



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Patrícia Jamile de Oliveira Martins

Framework de Diagnóstico de Discos Rígidos em Sistemas Linux.

FORTALEZA – CEARÁ
NOVEMBRO 2011

Autor:

Patrícia Jamile de Oliveira Martins

Orientador:

Prof. Msc. Ricardo Jardel Nunes da Silveira

Framework de Diagnóstico de Discos Rígidos em Sistemas Linux.

Monografia de Conclusão de Curso
apresentada à Coordenação do Curso
de Pós-Graduação em Engenharia
de Teleinformática da Universidade
Federal do Ceará como parte dos
requisitos para obtenção do grau de
Engenheira de Teleinformática.

FORTALEZA – CEARÁ

NOVEMBRO 2011

PATRÍCIA JAMILE DE OLIVEIRA MARTINS

Framework de Diagnóstico de Discos Rígidos em Sistemas Linux.

Esta Monografia foi julgada adequada para a obtenção do diploma de Engenharia do Curso de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Patrícia Jamile de Oliveira Martins

Banca Examinadora:

Prof. Msc. Ricardo Jardel Nunes da Silveira
Orientador

Prof.

Prof.

Fortaleza, 27 de novembro de 2011

Resumo

Discos rígidos são elementos fundamentais em qualquer sistema computacional moderno e é importante monitorar a saúde destes dispositivos, pois são utilizados desde computadores pessoais a aplicações de risco. Neste trabalho se objetiva a construir um *framework* em sistema Linux, utilizando comandos dos padrões *Advanced Technology Attachment* (ATA) e *Small Computer System Interface* (SCSI), para dar suporte ao desenvolvimento de algoritmos de teste de discos rígidos. Para demonstrar o uso do *framework*, são implementados alguns algoritmos de teste rápido, que são comparados com ferramentas de diagnóstico do mercado. Por último estes algoritmos são embarcados em uma versão inicializável do Linux, o que permite realizar testes em quais quer computadores que suportem inicialização por *pendrive* ou *live cd*, independentemente do sistema operacional instalado. Os resultados dos testes dos algoritmos se mostram bastante satisfatórios, inclusive melhores em alguns algoritmos, demonstrando a eficácia do *framework*.

Para o desenvolvimento do trabalho todos os comandos foram desenvolvidos em linguagem C++ ANSI. Vários comandos ATA e SCSI foram implementados para dar suporte à leitura, auto-teste e outras funcionalidades e a partir deles os algoritmos são implementados.

Palavras-chaves: Framework de Diagnóstico, Discos Rígidos, HDD, SSD, SCSI, ATA.

Abstract

Hard disks are key elements in any modern computer system and is important to monitor the health of these devices because they are used for applications from personal computers at risk. This work aims to build a framework in Linux system using the standard commands ATA and SCSI to support the development of algorithms for test drives. To demonstrate the use of the framework, algorithms are implemented some test that is compared with diagnostic tools on the market. Finally these algorithms are embedded in a bootable version of Linux, which allows you to perform during any computers that support booting from a live CD or USB stick, regardless of operating system installed. The test results show the algorithm is quite satisfactory and even better in some algorithms, demonstrating the effectiveness of the framework.

For the development of the work all commands have been developed in ANSI C + + language. Several commands ATA and SCSI were implemented to support reading, self-testing and other features from them and the algorithms are implemented.

Keywords: Diagnostic Framework, Hard Disk, Solid-State Drive, Hard Disk Drive, SCSI, ATA, Linux .

Dedico este trabalho primeiramente a Deus,
à minha mãe Graça, ao meu pai Genésio,
à minha família, ao meu namorado Jefferson
e aos amigos.

"...

All your life, You were only waiting for the moment to arise.

Black bird fly, black bird fly

Into the light of the dark black night

..."

Black Bird - The Beatles

Sumário

Lista de Figuras	viii
Lista de Tabelas	ix
Lista de Siglas	ix
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.2.1 Objetivos Gerais	3
1.2.2 Objetivos Específicos	4
1.3 Organização da Monografia	4
2 Discos Rígidos	5
2.1 Introdução	5
2.2 Histórico	6
2.2.1 Primeiros Dispositivos	7
2.2.2 Desenvolvimento dos <i>Desktops</i>	8
2.3 Padrões Adotados	9
2.3.1 <i>Form Factors</i>	10
2.3.2 Interfaces	10
2.4 <i>Hard Disk Drive</i>	11
2.5 <i>Solid-State Drive</i>	12

2.6	<i>Hard Disk Drive</i> versus <i>Solid-State Drive</i>	13
2.6.1	Desempenho	13
2.7	Resumo do Capítulo	14
3	Fundamentação Teórica	20
3.1	Fundamentos de Diagnóstico de Falhas	20
3.1.1	Falha, Erro e Defeito	20
3.1.2	Diagnóstico de Falhas	21
3.1.3	Falhas em Discos Rígidos	22
3.2	SMART	23
3.2.1	Atributos	24
3.2.2	Auto-testes	24
3.3	Protocolos	25
3.3.1	SCSI	25
3.3.2	ATA	27
4	Metodologia e Implementação	30
4.1	ATA, SCSI, SMART e o Linux	30
4.2	<i>Framework</i> Proposto	32
4.3	Camada da Aplicação	34
4.3.1	Algoritmos Implementados	35
4.4	Versão Inicializável	36
4.5	Resumo do Capítulo	37
5	Resultados	40
5.1	Bootable	40
5.2	Testes	40
5.3	Dados	42
6	Conclusão	45
6.1	Perspectivas Futuras	46
	Referências Bibliográficas	51

Lista de Figuras

2.1	Pirâmide Hierárquica de Armazenamento, adaptada de (MAMUN, 2007)	15
2.2	Primeiros Dispositivos de Armazenamento	16
2.3	IBM 350, adaptada de (MORIMOTO, 2007)	16
2.4	Seis configurações padrão de Discos Rígidos, adaptada de (POTTS, 2008)	17
2.5	Componentes normalmente encontrados em HDDs, adaptada de (MAMUN, 2007)	17
2.6	Componentes de um <i>Hard Disk Drive</i> (HDD)	18
2.7	Solid-State Drive, adaptada de (SYSADM, 2011)	18
2.8	<i>Benchmark realizado pela PCmark05</i> , adaptada de (COMPONENTS, 2009)	19
3.1	Modelo de três universos	21
3.2	SCSI, modelo de camadas	26
3.3	Família de padrões SCSI	27
3.4	ATA, modelo de camadas	28
3.5	Família de padrões ATA	29
4.1	Comando de 10 bytes típico.	32
4.2	Diagrama de Classes do Framework	35
4.3	Fluxograma dos algoritmos implementados	37
4.4	Fluxograma dos algoritmos implementados	38
4.5	Fluxograma dos algoritmos implementados	39

5.1	<i>Benchmark realizado pela PCmark05</i>	41
5.2	<i>Benchmark realizado pela PCmark05</i>	41
5.3	<i>Benchmark realizado pela PCmark05</i>	42
5.4	<i>Benchmark realizado pela PCmark05</i>	42

Lista de Tabelas

3.1	Exemplos de atributos <i>Self-Monitoring, Analysis and Reporting Technology</i> (SMART) para ATA	25
4.1	Comandos SCSI selecionados em ordem alfabética	33
4.2	Comandos ATA selecionados em ordem alfabética	34
5.1	Notação utilizada nas tabelas de resultados	43
5.2	Resultados Framework	43
5.3	Resultados PC-Doctor	44
5.4	Comandos SCSI selecionados em ordem alfabética	44

Lista de Siglas

ABS *Air Bearing Heads*

ANSI *American National Standards Institute*

ATA *Advanced Technology Attachment*

ATAPI *Advanced Technology Attachment Packet Interface*

BD *Blu-Ray Disk*

BIOS *Basic Input/Output System*

CD *Compact Disk*

CD-ROM *CD - Read Only Memory*

CCROS *Charged Capacitor Read Only Store*

CRC *Cyclic Redundancy Check*

DVD *Digital Versatile Disk*

DASD *Direct Access Storage Devices*

EIDE *Extended IDE*

EPROM *Erasable Programmable Read-Only Memory*

EEPROM *Electrically Erasable Programmable Read-Only Memory*

IBM *International Business Machines*

ISA *Industry Standard Architecture*

iSCSI *Internet SCSI*

LBA *Logical Block Addressing*

HD *Hard Disk*

HDD *Hard Disk Drive*

HD-DVD *High Definition DVD*

IDE *Integrated Drive Electronics*

MLC *Multi-Level Cell*

PATA *Parallel ATA*

PCB *Printed Circuit Board*

RAM *Random Access Memory*

RAMAC *Random Access Method of Accounting and Control*

ROM *Read Only Memory*

SAS *Serial Attached SCSI*

SATA *Serial ATA*

SCSI *Small Computer System Interface*

SLC *Single-Level Cell*

SMART *Self-Monitoring, Analysis and Reporting Technology*

SPI *SCSI Parallel Interface*

SSD *Solid-State Drive*

VCM *Voice Coil Motor*

Capítulo 1

Introdução

Vivemos em uma era da tecnologia da informação onde cada aspecto de nossa vida passa por algum tipo de processamento ou armazenamento de informações e os discos rígidos desempenham um grande papel neste desenvolvimento tecnológico. Eles são memórias não voláteis, fundamentais ao funcionamento de qualquer sistema computacional moderno, responsáveis por armazenar sistemas operacionais, programas e dados dos usuários.

Atualmente, os dois principais tipos de *Hard Disk* (HD) são o HDD, baseado em armazenamento magnético, que é a tecnologia que domina o mercado, e o *Solid-State Drive* (SSD) baseado em memória *flash*, popularizada recentemente. Estas duas tecnologias seguem dois padrões definidos pela *American National Standards Institute* (ANSI), são eles: ATA¹ e SCSI. Estes padrões definem protocolos de comunicação, conexões físicas, comandos, entre outros.

Nas últimas décadas, os discos rígidos passaram por grandes mudanças, houve aumento de densidade, mais dados são gravados no mesmo espaço físico, diminuição de tamanho e a sua capacidade de armazenamento cresceu enormemente.

Devido a este aumento da densidade, os discos rígidos estão mais sujeitos a defeitos de fabricação. Além disso, tais dispositivos sofrem desgaste natural decorrente do uso, aumentando a probabilidade de ocorrência de falhas. Estejam eles em sistemas que controlam aplicações críticas, tais como bolsa de valores, plantas de usinas nucleares ou sistemas hospitalares, ou em um *desktop* de um usuário comum,

¹ATA foi o nome dado ao padrão *Integrated Drive Electronics* (IDE) ao ser oficializado pela ANSI, ele passou a ser chamado por muitas pessoas como ATA, IDE, ou ainda IDE\ATA

é importante que um sistema de diagnóstico eficiente seja executado periodicamente, visando evitar perda de dados, corrupção de arquivos ou falhas catastróficas, as quais podem implicar inclusive na perda de vidas humanas.

1.1 Motivação

A maioria das ferramentas de diagnóstico de dispositivos de *hardware* disponíveis no mercado, foi desenvolvida para sistemas operacionais *Windows*. Em pesquisas em *sites* de busca, por exemplo, encontra-se 1.650.000 para *Windows* e 900.000 resultados para *Linux*², em busca em *sites* de *downloads*³ por "diagnóstico de hardware" encontra-se em média, 171 *softwares* para *Windows* e apenas 11 para *Linux*. Entretanto nestes resultados estão vários *softwares* que não tratam de discos rígidos, apenas de CPU ou medição de temperatura, por exemplo, e filtrando esses resultados por "discos rígidos" encontramos apenas 75 *softwares* de diagnóstico para *Windows* e 5 para *Linux*.

Estes *softwares* de diagnóstico realizam diferentes algoritmos de teste, para detectar erros de leitura e escrita e avaliar a saúde do dispositivo, se há setores com falha ou iminência de falha, dando ao usuário tempo para realizar *backup* ou substituir o dispositivo em casos mais graves.

O *Windows* é o sistema operacional utilizado na maioria dos computadores pessoais. No entanto, *Linux* é o sistema operacional preferido para servidores e computadores de grande porte, responsáveis por prover os mais diversos serviços. O *Linux* roda de servidores a sistemas embarcados e possui várias distribuições com diferentes propostas de uso como *Fedora*, *SUSE*, *Ubuntu*, *Debian* e *Red Hat*, por exemplo, sendo a maioria gratuita. Ele dispõe de código aberto, possibilitando ao desenvolvedor amplo acesso ao sistema e a capacidade para adaptá-lo às suas necessidades. E dispõe ainda de várias funcionalidades úteis, como versões inicializáveis, que permitem a sua execução em qualquer computador cujo o *Basic Input/Output System* (BIOS) suporte inicialização por *Pendrive* ou *Compact Disk* (CD), sem requerer instalação e independentemente do sistema operacional instalado.

²Busca por "diagnóstico de hardware para Linux" e "diagnóstico de hardware para Windows" realizada em www.google.com em Setembro de 2011

³Site www.baixaki.com.br, 175 resultados para *Windows* e 6 para *Linux*
Site www.superdownloads.com.br 168 resultados para *Windows* e 17 para *Linux*, em Setembro de 2011

Há duas maneiras de acessar um HD via *software*: Acessando diretamente a controladora do disco ou através da BIOS. Acessar diretamente a controladora, através de comandos ATA ou SCSI, é o meio mais rápido de ter acesso disco, mas requer conhecimento sobre o *hardware*, (CARRIER, 2005). Este normalmente é o meio utilizado pelos *softwares* de diagnóstico, que enviam comandos ATA e SCSI para realizar algoritmos de teste e também ter acesso ao SMART, sistema de predição de falhas adotado pela indústria com o intuito de aumentar a confiabilidade dos discos rígidos, disponível na maioria dos HDs.

No Linux há alguns programas que implementam comandos ATA e SCSI e acessam o SMART tais como *smartmontools*, *hdparm*, *sg3-utils*, *util-linux*, mas que dispõem de poucas opções para o usuário, que para utilizá-los em um algoritmo de teste, por exemplo, teria de executá-los através de chamadas ao sistema, que pode levar a perda de desempenho e funcionalidade (TANENBAUM, 2001).

Neste sentido, o desenvolvimento de um *framework* para auxiliar no diagnóstico de discos rígidos é proposto neste trabalho. Ele é baseado na implementação de comandos ATA, SCSI e SMART, foi projetado para sistemas Linux, e voltado para execução em *desktops* e *laptops*. A partir deste *framework*, uma camada de aplicação com os algoritmos de testes mais importantes e mais frequentes nas ferramentas de diagnóstico do mercado é implementada. Esta camada foi portada para uma versão inicializável do Linux, chamada neste trabalho de *Bootable*.

1.2 Objetivos

Os objetivos gerais e específicos desta monografia, são apresentados a seguir.

1.2.1 Objetivos Gerais

Desenvolver um *framework* para auxiliar no desenvolvimento de testes de diagnóstico de discos rígidos em sistemas Linux para ser executado em *desktops* e *laptops*. Criar uma camada de aplicação, a partir do *framework* desenvolvido, com a implementação de alguns algoritmos de teste. Comparar a eficácia dos algoritmos implementados na camada de aplicação com os testes de ferramentas de diagnóstico do mercado. Portar os testes criados na camada de aplicação para uma versão inicializável do Linux, um *Bootable*.

1.2.2 Objetivos Específicos

Para o desenvolvimento da monografia alguns objetivos específicos foram necessários, eles estão listados a seguir:

- i. Revisão bibliográfica.
- ii. Pesquisa e estudo dos padrões ATA e SCSI.
- iii. Pesquisa e estudo sobre diagnóstico de falhas.
- iv. Implementação e validação de comandos ATA e SCSI em linguagem C++.
- v. Escolha e implementação de algoritmos de teste em C++, baseados nos testes das ferramentas de diagnóstico de mercado.
- vi. Criação de uma versão inicializável do Linux contendo os algoritmos desenvolvidos neste trabalho.
- vii. Avaliação da eficácia dos algoritmos implementados neste trabalho em comparação com os testes realizados por ferramentas de diagnóstico.
- viii. Análise dos resultados.

1.3 Organização da Monografia

Esta monografia está organizada em 6 capítulos. O capítulo 2 apresenta um apanhado sobre a história e o funcionamento das tecnologias nas quais os discos rígidos atuais se baseiam, os padrões adotados pela indústria são apresentados e um comparativo entre as tecnologias HDD e SSD é realizado. O capítulo 3 detalha os fundamentos de diagnóstico de falha, as principais falhas em discos rígidos e os padrões ATA e SCSI.

No capítulo 4, a arquitetura do *framework* desenvolvido neste trabalho é descrita, bem como os algoritmos de testes implementados, o processo de criação de uma versão inicializável do Linux e a metodologia de testes. No capítulo 5 são descritos e discutidos os resultados obtidos no trabalho, a implementação da camada de aplicação e os resultados obtidos nos testes com os algoritmos implementados. Por fim, o último capítulo apresenta as conclusões, considerações finais e as perspectivas futuras deste trabalho.

Capítulo 2

Discos Rígidos

Neste capítulo, a evolução dos discos rígidos é contada de maneira breve para contextualizar o leitor quanto às tecnologias utilizadas atualmente e ao seu funcionamento.

2.1 Introdução

Sistemas modernos de computação usam diferentes tecnologias para armazenar estas informações, seja temporariamente ou permanentemente. Essas tecnologias podem ser memórias de estado sólido, como *Read Only Memory* (ROM), *Random Access Memory* (RAM) e memórias *Flash*, armazenamento magnético, como HDD, *floppy disk* e *tape*, ou armazenamento óptico, como *CD - Read Only Memory* (CD-ROM), *Digital Versatile Disk* (DVD), *Blu-Ray Disk* (BD) e *High Definition DVD* (HD-DVD).

Os discos rígidos desempenham um grande papel neste desenvolvimento tecnológico. Atualmente, eles tanto podem ser dispositivos de armazenamento magnético como os HDDs, como circuitos de memória *Flash*, no caso dos SSDs atuais, quanto dispositivos híbridos,(SEAGATE, 2011). Os principais fatores considerados na escolha entre essas tecnologias são o preço, a taxa de transferência de dados, o tempo de acesso e a confiabilidade (MAMUN, 2007).

A pirâmide de hierarquia de armazenamento, Figura 2.1, adaptada de (MAMUN, 2007), descreve uma hierarquia de tecnologias de memória em função do custo, da capacidade de armazenamento e do tempo de acesso. Nela os dois principais tipos de discos rígidos,HDD e SSD, encontram-se na camada

intermediária.

A camada superior da pirâmide contém registradores do processador, memória cache e memória RAM, que se caracterizam pelo alto custo, rapidez no acesso aos dados e baixa capacidade de armazenamento. Já a camada intermediária contém memórias Flash, das quais são feitos os SSDs, discos magnéticos e discos ópticos, que possuem um custo menor em comparação às memórias internas, maior capacidade de armazenamento e menor velocidade no acesso aos dados. Na camada inferior se encontram fitas magnéticas, dispositivos de acesso sequencial usados para cópias de segurança.

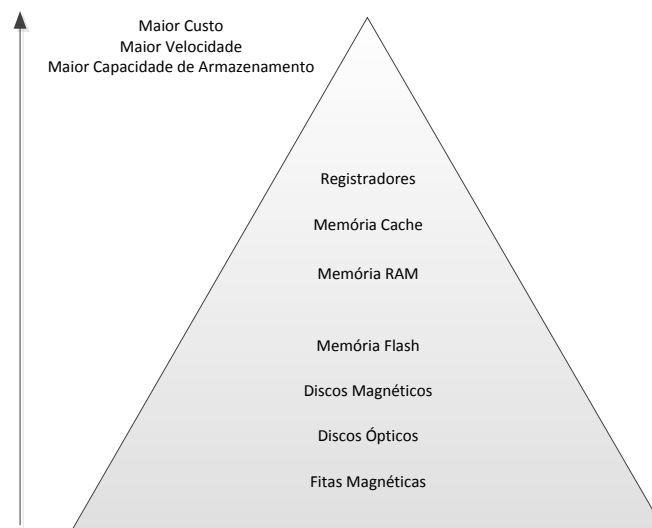


Figura 2.1: Pirâmide Hierárquica de Armazenamento, adaptada de (MAMUN, 2007)

2.2 Histórico

O disco rígido, também conhecido como *Direct Access Storage Devices* (DASD), em referência à capacidade de acesso direto, foi um dos dispositivos que mais evoluiu na história da computação.

Em pouco mais de 5 décadas, a indústria de armazenamento de dados evoluiu de um grande aparato de 50 discos de 24 in de diâmetro (aproximadamente 61 cm), com uma capacidade total de 5 MB para discos que armazenam até 4 TB em discos de 3,5 in de diâmetro (pouco menos de 9 cm). Esta enorme evolução se tornou possível graças a avanços em diversas áreas do conhecimento incluindo: materiais, tribologia¹, mecânica, mecatrônica, processamento de sinais e eletrônica, (HARKER

¹Tribologia é a ciência que estuda a interação de superfícies, submetidas a cargas e a movimentos

et al., 1981).

Os primeiros HDs eram significativamente diferentes dos que usamos agora não apenas em relação ao tamanho, mas em aspectos como capacidade de armazenamento e taxa de transferência de dados, bem como na tecnologia utilizada.

Os HDDs e SSDs são contemporâneos e foram desenvolvidos pela *International Business Machines* (IBM)², empresa que começou a se destacar no início do século XX através do desenvolvimento de máquinas mecânicas e elétricas como calculadoras de repetição e máquinas datilografia e que foi responsável por muitos avanços na computação.

As primeiras máquinas utilizavam cartões e fitas de papel para codificar e para armazenar informações, mas, entre outras desvantagens, não havia memória RAM suficiente e o *software* e dados iniciais precisavam ser carregados a cada vez que a máquina era ligada.

Havia a necessidade de memórias mais eficientes, temporárias e permanentes. Então a IBM começou a desenvolver memórias alternativas, principalmente baseadas em magnetismo. Ela desenvolveu os discos magnéticos, exemplificados na Figura 2.2.a, adaptada de (ARCHIVES, 1970), e com a popularização dos transistores na década de 1950 ela desenvolveu o primeiro dispositivo de memória não volátil de estado sólido chamado *Charged Capacitor Read Only Store* (CCROS), (RENT, 2010a), o antecessor das memórias *Erasable Programmable Read-Only Memory* (EPROM), *Electrically Erasable Programmable Read-Only Memory* (EEPROM) e *Flash*, mostrado na Figura 2.2.b, adaptada de (RICHARDS, 1965).

Na mesma época, outro método de SSD foi desenvolvido usando magnetismo, era o Núcleo de Memória, que utilizava pequenos núcleos individuais de ferrite³ unidos por cobre, como exemplificado na Figura 2.2.c, adaptada de (RENT, 2010a).

2.2.1 Primeiros Dispositivos

O primeiro equipamento a utilizar um DASD não volátil foi o *Random Access Method of Accounting and Control* (RAMAC), lançado pela IBM em 1956. A unidade de disco do RAMAC, chamada IBM 350, mostrada na Figura 2.3,

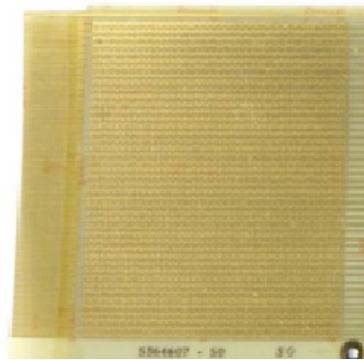
relativos.

²A IBM foi o novo nome dado à C-T-R (Computing-Tabulating-Recording Company), empresa fruto da fusão de outras três empresas entre elas a *Tabulating Machine Company*, fundada por Herman Hollerith, especializada no processamento de dados de cartões perfurados.

³Ferrite, material feito de cerâmica com propriedades eletromagnéticas



(a) Disco Magnético, adaptada de (ARCHIVES, 1970)



(b) Cartão CCROS, adaptada de (RICHARDS, 1965)



(c) Matriz de Núcleos de Memória, adaptada de (RENT, 2010a)

Figura 2.2: Primeiros Dispositivos de Armazenamento

adaptada de (MORIMOTO, 2007), continha 50 discos, cada um com 24 in de diâmetro e que podiam armazenar 5 MB a uma densidade de gravação de $2Kbits/in^2$, (STEVENSON, 1981) e taxa de transferência de dados de 8.8 kilobytes por segundo (KB/s) (HARKER *et al.*, 1981). Ela não possuía sistema de malha fechada para posicionamento da cabeça de leitura e escrita.

Em 1961, o primeiro dispositivo HDD a usar de rolamentos de ar, *Air Bearing Heads* (ABS), (STROM *et al.*, 2007), foi lançado e em 1963, o primeiro disco removível.

Em 1964, o primeiro equipamento a utilizar cartões CCROS⁴, é lançado pela IBM, era a família IBM System 360, família de *mainframes*.⁵

⁴Cartões CCROS foram os primeiro dispositivos SSD

⁵Mainframe, computador de grande porte, dedicado normalmente ao processamento de um volume grande de informações.

Em 1971, a IBM lançou o primeiro HDD com malha fechada, o IBM 330 *Merlin Drive*. Ele mantinha a posição da cabeça relativa à trilha sensorizada do disco.

Em 1973, a IBM introduziu outro modelo de HDD, o IBM 3340, que usava cabeça e braço de ferrite. Ele continha 2 ou 4 discos de 14 polegas de diâmetro, aproximadamente 35,56 cm, possuía capacidade de armazenamento de 35 MB para um dispositivos com 2 discos, taxa de transferência de dados de 0.8MB/s e densidade de gravação média de 1,69 Mbits/in². Este ficou conhecido como disco Winchester, o primeiro HDD a usar a malha de controle dos dispositivos atuais.



Figura 2.3: IBM 350, adaptada de (MORIMOTO, 2007)

2.2.2 Desenvolvimento dos *Desktops*

O desenvolvimento de computadores *desktop*, no início da década de 1980, representou uma nova era da tecnologia da informação. Este tipo de computador iniciou uma nova demanda na indústria de armazenamento de dados, pois os HDs, até então usado apenas para soluções corporativas, teriam de se adequar a equipamentos mais compactos.

A necessidade de DASD com tamanho físico adequado a desktops se tornou evidente. E foi quando os discos de 5 $\frac{1}{4}$ polegadas surgiram. Estes dispositivos tinham 5 MB de capacidade de armazenamento e traziam a primeira interface de disco, eram da série ST506, introduzida pela Seagate Technology.

A série ST506 usava motor de passo como atuador. Esse motor, ao receber um impulso elétrico, movimenta o braço por uma curta distância, correspondente ao comprimento de uma trilha. Eles eram muito suscetíveis a problemas de

desalinhamento e não permitiam densidades de gravação muito altas. E logo nos primeiros anos da década de 1980, o primeiro dispositivo de $5\frac{1}{4}$ polegadas com atuador do tipo *Voice Coil Motor* (VCM)⁶, dispositivo que atua através de atração e repulsão magnética, foi produzido em grande escala.

Em 1983, o primeiro HDD com discos de $3\frac{1}{2}$ polegadas de diâmetro foi lançado pela Rodime.

Até então, não havia interfaces de disco padronizadas, as novas interfaces eram vendidas junto com os HDs e instaladas em slots *Industry Standard Architecture* (ISA) disponíveis na placa-mãe, (MORIMOTO, 2007). E em 1985 a Quantum Corporation lançou o *Plus HardCard*, em que tanto a controladora quanto o HD eram integrados em um único slot ISA. Esta integração se mostrou vantajosa pois diminuía problemas de sincronismo e simplificava o projeto e isto deu origem ao padrão IDE.

Apenas em 1990 o padrão IDE foi ratificado pelo ANSI, organização que tem por objetivo facilitar a padronização dos trabalhos de seus membros, dando origem ao padrão ATA.

Nos anos seguintes, os HDs de $5\frac{1}{4}$ polegadas começaram a perder mercado para os HDs de $3\frac{1}{2}$ e $2\frac{1}{2}$ polegadas, posteriormente usados no mercado de laptops. Estes formatos foram bem aceitos pelo mercado, sendo utilizados até hoje.

Paralelamente a estes avanços, o desenvolvimento de memórias *Flash* começava a dar os primeiros passos. As memórias *Flash* foram desenvolvidas a partir das memórias EEPROM. Em 1984, o Dr. Masuoka publicou sua invenção no artigo ”*A New Flash EEPROM Cell Using Triple Polysilicon Technology*”, (MASUOKA *et al.*, 1984).

Em 1987, a Toshiba anuncia as memórias *Flash* do tipo *NAND* e em 1988 a Intel anuncia as memórias *Flash* do tipo *Nor*, (BEZ *et al.*, 2003). Hoje, estes dois tipos são considerados os padrões da indústria. A memória *Flash* do tipo *Nand* é otimizada para o armazenamento de dados em massa e a do tipo *Nor* para o armazenamento de dados e código⁷. E em 1988, o primeiro produto *Flash* foi apresentado, (KUNETT

⁶Motores Voice Coil são atuadores cujo funcionamento baseia no de um alto-falante, onde uma bobina é excitada por corrente (sinal de áudio), e um ímã permanente interage com o campo dessa bobina, deslocando um cone e produzindo ondas mecânicas de som.

⁷*Flash* do tipo *Nor* são usadas para armazenar informações na BIOS da placa-mãe e *firmwares* em dispositivos diversos, pois, embora requeiram pouco tempo para a leitura, apresentam um tempo de gravação alto.

et al., 1988), tratava-se de um *chip* de memória *Flash* do tipo *Nor* de 256K.

Inicialmente, as memórias *Flash* foram usadas para a substituição de memórias EPROM ⁸, (BEZ *et al.*, 2003) e só passou a ser utilizada em larga escala no final dos anos 1990. E os primeiros SSDs baseados em *Flash* só chegaram ao mercado em 2007.

2.3 Padrões Adotados

Com o passar dos anos, várias empresas começaram a fabricar discos rígidos e visando a compatibilidade entre diferentes discos rígidos e computadores padrões de tamanho e de interfaces foram adotados.

2.3.1 Form Factors

Os discos rígidos foram projetados para serem instalados dentro de computadores, e são produzidos em poucos padrões de tamanho e forma. Estes padrões são chamados de *form factors* de discos rígidos, algo como configurações padrão, e se referem principalmente às dimensões externas dos discos rígidos, (MAMUN, 2007).

A Figura 2.4, adaptada de (POTTS, 2008), mostra seis HDDs com as tampas removidas deixando à mostra os pratos e as cabeças, onde os padrões de discos de 8, 5.25, 3.5, 2.5, 1.8 e 1 in de diâmetro são representados.

Nos HDDs, os *form factors* geralmente são limitados pelo tamanho do disco e os SSDs, embora não tenham esta limitação, geralmente seguem o mesmo padrão para serem compatíveis com as estruturas existentes em *laptops* e *desktops*.

2.3.2 Interfaces

Da mesma forma que outros componentes dos HDs as interfaces que os interligam passaram por muitas mudanças. Atualmente existem dois tipos predominantes no mercado: ATA e SCSI, ambas são padrões ANSI. Estes padrões definem conectores, conexões físicas, como sinais e temporização, protocolos de transporte, modelos de dispositivos e comandos. Estes modelos e comandos são a base do desenvolvimento do *Framework* proposto neste trabalho e serão detalhados no Capítulo 3.

⁸EPROM, *Erasable Programmable Read-Only Memory*



Figura 2.4: Seis configurações padrão de Discos Rígidos, adaptada de (POTTS, 2008)

O padrão ATA, também conhecido como IDE, foi projetado para ser uma interface entre computadores e discos rígidos. Diversas versões foram definidas até agora, são elas: ATA 1 ou IDE, ATA 2 ou *Extended IDE* (EIDE), ATA 3, ATA\Advanced Technology Attachment Packet Interface (ATAPI) 4, ATA\ATAPI 5, ATA\ATAPI 6, ATA\ATAPI 7 e ATA\ATAPI 8.

A versão ATA 3 foi a primeira a trazer o sistema SMART e a partir da versão ATA 4, o padrão ATA se tornou capaz de enviar comandos SCSI encapsulados em comandos ATA, possibilitando a comunicação com outros periféricos como unidades de discos ópticos. Esta extensão do protocolo ATA foi chamada de *Advanced Technology Attachment Packet Interface* (ATAPI).

O padrão SCSI foi desenvolvido para ser um padrão genérico para periféricos, como *scanners*, unidades de discos ópticos, mídias removíveis e discos rígidos. Ele foi padronizado pela ANSI em 1986 e dele foram desenvolvidas as seguintes versões: SCSI 1, SCSI 2, com suporte à comandos *multi-thread*, e SCSI 3, com separação entre conexão física, protocolos de transporte e conjuntos de comandos SCSI.

O SCSI oferece o melhor desempenho em aplicações que requerem velocidade

de processamento e transferência de dados (MASON, 2000), por isso os discos SCSI são tipicamente mais caros e, em geral, usados em servidores, onde o alto custo é justificável. Já os discos ATA são mais utilizados nos computadores pessoais.

Os computadores se tornaram mais rápidos e as interfaces precisavam atender a esta demanda e, neste intuito, tanto o padrão ATA quanto o SCSI, antes interfaces paralelas, tenderam à serialização visando alcançar maiores velocidades de transmissão e evitar problemas de interferência entre as vias. Desta iniciativa surgiram os padrões *Serial ATA* (SATA) e *Serial Attached SCSI* (SAS), protocolos seriais de transporte e interconexão física das interfaces ATA e SCSI, respectivamente, e as interfaces anteriores passaram a ser chamadas de *Parallel ATA* (PATA) e *SCSI Parallel Interface* (SPI).

2.4 Hard Disk Drive

De maneira resumida os HDDs são a integração dos seguintes componentes, descritos na Figura 2.5, adaptada de (MAMUN, 2007): Discos magnéticos (*platters*⁹), onde os dados são armazenados, um motor de rotação, responsável por girar os *platters* a uma velocidade angular constante adequada à leitura e gravação de dados, pares de cabeças de leitura e escrita de dados, há duas para cada lado de um *platter*, um braço metálico, onde se encontram as cabeças de leitura e escrita, um atuador, mecanismo composto por um motor VCM e pelo braço, um conector, atualmente a maioria são do tipo SATA ou SCSI, e uma placa controladora *Printed Circuit Board* (PCB), que é responsável por acionar o motor que gira os *platters*, posicionar as cabeças de leitura e escrita, através do atuador, e executar os comandos recebidos pelo sistema através de um conector, tais comandos podem ser ATA ou SCSI. A controladora pode ser interna, como nos discos SATA, ou externa, como nos discos SCSI, (CARRIER, 2005).

Ao contrário do que se possa pensar o HDD não é vedado ou blindado. Seu funcionamento se baseia em aerodinâmica. As cabeças não devem entrar em contato com os *platters*, pois isso os danificaria, por isso eles são girados a uma alta velocidade de rotação e só então o braço é movimentado, isto gera uma "bolsa" de ar, uma folga

⁹ *Platters* são discos magnéticos compostos por duas camadas, o substrato, um disco feito de liga de alumínio ou vidro, extremamente polidos para tentar evitar qualquer rugosidade, e a superfície magnética, que possui apenas alguns micrômetros de espessura (MORIMOTO, 2007). Neles cada bit é armazenado em um setor de fino segmento destas superfícies através da magnetização do meio pela cabeça de gravação.

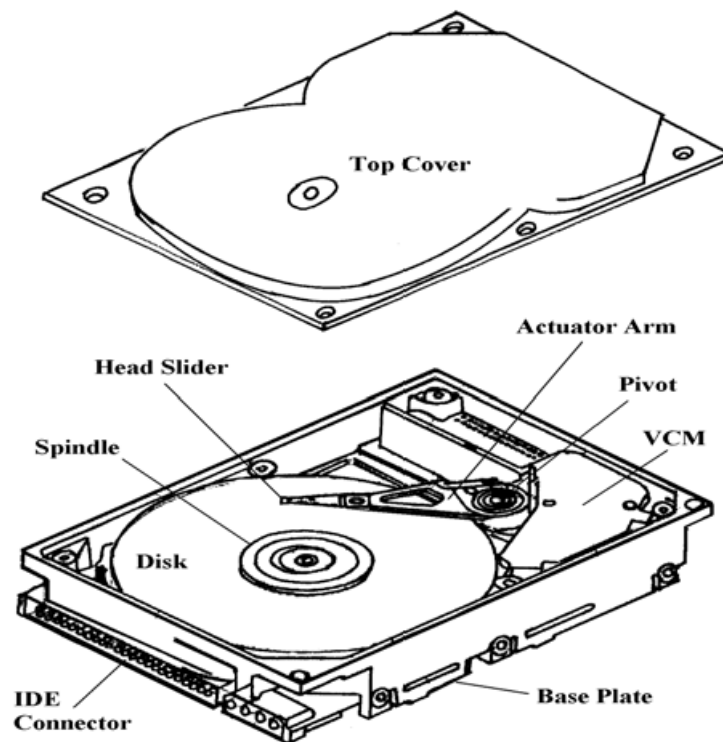


Figura 2.5: Componentes normalmente encontrados em HDDs, adaptada de (MAMUN, 2007)

entre a cabeça e o *platter*¹⁰, fazendo com que as cabeças flutuem¹¹ sobre eles. Esta folga pode ser alterada por parâmetros ambientais como temperatura, a pressão do atmosférica e a umidade (STROM *et al.*, 2007).

Nos HDDs existem uma entrada de ar que passa por um filtro entrar em contato com os *platters*. E embora não seja blindado, abrir um HDD fora de uma sala limpa¹² pode acarretar na inutilização do mesmo, pois qualquer partícula de poeira presente no ambiente poderia danificar a superfície dos *platters* (ZHAO; XIE; WANG, 2007).

Os HDDs atuais contém normalmente 1, 2 ou 4 *platters*, para cada lado de um *platter* há duas cabeças acopladas, uma de escrita e outra de leitura, as cabeças estão presas por espaçadores ao braço e embora haja um conjunto de cabeças para cada lado do *platter*, elas não são independentes. Todas se movimentam como uma só, guiadas pelo braço, como mostrado na Figura 2.4.a, adaptada de (VASCONCELOS, 2002) . Os *platters* são organizados em trilhas concêntricas e

¹⁰Esta folga é chamada de *Head-Disk Clearance*, (STROM *et al.*, 2007)

¹¹Esta bolsa de ar criada funciona como rolamento de ar, o sistema ABS descrito anteriormente

¹²Sala Limpa, ambiente controlado utilizado para testes ou manufatura de produtos onde a contaminação por partículas presentes no ar interfere no resultado

estas trilhas organizadas em setores. Como descrito na Figura 2.4.b, adaptada de (VASCONCELOS, 2002).

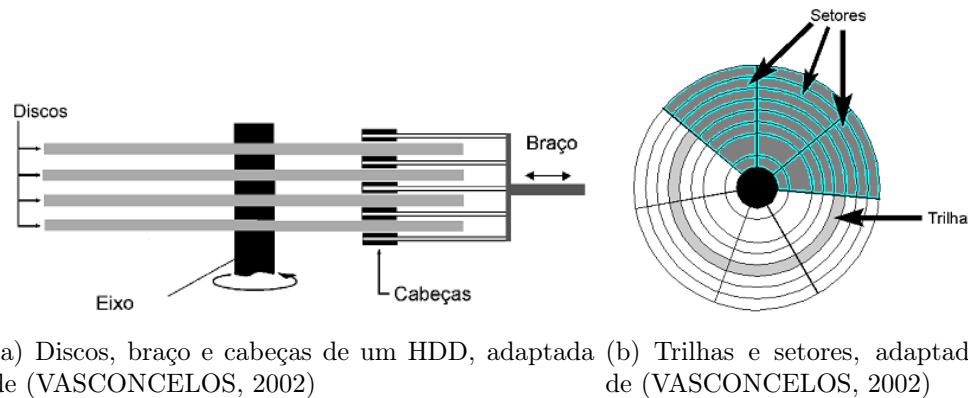


Figura 2.6: Componentes de um HDD

2.5 Solid-State Drive

Os Solid-State Drives (SSD) podem ser definidos como dispositivos que armazenam dados em *chips*, na maioria dos casos memórias *Flash*, e sem partes mecânicas, como mostrado na Figura 2.7, adaptada de (SYSADM, 2011).



Figura 2.7: Solid-State Drive, adaptada de (SYSADM, 2011)

Os principais componentes de um SSD são a controladora, os módulos de memória *cache* e memória *Flash*, normalmente 10 ou 20, e o conector, que pode ser SATA, SAS, entre outros, (RENT, 2010b). A controladora consiste em um processador embarcado, normalmente um microcontrolador de 32 *bits*, que executa

código a nível de *firmware*, como o SMART, gerenciamento automático de memória¹³ e TRIM¹⁴, por exemplo, em controladores de memória *Flash*, DRAM, ROM e interfaces de entrada e saída.

2.6 *Hard Disk Drive versus Solid-State Drive*

Os HDDs e SSDs utilizam as mesmas interfaces, possuem controladora, seja ela interna ou externa, organização dos dados em setores, códigos de correção de erro, e área reservada de memória (*spare area*).

No entanto, as duas tecnologias usam princípios de armazenamento diferentes, e há muitas diferenças de desempenho e fontes de falha.

2.6.1 Desempenho

Em comparação aos HDDs, os SSDs são menos afetados por choques físicos, pois não contêm partes móveis, são mais silenciosos, tem menor consumo de energia elétrica.

Os SSDs tem um tempo de inicialização menor, são mais rápidos na escrita e leitura aleatória e na leitura sequencial de arquivos grandes. Na Figura 2.8, adaptada de (COMPONENTS, 2009), um *benchmark*¹⁵ é realizado entre HDDs de 7200 rpm e 5400 rpm e um SSD Toshiba MLC NAND¹⁶. Nele, o dispositivo SSD apresenta resultados bem superiores em termos de taxa de transferência de dados, especialmente na inicialização de aplicações.

Entretanto, os *chips* de memória *flash*, usados nos SSDs, tem limites de regravação estimados em 10.000 vezes, se *Multi-Level Cell* (MLC), ou 100.000, se *Single-Level Cell* (SLC), (MORIMOTO, 2010), o que compromete a sua longevidade. Enquanto que nos HDDs, a quantidade de escritas está associada apenas ao desgaste do material ou à danos físicos.

A capacidade de armazenamento e o custo por GB também são pontos de divergência entre as duas tecnologias. Até setembro de 2011, o dispositivo SSD de

¹³*Garbage Collector*

¹⁴TRIM é um comando ATA para gerenciamento de dados, responsável por informar quais blocos de dados não estão mais em uso, (INTEL, 2010).

¹⁵Execução séries de testes e ensaios visando avaliar o desempenho relativo de um produto

¹⁶O *Benchmark* foi realizado usando processador Intel Core2Duo 2.2GHz E4500, memória cache L2 de 2MB, memória RAM de 2GB DDR SDRAM com 800MHz, placa gráfica Intel GMA3100 e sistema operacional Windows® Vista

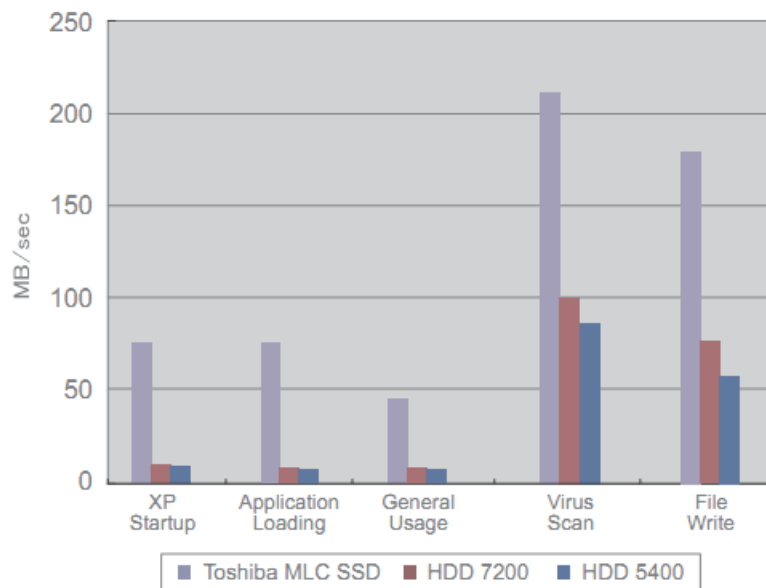


Figura 2.8: *Benchmark realizado pela PCmark05, adaptada de (COMPONENTS, 2009)*

maior capacidade tinha 1.6TB, *form factor* de 2.5 in, e o HDD de maior capacidade possuía 4TB, *form factor* de 3.5 in.

No dispositivo HDD, o custo por GB era de 0,0625 dólares (PARRISH, 2011). E no dispositivo SSD, embora não tivesse o preço anunciado, estimava-se um custo por GB superior a 6 dólares, custo por GB de um SSD de 1 TB (SHIMPI, 2011), o segundo maior dispositivo do tipo. Nestes quesitos, os dispositivos HDD, com custos quase 100 vezes menores, são bastante superiores e sua predominância no mercado é justificada.

2.7 Resumo do Capítulo

Este capítulo mostrou um pouco da história dos discos rígidos, os principais padrões, a evolução das tecnologias SSD e HDD e um comparativo entre elas.

No próximo capítulo serão abordados fundamentos de diagnóstico de falha, o sistema SMART, e as principais características dos padrões ATA e SCSI.

Capítulo 3

Fundamentação Teórica

Neste capítulo os alicerces desta monografia serão explorados: os fundamentos de diagnóstico de falha, os padrões adotados em discos rígidos e como interagir com eles através do Linux.

3.1 Fundamentos de Diagnóstico de Falhas

Primeiramente, para explicar o conceito deste trabalho, é necessário que se defina o conceito de falha, o conceito de diagnóstico e as origens das falhas no dispositivo em estudo, os discos rígidos.

3.1.1 Falha, Erro e Defeito

Embora sejam usadas como sinônimo, estas palavras tem significados diferentes para sistemas de computação. A falha está associada ao universo físico, o erro é a representação da falha no universo da informação, causado por uma falha, e o defeito é um desvio da especificação e ocorre em consequência de um erro, (LEE; ANDERSON, 1990). Descritos na Figura 3.1, adaptada de (WESZ, 2007).

Uma falha pode ser ocasionada por um problema em nível de *hardware*, uma flutuação de tensão ou impacto que corrompa um setor de um disco rígido, por exemplo, e não necessariamente leva a um estado de erro. O erro acontece quando uma informação é corrompida devido a um setor falho e também não necessariamente leva a um estado de defeito, que é detectado no nível de usuário, pois talvez a informação corrompida nunca seja usada, ou possa ser recuperada.

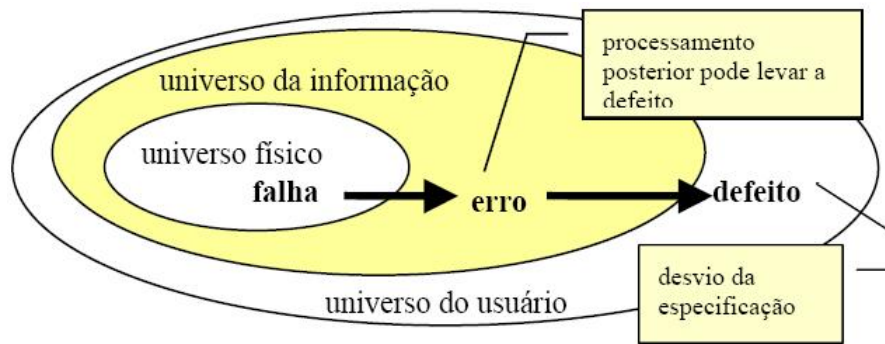


Figura 3.1: Modelo de três universos

3.1.2 Diagnóstico de Falhas

Com o propósito de evitar que um sistema se torne defeituoso, as falhas devem ser detectadas e diagnosticadas. A detecção de falhas se baseia em sistemas teóricos, modelos de processo ou funções matemáticas para reconhecer sintomas de falha. Estes sintomas são desvios das características do processo modelado. O diagnóstico de falhas é diferente da detecção de falhas, que consiste em reconhecer que uma falha aconteceu.

Diagnosticar falhas consiste na identificação do tipo de falha com a maior quantidade de detalhes possíveis, bem como o tamanho da falha, localização e o momento da detecção, (ISERMANN, 2011). O procedimento de diagnóstico é baseado na observação e análise de sintomas e no conhecimento prévio sobre o processo. Os sintomas disponíveis podem ser:

Sintomas Analíticos são o resultado da comparação entre os sinais de um processo e do seu modelo de detecção de falha. Verifica-se se estes sinais excederam algum limiar pré-determinado.

Sintomas Heurísticos são resultado da análise de um operador humano, trata-se de impressões quanto a ruído, oscilações, cores e fumaça obtidas por inspeção. Tratados muitas vezes como medidas qualitativas.

Histórico de processos e estatísticas de falha este histórico inclui informações sobre tempo de execução e reparo e descrevem a frequência de certas falhas no processo.

Representação unificada dos sintomas é a observação conjunta de sintomas analíticos e heurísticos, através de mecanismos de inferência de sistemas *fuzzy*.

Relações Falha-Sintomas consistem na análise inversa das relações de causa e efeito. Na análise direta, observam-se falhas que geram eventos que geram sintomas. Na análise inversa, as falhas são deduzidas a partir dos sintomas observados.

3.1.3 Falhas em Discos Rígidos

Os discos rígidos são organizados em pequenas módulos de armazenamento, chamados de setores, nos HDDs, ou blocos, nos SSDs. Aqui estes módulos serão chamados genericamente de setores. Estes setores dispõem das seguintes informações: identificação, usada para localizar o setor, dados, a informação em si, campos de sincronização e espaçadores, usados pela internamente controladora, e códigos de correção de erro, como *Cyclic Redundancy Check* (CRC), por exemplo (CORPORATION, 2001). Estes códigos de correção de erro são usados pela controladora para detectar a ocorrência de falhas.

As falhas podem ser geradas no processo de fabricação, alguns HDs novos já saem de fábrica com setores falhos¹ e outros setores se danificam em decorrência do uso e desgaste natural. Os setores falhos são remapeados em uma área reservada. Enquanto que nos HDDs isto pode gerar perda de desempenho, pois gera mais movimentos de busca para ler um arquivo, por exemplo, nos SSDs este mapeamento é praticamente transparente.

A área reservada tem duas funções principais: servir de armazenamento temporário de novos dados e ser uma espécie de memória "reserva", onde os setores com falha são remapeados, aumentando a longevidade do disco rígido (MORIMOTO, 2010). Ela é gerenciada pela controladora e provém da diferença entre a nomenclatura decimal utilizada pelos fabricantes e a nomenclatura binária usada pelos sistema operacional. Por exemplo, 1GB de memória na nomenclatura do fabricante corresponde a 10^9 bytes, o que para um sistema operacional corresponde a 2^{30} bytes e a diferença entre as duas notações, igual a 73.741.824 bytes, é a quantidade de memória a ser usada como área reservada.

O remapeamento de setores com falha é transparente para o sistema operacional. Quando um setor falho é requisitado, a controladora intercepta e devolve o dado correto do novo bloco atribuído àquele endereço na área reservada. Apenas quando

¹Em inglês *Bad blocks* ou *Bad sectors*

esta área está totalmente cheia, o sistema operacional começa a indicar a existência de setores falhos.

Nos SSDs os tipos mais comuns de falha em células de memória estão relacionados a distúrbios de gravação, problemas na retenção de dados e defeitos na fabricação. Cada célula armazena um bit de memória. Os distúrbios de gravação consistem na corrupção de dados de células escritas durante a escrita de outras células na mesma matriz de memória (BEZ *et al.*, 2003). Estes distúrbios podem acontecer nas células de mesma coluna ou linha.

Nos HDDs, assim como nos SSDs, existem distúrbios de gravação, problemas na retenção de dados e defeitos de fabricação. Os distúrbios de gravação podem ocorrer entre setores adjacentes, trilhas vizinhas e até mesmo entre *platters*, já que há um movimento unificado à todas as cabeças. Um exemplo de distúrbio de gravação é a propagação radial do erro (ZHAO; XIE; WANG, 2007), onde por algum motivo uma trilha não é escrita como um círculo perfeito² e quando esta é referenciada para leitura a cabeça segue um círculo perfeito assim como para as trilhas seguintes.

Mas, além das falhas originadas pelo desgaste ou defeitos de fabricação do meio magnético, nos HDDs há várias fontes de falha inseridas pelas partes móveis assim como por fatores ambientais e impactos físicos. Por exemplo, quando uma tentativa de leitura de um setor falha, por exemplo, a causa tanto pode ser um distúrbio de gravação, quanto uma falha no atuador, que não conseguiu posicionar a cabeça corretamente.

Outro exemplo é a folga existente entre as cabeças e os *platters*, que diminui com o aumento da temperatura e com a diminuição da pressão atmosférica, e também com o aumento da humidade do ar. Esta diminuição da folga pode levar ao choque das cabeças com pontos de maior rugosidade nos *platters* e estes choques podem gerar erros de leitura e gravação.

3.2 SMART

A indústria de HDs adotou um sistema de monitoramento para discos rígidos, o sistema SMART, visando detectar falhas, disponibilizar diferentes tipo de auto-teste e relatar medidas indicadoras de confiabilidade. O SMART prevê falhas em discos rígidos individualmente, diferentemente dos modelos de confiabilidade, que depois

²Este fenômeno é chamado *track misregistration* (TMR)

de análises estatísticas calculam a probabilidade de falha de uma população de discos rígidos, supondo que todos são iguais.

Um algoritmo de alerta de falha é executado pelo *firmware* da controladora do disco rígido e checa se as medidas excederam o limiar máximo e gera um alerta binário do tipo: Irá falhar, Não Irá Falhar. Os limiares máximos são definidos pelos fabricantes. O objetivo é gerar o alerta 24 horas antes de o *drive* falhar, (HUGHES *et al.*, 2002).

Há comandos padronizados para habilitação e leitura desses alertas de falha pelos sistemas operacionais. Alguns computadores checam o SMART durante a inicialização do sistema operacional e alguns fabricantes disponibilizam programas de diagnóstico que leem os alertas do SMART (HUGHES *et al.*, 2002). O sistema SMART foi especificado para o padrão ATA, detalhado na próxima seção e o seu equivalente em discos SCSI é a combinação dos comando SEND DIAGNOSTIC e RECEIVE DIAGNOSTIC RESULTS.

3.2.1 Atributos

Na tabela 3.1 alguns atributos monitorados pelo SMART são exemplificados.

3.2.2 Auto-testes

O SMART também disponibiliza testes para verificar as condições do disco rígido, (COMMITTEE, 2007),(MCLEAN, 2001). Estes testes são descritos a seguir.

SMART *Short Self-Test* avalia o desempenho elétrico, mecânico e de leitura do disco. Realiza buscas em pequenas áreas do disco e checa a lista de setores pendentes³, com prováveis erros de leitura.

SMART *Extended Self-Test* mesmo princípio do SMART *Short Self-Test*, mas realiza a leitura de toda a superfície do disco, geralmente é um processo bastante demorado.

SMART *Conveyance Self-Test* pretende identificar danos ocorridos durante o transporte do disco rígido.

SMART *Selective Self-Test* executa teste apenas em alguns trechos do disco.

³A controladora armazena informações como esta em *logs*, que podem ser acessados por comandos específicos, ATA ou SCSI

Identificador	Atributo	Descrição
01	Taxa de erro de leitura (<i>Read Error Rate</i>)	Frequência de erros durante a leitura de dados do disco.
02	Desempenho (<i>Throughput Performance</i>)	Eficiência média do disco.
05	Contagem de setores realocados (<i>Reallocated Sectors Count</i>)	Contagem dos setores realocados na área reservada.
08	Desempenho das operações de busca (<i>Seek Time Performance</i>)	Desempenho médio das operações de busca nos discos magnéticos. A diminuição deste atributo pode indicar problemas no sistema mecânico.
191	Taxa de erro devido a impactos (<i>G-sense Error Rate</i>)	Frequência de erros resultantes de impactos e vibrações.
194	Temperatura (<i>Temperature</i>)	Atual temperatura interna do disco.
197	Contagem atual de setores pendentes (<i>Current Pending Sector Count</i>)	Número de setores esperando para serem remapeados.

Tabela 3.1: Exemplos de atributos SMART para ATA

3.3 Protocolos

Como dito anteriormente, ATA e SCSI são os padrões de comunicação utilizados para discos rígidos. Eles são planejados em camadas, são elas: camada de aplicação, onde são definidos conjuntos de comandos e especificações, camada de transporte, onde são definidos protocolos de transporte e serviços, e camada de interconexão física, que definem conectores, cabeamento, mecanismos de sinalização (COMMITTEE, 2006) (COMMITTEE, 2008b).

3.3.1 SCSI

SCSI é um padrão genérico permite conectar computadores, discos rígidos, *scanners*, leitores ópticos, entre outros dispositivos. A conexão SCSI pode acontecer entre quais quer dois dispositivos SCSI. Um deles é o *iniciador*, o que começa a transmissão, e o outro é o *alvo*, que recebe os comandos e os executa, (ZIELSKI; SOSNOWSKI, 2007).

A Figura 3.2, adaptada de (COMMITTEE, 2008b), mostra o caminho percorrido pelos comandos enviados pelo dispositivo *iniciador* até chegar ao dispositivo *alvo*. O *iniciador* envia um comando, que passa por um protocolo de transporte, de onde é transmitido para o dispositivo *alvo* através de uma conexão física. Ao chegar no dispositivo *alvo* o comando passa por um protocolo de transporte e é então processado e executado por ele.

Note que os protocolos de transporte e interconexões físicas não interferem no comando enviado. Logo, outros padrões podem ser utilizados para transmitir comandos SCSI, como acontece com o padrão ATAPI, onde comandos SCSI são encapsulados e mandados através do SATA ou PATA, protocolos de transporte e de interconexão física ATA.

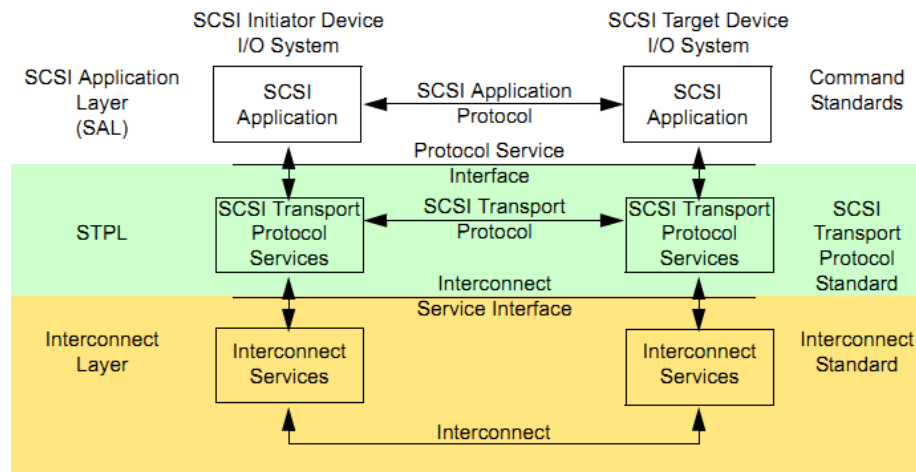


Figura 3.2: SCSI, modelo de camadas

Na Figura 3.3, adaptada de (COMMITTEE, 2008b), a família de padrões SCSI é apresentada (COMMITTEE, 2011). No topo da figura estão o conjunto de comandos primários⁴, compartilhado por todos os tipos de dispositivos, e os conjuntos de comandos específicos, para os diferentes grupos de dispositivo, como comandos de blocos⁵, caso dos HDs, de *stream*⁶, multimídia⁷, controladora⁸, entre outros.

No meio da figura estão os protocolos de transporte, SAS⁹, *fibre channel*¹⁰,

⁴SCSI Primary Commands - 4 (SPC-4), versão mais recente.

⁵SCSI Block Commands - 3 (SBC-3), versão mais recente.

⁶SCSI Stream Commands - 4 (SSC-4), versão mais recente.

⁷MultiMedia Command Set - 6 (MMC-6), versão mais recente.

⁸SCSI Controller Commands-2 (SCC-2), versão mais recente.

⁹SAS Protocol Layer - 3 (SPL-3)

¹⁰Fibre Channel Protocol - 4 (FCP-4), versão mais recente.

Internet SCSI (iSCSI), entre outros. Na base estão as conexões físicas, SAS¹¹, *fibre channel*, internet, **USB! (USB!)**, *PCI express*, entre outros. Já na lateral da figura está o modelo de arquitetura SCSI¹².

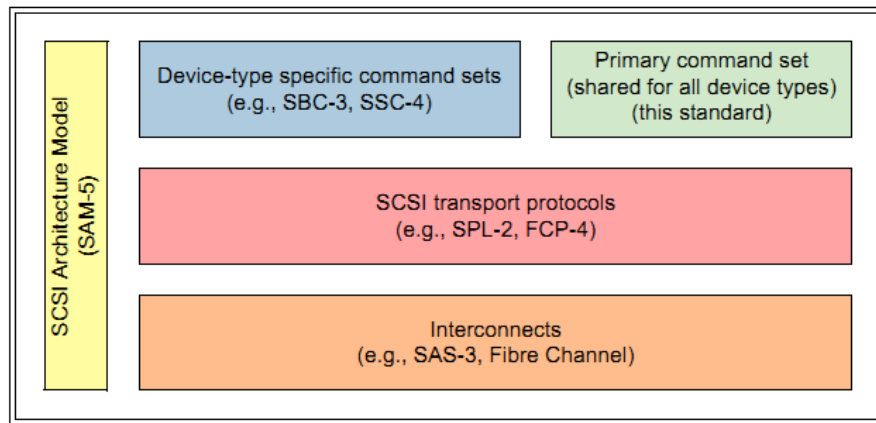


Figura 3.3: Família de padrões SCSI

3.3.2 ATA

O padrão ATA foi desenvolvido para ser uma interface entre computadores e discos rígidos. Inicialmente projetado para conectar placas-mãe e discos rígidos com as novas versões passou a conectar dispositivos ópticos, como leitoras de CD, DVD, e BD, e também fitas magnéticas, através do padrão ATAPI, que permite a transmissão de comandos SCSI por conexões ATA.

O ATA permite comunicar um *host*, que pode ser um *software*, o BIOS ou um *driver* de um sistema operacional, a um dispositivo, que pode ser um disco rígidos ou um leitor óptico, através de um subsistema de entrega. A Figura 3.4, adaptada de (COMMITTEE, 2006), mostra como essa comunicação é realizada. O *host* envia um comando, que passa por um protocolo de transporte, de onde é transmitido para o dispositivo através de uma conexão física. Ao chegar no dispositivo, o comando passa por um protocolo de transporte e é então processado e executado por ele.

Assim como no padrão SCSI, os protocolos de transporte e interconexões físicas não interferem no comando enviado. Logo, outros padrões podem ser utilizados para transmitir comandos ATA, como acontece com o padrão SCSI, que pode

¹¹ *Serial Attached SCSI - 3 (SAS-3)*, versão mais recente.

¹² *SCSI Architecture Model - 5 (SAM-5)*, versão mais recente.

transmitir comandos ATA através de uma camada de tradução¹³, comandos ATA são encapsulados em comandos ATA PASS-THROUGH e mandados através do SATA ou PATA, protocolos de transporte e de interconexão física SCSI.

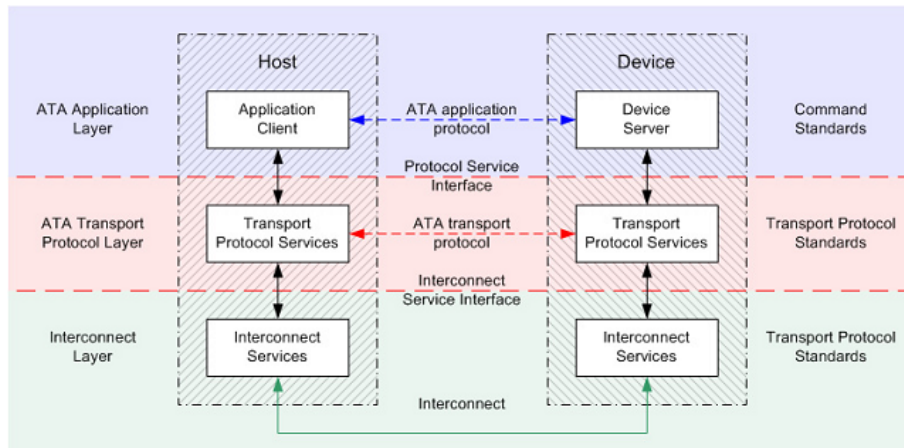


Figura 3.4: ATA, modelo de camadas

O desenvolvimento do ATA é guiado por um modelo abstrato, especificado através de diversos documentos, também chamados de padrões, como descrito na Figura 3.5, adaptada de (COMMITTEE, 2006). Nela tem-se o modelo de arquitetura¹⁴, que define o modelo de sistema e especificações, o conjunto de comandos¹⁵, o conjunto de comandos de entrega, que são documentos SCSI e ATAPI, e os padrões de transporte PATA e SATA.

¹³SCSI-ATA Translation

¹⁴ATA/ATAPI Architecture Model (ATA8-AAM), versão mais recente.

¹⁵ATA/ATAPI Command Set (ATA8-ACS), versão mais recente.

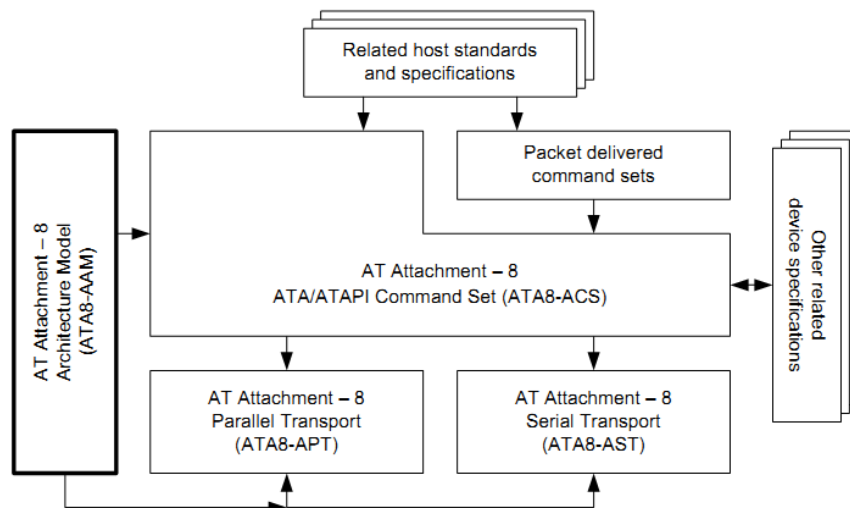


Figura 3.5: Família de padrões ATA

Capítulo 4

Metodologia e Implementação

Este capítulo apresenta a metodologia de desenvolvimento utilizada para este projeto, bem como os detalhes do *framework* proposto e os algoritmos implementados. Na seção 4.1 as estratégias de implementação dos comandos ATA e SCSI são apresentadas. A seção 4.2 detalha os comandos selecionados, bem como a representação do *framework* como diagrama de classes. A seção 4.3 detalha os algoritmos implementados, enquanto a seção 4.4 mostra a descrição da versão inicializável do Linux utilizada.

4.1 ATA, SCSI, SMART e o Linux

No Linux, lidar com dispositivos SCSI e ATA é bastante simples, similar ao trato de arquivos de entrada e saída padrão. Os comandos são enviados aos dispositivos através dos *drivers*, *softwares* que conectam o sistema operacional ao *hardware* do computador (TANENBAUM, 2001).

Os dispositivos correspondem a arquivos no diretório */dev* e utiliza-se chamadas ao sistema para comandá-los. Usa-se a chamada *open()* para criar um descritor para o dispositivo, como mostrado no próximo parágrafo, uma vez criado o descritor, os comandos podem ser escritos e lidos através dele com as chamadas *read()*, *write()* ou *ioctl()*, como descrito em (KINNEY, 2004), (EIWFELDT, 1996).

```
int fd = open (device_name, O_RDWR);
```

Discos SCSI são acessados através de *drivers* como *sd*, para discos rígidos, e *sg*, para quais quer dispositivos SCSI, ambos disponibilizados pelo pacote *sg3-utils*.

Neste trabalho, os dados serão escritos e lidos usando a estrutura `sg_io_hdr_t`, descrita em (*scsi/sg.h*) e a seguir, através do *driver sg*.

`int dxfer_direction`: especifica a direção da transferência de dados.
`unsigned char *dxferp`: ponteiro para os dados transferidos.
`unsigned char *cmdp`: ponteiro para o comando.
`unsigned char *sbp`: ponteiro para o sense buffer.
`unsigned char masked_status`: status GOOD ou CHECK CONDITION.

Os comandos SCSI tem em geral 6, 10, 12 ou 16 bytes de comprimento e são enviados no formato de um bloco descritor de comando, que é um dos campos da estrutura `sg_io_hdr_t` e contem *opcode* do comando, *Logical Block Addressing* (LBA), comprimento em *bytes* a ser transferido e outras flags de comando, como descrito na tabela 4.1, um comando de 10 bytes.

Depois de receber um comando o dispositivo *alvo* responde com um bytes de status good condition e retorna uma estrutura chamada sense buffer com mais informações. Algumas abordagens são tomadas em (KINNEY, 2004), (TAO, 2009), quanto ao uso das chamadas *read()/write()* ou *ioctl*. Neste trabalho foi adotado o uso da chamada *ioctl*, equivale ao uso de uma chamada *write()* seguida de uma chamada *read()*. No caso de um comando que retorne erro, usando *read()/write()*, é necessário o envio de outro comando, o REQUEST SENSE para obter mais informações sobre o problema, com o uso do *ioctl()* a própria estrutura já retorna todas as informações, no campo *sense_buffer*. No próximo parágrafo o envio de um comando por *ioctl()* é descrito, onde *fd* é o descritor do dispositivo, *p_hdr* é uma instanciação da estrutura `sg_io_hdr_t` e *SG_IO* é o parâmetro referente ao *driver sg*.

```
int ret = ioctl(fd, SG_IO, p_hdr);
```

Já com discos SATA/ATA, o processamento é um pouco diferente, pois algumas vezes, as controladoras SATA estão conectadas através de um barramento do tipo PCI, o que leva o sistema operacional à interpretá-lo como um adaptador de barramento¹ SCSI (TOOLBOX, 2008). Isto impossibilita o envio de comandos ATA aos discos, para ter acesso aos discos utiliza-se a camada de tradução SCSI-ATA, que encapsula comandos ATA em comandos SCSI, provida no Linux

¹Host Bus Adapter (HBA).

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB) LOGICAL BLOCK ADDRESS (if required)							
3								
4								
5								
6	Miscellaneous CDB information							
7	(MSB) TRANSFER LENGTH (if required)							
8	PARAMETER LIST LENGTH (if required)							
	ALLOCATION LENGTH (if required)							(LSB)
9	CONTROL							

Figura 4.1: Comando de 10 bytes típico.

pela biblioteca libATA (TROJANOWSKI, 2010), fazendo uso do comando ATA PASS-THROUGH. A camada de tradução é descrita em (COMMITTEE, 2008a) e o comando é detalhado em (STEVENS, 2005). Como o sistema SMART é especificado pelo padrão ATA, os comandos de SMART também são enviados via ATA PASS-THROUGH.

4.2 Framework Proposto

O intuito deste trabalho é dar suporte ao desenvolvimento de testes rápidos de diagnóstico para serem executados em *desktops* e *laptops*. Como dito anteriormente, a capacidade de armazenamento dos discos rígidos vem crescendo fortemente, testar todos os setores de um disco rígido pode levar muitas horas e *softwares* de diagnóstico como PC-Doctor, Aida32 e Hard Disk Sentinel, oferecem testes rápidos, que analisam amostras de setores, que levam de 5 a 10 minutos, e é neste tipo de teste que o *framework* será voltado.

Os algoritmos de teste foram pensados para serem executados em computadores pessoais. Logo, não é apropriado que eles realizem escritas, pois isto poderia corromper arquivos ou o sistema de arquivos do usuário. A maior parte dos computadores pessoais utilizam discos do padrão ATA, especialmente os SATA, por isso também é necessário que os algoritmos realizem comandos ATA, especialmente os comandos SMART, que oferecem auto-testes.

Tendo em vista essas considerações os seguintes comandos SCSI, descritos na tabela 4.1, foram implementados:

Na tabela 4.2 os comandos ATA implementados, enviados via ATA

Comando	Código	Descrição
ATA PASS-THROUGH (16)	85h	Este comando permite enviar comandos ATA através de comandos SCSI
INQUIRY	12h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
LOG SELECT	4Ch	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
LOG SENSE	4Dh	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
MODE SENSE (6)	1Ah	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
READ (10)	28h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
READ (12)	A8h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
READ CAPACITY (10)	25h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
READ CAPACITY (12)	9Eh	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
READ DEFECT DATA (10)	37h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
REASSIGN BLOCKS	07h	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
RECEIVE DIAGNOSTIC RESULTS	1Ch	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
SEND DIAGNOSTIC	1Dh	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
TEST UNIT READY	00h	Testa se a unidade está pronta para receber comandos
VERIFY (10)	2Fh	Permite coletar informações sobre o dispositivo, o fabricante entre outros.
VERIFY (12)	AFh	Permite coletar informações sobre o dispositivo, o fabricante entre outros.

Tabela 4.1: Comandos SCSI selecionados em ordem alfabética

Comando	Código	Descrição
WRITE UNCORRECTABLE	45h	Usado para inserir erros
READ SECTOR	20h	Usado para ler o conteúdo de um ou mais setores
READ SECTOR EXT	24h	Usado para ler o conteúdo de um ou mais setores
WRITE SECTOR	30h	Usado para escrever dados em um ou mais setores
WRITE SECTOR EXT	34h	Usado para ler o conteúdo de um ou mais setores
READ VERIFY SECTORS	40h	Usado para verificar o conteúdo de um ou mais setores
READ VERIFY SECTORS EXT	40h	Usado para verificar o conteúdo de um ou mais setores
EXECUTE DEVICE DIAGNOSTIC	90h	Usado para verificar o conteúdo de um ou mais setores
READ VERIFY SECTORS EXT	40h	Usado para verificar o conteúdo de um ou mais setores

Tabela 4.2: Comandos ATA selecionados em ordem alfabética

PASS-THROUGH, são descritos:

O *framework* é descrito como um diagrama de classes na figura 4.2. Nela vê-se a classe *ScsiCmd* que implementa a estrutura de um comando SCSI, bem como as suas funcionalidades. Várias classes são herdeiras da classe *ScsiCmd*, entre elas a classe *ATAPassThrough*, de onde os comando ATA e SMART herdam suas características. Neste diagrama são descritas apenas as classes que tratam da implementação de comandos.

4.3 Camada da Aplicação

Na camada de aplicação os algoritmos de teste são implementados e também outras funções para auxiliar na execução do teste como funções de listagem de dispositivo, filtragem de *pendrives* e discos rígidos externos, tratamento de *String* e uso de expressão regular, que não serão detalhados aqui.

Os algoritmos implementados são de dois tipos, os SMART e os de busca, que

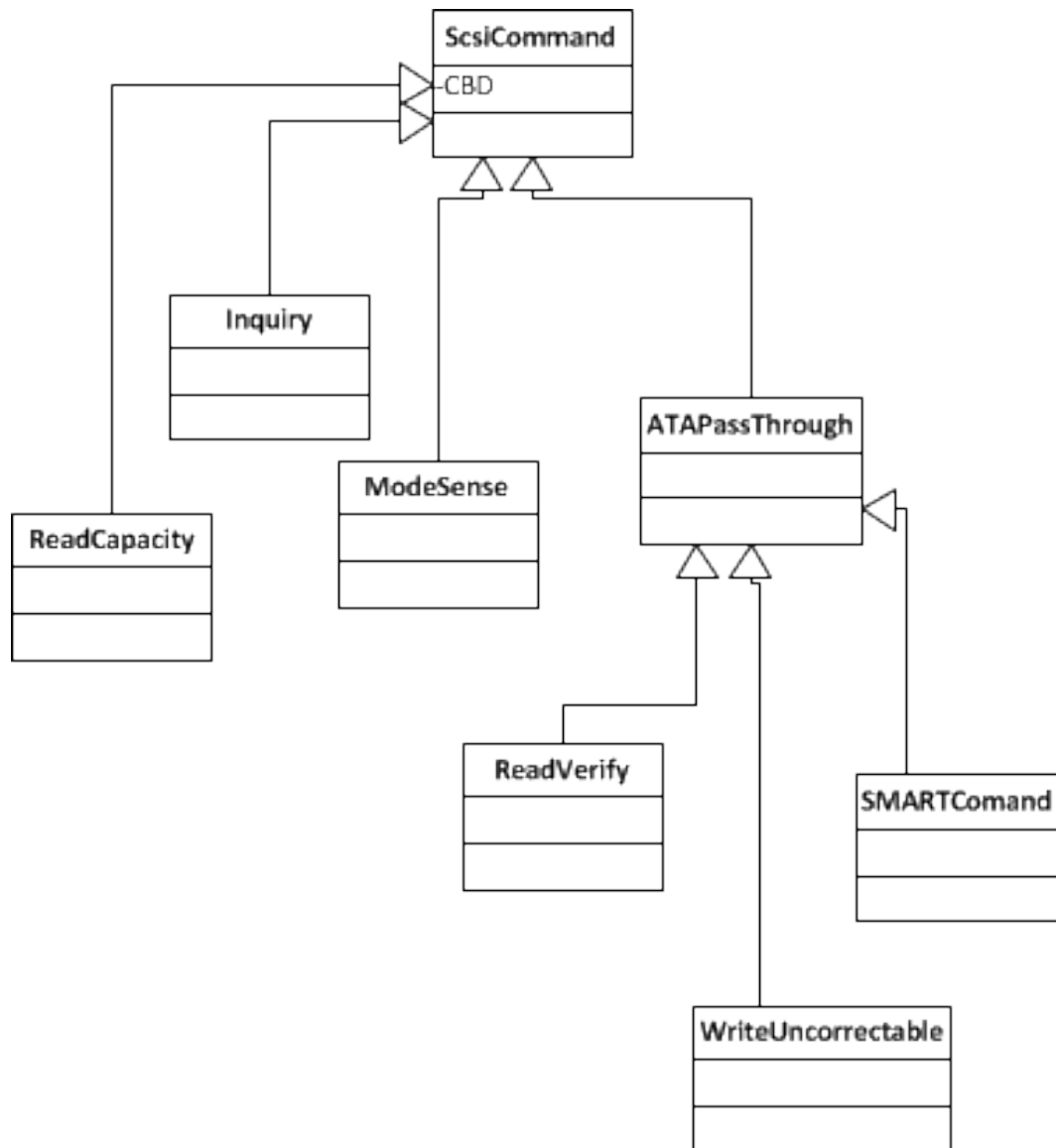


Figura 4.2: Diagrama de Classes do Framework

se baseiam nas descrição dada nos manuais de ajuda pelos *softwares* de diagnóstico, pois como trata-se de uma linha de estudo bastante comercial, há pouca ou nenhuma informação sobre eles.

4.3.1 Algoritmos Implementados

Aqui os algoritmos são descritos a seguir, nele se pelo menos um setor falho for encontrado o algoritmo reportará falha, do contrário o algoritmo terminará com status de sucesso:

- SMART *Status Test* executa o comando SMART *Return Status*.

- ▶ SMART *Short Self-Test* executa o comando SMART *Execute off-line immediate*, com o subcomando *Execute SMART short self-test routine in captive mode*.
- ▶ SMART *Extended Self-Test* executa o comando SMART *Execute off-line immediate*, com o subcomando *Execute SMART extended self-test routine in captive mode*.
- ▶ SMART *Conveyance Self-Test* executa o comando SMART *Execute off-line immediate*, com o subcomando *Execute SMART conveyance self-test routine in captive mode*.
- ▶ *Linear Seek* executa leituras com o comando READ VERIFY de modo que as leituras sejam feitas com o mesmo espaçamento, como descrito na figura 4.3.1.
- ▶ *Funnel Seek* executa leituras com o comando READ VERIFY de modo que as leituras sejam feitas com o em dois sentidos, o algoritmo alternará leituras da menor LBA para a maior e da maior para a menor LBA , como descrito na figura 4.3.1.
- ▶ *Random Seek* executa leituras com o comando READ VERIFY de modo que um percentual do disco seja analisado e as leituras sejam feitas de maneira aleatória, como descrito na figura 4.3.1.
- ▶ *Surface Scan*, executa leituras com o comando READ VERIFY de modo que as leituras sejam feitas com o mesmo espaçamento, para cada passo dado um grupo de setores será analisado, como descrito na figura 4.3.1.
- ▶ *Targeted Read Test*, este teste realiza a leitura do *log* de falhas de leitura da controladora, identifica os setores listados e realiza novas leituras para determinar se foi um *hard* erro, erro irreversível, ou um *soft* erro, apenas temporário, descrito na figura 4.3.1.

4.4 Versão Inicializável

Para criar uma versão inicializável do Linux, utilizou-se uma versão de 64 bits do Debian 6.0 Squeeze. Ele foi criado usando o sistema de arquivos *squashfs*, somente de leitura e que implementa compressão, sendo o mais indicado para

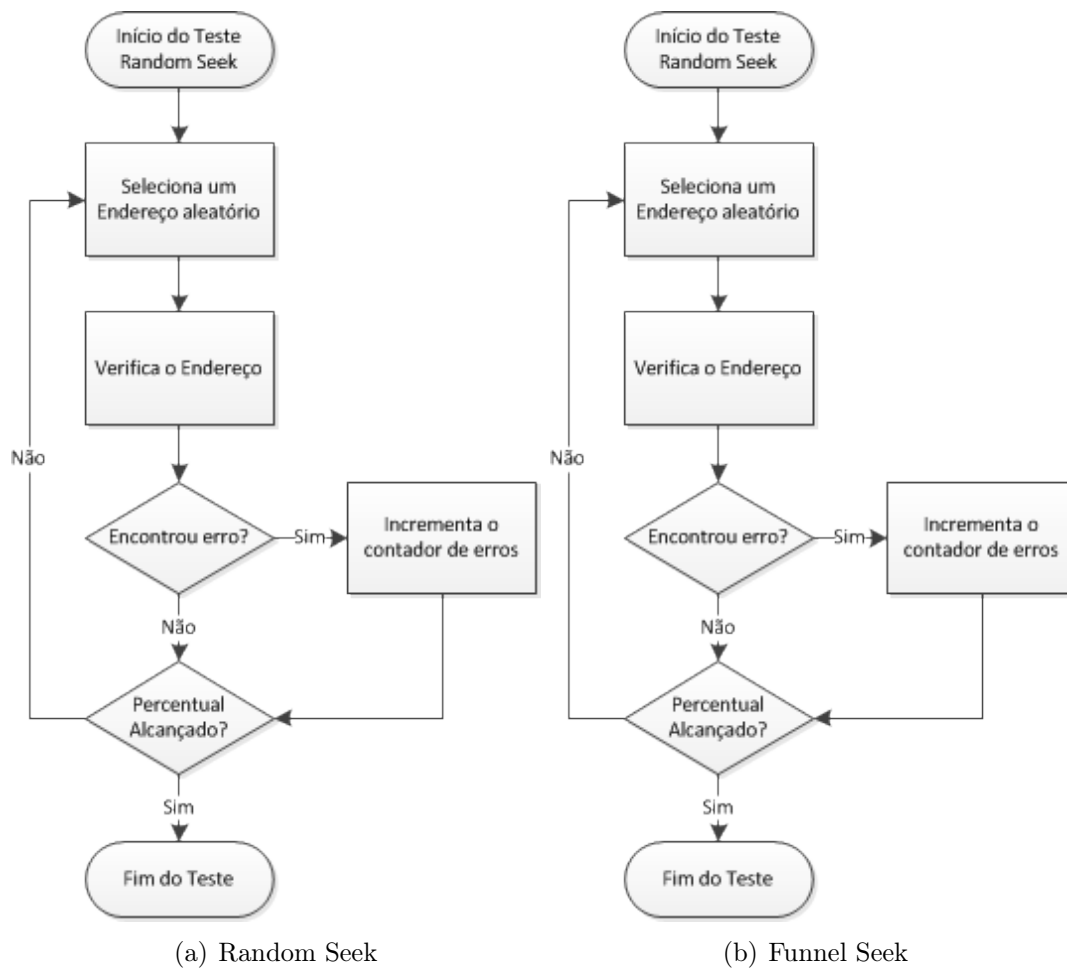


Figura 4.3: Fluxograma dos algoritmos implementados

versões inicializáveis, *live cd* ou *pendrive*. No total este "*bootable*", ocupa cerca de 200 MB e pode ser utilizado em qualquer computador que suporte inicialização por *cd* ou *pendrive*.

O processo de criação e atualização do *bootable* é descrito em (??).

4.5 Resumo do Capítulo

Este capítulo detalhou a implementação do *framework* e algoritmos propostos, bem como as razões pelos quais foram escolhidos. No próximo capítulo os resultados do algoritmos implementados serão comparados com os testes de ferramentas de diagnóstico do mercado.

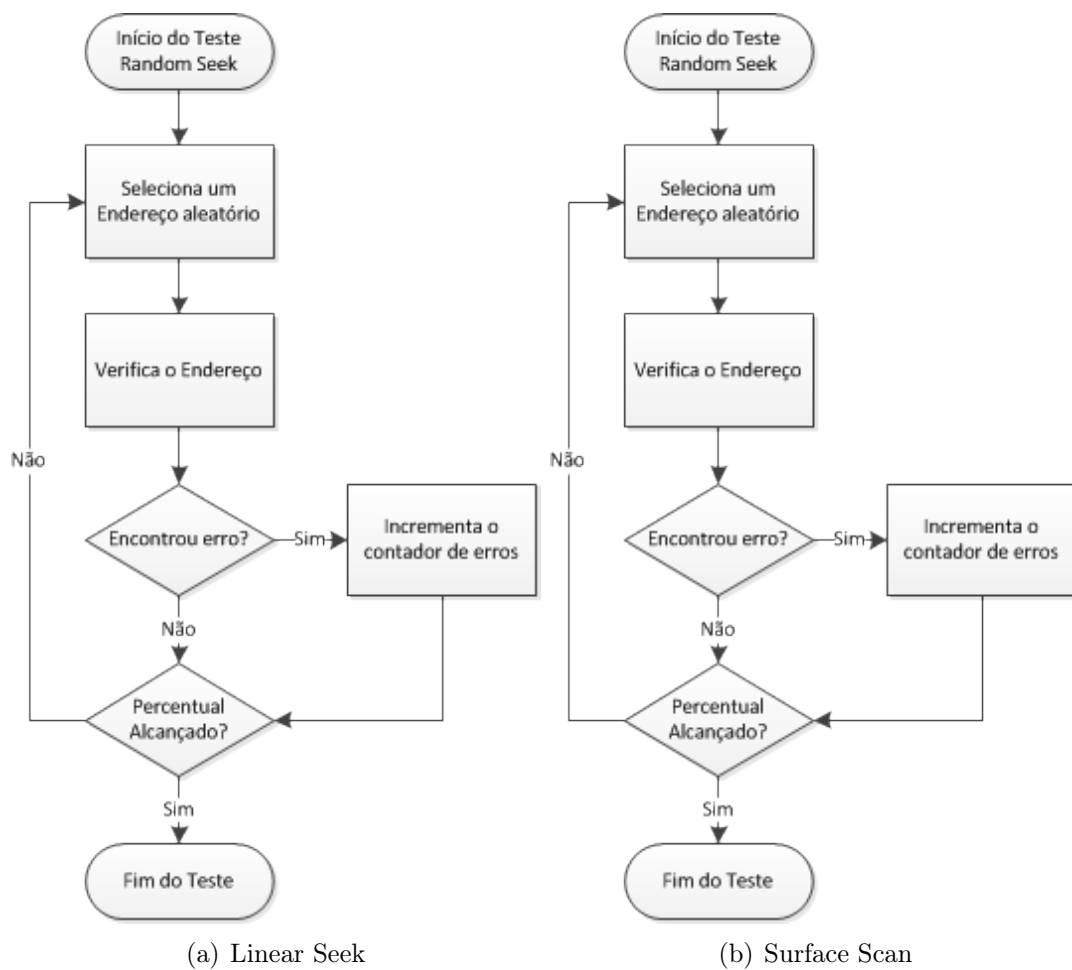
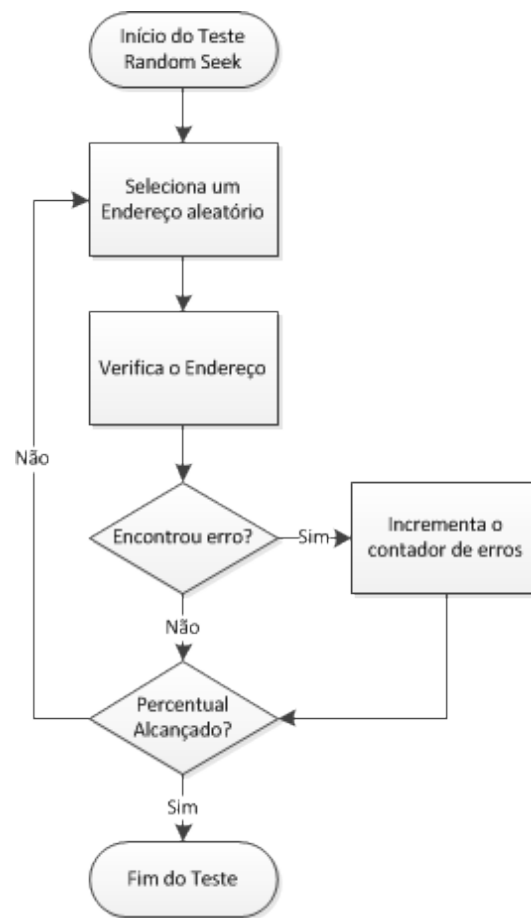


Figura 4.4: Fluxograma dos algoritmos implementados



(a) Targeted Read

Figura 4.5: Fluxograma dos algoritmos implementados

Capítulo 5

Resultados

Os resultados dos testes realizados são descritos nesta seção e resumidamente apresentados na tabela 5.4. Para a realização dos testes foram utilizados 12 discos rígidos, 8 do tipo HDD e 4 do tipo SSD, com defeitos conhecidos. É feita uma comparação entre os resultados obtidos utilizando os algoritmos implementados com o framework proposto, tabela 5.2, e os resultados obtidos com o *software* da PC-Doctor, tabela 5.3. Cada teste descrito nas tabelas é o resultado da maioria dos resultados iguais obtidos em 3 execuções.

5.1 Bootable

Note que a ordem de execução dos algoritmos adotada nos testes do *framework* é diferente da ordem adotada pelo *software* do PC-Doctor. O algoritmo *Targeted Read* é o último a ser executado, e não o segundo como no PC-Doctor, pois ele realiza a leitura dos *logs* de erro, e status dos últimos testes de SMART realizados.

5.2 Testes

A notação utilizada em todas as tabelas de resultados é descrita na tabela 5.1.

O desempenho dos algoritmos SMART é similar nos dois testes, pois trata-se da execução dos mesmos comandos.

Na tabela 5.4, os resultados dos algoritmos do PC-Doctor e do *Framework* são comparados a notação de falha e sucesso nesta tabela consideram todos os resultados, se o dispositivo falhou em pelo menos um teste ele será considerado falho. Nota-se

```

root@jamile-laptop: /home/jamile/Desktop/tcc/jamile/source
File Edit View Search Terminal Help

=====
=== Framework de Diagnóstico de Discos Rígidos ===
=====

          Camada de Aplicação

Algoritmos Disponíveis:

    1 - Smart Status           [status]
    2 - Targeted Read          [target]
    3 - Smart Short Self-Test  [short]
    4 - Extended Self-Test     [extented]
    5 - Conveyance Self-Test   [conv]
    6 - Random Seek Test 1     [random1]
    7 - Random Seek Test 2     [random2]
    8 - Random Seek Test 3     [random3]
    9 - Funnel Seek Test 1     [funnel1]
   10 - Funnel Seek Test 2     [funnel2]
   11 - Linear Seek Test 1     [linear1]
   12 - Linear Seek Test 2     [linear2]
   13 - Surface Scan Test 1    [surface1]
   14 - Surface Scan Test 2    [surface2]

Dispositivos Disponíveis:

    1 - WDC WD1200BEVS-0      [/dev/sdb]

root@jamile-laptop:/home/jamile/Desktop/tcc/jamile/source#

```

Figura 5.1: Benchmark realizado pela PCmark05

```

root@jamile-laptop: /home/jamile/Desktop/tcc/jamile/source
File Edit View Search Terminal Help

root@jamile-laptop:/home/jamile/Desktop/tcc/jamile/source# ./bin/x64/HDD -h

Usage: HDD [HELP] [PARAMETERS]

If HELP argument is used, just print this text.
If PARAMETERS argument are used, execute the test.
If PARAMETERS argument are NOT used, show a list of devices and algorithms available.

HELP:

    -h, --help                Prints this help.

PARAMETERS:

    -d, --device device       Specifies which device will be tested.
    -a, --algorithm algorithm Specifies which algorithm will be executed.

Examples:

    HDD                        : Show devices and algorithms available.
    HDD -d /dev/sdb -a short  : Execute Short Self Test on /dev/sdb device.

root@jamile-laptop:/home/jamile/Desktop/tcc/jamile/source#

```

Figura 5.2: Benchmark realizado pela PCmark05

que dois dispositivos só foram detectados pelo *framework*.

Abreviação / Símbolo	Definição
×	Teste falhou
✓	Teste passou
∅	Teste não pode ser executado
Short	SMART <i>Short Self-Test</i>
Status	SMART <i>Status Test</i>
Conv	SMART <i>Conveyance Self-Test</i>
Linear	<i>Linear Seek Test</i>
Random	<i>Random Seek Test</i>
Funnel	<i>Funnel Seek Test</i>
Surface	<i>Surface Scan Test</i>
Target	<i>Targeted Read Test</i>

Tabela 5.1: Notação utilizada nas tabelas de resultados

Dispositivo	Status	Short	Linear	Random	Funnel	Surface	Conv	Target
HDD 01	×	×	×	✓	×	✓	✓	×
HDD 02	✓	✓	✓	✓	✓	✓	✓	✓
HDD 03	✓	✓	✓	✓	✓	✓	✓	✓
HDD 04	✓	×	✓	✓	×	✓	✓	×
HDD 05	×	×	×	✓	×	✓	✓	×
HDD 06	✓	✓	✓	✓	✓	✓	✓	×
HDD 07	✓	×	✓	×	×	✓	✓	×
HDD 08	✓	✓	✓	✓	✓	✓	✓	✓
SSD 01	✓	×	✓	✓	×	✓	✓	×
SSD 02	✓	✓	✓	✓	✓	✓	✓	✓
SSD 03	✓	✓	✓	×	✓	✓	✓	×
SSD 04	✓	✓	✓	✓	✓	✓	✓	✓

Tabela 5.2: Resultados Framework

Dispositivo	Status	Target	Random	Funnel	Short	Surface-1	Surface-2
HDD 01	×	×	×	✓	×	✓	✓
HDD 02	✓	✓	✓	✓	✓	✓	✓
HDD 03	✓	✓	✓	✓	✓	✓	✓
HDD 04	✓	×	✓	✓	×	✓	✓
HDD 05	×	×	×	✓	×	✓	✓
HDD 06	✓	✓	✓	✓	✓	✓	✓
HDD 07	✓	×	✓	×	×	✓	✓
HDD 08	✓	✓	✓	✓	✓	✓	✓
SSD 01	✓	×	✓	✓	×	✓	✓
SSD 02	✓	✓	✓	✓	✓	✓	✓
SSD 03	✓	✓	✓	×	✓	✓	✓
SSD 04	✓	✓	✓	✓	✓	✓	✓

Tabela 5.3: Resultados PC-Doctor

Dispositivo	Modelo	Capacidade	Form Factor PC-Doctor	F
HDD 01	Seagate ST3500418AS	500 GB	3.5"	
HDD 02	Western Digital WD2500AAKX083CA0	250GB	3.5"	
HDD 03	Seagate ST31000528AS	1 TB	✓	
HDD 04	Seagate ST3500418AS	500 GB	×	
HDD 05	Hitachi HTS725050A9A364	500 GB	×	
HDD 06	Hitachi HTS545032B9A300	320 GB	✓	
HDD 07	Seagate ST9250315AS	250 GB	✓	
HDD 08	Toshiba MK5061GSY	500GB	✓	
HDD 09	Hitachi HTS723232A7A364	320 GB	×	
SSD 02	Kingston SV100S264G	64 GB	✓	
SSD 03	Intel	160 GB	×	
SSD 04	Intel	160 GB	✓	

Tabela 5.4: Comandos SCSI selecionados em ordem alfabética

Capítulo 6

Conclusão

Neste trabalho é realizada a implementação de um *framework* para dar suporte ao desenvolvimento de testes de diagnóstico de discos rígidos. Utilizou-se o sistema Linux, pois mesmo sendo utilizado por uma pequena parcela do mercado possibilita a realização de testes em computadores com qual quer sistema operacional instalado, bastando apenas a reinicialização.

Os algoritmos sugeridos mostraram bons resultados, pois detectaram mais falhas que os das ferramentas de mercado com o mesmo tempo a que se propos, de 5 a 10 minutos, variando de acordo com a capacidade de armazenamento. Mesmo não sendo o foco deste trabalho é possível realizar uma análise quanto a eficiência destes algoritmos.

Nota-se, com base nas tabelas do capítulo 5, que mesmo em dispositivos com o falha o algoritmo de *Surface Scan* não foi capaz de detectar qual quer falha, foi o de pior resultado. Logo melhorias devem ser propostas para melhorá-lo, como por exemplo diminuir a janela de execução ou concentrar as verificações nos primeiros setores do disco, pois por causa da tabela de partição, que normalmente utiliza MBR¹, os primeiros setores tendem a sofrer um desgaste maior.

Os algoritmos de melhores desempenho foram o SMART *short self-test* e o *Targeted Read* pois detectaram a maior quantidade de dispositivos defeituosos, vale ressaltar que o *Targeted Read* tem seu desempenho afetado pela ordem em que é executado, no PC-Doctor, por exemplo, ele é o segundo a ser executado e detectou menos falhas do que o implementado utilizando o *framework*.

1

A grande contribuição deste trabalho está em fornecer os recursos necessários para a implementação de algoritmos de teste de maneira simples, facilitando o análises de desempenho e implementação de novos algoritmos.

6.1 Perspectivas Futuras

Como uma proposta futura tem-se a implementação de mais comandos SCSI para abranger também testes servidores e discos SAS. Pretende-se também focar no desenvolvimento de novos algoritmos, utilizando fundamentos de inteligência artificial, algoritmos genéticos e reconhecimento de padrões, visando uma maior eficiência dos testes de discos rígidos.

Referências Bibliográficas

ARCHIVES, I. *Magnetic Disk*. 1970. Disponível via Internet.

Acessado em: 15 de Outubro de 2011. Disponível em:

<http://www-03.ibm.com/ibm/history/exhibits/storage/storage_PH4-15.html>.

BEZ, R. *et al.* Introduction to flash memory. *Proceedings of the IEEE*, v. 91, n. 4, p. 489 – 502, abril 2003. ISSN 0018-9219.

CARRIER, B. *File system forensic analysis*. Boston, Mass. London:

Addison-Wesley, 2005. ISBN 0321268172.

COMMITTEE, I. C. for I. T. S. I. T. T. *AT Attachment 8 - ATA/ATAPI Architecture Model(ATA8-AAM)*. Projeto t13/1700-d. [S.l.], Maio 2006.

COMMITTEE, I. C. for I. T. S. I. T. T. *AT Attachment 8 - ATA/ATAPI Command Set (ATA8-ACS)*. Projeto t13/1699-d. [S.l.], Maio 2007.

COMMITTEE, I. C. for I. T. S. I. T. T. *SCSI / ATA Translation - 2 (SAT-2)*. Projeto t10/1826-d. [S.l.], Junho 2008.

COMMITTEE, I. C. for I. T. S. I. T. T. *SCSI Architecture Model - 5 (SAM-5)*. Projeto t10/2104-d. [S.l.], Setembro 2008.

COMMITTEE, I. C. for I. T. S. I. T. T. *T10 Working Drafts*. 2011. Acessado em 15 de Agosto de 2011. Disponível em: <<http://www.t10.org/drafts.htm>>.

COMPONENTS, I. T. A. E. *Solid State Drives, Separating myths from facts*. Junho 2009. Disponível via Internet. Acessado em 04 de Novembro de 2011. Disponível em: <http://www.toshiba.com/taec/components/Generic/SSD_Myths.pdf>.

CORPORATION, D. A. *Hard Disk Sector Structures*. Dezembro 2001. Disponível via Internet. Acessado em 02 de Novembro de 2011. Disponível em: <<http://www.dewassoc.com/kbase/hard-drives%2Fhard-disk-sector-structures.htm>>.

EIWFELDT, H. *The Linux SCSI programming HOWTO*. Maio 1996. Disponível via Internet. Acessado em 20 de Maio de 2011. Disponível em: <<http://tldp.org/HOWTO/archived/SCSI-Programming-HOWTO/>>.

HARKER, J. M. *et al.* A quarter century of disk file innovation. *IBM Journal of Research and Development*, v. 25, n. 5, p. 677 –690, sep. 1981. ISSN 0018-8646.

HUGHES, G. *et al.* Improved disk-drive failure warnings. *Reliability, IEEE Transactions on*, v. 51, n. 3, p. 350 – 357, sep 2002. ISSN 0018-9529.

INTEL. *What are the advantages of TRIM and how can I use it with my SSD?* Setembro 2010. Disponível via Internet. Acessado em 04 de Novembro de 2011. Disponível em: <[http://www.intel.com/support/ssdc/hpssd%2Fsb/CS-031846.htm?wapkw=\(TRIM\)](http://www.intel.com/support/ssdc/hpssd%2Fsb/CS-031846.htm?wapkw=(TRIM))>.

ISERMANN, R. *Fault-diagnosis applications : model-based condition monitoring : actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Heidelberg New York: Springer, 2011. ISBN 9783642127663.

KINNEY, M. H. A. *Generic SCSI Command Generator*. Dissertação — Worcester Polytechnic Institute, Março 2004. Disponível em: <<http://web.cs.wpi.edu/~claypool/mqp/gd-scsi/final.pdf>>.

KUNETT, V. *et al.* An in-system reprogrammable 256k cmos flash memory. In: *Solid-State Circuits Conference, 1988. Digest of Technical Papers. ISSCC. 1988 IEEE International*. [S.l.: s.n.], 1988. p. 132.

LEE, P.; ANDERSON, T. *Fault tolerance, principles and practice*. Wien New York: Springer-Verlag, 1990. ISBN 0387820779.

MAMUN, A. *Hard disk drive : mechatronics and control*. Boca Raton, FL: CRC Press, 2007. ISBN 9780849372537.

MASON, H. Scsi, the industry workhorse, is still working hard. *Computer*, v. 33, n. 12, p. 152 – 153, dec 2000. ISSN 0018-9162.

MASUOKA, F. *et al.* A new flash e2prom cell using triple polysilicon technology. In: *Electron Devices Meeting, 1984 International*. [S.l.: s.n.], 1984. v. 30, p. 464 – 467.

MCLEAN, P. *Proposal for a Selective Self-test*. Outubro 2001. Disponível via Internet. Acessado em 15 de setembro de 2011. Disponível em: <<http://www.t10.org/t13/technical/e01139r0.pdf>>.

MORIMOTO, C. *Hardware - O Guia Definitivo*.
SULINA, 2007. ISBN 9788599593103. Disponível em:
<<http://books.google.com/books?id=P0ZDbwAACAAJ>>.

MORIMOTO, C. *Hardware, V.2 - O Guia Definitivo*.
SULINA, 2010. ISBN 9788599593165. Disponível em:
<<http://books.google.com/books?id=vPpBbwAACAAJ>>.

MORIMOTO, C. E. *Ciclos de Gravação e a questão da longevidade*. Outubro 2010. Disponível via Internet. Acessado em 20 de Outubro de 2011. Disponível em: <<http://www.hardware.com.br/tutoriais/entendendo-ssd/ciclos-gravacao-longevidade.html>>.

PARRISH, K. *Seagate Launching "Industry's First" 4TB HDD*. Setembro 2011. Disponível via Internet. Acessado em 02 de Novembro de 2011. Disponível em: <<http://www.tomshardware.com/news/GoFlex-Desk-4TB-3.5-inch-industrial-design-USB-3.0,13371>>.

POTTS, P. R. *SixHardDriveFormFactors.jpg*. Março 2008. Disponibilizada via Internet. Six hard disk drives with cases opened showing platters and heads; 8, 5.25, 3.5, 2.5, 1.8 and 1 inch disk diameters are represented. Disponível em: <<http://en.wikipedia.org/wiki/File:SixHardDriveFormFactors.jpg>>.

RENT, T. M. *Origin of Solid State Drives*. Março 2010. Disponível via Internet. Acessado em 12 de Outubro de 2011. Disponível em: <<http://www.storagereview.com/origin-solid-state-drives>>.

RENT, T. M. *SSD Architecture*. Abril 2010. Disponível via Internet. Acessado em 20 de Agosto de 2011. Disponível em: <http://www.storagereview.com/ssd_architecture>.

RICHARDS, M. *Card Capacitor Read Only Store (CCROS) for microprogram*. 1965. Acessado em 22 de Outubro de 2011. Disponível em: <<http://www.computerhistory.org/revolution/memory-storage/8/264/1103>>.

- SEAGATE. *The Case for Solid-State Hybrid Drives*. 2011. White Paper. Acessado em 14 de Setembro de 2011. Disponível em: <http://www.seagate.com/docs/pdf/corporate/seagate_view_solid_state_hybrid_drive_gen.pdf>.
- SHIMPI, A. L. *OCZ Z-Drive R4 CM88 (1.6TB PCIe SSD) Review*. Setembro 2011. Disponível via Internet. Acessado em 02 de novembro de 2011. Disponível em: <<http://www.anandtech.com/show/4879/ocz-zdrive-r4-cm88-16tb-pcie-ssd-review>>.
- STEVENS, C. E. *ATA Command Pass-Through*. 04-262r8a. ed. [S.l.], Março 2005.
- STEVENS, L. D. The evolution of magnetic storage. *IBM Journal of Research and Development*, v. 25, n. 5, p. 663 –676, sep. 1981. ISSN 0018-8646.
- STROM, B. *et al.* Hard disk drive reliability modeling and failure prediction. *Magnetics, IEEE Transactions on*, v. 43, n. 9, p. 3676 –3684, sept. 2007. ISSN 0018-9464.
- SYSADM, H. *Solid State Drives, some theory and a selection of videos*. Julho 2011. Disponível via Internet. Acessado em 20 de Agosto de 2011. Disponível em: <<http://www.happysysadm.com/search?q=ssd>>.
- TANENBAUM, A. *Modern operating systems*. Upper Saddle River, N.J: Prentice Hall, 2001. ISBN 0130313580.
- TAO, M. Y. *Tour the Linux generic SCSI driver*. Fevereiro 2009. Disponível via Internet. Acessado em 8 de maio de 2011. Disponível em: <<http://www.ibm.com/developerworks/linux/library/l-scsi-api/index.html>>.
- TOOLBOX, L. S. *SCSI TOOLBOX, LLC - Using SAT to Access SATA drives*. Agosto 2008. Disponível via Internet. Acessado em 28 de Julho de 2011. Disponível em: <<http://scsitools.com/pdfs/UsingSAT.pdf>>.
- TROJANOWSKI, B. *ATA messages via SCSI Layer*. Janeiro 2010. Disponível via Internet. Acessado em 02 de Novembro de 2011. Disponível em: <<http://www.jukie.net/bart/blog/ata-via-scsi>>.
- VASCONCELOS, L. *Hardware Total*. [S.l.]: MAKRON BOOKS, 2002.
- WESZ, R. *Falha, erro ou defeito?* Junho 2007. Disponível via Internet. Acessado em 20 de Outubro de 2011. Disponível em: <<http://testsnews.wordpress.com/2007/06/24/falha-erro-ou-defeito/>>.

ZHAO, X. *et al.* A novel method to restrain the radial error propagation in self-servowriting in hard disk. In: *Automation and Logistics, 2007 IEEE International Conference on*. [S.l.: s.n.], 2007. p. 2736 –2739.

ZIELSKI, S.; SOSNOWSKI, J. The scsi interface conformance tests generation. In: *Dependability of Computer Systems, 2007. DepCoS-RELCOMEX '07. 2nd International Conference on*. [S.l.: s.n.], 2007. p. 360 –367.