MiniFS文件管理系统

软件测试方案

Nullptr小组

2018年9月10日

目录

[1. 文档说明 3](#_Toc524372625)

[1.1变更记录 3](#_Toc524372626)

[1.2文档目的 3](#_Toc524372627)

[2. 引言 3](#_Toc524372628)

[2.1 项目背景 3](#_Toc524372629)

[2.2 测试目标 3](#_Toc524372630)

[2.3 参考文档 3](#_Toc524372631)

[2.4 术语和缩略语 3](#_Toc524372632)

[3. 测试要求 4](#_Toc524372633)

[3.1 测试环境 4](#_Toc524372634)

[3.2 测试需求分析 4](#_Toc524372635)

[3.3 测试策略 4](#_Toc524372636)

[3.4 测试过程 4](#_Toc524372637)

[4. 测试用例通过记录 5](#_Toc524372638)

[4.1功能测试 5](#_Toc524372639)

[4.2 性能测试 5](#_Toc524372640)

[4.3 安全性测试 5](#_Toc524372641)

## 1. 文档说明

### 1.1变更记录

|  |  |  |  |
| --- | --- | --- | --- |
| 变更时间 | 变更人 | 版本 | 变更备注 |
| 2018年9月10日 | 全体 | 1.0 | 第一次合代码,v1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### 1.2文档目的

本测试方案针对《MiniFS文件管理系统》，为开发者提供关于MiniFS项目整体系统功能、性能水平和程序安全健壮性的测试指导。同时，本文档也是记录测试结果和判断测试是否完整的重要依据。

## 2. 引言

### 2.1 项目背景

MiniFS是在Windows环境下实现的文件管理系统。在创建的硬盘空间中，实现命令行下对文件及目录进行操作，具体功能包括创建、删除新文件，创建文件目录及复制、移动文件及目录等操作。

### 2.2 测试目标

在交付软件之前，尽可能的发现软件中存在的功能错误并及时修复，基本判断软件的性能水平，同时尽可能排除软件中潜在的由于非预期输入所导致的严重错误。最终提高软件的可用性、性能水平及健壮性。

系统测试目标：软件功能、软件性能、软件健壮性。

### 2.3 参考文档

<https://wenku.baidu.com/view/23450d7bcc1755270622083d.html>

<https://wenku.baidu.com/view/e355a1af8662caaedd3383c4bb4cf7ec4bfeb656.html>

<https://www.cnblogs.com/shengs/p/4094327.html>

<https://www.cnblogs.com/zelmeli/archive/2012/06/08/2542032.html>

<https://www.cnblogs.com/studyskill/p/7115913.html>

<https://blog.csdn.net/qq_27446553/article/details/52282480>

https://wenku.baidu.com/view/f4bd9082f705cc17552709ec.html

### 2.4 术语和缩略语

|  |  |
| --- | --- |
| 术语/缩略语 | 说明 |
| Fuzz | 模糊测试技术，向程序进行随机输入，并  监测程序异常 |
| AFL | American Fuzz Loop，一种高效的自动化漏洞挖掘工具 |

## 3. 测试要求

### 3.1 测试环境

#### 3.1.1 硬件环境

|  |  |
| --- | --- |
| 主机 | 个人微机 |
| CPU | 英特尔处理器 |
| 内存 | 8G DDR4 |
| 硬盘 | 机械硬盘1TB |

#### 3.1.2 软件环境

|  |  |
| --- | --- |
| 操作系统： | Windows10 |
| 软件工具 | Visual Studio， 010 Editor |

### 3.2 测试需求分析

#### 3.2.1测试的功能：

create：建立一个新mini-FS空间

mount：命令安装要操作的空间，在硬盘中开辟一块外存空间进行操作利用

fmt：初始化当前的空间

close：退出mini-FS

help：显示帮助信息

dr：显示空间的文件目录

md：创建子目录

cp：复制文件或者子目录

dl：删除空间中的文件或者子目录

cd：打开目标目录

tp：显示空间中的文本文件

att：显示文件或者目录属性

more：分页显示空间的文本文件

#### 3.2.2 性能测试内容

多文件查找：能够在1-2s内响应文件查找。

大文件读写：在一分钟内完成500M文件的载入。

#### 3.2.3 安全性测试

健壮性：能够对非预期的非法输入进行处理，保证程序正常运行。

### 3.3 测试策略

针对系统功能测试和性能测试，采用手工测试的方法。按照边界测试、等价类划分等手段生成测试用例，采用白盒测试，通过判断输入是否能够实现预期的效果和记录软件性能情况等进行测试。

对于系统安全性测试，采用自动化测试工具Winafl进行测试。以功能测试和性能测试生成的测试用例作为种子，通过对种子输入的变异，实现多种、大量、非预期、非法输入用例，对软件进行“攻击性”输入测试，并统计程序崩溃数量，判断程序对异常情况处理的能力，即程序健壮性。

### 3.4 测试过程

#### 3.4.1 功能测试

|  |  |
| --- | --- |
| 测试范围 | 验证软件全部操作功能实现 |
| 测试目标 | 核实所有功能均已正常实现，是否满足需求 |
| 采用技术 | 主要采用白盒测试、边界测试、等价类划分等方法。 |
| 工具与方法 | 手动测试 |
| 测试用例 | 见测试用例文件 |
| 通过标准 | 全部功能正常实现，无不正常结果 |
|  |  |

#### 3.4.2 性能测试

|  |  |
| --- | --- |
| 测试范围 | 检测软件进行多文件读取、大文件存储、读取的性能情况 |
| 测试目标 | 基本确定软件对大文件、多文件的耗时、内存占用等性能水平。 |
| 采用技术 | 采用白盒测试、边界测试、等价类划分等方法。 |
| 工具与方法 | 手动测试 |
| 测试用例 | 见测试用例文件 |
| 通过标准 | 记录不同操作的耗时并分析 |
|  |  |

#### 3.4.3 安全性测试

|  |  |
| --- | --- |
| 测试范围 | 大量多种非预期非法输入下程序正常运行 |
| 测试目标 | 尽可能排除潜在的处理错误导致的程序崩溃 |
| 采用技术 | Fuzz， AFL |
| 工具与方法 | 使用Winafl进行自动化白盒测试 |
| 测试用例 | 使用功能测试和性能测试的用例作为种子进行变异 |
| 通过标准 | 运行Winafl一小时内出现不超过20个crash |
|  |  |

## 4. 测试用例通过记录

### 4.1功能测试

|  |  |
| --- | --- |
| 测试用例 | 通过情况 |
|  |  |
|  |  |
|  |  |
|  |  |

### 4.2 性能测试

### 4.3 安全性测试