### 设置和测试程序

### dd（GNU fileutils）4.0.36取证测试

### 摘要

本文档介绍了在Linux平台上测试dd（GNU fileutils）4.0.36作为磁盘映像工具的情况。使用的Linux版本是Linux版本2.4.2-2（Red Hat Linux 7.1 2.96-79）。磁盘映像工具规范版本3.1.6中介绍了应用的测试用例。使用的Linux版本是Linux版本2.4.2-2（Red Hat Linux 7.1 2.96-79）。磁盘映像工具规范版本3.1.6中介绍了应用的测试用例。

测试运行在五台933Mhz的电脑。测试中使用了各种硬盘（7种不同型号，5种主要品牌）。源磁盘（被复制的磁盘）使用FAT16，FAT32，NTFS或Linux EXT2类型分区进行设置，以表示最常见的分区类型。

本文档的主要目标是提供关于测试过程的足够信息，以便对过程进行独立评估或独立复现结果。本文档的目标读者应该熟悉DOS操作系统，Linux（或类似UNIX）操作系统，计算机操作，计算机硬件组件（如硬盘驱动器），硬盘驱动器接口（如IDE或SCSI）和计算机取证。

介绍

计算机取证工具测试项目的目标是提供一种保证措施，确保计算机取证调查中使用的工具能够产生准确的结果。这是通过开发计算机取证工具的规格和测试方法，然后测试特定的工具来完成的。测试结果为工具制造商改进工具提供了必要的信息，使用户能够在获取和使用计算机取证工具方面做出明智的选择，并使法律界和其他人了解工具的能力。我们的测试计算机取证工具的方法基于公认的一致性测试和质量测试方法。

CFTT是国家司法研究所，国家标准与技术研究院和其他机构如技术支持工作组的联合项目。整个计算机取证社区通过评论草案，并在NIST网站http://www.cftt.nist.gov/上发布，帮助制定规范和测试方法。

本文档描述了用于在Linux平台上测试dd（GNU fileutils）4.0.36磁盘映像工具的过程。使用的Linux版本是Linux版本2.4.2-2（Red Hat Linux 7.1 2.96-79）。磁盘映像工具规范版本3.1.6中介绍了应用的测试用例。本文档的主要目标是提供关于测试过程的足够信息，以便对过程进行独立评估或独立复现结果。如要尝试对报告的测试结果进行独立复现，NIST以外的机构或实验室需要足够的硬件和软件资源来执行测试用例以及本文档，dd测试报告和磁盘映像工具规范版本3.1.6。由于NIST所使用的硬件可能存在无法获取的情况，因此必须进行调整和替换才能运行测试用例。第7部分提供了适应不同于NIST环境的测试环境的建议。

本文档的目标读者应该熟悉DOS操作系统，Linux（或类似UNIX）操作系统，计算机操作，计算机硬件组件（如硬盘驱动器），硬盘驱动器接口（如IDE或SCSI）和计算机取证。

1.1 测试概述

为了完成测试，必须安装和准备几个工具。这些包括执行测试的计算机，硬盘驱动器，可移动媒体，支持软件（FS-TST版本1.0）以及控制测试过程的脚本。支持软件FS-TST：Forensic软件测试支持工具可从以下网站获得：<http://www.cftt.nist.gov。>

选择硬盘的一个子集作为初始设置，作为测试用例的源驱动器。源驱动器设置一次，然后用于多个测试用例。所有的组件准备好了测试用例方可运行。所有的测试案例都遵循包含三个步骤的执行计划。

1.准备目标驱动器。指定的值写入目标驱动器的每个扇区。如果需要分区，则会创建并格式化该分区。这一步在DOS环境下执行。

2.运行dd，这一步在Linux环境下执行。

3.评估结果。复制的准确性和完整性通过逐个的比较来检查。通过比较执行dd前执行的SHA-1和执行dd后执行的SHA-1，检查源驱动器是否有任何更改。这一步在DOS环境下执行。

对于每个测试用例，需要执行的命令都包含在每个步骤的一个脚本文件中。除了创建分区和格式化之外，设置每个测试用例和评估结果所需的程序都包含在FS-TST包中。

1.2 测试用例选择

并非所有在“磁盘映像工具规范”3.1.6版中定义的测试用例都适用于所有工具。表1.2-1列出了Disk Imaging Tool Specification，Version 3.1.6中定义的每个测试用例的参数。测试用例是为基于Linux的通用映像工具编写的。并非所有情况都适用于dd。在dd的情况下，运行32个测试用例，20个测试用例不适用。表1.2-2给出了测试用例到测试需求的映射。

每个参数的解释如下：

Case：测试实例编号

Operation该参数有三个可选值。Copy表示正在测试的工具是从源设备直接复制到目标设备。Images指定被测工具将从源设备复制到镜像文件，然后将镜像文件恢复到目标设备。Image-RM表示被测试的工具应将镜像文件写入可移动介质（例如磁带）。只有在被测工具具有固有可移动介质功能的情况下，才会执行带有image-RM的测试用例。由于dd没有固有的可移动媒体功能，因此这些情况未被执行。

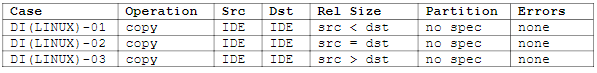
Src,Dst 指定源驱动器和目标驱动器的磁盘访问接口类型。

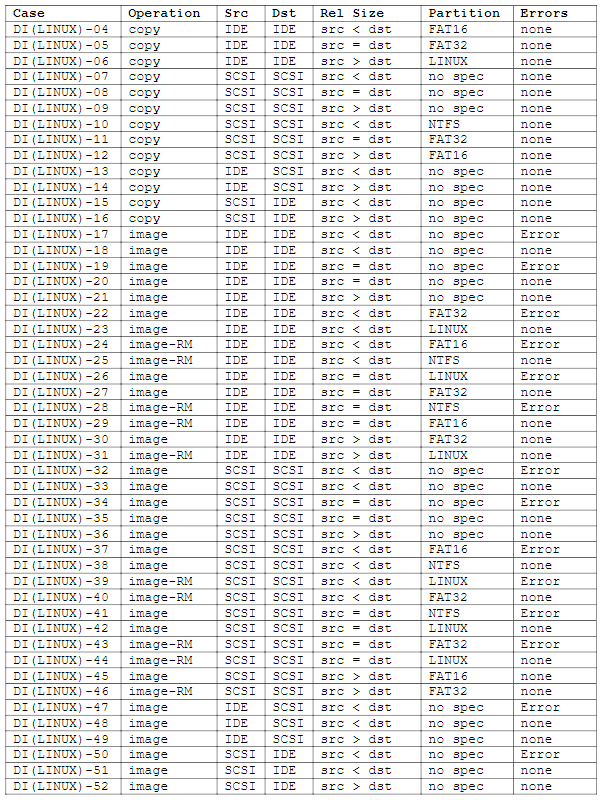
Rel Size 指定源和目标之间的大小关系。

Partition 分区类型的规范表示测试用例是分区操作。参数值no spec 表示完整的磁盘操作。

Errors Error值表示应该使用损坏的镜像文件。由于dd没有检测损坏的镜像文件的功能，因此不会执行这些情况。

表 1.2-1 测试实例参数





文档概述

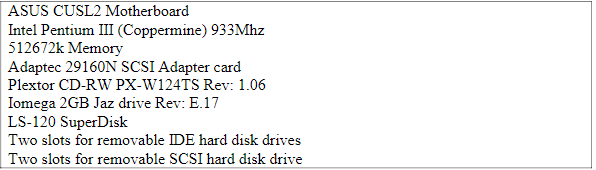
第2节介绍了测试硬件，所使用的硬盘驱动器和用于运行测试的系统配置。创建或设置源磁盘，DOS启动软盘，Linux启动驱动器和Jaz磁盘的过程在第3节中介绍。每个步骤的脚本文件在第4节中描述。第5节描述了执行过程。测试过程中遇到的技术困难在第6节中讨论。在上一节中介绍了在不同环境下执行测试的准则。

测试硬件

测试在五台主机上运行：Cadfael，Morse，Rumpole，Wimsey和JudgeDee。测试中使用了20多个硬盘（7种不同型号，5种不同品牌）（表2.2-1）。按照测试参数（表1.2-1）的要求，硬盘驱动器按照五种可能的配置之一（表2.3-1）进行测试。

2.1 主机

表2.1-1中列出的硬件组件安装在五台计算机中。



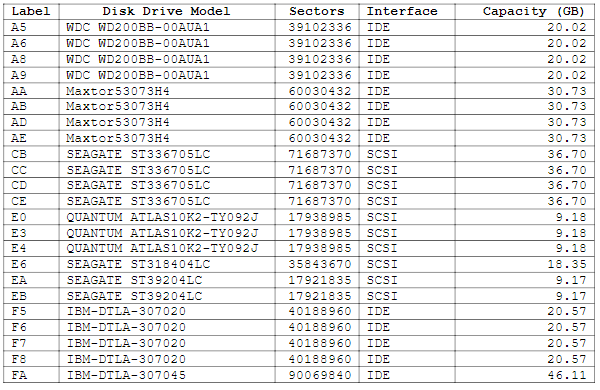
计算机Morse和Rumpole每台也有一个30GB的OnStream SC30磁带驱动器（未在测试过程中使用）。计算机JudgeDee有一个可移动IDE硬盘驱动器的第三个插槽。

2.2 硬盘驱动器

表2.2-1列出了所使用的硬盘驱动器。这些硬盘安装在CRU DataPort可移动存储模块中。根据测试的要求，可以在主机中安装零个，一个或两个IDE硬盘驱动器以及零个，一个或两个SCSI硬盘驱动器的任意组合。

IDE磁盘的跳线设置为选择电缆。根据测试用例的要求，SCSI磁盘的SCSI ID被设置为0或1。除指出的特例外，SCSI源磁盘设置为ID 0，SCSI目标磁盘设置为ID 1。

表2.2-1使用的硬盘驱动器



测试配置

硬盘驱动器设置由表1.2-1中介绍的测试案例参数根据磁盘映像工具规范版本3.1.6改编。每个测试用例，源，目标和引导/媒体都需要三个磁盘。源磁盘提供了一些要复制的内容。目标磁盘提供了放置副本的地方。引导/介质磁盘有两个功能。首先，它提供了执行dd的Linux环境。其次，它提供了一个放置需要创建镜像文件的测试用例的的地方。

决定源磁盘选择的因素是源磁盘接口和要复制的分区类型。使用测试用例所需的匹配接口和分区类型来选择磁盘。

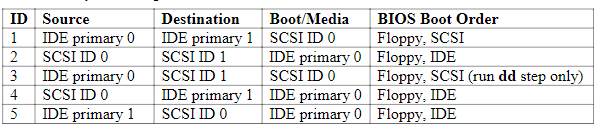
选择目标驱动器的因素是目标接口和相对大小参数。使用指定的接口选择驱动器，对于整个磁盘副本，选择相对于源的大小。对于分区副本，目标驱动器的实际大小并不重要，因为它是与目标相关的分区的大小。

选择源驱动器和目标驱动器后，从剩余的两个驱动器插槽中选择一个作为启动/介质磁盘。

表2.3-1给出了用于测试的五个系统硬盘驱动器和引导配置。Source列指示源驱动器的安装位置。只有主要的IDE通道被使用。驱动器通常定位为drive 0;但是，少数测试用例将源驱动器定位为drive 1. SCSI源驱动器设置为SCSI ID 0.Destination列指示目标驱动器的位置。Boot//Media列指示目标驱动器的位置。BIOS Boot Order指示测试所需的引导顺序的BIOS设置，以便为dd执行建立Linux环境。

对于使用系统配置3的测试用例，使用两台主机，在DOS环境中执行的测试过程的步骤是在BIOS启动顺序为Floppy IDE的主机系统上执行的。执行dd步骤在Linux环境中执行，该环境是将BIOS启动顺序为Floppy，SCSI的不同主机上的SCSI磁盘（ID 0）引导而获得的。

### 表2.3-1系统配置



### 3 媒介设置

测试用例需要创建几个媒体组件并为要执行的测试用例进行设置。测试时以下项目需要进行一次设置。

1. 源硬盘驱动器的测试用例。

2. DOS启动软盘，为测试用例设置和测量创建运行时环境。

3. 包含可启动的Linux系统的可移动硬盘。

4. 包含支持软件，控制脚本，日志文件和实用程序软件的Jaz磁盘。

3.1 源硬盘

源磁盘指的是保留的原始数字证据的硬盘驱动器。测试中使用的磁盘布局可能太多。选择了三种覆盖最常见分区类型的配置。第一种配置是双系统Red Hat Linux 7.1和Windows Me。该配置还包括FAT16，Linux EXT2，隐藏分区和删除分区（表3.1-1）。第二种配置是具有FAT32和NTFS文件系统的Windows 2000系统（表3.1-2）。第三个配置不包含有效的分区表。所有分区都使用partition magic Pro 6.0创建。

表3.1-1 Windows Me / Linux源驱动器布局

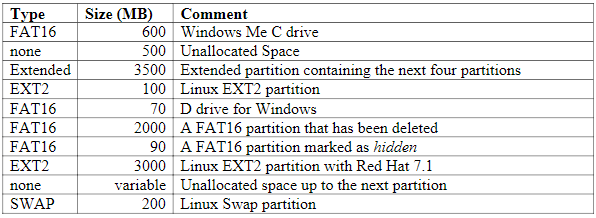
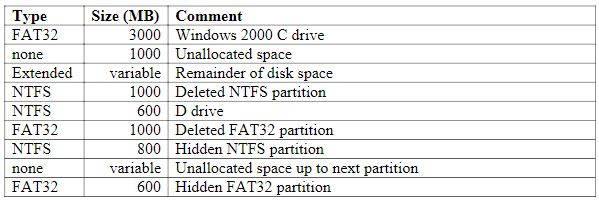


表3.1-2 Windows 2000 源驱动器布局



源磁盘的设置过程如下所示：

1. 选择安装类型：Windows Me / Linux，Windows 2000或无。磁盘E3和F5被设定为了Windows Me / Linux的设置;磁盘E4和F6被设定为Windows 2000设置;磁盘CC安装时没有分区表。

2. 选择一个硬盘

3. 选择计算机，安装驱动器，从启动软盘启动引导到PC DOS 6.3

4. 运行LOGSETUP以记录设置。

5. 运行 DISKWIPE 来初始化硬盘的内容

6. 如果安装中选择使用了操作系统，请执行步骤7-9

7. 运行 partition magic 来对硬盘进行分区。即对于Windows Me / Linux源磁盘，使用表3.1-3中的脚本，Windows 2000的源磁盘使用表3.1-4中的脚本。

8. 按照应安装的不同的操作系统的安装说明进行操作。对于Windows Me / Linux配置，首先安装Windows Me，然后安装Linux。对于Windows 2000配置，请安装Windows 2000。

9. 创建删除的文件。这是通过一个脚本（DOS批处理文件）完成的，该脚本创建一个包含已删除文件和已删除子目录（表3.1-5）的目录（X：\ UDT，其中X是一个驱动器盘符）。

10. 运行 DISKHASH 来计算源磁盘的 SHA-1哈希值作为参考。

11. （可选）如果可行，请创建一个备份到另一个可用于还原该磁盘的磁盘。如果需要恢复磁盘，则可以重新计算散列值以验证备份和恢复是否成功。

表3.1-3 Windows Me/Linux下Partition Magic脚本（FAT-SRC.TXT）

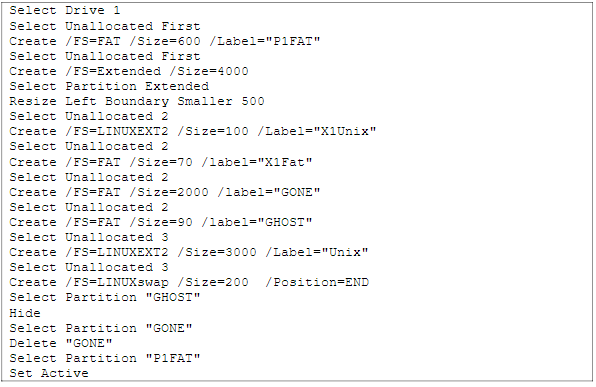


表3.1-4 Windows 2000 下Partition Magic 脚本(NT-SRC.TXT)

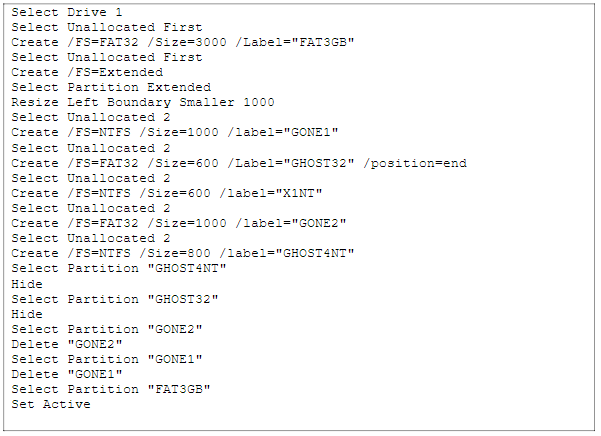
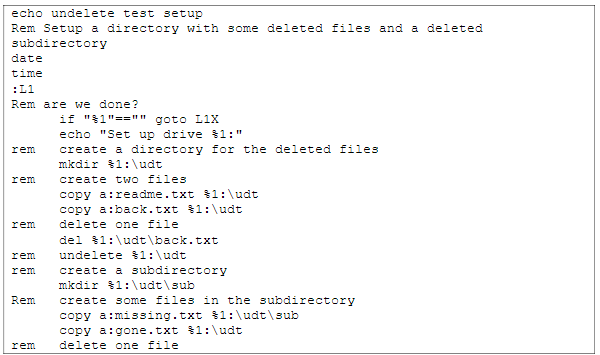
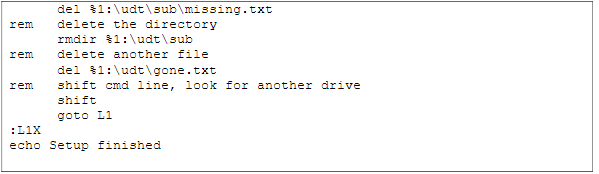


表3.1-5 创建删除文件的脚本(UDT-SET.BAT)





3.2 DOS 启动软盘

DOS软盘为支持软件提供执行环境。表3.2-1给出了设置DOS软盘的命令。DOS引导软盘用于源驱动器设置，目标驱动器设置和测量运行结果。

表3.2-1 DOS启动软盘设置步骤

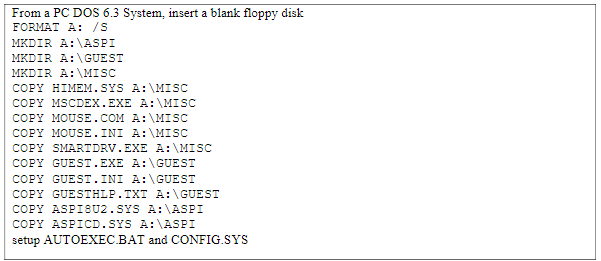


表3.2-2中提供的AUTOEXEC.BAT文件是基于SafeBack 2.0手册中建议的典型取证引导软盘的简化版本。

表3.2-2 DOS AUTOEXEC.BAT

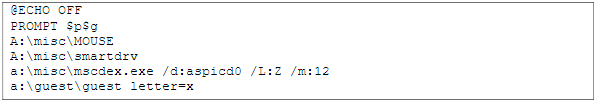
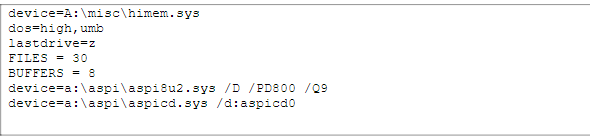


表3.2-3 DOS CONFIG.SYS



Jaz Disk

Jaz磁盘提供许多功能。测试用例脚本，支持程序和实体程序（例如partition magic）位于Jaz磁盘上。每个测试用例的日志文件都保存在Jaz磁盘上。该磁盘由表3.3-1中的脚本设置。标记为FA的磁盘不用于任何测试用例，而是用作执行测试用例所需文件的存储库。表3.3-1中提到的Drive％2是FA。驱动器％1是Jaz磁盘。每个主机都设置了一个Jaz磁盘。

Jaz磁盘安装有几个子目录。表3.3-2描述了每个子目录的内容。除了运行测试所需的工具外，Jaz磁盘上还放置了一些附加的实体程序。

表3.3-1 Jaz 磁盘配置（SETUPJAZ.BAT）

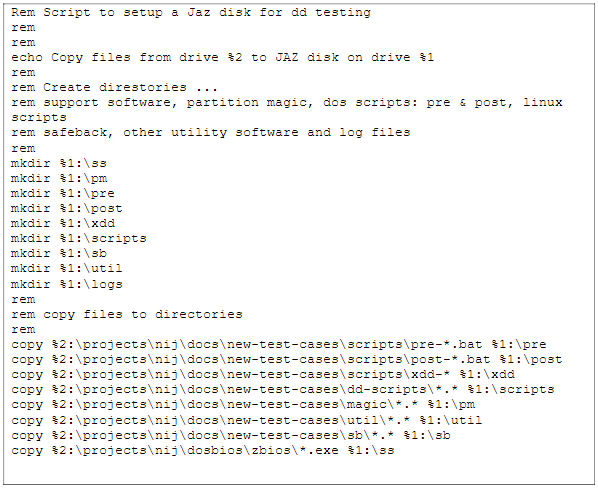
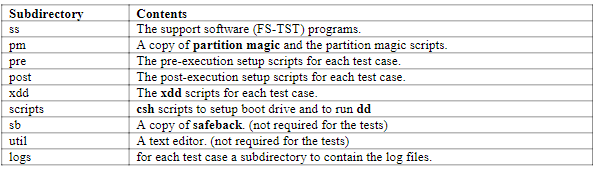


表3.3-2 Jaz磁盘内容



3.4 Linux 启动盘

Linux硬盘驱动器用于为dd提供执行环境，并为需要映像文件的测试用例提供存储映像文件的位置。以下过程用于设置Linux启动盘。

1. 运行LOGSETUP以记录设置。

2. 运行DISKWIPE以确保磁盘没有残留信息。

3. 加载Linux 系统(Red Hat 7.1)

4. 使用安装脚本挂载Jaz盘。

5. 从 Jaz磁盘中运行 make-test-account

make-test-account（表3.4-1）脚本是一个csh（C shell）脚本，它为给定的测试用例创建一个执行dd的帐户。必须将第3.3节中创建的Jaz磁盘挂载为/ x。该帐户由第8行（useradd程序）创建。该帐户是使用root（用户标识为0）特权创建的。第10行和第11行为该帐户创建登录脚本（.login）。第12行复制登录脚本用于定位和安装Jaz驱动器以进行测试运行的两个脚本。

表3.4-1 创建测试账户的脚本:make-test-account(make-test-account)

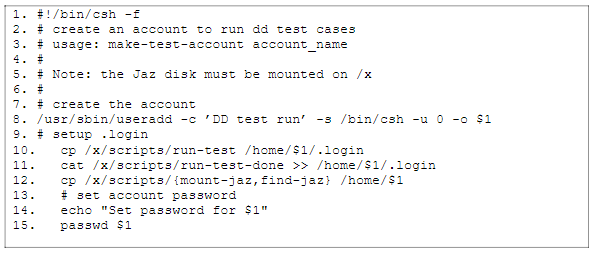
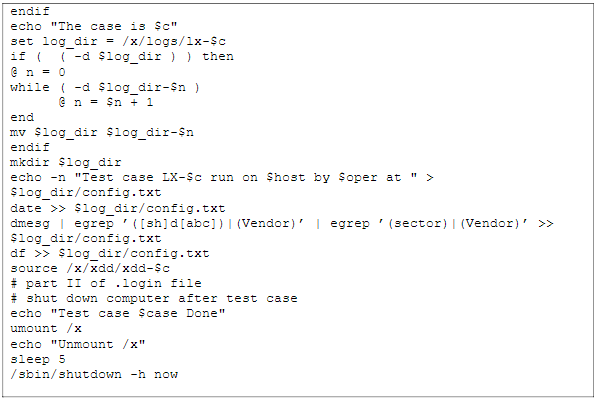
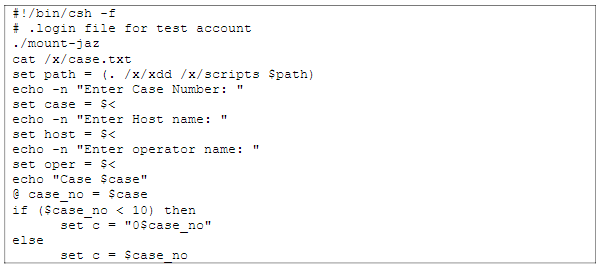


表3.4-2 测试账户登录脚本(.login)



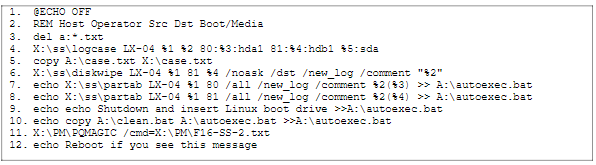
4 测试执行脚本

测试用例按照三个步骤的顺序执行。第一步（安装程序）从DOS环境设置目标驱动器并记录源磁盘和目标磁盘的分区表。第二步（执行）是从Linux环境执行dd。在DOS环境中执行的最后一步（度量）将源与扇区逐个进行比较，并使用SHA-1哈希来检查源的任何更改。除了创建分区和格式化之外，设置每个测试用例和测量结果所需的程序都包含在FS-TST包中。对于每个测试用例，Shell脚本和DOS批处理文件都存储在Jaz磁盘上。

4.1 预执行设置

目标驱动器设置脚本存储在目录X：\ PRE中的Jaz驱动器中。文本案例XX的脚本名为X：\ PRE \ PRE-XX.BAT。表4.1-1是在目标上创建FAT16分区的预执行脚本示例。

表4.1-1 示例预执行脚本(PRE-04.BAT)



该脚本可以通过下列命令运行

x：\ pre \ pre-04 Cadfael JRL F5 A6 CD

所有预执行脚本都需要五个命令行参数：主机名，操作员，源磁盘驱动器十六进制标签，目标磁盘驱动器十六进制标签和Linux启动驱动器十六进制标签。脚本逐行解释如下：

1.关闭回显，以便命令不会回显到显示屏幕。

2.记录命令行参数的注释。Host（在脚本中称为％1）是主机的名称。Operator（％2）标识运行测试的工作人员。Src，Dst和Boot / Media（％3，％4和％5）是分配给每个硬盘驱动器的外部两个十六进制数字标签。

3.删除所有文本文件。所有日志文件都创建为文本文件。在运行测试用例所需的所有程序之后，最后一步是将所有文本文件复制到为给定测试用例创建的目录中。删除文本文件会从先前的测试用例中删除日志文件。

4. LOGCASE程序记录测试用例的驱动器分配。

5.在Jaz驱动器上放置CASE.TXT文件的副本允许下一步（在Linux环境中）自动确定正在运行的测试用例。

6. DISKWIPE向目标磁盘的每个扇区写入已知的唯一内容。

7.修改AUTOEXEC.BAT，以便下次启动系统时记录源磁盘的分区表。

8.修改AUTOEXEC.BAT，以便下次系统启动时记录目标磁盘的分区表。

9.修改AUTOEXEC.BAT，以便下次系统启动并记录源和目标的分区表时，会打印一条消息，指示操作员关闭并继续下一步。

10.修改AUTOEXEC.BAT，以便下次系统启动时，AUTOEXEC.BAT文件恢复为干净版本。11.使用partition magic在测试用例所需的目标磁盘上创建分区。不需要目标分区的测试用例省略此步骤。

12.partition magic通常会在创建分区后自动重启计算机，但并非总是如此。此消息用于提醒操作员重新启动，以便在计算机未自动重新启动时记录分区表。

4.2 运行 dd

dd执行步骤是通过登录到Linux测试帐户自动调用的。登录脚本（表3.4-2）在设置步骤中从复制到Jaz磁盘的CASE.TXT文件中打印测试用例编号。提示操作员输入测试用例编号，操作员标识和主机名，然后从与测试用例对应的脚本执行dd。对于测试用例xx，脚本名称为/x/xdd/xdd-xx。执行完成后，计算机会自动关闭。

dd的实际执行是通过几层csh脚本完成的。首先是xdd脚本。它是一个一行脚本，它设置四个参数并通过run\_dd调用下一个级别的脚本。例如，run\_dd hda hdb copy LX-01是测试用例01的xdd脚本。这四个参数如下所示：

1.源设备是第一个参数。在这个例子中，完整的设备名称是/ dev / hda。一些可能的值是第一个IDE通道上磁盘0的hda，SCSI ID为0的磁盘的sda。

2.第二个参数是目标设备的名称，在实例中为/ dev / hdb。

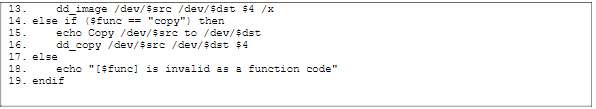
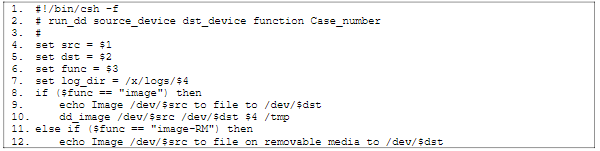
3.第三个字段应该是测试用例要求的复制或镜像。

4.最后一个字段是测试用例ID。

run\_dd脚本（表4.2-1）选择操作类型（copy或imgae）并调用下一层。如果第三个参数是copy，则调用脚本dd\_copy（表4.2-2）直接将源复制到目标。如果第三个参数是image，则调用脚本dd\_image（表4.2-3），以便首先将源复制到镜像文件，然后将镜像文件复制到目标。

dd\_image脚本调用dk-backup（表4.2-4）将源复制到映像文件和dk-restore（表4.2-5）以将映像文件复制到目标。脚本dk-backup创建一组源文件的压缩镜像文件。每个镜像文件通常包含最多1,000,000个扇区。 dk-backup和dk-restore输出状态信息，以便操作员跟踪测试运行进度。

表4.2-1 选择操作类型的脚本（run\_dd）



逐行解释run\_dd的步骤如下：

1.脚本在csh环境中执行，跳过.cshrc。

2.描述命令行的注释。

3.注释

4.将一个变量src设置为第一个命令行参数（源设备名称）。

5.将变量dst设置为第二个命令行参数（目标设备名称）。

6.将一个变量func设置为第三个命令行参数（操作：copy或image）。

7.设置一个变量log\_dir，以在Jaz驱动器上为测试用例命名的目录中查找任何日志文件。

8.测试操作类型

9.向操作员提供反馈。

10.调用dd\_image脚本来创建源镜像并将镜像恢复到目标位置。

11.测试可移动介质测试。但是，可移动媒体测试未使用。

12.向操作员提供反馈。

13.调用dd\_image脚本在可移动媒体上制作镜像。

14.测试复制操作。

15.向操作员提供反馈。

16.调用dd\_copy将源复制到目标。否则表示该功能不正确。

18.告诉操作员脚本运行有问题。

19.完成。

表4.2-2 调用dd进行直接复制的脚本(dd\_copy)



dd\_copy的逐行解释如下：

1.脚本在csh环境中执行，跳过.cshrc。

2.注释

3.描述命令行的注释。

4.注释。

5.注释。

6.将一个变量src设置为第一个命令行参数（源设备名称）。

7.将变量dst设置为第二个命令行参数（目标设备名称）。

8.设置一个变量log\_dir，以在Jaz驱动器上为测试用例命名的目录中查找任何日志文件。

9.记录开始时间。

10.记录日期。

11.注释。

12.将一个变量cmd设置为dd命令来执行。

13.将dd命令记录到文件中。

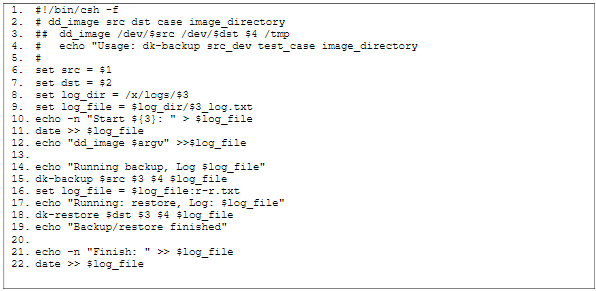
14.执行dd命令（在步骤12中创建）。

15.注释。

16.记录完成时间。

17.记录日期。

表4.2-3 脚本使用dd通过镜像文件进行复制(dd\_image)



dd\_image的逐行解释如下：

1.脚本在csh环境中执行，跳过.cshrc。

2.注释。

3.描述命令行的注释。

4.注释。

5.注释。

6.将一个变量src设置为第一个命令行参数（源设备名称）。

7.将变量dst设置为第二个命令行参数（目标设备名称）。

8.设置一个变量log\_dir，以在Jaz驱动器上为测试用例命名的目录中查找任何日志文件。

9.设置备份的日志文件的名称。

10.记录开始时间。

11.记录日期。

12.在日志文件中记录命令。

13.注释。

14.向操作者提供反馈意见。

15.创建映像文。

16.设置还原的日志文件。

17.向操作者提供反馈。

18.从镜像恢复到目的地。

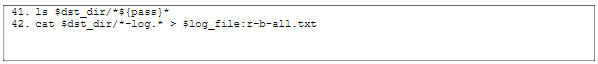
19.向操作者提供反馈。

20.注释

21.记录完成时间。

22.记录日期。

表4.2-4 使用dd创建镜像文件的脚本(dk\_backup)



dk\_backup的逐行解释如下：

1.脚本在csh环境中执行，跳过.cshrc。

2.测试正确的参数数量

3.描述命令行的操作员反馈。

4.退出。

5. if结束。

6.声明注释。

7.将变量src设置为第一个命令行参数（源设备名称）。

8.将变量dst\_dir设置为第三个命令行参数（映像文件目录名称）。

9.将一个变量log\_file设置为日志文件名。

10.记录操作。

11.向操作者提供反馈。

12.注释。

13.为镜像文件创建目录。

14.无用代码。可以删除。

15.设置镜像文件的最大大小（1,000,000个扇区）。

16.将n设置为max.

17.设置第一遍跳过的扇区数。

18.开始编号为100，强制设置趟数值为三位数。

19.无用代码。可以删除。

20.无用代码。可以删除。

21.more表示是否有更多来源获得，是（1）或否（0）。

22.只要有更多的来源可以循环。

23.设置当前的遍数。

24.向操作员提供反馈。

25.设置镜像文件的完整路径。

26.记录日期和时间。

27.创建执行dd的命令。镜像使用gzip压缩以节省空间。

28.记录命令。

29.执行dd命令。

30.记录趟数。

31.向操作者提供反馈。

32.为此趟添加日志。

33.将此传递的日志添加到调试文件。

34.死码。可以删除。

35.记录日期。

36.调整下一次跳过的扇区数。

37.测试不再有来源的部门。

38.如果源完全镜像完成，则将more设置为零。

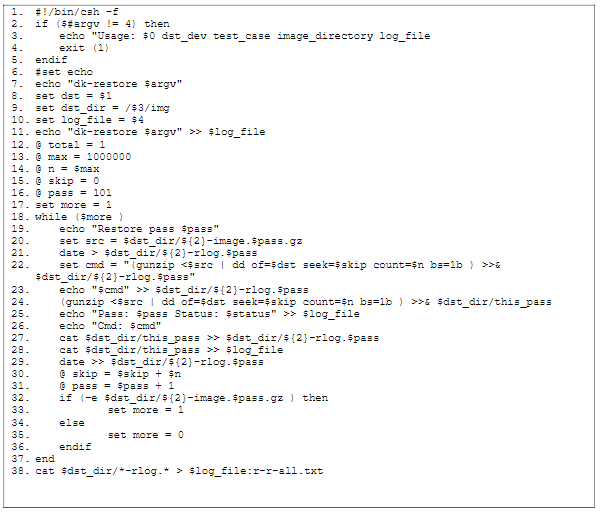
39. while循环结束。

40.向操作者提供反馈

41.列出创建的镜像文件。

42.完成备份日志文件。

表4.2-5 脚本使用dd恢复镜像文件



dk\_restore的逐行解释如下：

1.脚本在csh环境中执行，跳过.cshrc。

2.测试正确的参数数量。

3.描述命令行的操作员反馈。

4.退出。

5. if结束。

6.注释声明。

7.给操作员提供反馈。

8.将变量dst设置为目标设备名称。

9.将变量dst\_dir设置为包含镜像文件的目录的名称。

10.设置日志文件的名称。

11.记录行动。

12.无用代码。可以删除。

13.设置镜像文件的最大大小（1,000,000个扇区）。

14.将n设置为max。

15.设置第一遍跳过的扇区数。

16.在101开始编号以强制遍数值为三位数。

17.more表示是否有更多来源获得是（1）或否（0）。

18.只要有更多的镜像文件循环。

19.向操作者提供反馈。

20.设置镜像文件的完整路径。

21.记录日期和时间。

22.创建执行dd的命令。镜像使用gzip压缩以节省空间。

23.记录命令。

24.执行dd命令。

25.记录遍数。

26.向操作者提供反馈。

27.为此趟添加日志。

28.将此趟的日志添加到调试文件。

29.记录日期。

30.调整下一次跳过的扇区数。

31.调整通过号码。

32.测试更多镜像文件。

33.如果有更多镜像文件存在，将其设置为1。

34. else。

35。如果没有更多的镜像文件，将其设置为0。

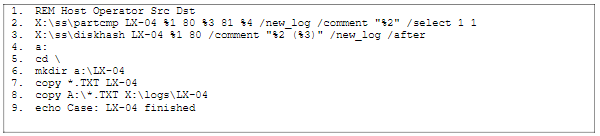
36.if结束。

37.合并日志文件。

4.3 执行后评估

脚本存储在Jaz驱动器的目录X：\ POST中。 案例XX的脚本名为X：\ POST \ POST-XX.BAT。表4.3-1是一个后处理脚本的例子。执行后脚本有三个功能，将源与目标进行比较，计算源的SHA-1散列并将所有日志文件保存到目录。

表4.3-1 示例后执行评估脚本（POST-04.BAT）



该脚本由以下命令执行：

x:\post\post-04 Cadfael JRL F5 A6

所有执行后脚本都需要四个命令行参数：主机名，操作员，源磁盘驱动器十六进制标签和目标磁盘驱动器十六进制标签。脚本逐行解释如下：

1.注释以解释命令行参数。首先是主机名称（％1），然后是操作员标识（％2），并为源（％3）和目标（％4）驱动器保留外部十六进制标签。

2.将来源与目的地进行比较。如果测试用例是分区副本，则执行PARTCMP。如果测试用例是磁盘拷贝，则执行DISKCMP。

3.计算源磁盘的SHA-1哈希。

4.使软盘成为当前驱动器。

5.使根目录成为当前目录。

6.为测试用例日志文件创建一个目录。

7.将日志文件从软盘复制到测试用例目录。

8.将日志文件从软盘复制到Jaz驱动器

9.通知操作员测试案例已完成。测量脚本完成后，应将日志文件复制到永久位置。

评估脚本完成后，应将日志文件复制到永久位置。

5.测试用例执行

本节介绍运行测试用例的过程。已假设读者熟悉电脑的基本操作。

5.1 执行过程

执行dd测试用例的过程如下所示：

1.选择要运行的测试用例。

2.收集可移动媒体：DOS启动软盘，Jaz磁盘。

3.根据测试用例参数选择源磁盘。源接口参数确定磁盘是IDE还是SCSI。如果指定了分区类型，则选择具有指定设置的源磁盘。

4.根据测试用例参数为测试选择一个目标磁盘。目标接口确定磁盘是IDE还是SCSI。相对大小参数确定在整个磁盘上运行的测试用例的可接受选择。对于分区操作，可以使用任何大小的磁盘。

5.根据源接口和目标接口测试用例参数选择系统配置（表2.3-1系统配置）。

6.选择一台主机来运行测试。确保BIOS引导顺序按照所选系统配置的要求进行设置。

7.根据所选系统配置选择引导/介质磁盘。

8.确保主机关闭。安装DOS启动盘，Jaz磁盘，源磁盘和目标磁盘。不要安装引导/媒体硬盘驱动器。

9.打开主机从取证DOS软盘启动。

10.运行预执行脚本（第4.1节）以设置目标硬盘驱动器并记录源驱动器和目标驱动器的分区表。脚本的完整路径名是X：\ PRE \ PRE-xx，其中xx是两位数测试用例编号。该脚本位于Jaz磁盘上（DOS驱动器X :)。如果在目标上创建分区，主机可能会重新启动。如果主机没有自动重启，那么应该手动重启主机（记录分区表）。此步骤有问题需要进行程序修改，详情请参阅第6节。

11.关闭系统并取出DOS启动软盘。

12.使用Linux启动/执行环境安装启动/介质硬盘驱动器，将BIOS（如果需要）设置为从该磁盘启动。打开主机并启动到Linux。

13.登录到测试运行帐户。登录脚本提示输入测试用例编号，然后自动运行测试用例的脚本。计算机将在案件结束时关闭。

14.移除Linux引导/媒体硬盘。

15.插入DOS启动软盘。

16.打开主机从启动软盘启动到DOS。

17.运行执行后脚本（第4.3节）以评估结果。该脚本将源与目标进行比较并计算源磁盘的SHA-1。脚本的完整路径名是X：\ POST \ POST-xx，其中xx是两位数测试用例编号。该脚本位于Jaz磁盘上（DOS驱动器X :)。

18.评估脚本完成后，应将日志文件复制到永久位置。

有关BIOS引导顺序的注意事项：支持软件（pre-xx和post-xx）要求IDE驱动器的驱动器数量低于任何已安装的SCSI驱动器。这可以通过确保在启动顺序中的SCSI驱动器之前确认IDE驱动器来完成。因此，如果IDE和SCSI驱动器都存在，但是需要从SCSI磁盘引导dd（Linux环境）（即，SCSI第一引导顺序），则必须以IDE第一引导顺序运行pre-xx步骤。

5.2 非FAT分区的特殊过程

用于确定工具是否修改了目标大于源的分区副本的任何超出扇区的过程取决于所复制的分区的类型。如果复制了FAT分区，则以下过程足以确定工具是否更改了任何超出的扇区：

1.使用diskwipe程序初始化目标磁盘。

2.创建目标分区。

3.运行正在测试的工具。

4.使用partcmp程序将源分区与目标分区进行比较。

创建一个FAT分区用文件分配表覆盖分区开始附近的一些扇区，但是大部分扇区都保留在分区中，即分区的多余扇区保留了diskwipe产生的原始内容。partcmp计划将检查目的地的多余部分，并将每个部分分配到几个可能类别中的一个。partcmp程序然后将每个类别中的扇区数写入日志文件。由工具更改的任何部分都计入除目的地填写类别以外的类别中。如果该工具没有更改任何超额部分，则将所有超出部分分配到目标填充类别。

此过程可能不足以确定非FAT分区类型的多余扇区是否由工具保持不变。如果创建分区修改了分区的任何超出扇区，则需要额外的步骤来确定工具是否更改了目标分区的任何超出扇区。以下过程可用于确定工具是否修改了任何超出扇区：

1.使用diskwipe程序初始化目标磁盘。

2.创建目标分区。

3.用sechash程序计算多余扇区上的SHA-1散列值。

4.运行正在测试的工具。

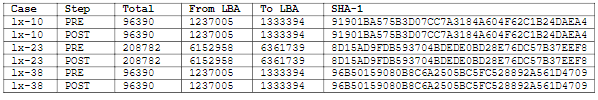
5.计算多余扇区上的第二个SHA-1散列值。

6.使用partcmp程序将源分区与目标分区进行比较。

如果该工具更改了任何多余的扇区，则两个SHA-1值将有所不同。如果两个SHA-1值相同，则该工具没有更改任何超出扇区。

测试用例10和38涉及NTFS分区;测试用例23涉及Linux（EXT2）分区。第二种过程适用于测试案例10,23和38.在所有三种情况下，超额部分的散列在运行dd之前和之后都是相同的。结果列于表5.2-1。标记为Case的列是测试用例。步骤栏指示散列值是在运行该工具之前（PRE）还是之后（POST）计算的。总列表示扇区总数。From LBA和To LBA列表示超出扇区的起始和结束LBA地址。SHA-1的值显示在SHA-1列中。sechash的日志文件名为PRELOG.TXT和POSTLOG.TXT。

表5.2-1运行dd之前和之后的过度分区扇区的SHA-1值



5.3 检查日志文件的指导

测试用例完成后，结果将包含在一组名为LX-xx（xx是测试用例编号）的日志文件中。在设置和评估步骤中执行的每个支持程序都会生成一个可以检查的日志文件。对于源驱动器安装日志文件，有一个名为setup的目录，其中包含每个源磁盘的子目录。在每个子目录中都有来自相应源磁盘设置的日志文件。有执行logsetup，diskwipe和diskhash的日志文件。

本节的其余部分讨论由dd测试中使用的每个支持程序生成的日志文件的相关内容。

5.3.1 LOGSETUP: 设置源驱动器

关于源磁盘驱动器设置的管理细节记录在logsetup的日志文件SETUP.TXT中。记录磁盘驱动器标签，使用的主机，操作员，装载的操作系统（如果有）和日期。

5.3.2 LOGCASE：启动一个测试用例

关于测试用例执行的管理细节从日志文件记录在日志文件CASE.TXT中。记录所用磁盘驱动器的标签，分配给每个磁盘的角色，每个磁盘的BIOS驱动器号，使用的主机，操作员和日期。

5.3.3 PARTAB：文档分区表

partart程序记录源和目标磁盘驱动器的分区表。源磁盘的日志文件应显示驱动器具有第3.1节中的三个初始设置之一。目标驱动器的日志文件应该显示，对于整个磁盘驱动器上的操作，不存在分区表，但对于分区上的操作，目标驱动器上的测试用例需要该类型的分区。

5.3.4 DISKCMP和PARTCMP：检查重复的准确性

比较程序diskcmp和partcmp有两个功能：计算源到目的地的复制的精确度，以及比源更大的目的地的测量精度，并确定dd是否改变了任何超出的扇区。为了计算重复的准确性，日志文件中的两个值是相关的。标记为“扇区比较”的值表示检查的扇区数，标记扇区不同的值表示扇区数不符合预期。如果有少量扇区不匹配，则不匹配扇区的LBA地址将报告在Diffs范围下。seccmp程序可以详细检查不匹配的扇区。对于diskcmp，LBA地址相对于磁盘的开始位置，对于partcmp，LBA地址是相对于分区的开始位置的。

为了确定dd是否改变了多余部分的内容，比较程序将目的地的多余部分分类。在FAT分区的情况下，评估超额目标扇区的分类很简单，但是对于NTFS和Linux EXT2分区有一个复杂的情况。在FAT分区的情况下，所有多余的扇区应该归类为目标填充。目标扇区的数量是标记为较少扇区的值。该值应该与标记为Dst字节填充的值匹配。在NTFS或Linux EXT2分区的情况下，一些超出扇区的内容不是目的地填充，并且需要使用不同的过程（请参阅第5.2节）来评估dd是否更改了超出的目标扇区的内容。该过程是在执行dd之前识别多余的扇区。在执行dd之前和之后，用sechash程序计算多余扇区的SHA-1散列。如果散列值匹配，那么对于超出的扇区没有改变。

5.3.5 SECCMP：调查异常

seccmp程序不是通常测试程序的一部分。但是，它在某些测试用例（06和35）中使用，其中diskcmp或partcmp指示源或目标的最后一个扇区与复制操作中的对应扇区不匹配。seccmp程序显示目标上副本的最后一个扇区保留了目标设置中由diskwipe写入的原始值。这意味着dddid不会将最后一个源扇区复制到目标。

5.3.6 DISKHASH：验证没有更改到源

diskhash程序用于验证源磁盘未被dd更改。diskhash日志文件包含一个SHA-1散列值。验证是通过将测试用例日志文件HASHALOG.TXT的散列值与源磁盘设置HASHBLOG.TXT的散列值进行比较来完成的。如果该值一致，则dd没有更改源磁盘。

5.3.7 SECHASH：验证目标部分没有更改

sechash程序用于三个测试用例（06,23和38），以验证dd没有对目标分区的多余扇区进行任何更改。日志文件PRELOG.TXT在执行dd之前包含多余扇区的SHA-1散列，日志文件POSTLOG.TXT在执行dd之后包含多余扇区的SHA-1散列。

5.4 结果评估程序

在运行测试用例之后，必须检查结果以确定结果是否应该被接受，或者是否需要一些进一步的操作来完成测试用例。结果评估必须考虑明显的结果是否准确反映了被测工具。必须检查成功或不成功的测试结果，以确保没有发生错误。

第一个问题是如果测试看起来是成功的，那么我们应该接受该工具为特定测试用例产生预期结果的结果。有几种方法可以使测试看起来能够产生预期的结果，而实际上却没有这样做。这通常涉及整个步骤没有运行，并且评估留在最初成功测试的最终状态的磁盘。这可以通过始终确保目标磁盘在每次测试开始时已被擦除来缓解。diskwipe日志文件应该显示为给定的目标磁盘擦除了正确数量的扇区。

第二个问题是如果测试产生异常并且看起来失败，dd失败或者是其他错误。需审查每个异常测试运行以表征异常，然后选择一个操作过程。

1.如果可以找到硬件或程序问题，例如磁盘驱动器发生故障或测试配置不正确，则重新运行测试并进行适当的调整。

2.如果没有发现硬件或程序问题，并且异常与已知异常相匹配，则将异常视为真实。

3.如果异常是唯一的，则推迟一个决定，直到运行更多的测试用例。这些测试用例被称为延迟，直到更多。

4.如果异常匹配延迟中的异常直到更多的类别，则检查两个结果的共同因素。根据审稿人的判断，可能会建立一个新的已知异常情况，否则测试案例会一直推迟到更多时间。

在所有测试案例运行之后，任何剩余的测试案例将被推迟到更多类别，必须通过接受异常为真实或根据需要进行额外调查重新分类来解决。

6 技术难点

在测试过程中遇到了两个问题。第一个问题是，在某些情况下，在一个硬盘驱动器上创建分区伴随着对另一个硬盘驱动器的更改。第二个问题是Jaz磁盘的性能不可靠。这两个问题都没有对测试结果产生任何影响。

在目标驱动器上创建分区的原始过程是将源驱动器和目标驱动器都安装到主机中，启动到DOS中，擦抹目标并创建分区。此过程对于创建FAT16和Linux EXT2分区没有任何问题。但是，在创建FAT32或NTFS分区时，将会对源驱动器进行测试更改。该过程被修改为直到在目标驱动器上创建分区之后才安装源驱动器。

Jaz磁盘有时会变得不可读。Jaz磁盘对于几个测试用例可以很好地工作，但会变得不可读。尝试从磁盘读取将产生Jaz磁盘未格式化的消息。Jaz磁盘然后被重新格式化并重新加载。经与供应商讨论后，Jaz驱动在Cadfaeland rumpole上被替换。这显着改善了这个问题，但并没有完全消除这个问题。

7 适应不同的测试环境

测试在NIST的CFTT实验室进行。试图在另一个实验室重现测试结果可能需要对测试脚本和程序进行重大调整。测试用例也可用于调查其他问题。本节提供了适用于其他实验室环境的测试用例的支持组件的指导原则。将对硬件，源硬盘驱动器，执行环境和测试脚本依次进行讨论。

7.1硬件

可用的硬件决定组织测试过程的策略。至少需要六个磁盘驱动器。三个应该是IDE驱动器，这样两个驱动器的大小不同，第三个驱动器的大小与另外两个驱动器的大小相同。其他三个磁盘驱动器应该是具有相同大小关系的SCSI驱动器。另外，其中一个SCSI驱动器应该至少大于一个IDE驱动器，并且其中一个IDE驱动器应该至少大于一个SCSI驱动器。驱动器可以安装在一台计算机上（不可移动），也可以像安装在NIST的CFTT实验室中，每台硬盘驱动器都可以从一台计算机上卸下并放入另一台计算机中。

独立复制由NIST以外的机构或实验室完成，在他们自己的实验室环境中重复每个测试用例。由于NIST所使用的确切硬件不太可能存在，因此必须进行调整和替换才能运行测试用例。例如，NIST环境使用Iomega Jaz驱动器来包含作为驱动器X访问的支持软件。另一个没有Jaz驱动器的实验室可能会将支持软件放在软盘，LS-110驱动器（SuperDisk）或CD-ROM上。更重要的是测试中使用的实际硬盘驱动器。并不总是清楚所用硬盘的等效替代品是什么。

这里有一个关于复制的重要问题。发现的一个异常（从具有奇数个扇区的源中遗漏最后一个扇区）不取决于任何实际测试参数。但是，异常的复制取决于选择具有某些特征的驱动器。这个问题没有简单的答案。特别的答案是要求任何替代符合测试运行后确定的任何复制结果所需的条件。在测试过程中学到的信息应该适用于任何尝试的复制。在这种情况下，我们已经了解到源的奇怪性或均匀性是在测试过程中发现的隐藏测试参数。任何复制测试的尝试都必须使用这个隐藏参数。这意味着磁盘或分区替换的规则。如果确切的磁盘驱动器或分区不可用，那么替换必须是相同的接口类型（例如，IDE或SCSI），并且必须具有奇数个扇区如NIST进行的测试中使用的对象具有奇数个扇区。

7.2 源硬盘

可用的硬盘越多，提前组织和设置源硬盘就越容易。如果只有少量驱动器可用，则可能需要重新设置源驱动器，因为驱动器将转移到不同的角色中。这可能需要很长时间。这可以通过仔细选择测试用例顺序来缓解，以便一旦设置了源驱动器，就可以在将驱动器分配给其他角色之前运行所有需要该驱动器的测试用例。通过检查表1.2-1中的测试用例参数可以方便地确定给定测试所需的硬盘接口类型。

7.3 运行环境

有两种执行环境用于测试用例。测试设置和结果测量的支持软件在DOS环境下运行。dd程序在Linux环境中执行。

DOS环境是从DOS启动盘建立的。引导磁盘应该与第3.2节中描述的类似，除了反映实际硬件存在的更改。例如，如果不存在Jaz驱动器，则可以删除guest目录中的所有文件和autoexec.bat文件中对guest的引用。

Linux环境有更多的改变可能性。需要考虑的备选方案列表如下：

•如果Linux内核支持大型（大于2GB）文件，则可以修改dk\_backup和dk-restore以使用单个映像文件，而不是将映像文件拆分为多个小文件的当前版本。但是，将镜像文件分解为更小的块的一个优点是可以监视测试进度。

•当前的Linux环境从硬盘启动。环境可以从可启动的CD或其他媒体获得。

•这些测试是为Linux环境编写的，但可以适用于其他类似UNIX的操作系统，例如FreeBSD，只需稍作更改。例如，在FreeBSD中，磁盘驱动器设备名称将需要更改。

7.4 测试脚本

对前期和后期脚本进行一定修改可以改进测试过程。

•使程序和脚本的位置成为一个变量，该变量可以从配置文件中获得，也可以在计算机启动时设置。这简化了重定位文件或指定替代脚本。

•将测试用例使用的任何脚本复制到日志目录。如果对测试用例进行了修改，这简化了测试用例中实际使用的脚本的标识。

•对脚本前和脚本进行任何修改都需要确保驱动器分配保持正确。

•对每个测试用例都有一个独特的前置或后置脚本的替代方案是使用一个通用脚本，该脚本需要额外的参数来指定驱动器号分配，Linux设备名称和partition magic脚本。

•前脚本应该重新设计。而不是修改autoexec.bat来设置打印分区表，将脚本分成两部分。第一部分应安装目标驱动器，而不安装源驱动器。然后可以关闭系统，安装源驱动器并打印两个驱动器的分区表。在目标上创建分区时安装源驱动器和目标驱动器会导致第6节中讨论的问题。

•使用脚本设置源驱动器。