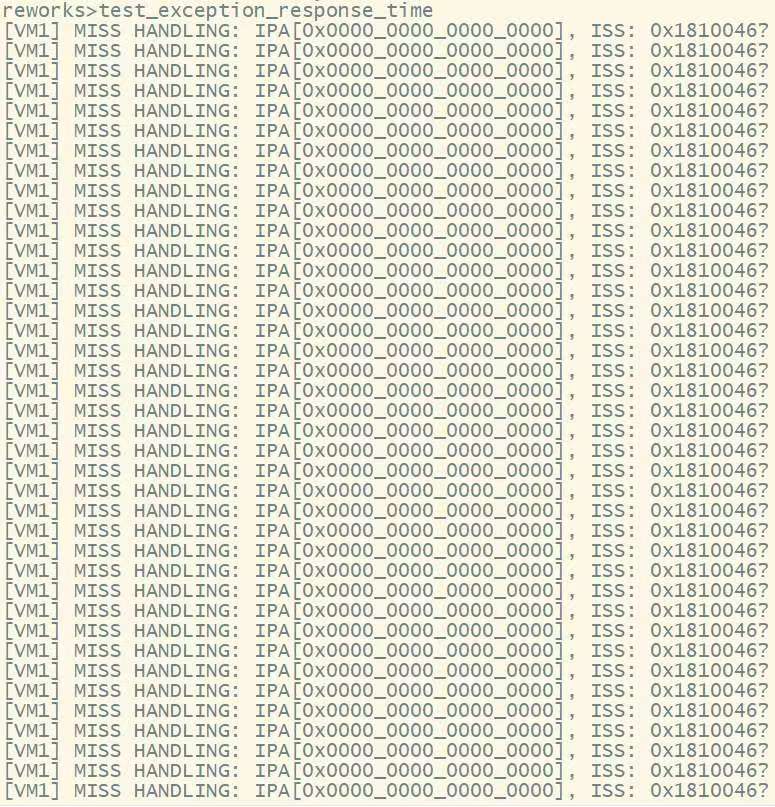
1. **异常捕获率测试**

结果与预期（异常触发后由注册的异常处理函数捕获，且计算捕获率）不符，原因是ReWorks 虚拟机监控器未能成功处理该异常，或者没有把它正确地转发给注册的信号处理器。结果如下图：



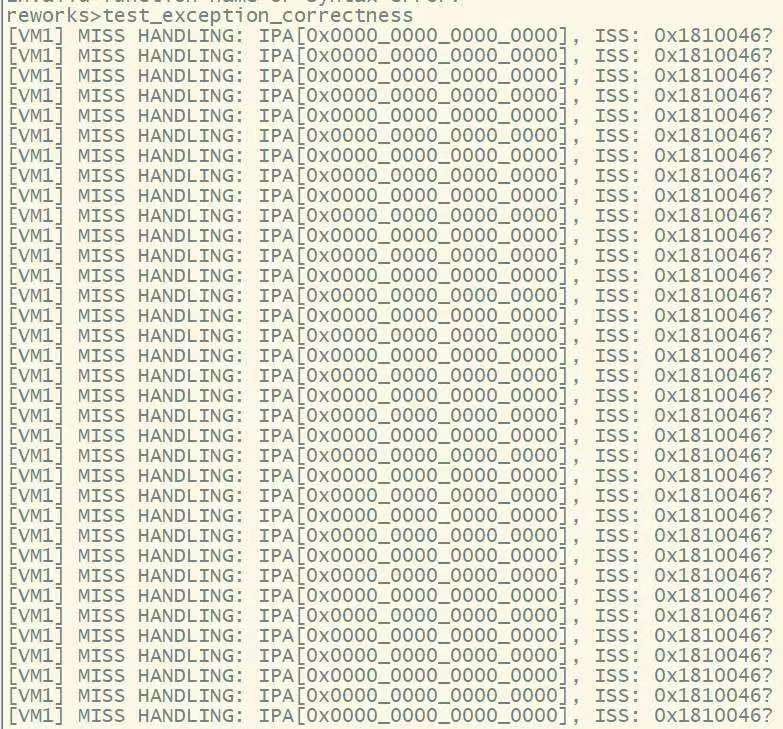
1. **异常相应时间**

如图，未能有效处理异常



1. **异常处理函数执行正确性**

如图，未能有效处理异常



# 结论与建议

# 附录

关键测试日志

测试用例表

测试方法的代码压缩包