

Sistemas Operacionais

4º período

Professora: Michelle Hanne

Sistema de Arquivo

O exFAT, introduzido em 2006, é outro sistema de arquivos criado pela Microsoft. O exFAT é usado, principalmente, em dispositivos de mídia portáteis, como pen drives e cartões de armazenamento. O exFAT suporta oficialmente dispositivos de armazenamento de mídia portáteis com até 512 Terabytes ($2^{40} + 2^{10} = 2^{50}$).

- Mas, teoricamente, poderia suportar unidades de até 64 Zebibytes (2^{70}).

O sistema exFAt possuía como recurso o suporte a nomes grandes de arquivos até 255 caracteres e suporte para até 2.796.202 arquivos por diretório. O sistema de arquivos exFAT é suportado por quase todas as versões do Windows (versões mais antigas com atualizações opcionais), Mac OS X (10.6.5+), bem como em muitos dispositivos de TV, mídia e outros.

O **New Technology File System (NTFS)** foi desenvolvido inicialmente para os sistemas operacionais de rede da **Microsoft, Windows NT**. Posteriormente foi adotado como sistema de arquivos padrão de todos os produtos Microsoft, tanto a linha de **redes Windows 2000, 2003 e 2012** quanto a linha pessoal **Windows XP, Vista, 7, 8, 10** (ZACKER, 2015).

Um sistema de arquivos **NTFS** possui **nomeação para arquivos e pasta até 255 caracteres**. Com relação ao caminho absoluto, esse suporte é de aproximadamente **32.000 caracteres**, desde o diretório raiz até o arquivo destino, como exemplo:

“c:/users/aluno/desktop/musica/sertanejo/sp/musica.mp3”.

O sistema de codificação é **UNICODE (ZACKER, 2015)**. Isso permite que países de línguas não latinas (Japão, Israel, etc.) possam gravar **nomes de arquivos em suas línguas nativas**. No Windows, não há diferença entre letras maiúsculas e minúsculas.

O sistema NTFS possui vantagens significantes com relação aos seus antecessores, como:

- **Melhoria de confiabilidade:** possui sistema de tolerância a falhas que faz reparo em erros de disco automaticamente sem exibir mensagem de erro.
- **Registros detalhados de transação:** mantém registros detalhados de transações, que rastreiam erros no disco rígido. Possibilita a recuperação de arquivos se o disco rígido falhar.
- **Permissões de acesso:** permite que permissões (como leitura, gravação e execução) sejam definidas para diretórios e arquivos individuais.
- **NTFS remove as limitações impostas pelo sistema FAT**, aumentando o tamanho dos volumes e arquivos:
 - pode suportar, na teoria, discos rígidos até pouco menos de 16 Exabytes (2^{18}) (ZACKER, 2015). O tamanho do arquivo individual é limitado a pouco menos de 256 Terabytes (2^{12}), no Windows 8, Windows 10 e versões mais recentes do Windows Server.

O sistema NTFS possui vantagens significantes com relação aos seus antecessores, como:

- **Cotas:** As cotas de uso de disco são definidas por um administrador para restringir a quantidade de espaço em disco que um usuário pode ocupar, muito utilizado em redes de computadores em que o espaço é compartilhado.
- **Atributos de arquivo anteriormente não vistos nos sistemas operacionais Windows,** como o atributo compactado e o atributo indexado, estão disponíveis em unidades formatadas com NTFS.
- **Criptografia:** suporta sistema de arquivos com criptografia (EFS). O EFS fornece criptografia em nível de arquivo e pastas.
- **Master File Table (MFT):** os dados de arquivo, diretório e metarquivo - nome do arquivo, data de criação, permissões de acesso e tamanho - são armazenados como metadados na tabela de arquivos mestre (MFT).

Sistema de Arquivo NTFS

Exemplo de uma *Master File Table* NTFS

7	\$Bitmap Mapa de <i>bits</i> de usuário
6	\$ Diretório Raiz
5	\$AttrDef Definições de atributo
	\$Volume Arquivo de volume
4	\$LogFile Arquivo de registro de eventos para recuperação
3	\$MtfMirr Cópia espelho da Mtf
	\$Mtf Tabela de arquivo mestre
2	
1	
0	

Essa tabela é linear com registro de tamanho fixo de *1kByte*. Ele em si, é uma arquivo e pode ter seu crescimento até 248 registros.

O sistema NTFS possui vantagens significantes com relação aos seus antecessores, como:

- **Hard links:** permite que diferentes nomes de arquivos se refiram diretamente ao mesmo conteúdo do arquivo. O hard links pode se vincular apenas a arquivos no mesmo volume, porque cada volume tem sua própria Master File Table (MFT).
- **Metafiles:** o NTFS possui diversos tipos de arquivos que permitem organizar o sistema de arquivos, como o \$Volume, que contém informações sobre o volume, ou seja, o identificador do objeto de volume, o rótulo do volume, a versão do sistema de arquivos, etc.
- **Tempo:** todos os timestamps NTFS estão em UTC (ZACKER, 2015).

Sistema de Arquivo NTFS



PUC Minas

No Windows, as FATs (*File Allocation Table*) e o NTFS determinam como os arquivos são nomeados, armazenados e organizados em um volume. Um sistema de arquivos gerencia arquivos e pastas e as informações necessárias para localizar e acessar esses itens por usuários locais e remotos dentro de um volume (ZACKER, 2015).

O volume pode ser particionado, ou seja, dividido em áreas distintas no mesmo volume, e todas as informações que descrevem a partição estão contidas na tabela de partição. **Um desses campos é o tipo de sistema de arquivo.**

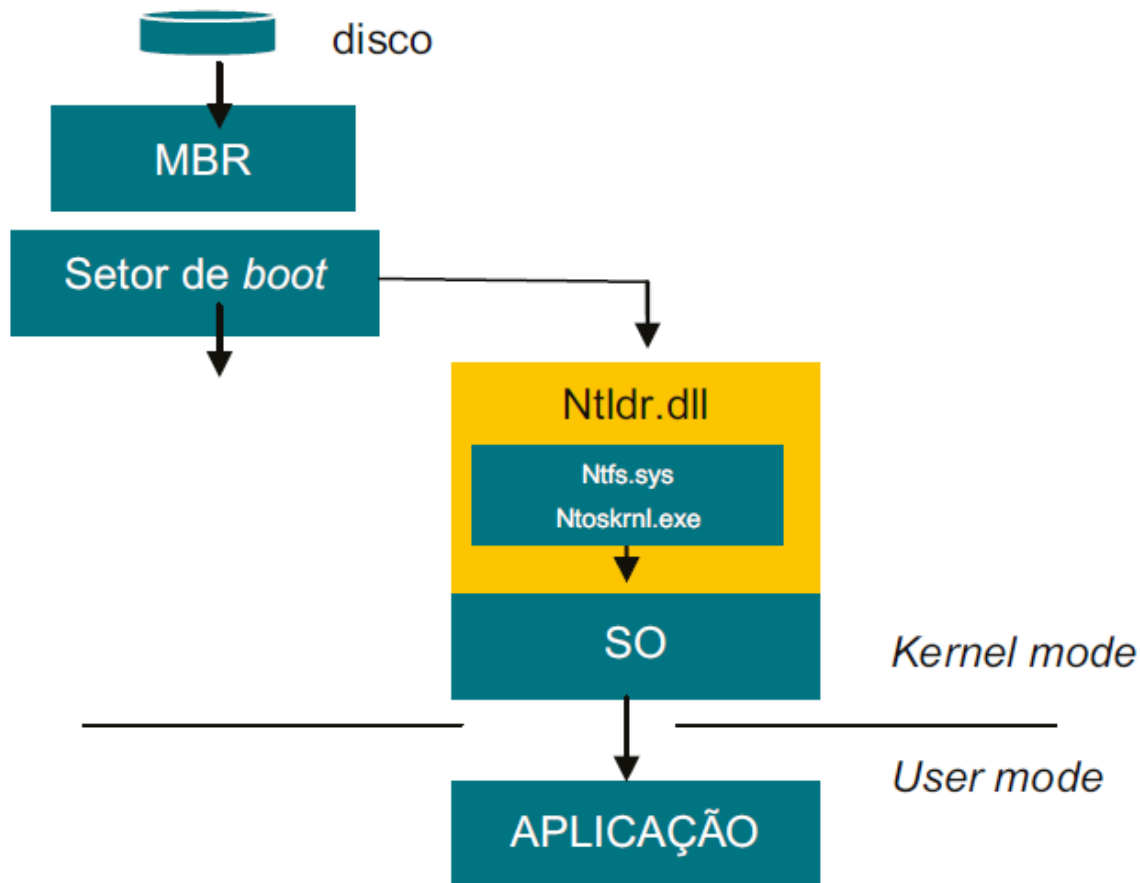
Para volumes/partições NTFS, é 0x07. Essa tabela de partição fica armazenada no início do disco rígido. **O volume/partição principal de disco, no setor zero da trilha zero de um HD**, armazena informações sobre Master Boot Record (MBR), os arquivos de inicialização do sistema, como a tabela de partição do disco.

Esse MBR é criado durante o processo de **formatação e configuração do volume do disco**. Na inicialização, o setor de MBR executa o código de inicialização principal e, posteriormente, transfere o controle para o setor de inicialização no disco, permitindo que **o computador execute a inicialização do sistema operacional no sistema de arquivos daquele volume específico.**

Sistema de Arquivo NTFS



PUC Minas



- O disco pode ter uma ou mais partições. MBR contém o código executável que o Basic Input Output System (BIOS) do sistema carrega na memória. No setor de boot, as informações sobre o layout do volume de disco (um nome de um grupo de segmentos físicos que aparecem no sistema como um dispositivo único e lógico), as estruturas do sistema de arquivos e o código de inicialização que carrega o Ntldr (**biblioteca responsável por carregar o sistema operacional**) são armazenados.
- A biblioteca altera para modo protegido a CPU, faz a iniciação do sistema e, em seguida, faz a leitura do conteúdo do **arquivo boot.ini** (ZACKER, 2015). O **boot.ini** contém duas partes: **boot loader**, seção que contém informações sobre o boot, por exemplo, o tempo de espera para inicialização, timeout; e o operating system, que compreende as entradas de um ou mais sistemas operacionais.

Sistema de Arquivo NTFS

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP
Professional"/fastdetect
C:\CMDCONS\BOOTSECT.DAT="Microsoft Windows Recovery Console" /cmdcons
```

Um sistema NTFS permite suporte a volumes grandes.

Tamanho do <i>cluster</i>	Maior volume	Arquivo maior
4 KB (tamanho padrão)	16 TB	16 TB
8 KB	32 TB	32 TB
16 KB	64 TB	64 TB
32 KB	128 TB	128 TB
64 KB (tamanho máximo)	256 TB	256 TB

Como no Windows, o sistema operacional Linux contém um sistema de arquivos. O Linux organiza essas informações em forma de arquivos, e esses arquivos podem conter códigos-fonte, textos, script Shell ou qualquer outro tipo de informação (NEGUS, 2014).

O **volume em Linux** é representado na forma de diretório, **ao contrário do Windows, que tem a representação por letra**. Sendo assim, a estrutura de diretório é definida no momento da montagem, tais como: /media/cdrom, /media/cd, /mnt/ide, etc.

A nomeação de arquivos no Linux está no formato **8.3 (oito caracteres de letras e três de extensão)**. Pode-se, como no Windows, ter arquivos com 255 caracteres de nomes, mas pode-se ter mais de um ponto, tais como **jdk-7u71-linux-x64.gz, hadoop-2.4.1.tar.gz**, etc. Existe diferença entre letras maiúsculas e minúsculas, ou seja, home é diferente de HOME, que é diferente de Home.

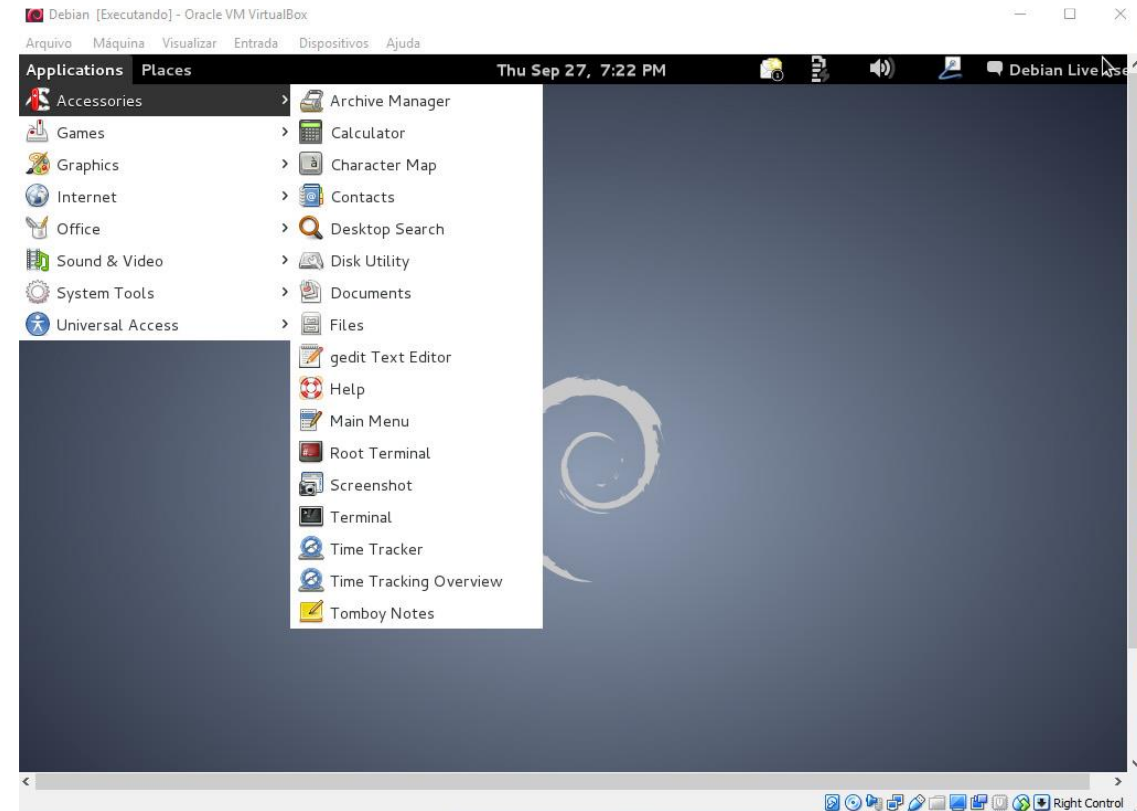
Linux

Alguns caracteres não podem ser usados em nomes de arquivos, tais como ! @ # \$ % ^* () { } " ' ? , etc.

Não há extensões compulsórias, a exemplo de .com e .exe, para os arquivos de aplicativos (programas), ou .bat (NEGUS, 2014).

Arquivos executáveis são marcados com "*" no final de arquivo e os arquivos de backups são marcados com "~" no final. Diretórios, que também são arquivos no Linux, são marcados com "/".

No prompt de comando do Linux, visualizam-se as informações com a digitação do comando `ls -F`.



Linux – Tipos de Arquivos

```
root@ubuntu-server:~/bin# ls -F
bash*      dir*      lesspipe*  open@     tempfile*
bunzip2*   dmesg*    ln*        openot*   touch*
busybox*   dnsdomainname* loadkeys*  pidof@    true*
bzipcat*   domainname* login*     ping*     ulockmgr_server*
bzcmp@     dumpkeys* ls*        ping6*    umount*
bzdiff*    echo*     lsnod*    plymouth* uname*
bzegrep@   ed*       nkdir*    ps*       uncompress*
bzeze*     egrep*    nknd*     h@        unicode_start*
bzfgrep@   false*    nktmp*    link*     vdir*
bzgrep*    fgconsole* more*      link*     which*
bzip2*     fgrep*    mount*    rn*       ypdomainname*
bzip2recover* fuser*    mountpoint* rndir*    zcat*
bzless@    fusermount* nt@        rnano@    zcmp*
bzmore*    gz*       nt-gnu*   run-parts* zdiff*
cat*       gzip*     nv*       sed*       zegrep*
chgrp*     hostname* nc@        setfont*  zfgrep*
chmod*     ip*       nc.openbsd* setupcon* zforce*
chown*     kill*     netcat@   sh@        zgrep*
cp*        kbd_mode* netstat*   sh.distrib* zless*
cpio*      kill*     nisdomainname* sleep*     znore*
dash*      less*     ntfs-3g*  stty*      znew*
date*      lessecho* ntfs-3g.probe* sync*
dd*        lessfile@ ntfs-3g.secaudit* tailf*
df*        lesskey*  ntfs-3g.usermap* tar*
```

Arquivos com "@" são links simbólicos

Arquivos com "*" são executáveis

Visualizando arquivos no prompt do Linux: comando ls -F

Há diferentes grupos de arquivos no sistema operacional Linux:

- **Arquivos comuns:** são os arquivos com base na tabela ASCII e outros não baseados. Exemplo: shell scripts e arquivos binários executáveis.
- **Diretórios:** são áreas lógicas de organização de conteúdo que contêm arquivos e subdiretórios.
- **Links:** é um arquivo que aponta para um arquivo em outro diretório ou um outro diretório.
- **Device file (arquivos de dispositivos):** representam dispositivos de hardware do computador com CD, DVD, HD. No Linux, tudo é um arquivo, como /dev/had, /dev/fd0.
- **Sockets:** são utilizados na comunicação entre processos.
- **Pipes:** usados também na intercomunicação entre processos.

Linux – Sistema de Arquivos

O Linux pode implementar diversos sistemas de arquivos, como **ext**, **reiserfs**, **ufs**, **iso9600 (CD-ROM)**, etc. Porém, o **Extension File System (ext)** foi adotado como padrão. O sistema de arquivo **ext** foi lançado no ano de 1992 para substituir **Minix**. Disponível em <http://www.minix3.org/doc/faq-brazilian-portuguese.html>

O **Minix** era o sistema de arquivos utilizado pelo Linux (NEGUS, 2014). Esse novo sistema suportava 2 GB de tamanho de sistema, arquivos com nomes de até 255 caracteres e tamanho de 2 GB, no máximo, e causava com, o uso intensivo, muita fragmentação.

Ext2: foi lançado em 1993 com o intuito de correção de algumas características do **ext** que causavam limitação do sistema, como limitação de tamanho e fragmentação. Esse sistema de **arquivo trabalha com blocos**. O volume é dividido em blocos e estes são agrupados em grupos de blocos.



Linux – Sistema de Arquivos – Ext2

As informações armazenadas, os dados, são geralmente guardadas no mesmo conjunto de blocos. Esse tipo de armazenamento favorece minimizar as tentativas do sistema operacional em ler grandes quantidades de dados contínuos. Cada grupo de blocos contém informações sobre a tabela de descritores de superblocos e grupos de blocos. Os grupos de blocos possuem um *bitmap* (mapa de bits) de bloco, um bitmap de *inode*, uma tabela de *inode*, os blocos de dados reais.

Bloco: Informações sobre o Sistema de Arquivos

Super Bloco	FS descritores	Bloco <i>Bitmap</i>	<i>Inode</i> <i>Bitmap</i>	<i>Inode</i> Tabela	Dados Bloco
----------------	-------------------	------------------------	-------------------------------	------------------------	----------------

Linux – Sistema de Arquivos – Ext2

- O uso de blocos garante uma confiabilidade do sistema, pois há uma replicação da estrutura de controle em cada grupo de bloco. Isso permite uma recuperação de um sistema de arquivos em que o superbloco foi corrompido, reduz a distância entre a tabela de *inodes* e os blocos de dados e também melhora as buscas durante a operação **de entrada e saída em disco**, pois a tabela de *inode* contém **informações sobre os blocos de discos**.
- Os diretórios são gerenciados por meio de uma lista vinculadas de entradas. Essa lista possui comprimento variável. Em cada entrada, **o número do *inode*, o tamanho da entrada, o nome do arquivo e seu comprimento estão descritos**.
- Nomes extensos de arquivos podem ser implementados sem desperdício de espaço em disco:

Lista da Tabela de Diretórios

<i>inode</i> nº	Tamanho da entrada	Tamanho do nome	nome
-----------------	--------------------	-----------------	------

Limite para o sistema de arquivos Ext2

<i>Extension File System</i>				
Tamanho de bloco:	1 KiB	2 KiB	4 KiB	8 KiB
Tamanho máximo de arquivo:	16 GiB	256 GiB	2 TiB	2 TiB
Tamanho máximo do FS:	4 TiB	8 TiB	16 TiB	32 TiB

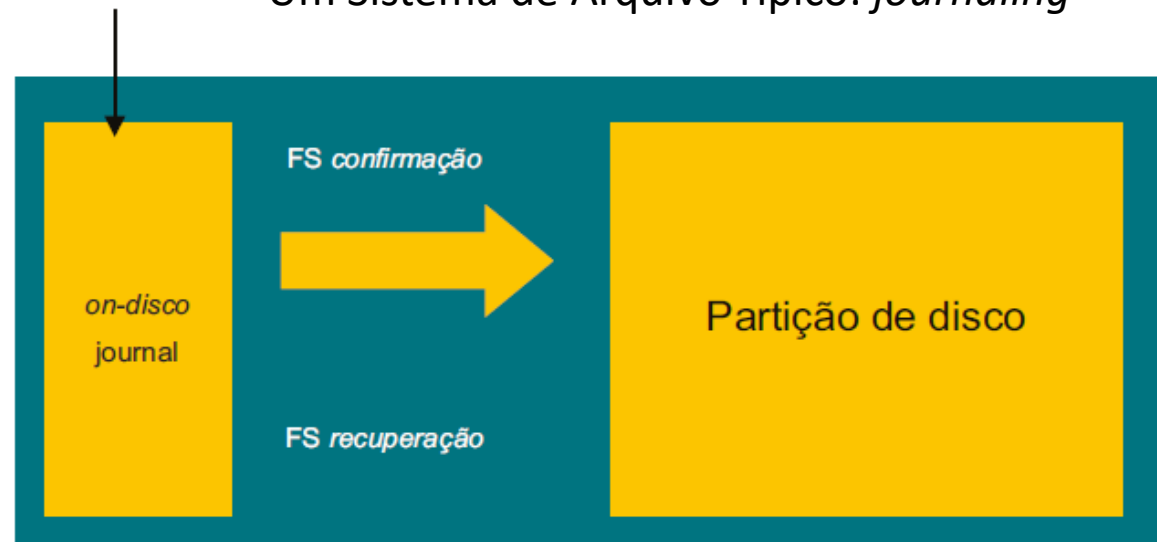
Linux – Sistema de Arquivos – Ext3



PUC Minas

Esse sistema de arquivos foi implementado **no Linux em 2001**. Sua característica mais marcante com relação ao **ext2** foi a **implementação do registro no diário, *journaling***. O sistema de *journaling* mantém um registro (log) de todas as mudanças no sistema de arquivos antes de escrever os dados no disco. Utilizando o *journaling*, somente as últimas transações são realizadas pelo sistema. Confiabilidade e eliminação da necessidade de verificar o sistema de arquivos após um desligamento não programado são melhoradas pela utilização do *journaling*.

Um Sistema de Arquivo Típico: *journaling*



O *journaling* registra as informações num diário, armazenando num buffer as alterações no sistema de arquivos. Esse registro diário é realizado por estratégias diferentes que determinam quando e o que é registrado. **Writeback, ordenadas e journal** são três das estratégias mais comumente utilizadas (NEGUS, 2014).

- **Journal:** as informações (os metadados e os dados) de cada arquivo são escritos no **journal**. Essa operação ocorre antes de qualquer escrita no sistema operacional. Pelo fato de a operação de escrita ocorrer duas vezes, há um aumento de confiabilidade e uma perda de desempenho.
- **Write-back:** essa técnica permite um melhor desempenho que o journal, pois somente os metadados são escritos no journaling, e não os dados. Os blocos de dados são gravados diretamente para sua localização no disco. Se o **sistema falhar após o registro dos metadados, mas antes de escrita dos blocos dos arquivos, pode haver uma falha nos arquivos**.
- **Ordenada:** semelhante ao **write-back**, porém, após o registro dos metadados, executa a escrita do conteúdo igual ao **journal**. Meio termo entre confiabilidade e desempenho.

Tamanho de bloco	Tamanho máximo de arquivo	Tamanho máximo do FS
1 KiB	16 GiB	4 TiB
2 KiB	256 GiB	8 TiB
4 KiB	2 TiB	16 TiB
8 KiB ^[limits 1]	2 TiB	32 TiB

Limites do ext3 FS

Linux – Sistema de Arquivos – Ext4

O ext4 utiliza vários conceitos úteis de muitos sistemas de arquivos concorrentes com JFS e XFS. Algumas características são:

- **Compatibilidade com versões anteriores:** foi desenvolvido para ter compatibilidade com versões anteriores, permitindo que as atualizações fossem feitas sem prejuízo.
- **Tempo do relógio:** versões anteriores ao **ext4**, a **marcação do relógio (tempo)** era feita em segundos, porém, no ext4, em **nanosegundos**.
- **Escalabilidade:** além dos limites, tabela Limites do **ext4 FS**, o limite do subdiretório foi expandido com o ext4, de diretórios de 32 kB para virtualmente ilimitados.
- **Utiliza extensões:** uma extensão é um intervalo de blocos físicos contíguos que melhora o desempenho de arquivos grandes e reduz a fragmentação.

Limites do ext4 FS

Tamanho de bloco	Tamanho máximo de arquivo	Tamanho máximo do FS
4 k	16TiB	1 EiB

Segurança

Segurança na área de TI possui três pilares: integridade, disponibilidade e confidencialidade.

O primeiro faz referência à integridade dos dados, ou seja, dados não corrompidos ou não alterados com relação à sua criação, origem (PAIXÃO, 2016). O segundo pilar faz referência ao acesso ao dado a quem é de direito, ou seja, o dado deve ser disponível a quem é permitido acessá-lo. Por último, a confidencialidade que diz respeito a garantir que a informação não será conhecida por pessoas que não estejam autorizadas para tal.



Nos Sistemas de Arquivos é essencial garantir que **os arquivos não sejam lidos ou modificados por pessoas não autorizadas** por meio de política, aspectos legais e também criar mecanismo de proteção para salvaguardar as informações no computador. Dois aspectos são mais importantes com relação ao computador: a **perda das informações e o acesso indevido (intruso)**.

Pra garantir uma cópia “original” dos dados, backups podem ser realizados. Backup é uma cópia de arquivos armazenados em diferentes dispositivos de armazenamento, como DVD, fita DA ou na nuvem (armazenamento on-line). Caso, por qualquer incidente de segurança, houver perda de informação, a cópia pode ser restaurada com os dados originais.

Outro aspecto importante é a privacidade, em sistemas multiusuários, como Windows e Linux, esse problema é resolvido criando seções individuais para cada usuário logado (autenticado) no sistema. Esses métodos são baseados em **login com senha e nome de usuário e verificação** (validação) no banco de dados dessas informações.

Na segurança de **ambientes operacionais**, há outras medidas para serem utilizadas para proteção, como o uso de firewall, antivírus, etc.

Segurança – Domínios de Proteção

- Num computador, **diversos objetos** estão sendo usados e/ou **compartilhados ao mesmo tempo**. Objetos de *hardware*, como DVD, HD, segmento de memória, impressores, ou de *software*, como processos, arquivos, banco de dados, semáforos.
- Cada objeto, além do nome único, possui uma série limitada de operações que os processos têm permissão para executar, por exemplo, para arquivos, operação de escrita e leitura, já para os semáforos, operações *UP* e *DOWN*.
- Então, o sistema operacional deve ter **mecanismo que faça corretamente a permissão de uso desses objetos**, ou seja, **que os objetos não sejam acessados indevidamente por outros objetos sem a permissão devida**. O mecanismo também deve determinar **quais tipos de operação podem ser realizadas**. Um processo X tem permissão de gravação num arquivo Y, mas não de leitura.
- O domínio é um conjunto de **pares (objetos, direitos)** que especifica um objeto e um subconjunto de permissão das operações que podem ser executadas.

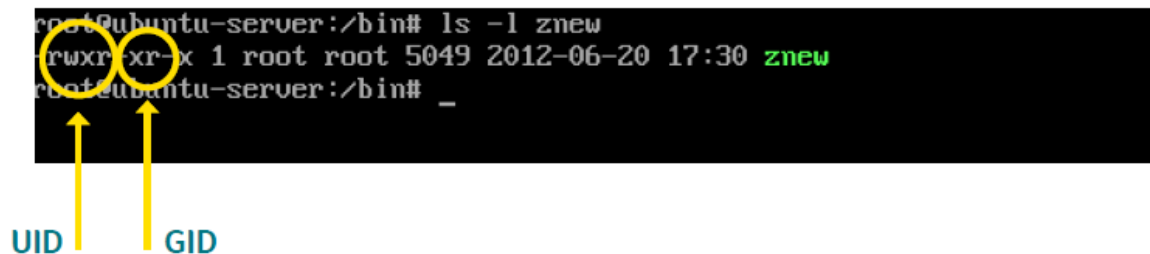


Os processos podem ser definidos **por somente leitura (R), somente escrita (W), somente execução (X)** ou por conjunto dos três: **leitura, escrita, execução (RWX)**. Além do mais, esse domínio é dividido em três: **uid, gid e others** (outros):

- **Uid** é a identificação relacionada ao usuário que o criou, **gid** está relacionado ao grupo a quem pertence o usuário, **others** é a **identificação dos demais usuários**, ou seja, que não são o proprietário do arquivo nem pertencem ao **grupo a que pertence o usuário proprietário**.

Proteção de um Arquivo no Linux

```
root@ubuntu-server:/bin# ls -l znew
-rwxr-xr-x 1 root root 5049 2012-06-20 17:30 znew
root@ubuntu-server:/bin# _
```

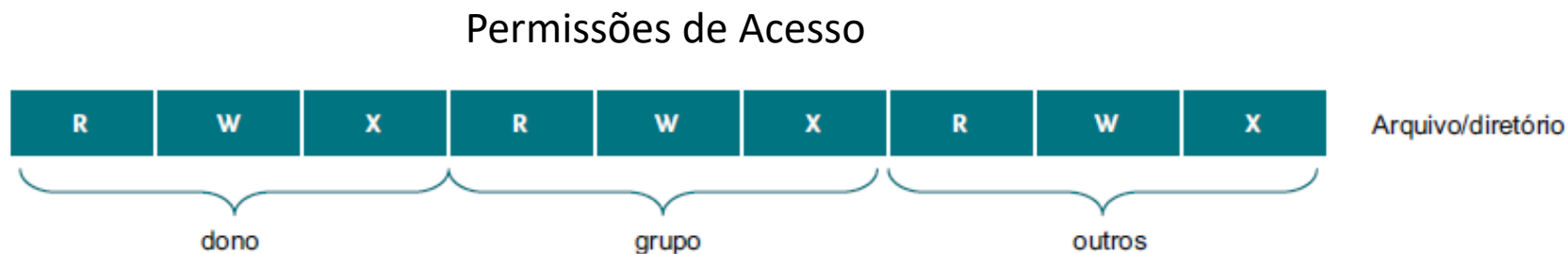


A figura ilustra as permissões *uid* (RWX), *gid* (R-X) e *others* (R-X), isso significa que:

- **o proprietário** possui permissão de leitura (*read*, R), gravação (*write*, W) e execução (*execute*, X), **RWX**;
- **o grupo** possui permissão de leitura (*read*, R) e execução (*execute*, X), gravação, não (-), **R-X**;
- **outros** possuem permissão de leitura (*read*, R) e execução (*execute*, X), gravação, não (-), **R-X**.

Segurança – Lista de acesso de controle (ACL)

Os sistemas de arquivos fazem uso do mecanismo de lista de acesso de controle para definir as permissões de **usuário e de grupo para os objetos**. Há, para **uid e gid**, três tipos de permissão padrão definidos em sistemas: **(R) leitura, (W) gravação e (X) execução** (PAIXÃO, 2016). Essas permissões são aplicáveis nos arquivos e diretórios para o dono, grupo e outros:



- O dono faz referência **ao proprietário do arquivo/diretório** e este pode **alterar essas informações a qualquer momento**. O grupo faz referência a agrupamento de usuários que possuem a mesma **permissão**. Um grupo pode **conter diversos objetos**, a exemplo, um grupo de administrador pode **conter diversos usuários e todos terão direito de administração com as regras colocadas na ACL** para todos os objetos que pertencerem a esse grupo.
- A parte de permissão relativa a **outros** faz referência às permissões que são dadas aos usuários que **não são o proprietário do arquivo nem pertencem ao grupo daquele objeto**.

Segurança – Lista de acesso de controle (ACL)

Por padrão, uma **restrição de usuário** que criou um determinado arquivo **se sobrepõe a uma permissão de grupo** ao qual esse usuário pertence. **As permissões desse grupo** se sobrepõem a uma restrição de **outros**. Porém, leitura faz sobreposição com relação à **gravação e escrita**. Isso pode ser visto pela ordem de valores que também são atribuídos a essas informações, **como leitura (4), escrita (2) e execução (1)**

Valores de Permissão de Arquivos

Permissão	Valores de permissão
R – <i>read</i> (leitura)	4
W – <i>write</i> (gravação)	2
X – <i>execute</i> (execução)	1

Um arquivo **com total permissão teria valor 7**, outro com permissão somente para leitura e execução, **valor 5**, e um terceiro com permissão somente para leitura, **valor 4**. Essas informações são para cada item na lista de **permissão, dono, grupo e outros**.

Exemplo: um arquivo com permissão de leitura (4), gravação (2) e execução (1) possui valor 7 (4+2+1). Esses valores são individualmente determinados para todos os três agrupamentos (dono, grupo, outros). Assim, um arquivo com permissão total (leitura, gravação e execução) possui valores 7 (dono), 7 (grupo) e 7 (outros). Nesse caso, o arquivo poderia ser acessado por qualquer usuário do sistema. Outros exemplos de permissão:

- **700** arquivo.txt: arquivo com permissão total para o dono (7) e negação total para grupo (0) e outros (0).
- **510** arquivo.txt: arquivo com permissão de leitura (4) e execução (1) para o dono, execução (1) para os membros do grupo e negação (0) total para outros.

Gerência de Entrada e Saída

Os dispositivos de entrada e saída (E/S) contido em qualquer computador são gerenciados pelo sistema operacional. Esses dispositivos são responsáveis por introduzir ou exibir as informações extraídas de um computador. Monitores, discos de armazenamentos, impressoras, terminais são exemplos de dispositivos de entrada e saída.

Gerenciamento de E/S

O sistema operacional controla todos os dispositivos de entrada e saída de um computador

- Emite comandos para os dispositivos.
- Intercepta interrupções.
- Trata erros.

Os dispositivos de E/S podem ser, de modo genérico, de dois tipos:

- Dispositivos de bloco
 - Armazena informação em blocos de tamanho fixo, que variam de 512bytes a 32.768 byte.
 - Todas as transferências estão em unidades de um ou mais blocos inteiros.
 - Cada bloco pode ser lido ou escrito independentemente de todos os outros, esta é uma propriedade essencial dos dispositivos de bloco.
- Dispositivos caractere
 - Envia ou recebe fluxos de caracteres e não dispõe de qualquer operação de posicionamento.

(TANENBAUM, 2009)

Gerenciamento de E/S

Dispositivos de E/S
apresentam uma ampla
variação de velocidade.

Dispositivo	Taxa de dados
Teclado	10 bytes/s
Mouse	100 bytes/s
Modem 56 K	7 KB/s
Scanner	400 KB/s
Filmadora <i>camcorder</i> digital	3,5 MB/s
Rede sem fio 802,11g	6,75 MB/s
CD-ROM 52x	7,8 MB/s
Fast Ethernet	12,5 MB/s
Cartão flash compacto	40 MB/s
FireWire (IEEE 1394)	50 MB/s
USB 2.0	60 MB/s
Padrão SONET OC-12	78 MB/s
Disco SCSI Ultra 2	80 MB/s
Gigabit Ethernet	125 MB/s
Drive de disco SATA	300 MB/s
Fita Ultrium	320 MB/s
Barramento PCI	— 528 MB/s

Tabela 5.1 Algumas taxas de dados típicas de dispositivos, placas de redes e barramentos.

(TANENBAUM, 2009)

Discos

Discos rígidos ou hard disks são dispositivos de armazenamento de dados permanentes. Referem-se a **dispositivos de entrada e saída** utilizados como memória secundária no sistema operacional. Os dados **no disco magnético são gravados em uma camada magnética muito fina**. Os discos rígidos possuem características como velocidade de rotação, quantidade de buffer e tipos de conexões IDE, serial ATA, SCSI, etc. (MACHADO; MAIA, 2013).



CD/DVD

Abreviação de compact disc, o CD é um dispositivo de armazenamento digital. O Digital Video Disc (DVD), **por sua vez, possui características semelhantes ao CD, sendo ambos dispositivos de entrada e saída.** O DVD possui, em média, **4,7 GB** de capacidade de armazenamento, e o **CD**, em média, **700 MB**. Ambos possuem tipos para **gravação e leitura**.

Monitores

Monitor é um dispositivo **de saída utilizado no computador para visualização do resultado das informações que são processadas na CPU.** Hoje, há monitores de diversos tipos (**plasma, LCD, LED, OLED**) e tamanhos (12", 24", 50") (MACHADO; MAIA, 2013).



Impressora

Impressora é um **dispositivo de saída** que tem como objetivo imprimir gráficos, textos e imagens em um papel. Elas evoluíram muito ao longo dos últimos tempos, existindo atualmente impressoras de diversos tipos, como jato de tinta, a laser e térmica, bem como impressoras específicas, como de cartão (MACHADO; MAIA, 2013).

Placa de Rede

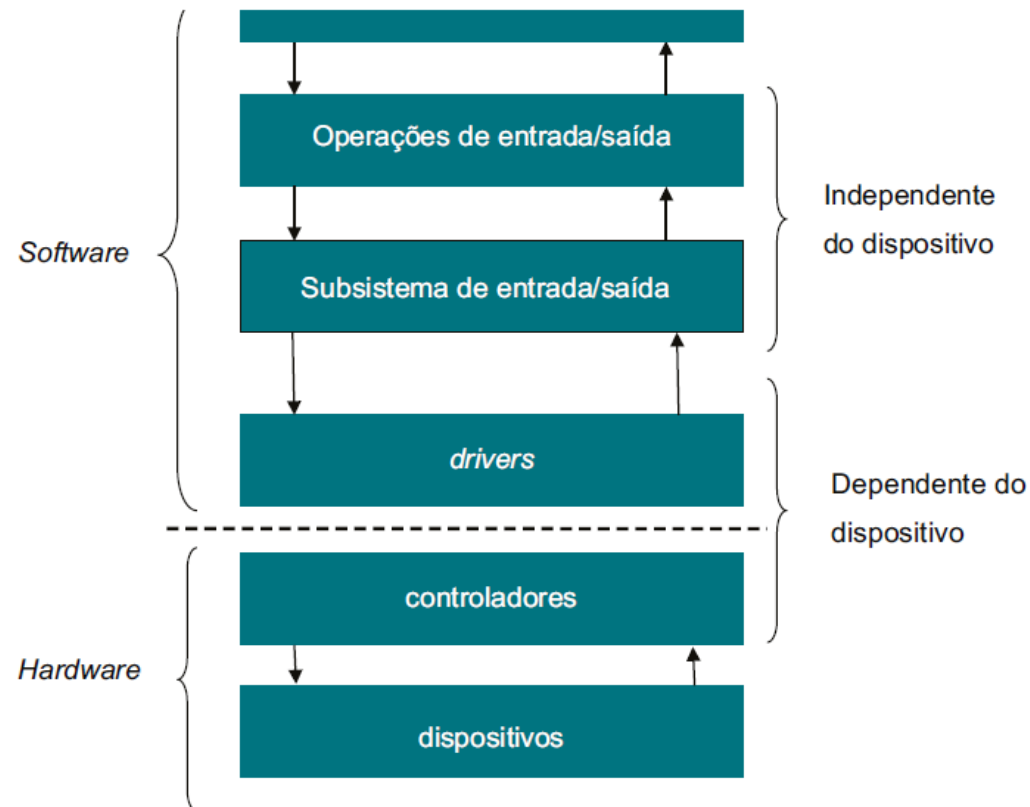
Placas de rede são dispositivos de *hardware* de entrada e saída que permitem que computadores ou quaisquer outros dispositivos eletrônicos, como **celulares, tablets e relógios**, façam a comunicação por meio de uma rede de computadores. Placas de rede podem ser de diversos tipos, como sistemas cabeados (cabos de cobre ou fibra) ou **sem cabo (*wireless*)**; além disso, possuem características de velocidade medidas em *bits*/segundo (10 Mb/s, 100 Mb/s, etc.).



Com os processadores e memórias, o sistema operacional possui uma tarefa árdua e complexa na gerência dos dispositivos de entrada/saída (input/output). Cada dispositivo de I/O é diferente do outro. O sistema operacional deve permitir que as informações passem de um dispositivo para outro, bem como o reconhecimento do dispositivo quando acoplado ao sistema operacional.

Princípios de hardware de entrada/saída

- O sistema operacional tem que fazer a comunicação com qualquer dispositivo que possa ser acoplado ao computador. Esse conceito chama-se independência de dispositivos. Geralmente, um dispositivo de **entrada e saída é menos confiável que outros componentes do sistema operacional**, como processador e memória. A maneira mais simples de fazer essa comunicação é por meio de **bibliotecas de comando de I/O**. Essas operações devem ser realizadas por chamada de sistema (*system calls*).



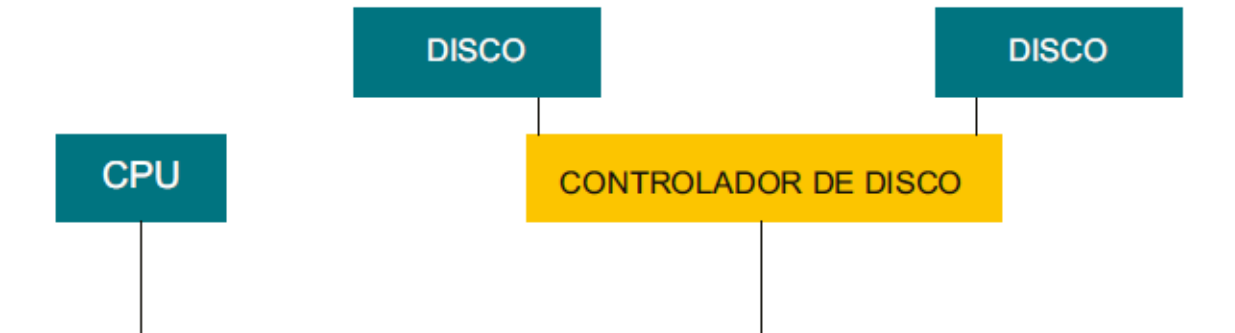
Subsistemas de entrada e saída

Um **sistema de entrada e saída** possui alguns **subsistemas**. Ao imaginar um dispositivo de entrada e saída, pensa-se em dispositivos como **disco rígido, monitores, impressoras, entre outros; porém, esses dispositivos não são controlados pelo sistema operacional, e sim pelo controlador de dispositivo** (STUART, 2011).

Esse controlador é uma interface entre a CPU e o dispositivo de hardware.

Esse controlador faz comunicação com o sistema operacional por meio de um **software chamado de driver de dispositivo**. A última parte de um subsistema é o tratador de interrupções, uma parte do driver de dispositivo que responde a sinais gerados pelo **controlador de dispositivo para solicitar serviço**.

Layout Típico de um Dispositivo de I/O



Dispositivos de entrada e saída podem ser divididos em dois subgrupos: **dispositivo de bloco** e **dispositivo de caractere**.

- **Dispositivos de bloco** são aqueles que têm o armazenamento em blocos de tamanho fixo, cada um com seu próprio endereço. **Os blocos possuem tamanhos variados (entre 512 bytes e 32.768 bytes)**. O conceito fundamental dos dispositivos de **bloco** é que **operações de leitura e gravação podem ser realizadas independentemente uma da outra**. **Exemplo:** Disco Magnético
- Os dispositivos de caractere enviam e **recebem um fluxo de informações com base em caracteres**, sem considerar a estrutura de blocos. **Impressoras, interfaces de rede, mouses e teclados** podem ser considerados dispositivos de caracteres.

Dispositivo	Taxa de dados (velocidade)
Disco IDE	5 MB/s
Barramento PCI	528 MB/s
<i>Mouse</i>	10 Bytes/s
USB	1,5 MB/s

Controladores de dispositivos

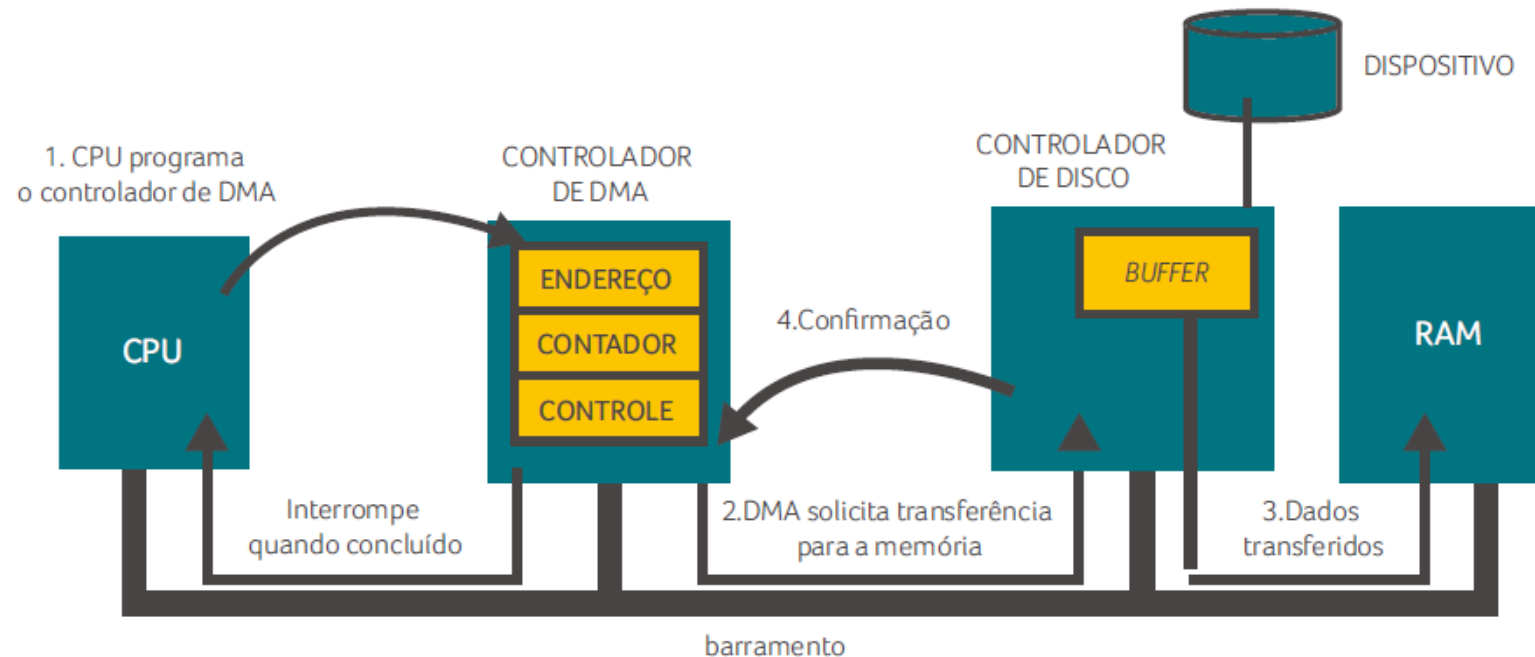
Unidade de armazenamento, por exemplo, possui dois controladores: um conjunto mecânico e uma placa eletrônica. Esse componente eletrônico é chamado de **controlador de disco** ou **controlador de dispositivo**.

A placa geralmente tem um conector onde é plugado o cabo que conecta o dispositivo. Esses conectores, chamados de interface, são padronizados por entidades internacionais (ANSI, IEEE ou ISO), e as empresas desenvolvedoras desses dispositivos somente precisam criar controladores que aceitem aquele padrão de interface. Para disco, por exemplo, o padrão muito usado atualmente é a serial ATA (STUART, 2011).

Existem interfaces universais que podem se conectar a diversos tipos de dispositivo. Esses conectores são chamados de **universal serial bus (USB)** ou **barramento universal**. Hoje, a maioria dos conectores é do tipo USB. Cada controlador possui alguns registradores usados para a comunicação com a CPU. Esses registradores ajudam o sistema operacional a enviar comandos aos dispositivos de aceitação dos dados para executar ou desligar qualquer outra tarefa. Muitos desses controladores possuem buffer, e uma memória secundária ajuda na melhoria de desempenho de leitura dos dados.

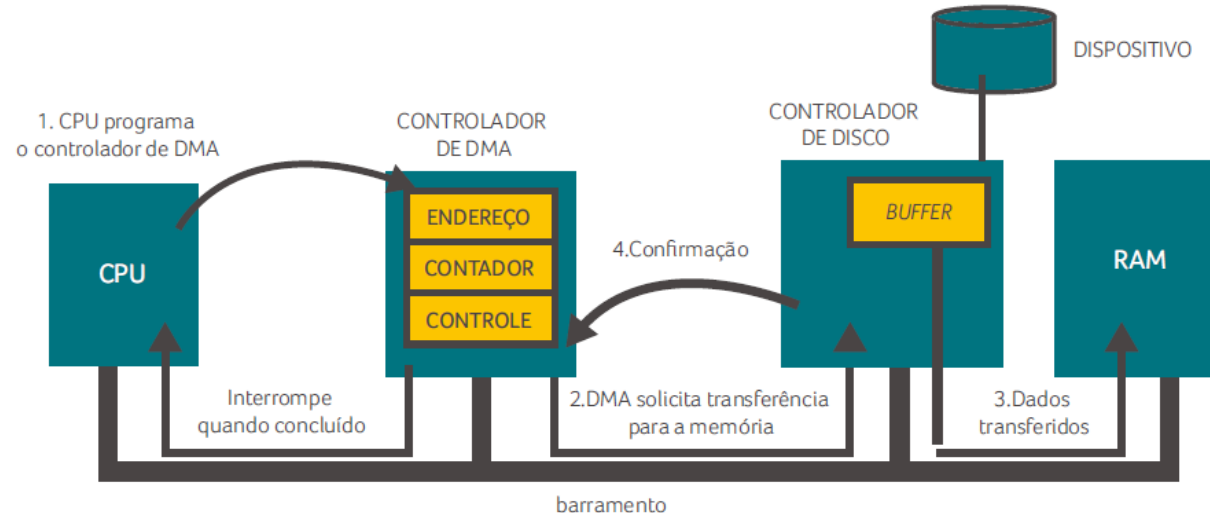
Acesso Direto à Memória

- Independentemente do **processo de mapeamento dos registradores feito pela CPU**, esta precisa endereçar os **controladores para trocar dados entre eles**. A **requisição de dados da CPU pode ser realizada em bytes por vez; porém, esse processo gera um custo alto de operação**.
- Muitos sistemas fazem uso de um recurso chamado **DMA (direct memory access)**, ou seja, o DMA permite que dispositivos de **hardware (placas de rede, vídeo, discos)** acessem a **memória independentemente da CPU**. São **permitidas operações tanto de leitura quanto de gravação**.
- Basicamente, uma transferência por DMA copia um bloco de memória de um dispositivo para outro (TANENBAUM, 2016). Esses controladores são constituídos de vários registradores, e muitos desses barramentos de DMA operam de dois modos: **bloco e modo palavra**.



Acesso Direto à Memória

- Nesse cenário, os **registradores de DMA** são inseridos com valores enviados pela CPU.
 - Esses valores registram **operações de transferência de informações** – o que transferir e para onde transferir essas informações (**passo 1**). O controlador DMA gera um sinal, envia-o para o controlador de disco com o objetivo de transferir **os dados do disco para seu buffer interno e analisa a somatória da verificação**. O controlador de DMA começa a transferir os dados quando os dados contidos no buffer do controlador são válidos. **No passo 2, um sinal de requisição de leitura para o controlador de disco é emitido pelo controlador DMA**. A partir deste momento, dá-se início à transferência. Os dados são transferidos **para RAM, ou seja, uma escrita na memória é realizada (passo 3)**. Por meio do barramento, **após a escrita completa, um sinal de confirmação para o controlador de DMA é enviado pelo controlador de disco (passo 4)**.
- A maioria dos controladores DMA utiliza endereçamento de **memória física para as suas transferências**. Esse modo faz com que o sistema operacional converta os endereços virtuais de buffer de memória pretendido em um endereço físico e escreva o endereço físico no registrador de endereço do DMA. O DMA tem uma velocidade bem mais baixa de que a velocidade da CPU. É Usado em computadores de baixo custo (embarcados).



- NEGUS, Christopher. **Linux** — A Bíblia — o Mais Abrangente e Definitivo Guia Sobre Linux. Alta Books, 2014. 852 p. ISBN: 9788576087991.
- MACHADO, Francis Berenger; MAIA, Luiz Paulo. Arquitetura de sistemas operacionais. 5a ed. Rio de Janeiro: LTC, 2013. 263 p. ISBN:9788521622109.
- PAIXÃO, Renato Rodrigues. **Arquitetura de Computadores** – PCs. São Paulo: Editora Érica, 2016. ISBN digital 9788536518848.
- SILBERSCHATZ, Abraham. Fundamentos de Sistemas Operacionais. 9ª ed. LTC, 2015. 524 p. ISBN: 9788521629399. ISBN digital 9788521630012.
- STUART, Brian L.. Princípios de sistema operacionais: projetos e aplicações. São Paulo: Cenage Learning, 2011. 637 p. ISBN: 9788522107339.
- TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 4a ed. São Paulo: Pearson, 2016. 864 p. ISBN: 9788543005676.
- ZACKER, Craig. **Instalação e Configuração do Windows Server 2012 R2**. Porto Alegre: Editora Bookman, 2015. ISBN digital 9788582603581.