

UNIDADE 4 – ARQUIVOS DE DADOS, EXECUTÁVEIS E BIBLIOTECAS, BAT E SCRIPTS.**MÓDULO 1 – ARQUIVOS DE DADOS****01****1 - O QUE SÃO ARQUIVOS DE DADOS?**

Já vimos conceitos básicos sobre arquivos de internet e multimídia, agora iremos aprender características sobre os arquivos de dados.

Arquivos de dados, arquivos de banco de dados, bases de dados, ou simplesmente banco de dados são sistemas e tecnologias capazes de armazenar informações de forma organizada e racionalizada.

Informação refere-se principalmente a informações relacionadas às empresas e organizações, acerca do dia a dia, tanto na área administrativa quanto na área finalística.

O termo organizada diz respeito à separação das informações de acordo com as características de cada tipo de assunto. Exemplo hipotético relacionado a uma empresa: um local para armazenar os dados dos produtos que ela vende, outro local para armazenar os dados dos clientes, outro para os dados das vendas, outro para fornecedores, contas a pagar a receber, etc.

O local onde os dados são armazenados são chamados de **tabelas**. Por isso, para cada assunto, geralmente há uma ou mais tabelas para armazenamento.

O termo racionalizada refere-se a técnicas que evitam o cadastro de informação duplicada, tornando os dados menos redundantes, mais organizados e ocupando menores espaços. Essa racionalização é chamada de **normalização de modelos de banco de dados**.

Todas as pequenas, médias e grandes empresas atualmente utilizam sistemas de informação que contemplam o registro (guarda) e a recuperação de informações.

Os sistemas que guardam e permitem a manipulação das informações são chamados de “Sistemas Gerenciadores de Bancos de Dados (SGBD)”.

02**2 - TECNOLOGIAS E CARACTERÍSTICAS DOS SGDBs****2.1 - Bancos de dados de primeira geração**

Os primeiros sistemas de bancos de dados eram constituídos por simples arquivos nos quais a informação era registrada sequencialmente. Cada arquivo recebia um conjunto de informações do mesmo tipo.

A principal desvantagem dessa tecnologia era a forma de localização e recuperação da informação, pois era preciso percorrer o arquivo desde o início até localizar a informação, de modo sequencial, e isso era extremamente lento. Tecnologias muito antigas como o COBOL utilizam sistemas de arquivos sequenciais.

- **Cobol**

Falando especificamente da parte de arquivos de dados do Cobol, por ser uma linguagem muito simples de recursos, as técnicas para acessar as informações são bastante complexas se comparadas com as tecnologias mais modernas.

Para acessar um arquivo em Cobol, o programador precisa saber:

- **Quais informações estão gravadas no arquivo (campos).**
- **Qual o tamanho de cada informação gravada no arquivo (em bytes).**
- **Qual o tamanho de cada bloco de informações (somatório dos tamanhos de cada campo).**
- **Determinando assim a largura de cada registro.**

De posse dessas informações, o programador, **para localizar e recuperar um registro**, precisa:

- **Localizar e acessar fisicamente o arquivo onde estão as informações.**
- **Abrir o arquivo para leitura.**
- **Posicionar a leitura no ponto onde está o campo a ser identificado o primeiro conjunto de informação.**
- **Ler as informações, separando-as em campos.**
- **Validar se aquela é ou não a informação que ele queria ler.**
- **Se não for, mover para frente, saltando tantos bytes do arquivo quanto for a largura do registro.**
- **Repetir o processo a partir do terceiro passo para validar o próximo registro.**
- **Quando conseguir ler o dado correto, então pode parar.**

Cobol

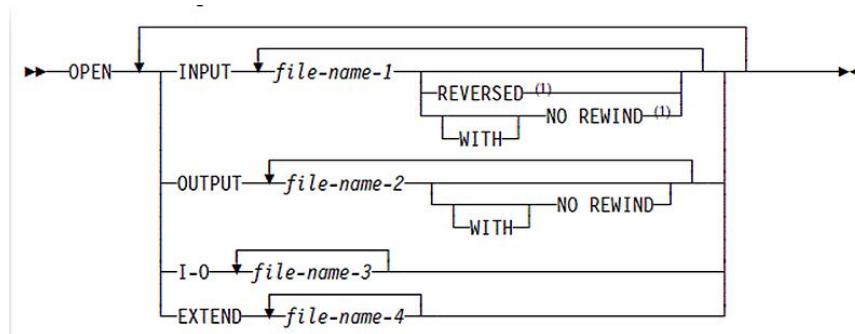
O Cobol - Common Business Oriented Language ou Linguagem Comum Orientada para os Negócios - é uma linguagem de programação orientada para o processamento de banco de dados comerciais. Foi criada em 1952, pelo Departamento de Defesa Norte Americano. Apesar de ser uma tecnologia bastante antiga, ela ainda é utilizada por uma grande quantidade de empresas. Em 1997, o Gartner Group relatou que 80% dos sistemas do mundo rodavam em Cobol. Aqui no Brasil, por exemplo, Banco do Brasil e Caixa Econômica Federal possuem sistemas que ainda rodam em Cobol. Uma das grandes vantagens do uso do Cobol é que ele exige uma quantidade mínima de recursos do computador, com isso, sistemas em Cobol são bastante rápidos. As maiores desvantagens são: a) o acesso exclusivo aos dados (somente uma aplicação ou usuário pode acessar os dados por vez) e b) pouquíssimos recursos visuais disponíveis (sistemas em Cobol funcionam em telas monocromáticas,

que apresentem apenas caracteres; nem mesmo o mouse é utilizável em sistemas Cobol).

03

Veja um exemplo de um programa Cobol aqui: <http://www.cadcobol.com/cadbalan.htm>.

Bastante complexo, não é mesmo? O gráfico a seguir resume os comandos técnicos que o programador precisa definir para ler as informações no arquivo:



Para saber um pouco mais sobre o Cobol, acesse aqui: <http://pt.wikipedia.org/wiki/COBOL>.

04

2.2 - Bancos de dados de segunda geração

A segunda tecnologia que surgiu ainda utilizava arquivos separados para cada assunto, mas a novidade criada à época foi o surgimento de índices.

Índices são pequenos arquivos que funcionam como um sumário de um livro, eles indicam o local exato dentro do arquivo onde estão localizadas as informações desejadas.

Os arquivos de índice são bem pequenos e as informações são ordenadas por algum parâmetro, como alfabeticamente ou por uma data, por exemplo. Como estão ordenados, podem ser consultados por meio de técnicas mais eficientes de localização de informação, como a B-tree (árvore binária). Ao localizar a informação no índice, pode-se posicionar a leitura da informação exatamente no ponto correspondente, e assim realizar as operações de busca e alteração de forma bem mais rápida.

Os arquivos dessa tecnologia também deixaram de ser arquivos de texto e passaram a ser arquivos binários, para facilitar a inclusão de informações de índices, por exemplo. Tecnologias como o DBASE e CLIPPER são baseadas nessas características.

05

- **dBase e Clipper**

Uma das grandes inovações (e vantagens) do dBase em relação ao Cobol é que o dBase incorporava funcionalidades de **controle de fluxo** (como o Do While / End Do) e **Lógica condicional** (como IF e End If). Para manipulação de dados, dBase oferecia comandos e funções procedurais sofisticados para abrir e pesquisar arquivos, manipular o valor de campos, e manipular texto, números e datas. Também possuía capacidade de abrir e manipular simultaneamente diversos arquivos.

O formato de arquivo do dBase (e seus variantes) é o DBF. O sistema de banco de dados dBase foi um dos primeiros a prover uma seção de cabeçalho (header) para descrever a estrutura dos dados dentro do arquivo. Isso significa que o programa não mais necessitava conhecimento prévio da estrutura de dados, mas poderia perguntar diretamente ao arquivo de dados como este estava estruturado.

O dBase também criou o conceito de **arquivos de índice**, extensão IDX (dBase) ou NTX (Clipper). Esses arquivos serviam como sumários para apontar o local exato do conteúdo da informação dentro do arquivo DBF, dessa forma, era extremamente rápido pesquisar uma informação indexada.

Imagine uma tabela de pessoas com mais de 1.000 registros, para o sistema localizar uma determinada pessoa sem um índice de nome, o sistema precisaria percorrer cada um dos clientes até encontrar o cliente desejado, ou seja, se seu cliente fosse o último do cadastro, o sistema iria realizar 1.000 consultas até conseguir localizá-lo. Já se houvesse um índice de nomes, o sistema iria executar no máximo 10 passos para localizar o cliente (a quantidade de passos para localização de registros em um índice é calculado por Log x base 2, onde x é a quantidade de registros. Dessa forma, Log 1000 base 2 = 9,96, que corresponde a 10 operações).

Uma tabela pode ter vários índices. Por exemplo, uma tabela de cadastro de clientes pode ter um índice para o nome do cliente, outro índice para o CPF da pessoa, outro para o CEP e outro para o Telefone. Dessa forma, realizar pesquisas por nome, CPF, CEP e telefone seria algo rápido de ser executado.

dBase

O dBase surgiu em 1978 e foi o primeiro sistema de banco de dados criado para microcomputador (o Cobol funcionava apenas em computadores de grande porte àquele época). Durante os anos 1980, o dBase foi um dos softwares mais vendidos mundialmente. Nessa época, muitas variações semelhantes ao dBase surgiram, entre elas, a mais famosa certamente foi o Clipper (criado em 1984). Para saber mais sobre dBase e Clipper, acesse aqui: <http://pt.wikipedia.org/wiki/DBASE>.

06

2.3 - Bancos de dados de terceira geração

A terceira forma de organizar as informações foi por meio de um arquivo único. Todas as informações são separadas internamente em tabelas, mas apenas um arquivo contempla tudo o que é necessário: dados, índices etc.

A grande vantagem dessa tecnologia foi a de simplificação técnica, ao invés de termos dezenas e dezenas de arquivos, apenas um era necessária. Com isso ficou mais fácil evitar problemas de perdas de dados, quebra de índices e outros. O Ms Access é um exemplo dessa tecnologia.

- **Ms Access**

Mais do que um simples sistema de banco de dados, o Ms Access incorpora:

- Ferramenta gráfica de modelagem de dados;
- Ferramenta gráfica e textual para criação de consultas padrão SQL;
- Ferramenta de criação de relatórios;
- Ferramenta de formulários de cadastro, com interface gráfica que permitia listas, tabelas, imagens, comandos de navegação e busca;
- Linguagem de programação Visual Basic Applications.

Ao utilizar o Ms Access você tem a opção de criar um arquivo que contenha somente um banco de dados ou mesmo um sistema completo, com as funcionalidades e recursos citados acima. Muitas empresas utilizam sistemas e bancos de dados criados em Ms Access dada a sua simplicidade de utilização.

Outra grande vantagem, é que tanto o banco de dados como o sistema completo fica dentro de um único arquivo. As extensões de arquivo para bancos de dados e sistemas feitos em Ms Access são: MDB (utilizada até a versão 2003) e ACCDB (a partir da versão 2007, inclusive).

MS Access

O Ms Access é um programa criado pela Microsoft em 1992 com o objetivo de criar e gerenciar banco de dados.

Para saber mais sobre o Ms Access, acesse aqui: http://pt.wikipedia.org/wiki/Microsoft_Access.

07

2.4 - Bancos de dados de quarta geração

A quarta forma de organização é aquela de deixou de ser apenas arquivos para guardar informações e serem sistemas para gerenciar as informações. Com essa tecnologia, o sistema gerenciador de banco de dados é o responsável por organizar, alterar, excluir e consultar as informações.

Os sistemas de informação fazem consultas e operações ao SGBD e ele decide a melhor forma de prover os resultados. Duas das grandes vantagens dessa tecnologia foram:

- a) o isolamento entre a aplicação e dos dados, os programadores não precisavam mais operar com arquivos diretamente, mas sim solicitar operações aos SGBD apenas;
- b) o melhor gerenciamento do trabalho concorrente em sistemas com vários usuários realizando operações no mesmo banco de dados. Com isso, tornou-se muito mais seguro (diminuíram os erros e problemas de acesso) gerenciar dados.

Oracle, SQL Server, MySQL e Postgree são exemplos de SGBDs que comportam essa tecnologia.

- **Arquivos de dados, logs e de índices**

A criação de arquivos de dados e índices dentro dessas tecnologias é algo simples e transparente para o programador. Ele cria o banco de dados e automaticamente os arquivos necessários são criados. Entretanto, em aplicações complexas ou bancos de dados muito grandes, é possível realizar a criação manual dos arquivos, possibilitando assim definir o conteúdo de cada arquivo, para segmentar os dados, logs e índices de acordo com as características do banco e do hardware onde o banco de dados está localizado. O profissional habilitado a realizar tais manobras é o DBA (Administrador de Banco de Dados). Foge da competência de um programador saber realizar tais atividades.

08

3 - PRINCIPAIS TERMOS ASSOCIADOS A UM BANCO DE DADOS

Atenção, nosso objetivo é apenas oferecer um contexto inicial sobre o tema banco de dados, você verá essas informações com bastante detalhes em matéria específica de banco de dados.

Veremos a seguir alguns termos associados a um banco de dados. O conhecimento desses é fundamental para que você entenda um pouco mais sobre os assuntos abordados.

3.1 – Tabela

Tabela é o local onde os dados semelhantes são armazenados.

Em uma empresa, por exemplo, podemos ter a tabela de estoque, de clientes, de fornecedores, de produtos, de contas a pagar, de vendas, de cidades, de Unidades da Federação, de CEPs, de usuários do sistema etc.

Cada tabela contempla informações semelhantes, ou seja, uma tabela de clientes contempla os nomes, endereços e telefones de todos os clientes. Já a tabela de produtos, contempla o nome, descrição, marca, modelo, preço e estoque disponível dos produtos.

09

3.2 - Campo

Campo refere-se a cada tipo de informação que é cadastrada em uma tabela.

Uma tabela de cadastro de pessoas, tem, por exemplo, os seguintes campos: nome, endereço, cidade, UF, CEP e telefone.

Observe o exemplo abaixo, uma tabela de Unidades da Federação (UF), os campos dessa tabela são “código” e “sigla”.

Código	Sigla
1	RJ
2	SP
3	DF

Exemplo de tabela de UFs**3.2.1 - Chave primária**

Chave primária é um código que identifica exclusivamente cada registro dentro da tabela. Esse código é necessário para permitir o relacionamento entre as informações, praticamente todas as tabelas contam com esse campo especial.

A chave primária é geralmente um número sequencial (1, 2, 3...). Entretanto, outras informações como número do CPF, número de série de um produto, poderiam ser chaves primárias em determinado contexto.

No exemplo da tabela de Unidades da Federação, a chave primária da tabela é o campo código. O código do Rio de Janeiro é 1, portanto, o número 1 identifica exclusivamente o RJ como a UF de código 1.

10**3.3 - Tipos de dado**

Tipos de dado referem-se ao tipo de informação de cada campo da tabela.

Um campo do tipo “nome de pessoa” é, por exemplo, um campo do tipo texto; já um campo como “idade” é um campo do tipo número inteiro; por fim, um campo como “pedido foi entregue”, é do tipo verdadeiro ou falso, pois só pode assumir esses dois valores. Os tipos mais comuns de dados são:

• Inteiro	Para representar um número inteiro, como 1, 2, 3 etc. São exemplos de números inteiros: o código de uma tabela, o número de filhos de uma pessoa, a idade de uma pessoa.
• Fracionado	Para representar um número fracionado, como um percentual, um valor monetário ou qualquer outro valor não inteiro. São exemplos de números fracionados: Valor de uma venda, percentual de juros, altura de uma pessoa, peso de uma carga.
• Texto	Para representar campos de texto. Geralmente esse tipo de dados inclui quantos caracteres o campo pode conter. Por exemplo, texto(50), representaria um limite de até 50 caracteres. São exemplos de campos de texto: o nome da pessoa, endereço, nome da cidade, número de telefone (pois inclui sinais especiais como parêntesis e traço), e-mail, número de CPF (pois incluir sinais como pontos, barra ou traço).

• Data	Para representar datas, horas ou data/hora. São exemplos de campos do tipo data: data de cadastro no sistema, data da venda, data do pagamento.
• Verdadeiro/ Falso	Também chamado de booleano, representa apenas duas opções de valores: verdadeiro ou falso. São muito utilizados para sinalizar status. Exemplos: para sinalizar se o cliente está ativo, para sinalizar se o usuário pode fazer uma compra, para sinalizar se um produto foi entregue.
• Binário	Para permitir o cadastro de imagens, fotos, arquivos de som, documentos ou arquivos do computador, ou qualquer outra informação em formato binário.

11

3.4 – Registro

Registro é cada informação cadastrada em uma tabela.

Olhando o exemplo da tabela de UFs, a informação “1 – RJ” é um registro, “2 – SP” é outro registro, “3 – DF” é o último registro dessa tabela.

3.5 – Normalização

Normalização é a técnica utilizada para eliminar redundâncias do banco de dados, e assim conseguir diminuir o espaço utilizado, bem como evitar duplicação de informações.

Vamos supor, por exemplo, que o nosso sistema possua uma tabela de cidades com as informações abaixo:

Código	Cidade	UF
1	Búzios	RJ
2	Araruama	RJ
3	Santos	SP
4	Brasília	DF

Exemplo de tabela de cidades

Observe que essa tabela de cidades contempla um campo “UF” com a sigla das unidades da federação onde estão localizadas as respectivas cidades. Veja que a sigla do RJ, por exemplo, foi repetida. E ao cadastrar uma quinta cidade, alguém poderia inadvertidamente cadastrá-la assim:

5	Teresópolis	Rio de Janeiro
---	-------------	----------------

Veja que não foi observado o padrão de sigla da UF, que ficou diferente da forma anterior. Para o usuário comum, RJ e Rio de Janeiro são a mesma coisa, mas para o banco de dados não. Essa diferença de padrão pode resultar em problemas futuros utilizando o *software*.

12

Ao aplicar a normalização (no caso anterior), iremos criar uma outra tabela auxiliar com as informações repetíveis e substituir o valor da informação na tabela de origem pelo código (chave primária) dessa segunda tabela. Dessa forma, o exemplo acima, normalizado, poderia ser algo assim:

Código	Cidade	Código da UF
1	Búzios	1
2	Araruama	1
3	Santos	2
4	Brasília	3

Tabela de cidades

Código	Sigla
1	RJ
2	SP
3	DF

Tabela de UFs

Dessa forma, por meio de comandos de relacionamento (veja a seguir), o sistema seria capaz de interpretar e substituir o código da UF pela sigla da UF, trazendo para o usuário um resultado assim:

Código	Cidade	UF
1	Búzios	RJ
2	Araruama	RJ
3	Santos	SP
4	Brasília	DF

Tabela vista pelo usuário: Cidades e UFs

13

Com a normalização, o usuário não tem como cadastrar novas cidades com divergências de informação de UF, por exemplo. O uso desse tipo de recurso é facilmente visto nos sistemas quando temos aquelas caixas de seleção de valor.

Veja a tela abaixo, o campo de “identificação de endereço” certamente é uma tabela à parte que leva seu identificador (chave primária) para a tabela de cadastro de endereço.

Meu Endereço

Os campos em **NEGRITO** são de preenchimento obrigatório, pois são impressos na Nota Fiscal do pedido.

Primeiro digite o CEP: Não sabe o CEP, [consulte Aqui](#)

***Identificação do endereço**

* endereço Seleccione ▾
Seleccione
Apartamento
Casa
Comercial
Outro

* número Ex: Apto 203

complemento

* bairro

* cidade

* estado

* informações de referência

Sistema de informação com tela de cadastro de endereço.

14

3.6 – Relacionamento

O relacionamento diz respeito à interligação entre determinados registros dentro de um banco de dados.

Uma das vantagens do relacionamento é permitir que vários registros de uma tabela sejam associados a um registro de outra. Por meio do relacionamento, existem técnicas que não permitem a existência de dados desvinculados da relação.

Um exemplo disso seria, por exemplo, uma pessoa que possui vários números de telefone, poderia haver uma tabela com os dados da pessoa e outra com os números do telefone e o código da pessoa que possui aqueles números de telefone. Veja o exemplo abaixo:

- Suponha que Marcelo tenha os números de telefone: 8888-7777 e 2222-3333;
- Suponha que Cláudio tenha os números de telefone: 9999-8888, 2222-4444 e 3333-5555.

Uma tabela para conter essas informações seria:

Código	Nome	Telefone 1	Telefone 2	Telefone 3
1	Marcelo	8888-7777	2222-3333	
2	Cláudio	9999-8888	2222-4444	3333-5555

Além de termos campos em branco, observe que estamos limitados a três números de telefone por pessoa. Caso outra pessoa possua quatro números de telefone, apenas três poderiam ser cadastrados.

Para resolver tal questão, utilizaríamos a técnica de normalização, criando uma outra tabela de telefones e alterando a tabela de pessoas, o resultado final seria algo como:

Código	Nome
1	Marcelo
2	Cláudio

Tabela de pessoas

Código	Código da Pessoa	Telefone
1	1	8888-7777
2	1	2222-3333
3	2	9999-8888
4	2	2222-4444
5	2	3333-5555

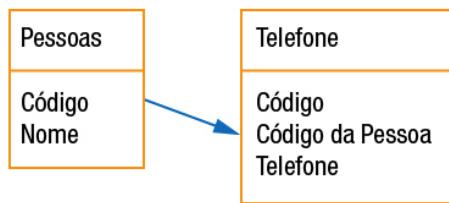
Tabela de telefones

Dessa forma, para saber quais são os números de telefone do “Marcelo”, o sistema de banco de dados procederia assim:

- Qual é o código do Marcelo na tabela de pessoas: é 1.
- Quais são os números de telefone na tabela de telefones que tem 1 como código de pessoa: 8888-7777 e 2222-3333.

Observamos que há um relacionamento entre os campos “Código” da tabela de pessoas e “Código da pessoa” na tabela de telefones. Esse relacionamento é que permite identificar quais números temos associados a uma pessoa.

O diagrama que representa relacionamentos é chamado de **Diagrama de Entidade Relacionamento** ou **Modelo de Entidade Relacionamento – MER**, ou ainda, modelo de dados. Para o exemplo acima, teríamos o seguinte diagrama de relacionamento:



Exemplo simples de Modelo de Entidade Relacionamento

Nesse diagrama vemos duas caixas, cada caixa representa uma tabela. Dentro de cada caixa vemos um texto na parte superior, esse é o nome da tabela. Depois, vemos textos logo abaixo, esses são os campos da tabela. A linha com a seta indica o relacionamento entre as tabelas. A direção da seta indica quem é a tabela que recebe a chave primária da outra tabela. Esse relacionamento é interpretado da seguinte forma: uma pessoa pode ter nenhum, um ou vários telefones cadastrados.

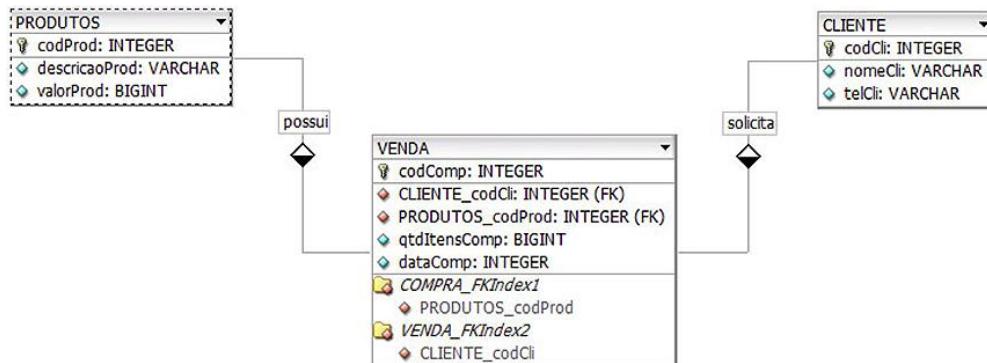
Dependendo do grau de sofisticação da ferramenta com a qual você está elaborando um diagrama MER, você poderá observar várias outras informações, como por exemplo:

- **Indicador gráfico de qual campo é a chave-primária (geralmente é o desenho de uma chave);**
- **Indicador se o campo é obrigatório ou não no cadastro;**
- **Tipo de campo (se é número inteiro, número fracionado, texto fixo, texto variável, data, verdadeiro/falso etc.).**
- **Relacionamento entre as tabelas (setas ou linhas).**
- **Regras de conteúdo e relacionamento (como, por exemplo, se é obrigatório possuir ao menos um registro no relacionamento).**

17

a) Exemplo 1

Observe o exemplo a seguir:



Exemplo de diagrama de relacionamentos e modelo de dados

Na imagem acima, observamos três tabelas: Produtos, Venda e Cliente. Conseguimos identificar os relacionamentos entre as tabelas (uma venda possui um ou vários produtos, um cliente solicita uma ou várias vendas) e também as informações de cada tabela:

- Tabela de produtos, campos:
 - codProd, que é do tipo inteiro, e é a chave primária.
 - descricaoProd, do tipo texto (varchar) e é a descrição do produto.
 - valorProd, do tipo inteiro (bigint) representa o preço do produto.
- Tabela de cliente, campos:
 - codCli, inteiro, chave primária.

- nomeCli, texto (varchar), nome do cliente.
- telCli, texto, telefone do cliente.
- Tabela de venda, campos:
 - codComp: inteiro chave primária da venda.
 - CLIENTE_codCli: chave primária do cliente, para identificar qual cliente fez a compra.
 - PRODUTOS_codProd: chave primária do produto, para identificar qual produto foi vendido.
 - qtdItensComp: inteiro, para identificar a quantidade de itens vendidos.
 - dataComp: inteiro, para identificar a data da compra.
- Tabela de venda, índices:
 - Compra_fkindex1 – índice para localizar o código do produto.
 - Venda_fkindex2 – índice para localizar o código do cliente.

Esse modelo acima possui 3 erros / limitações, você conseguiu identificá-los? Veja quais são.

Alguns erros e limitações observados:

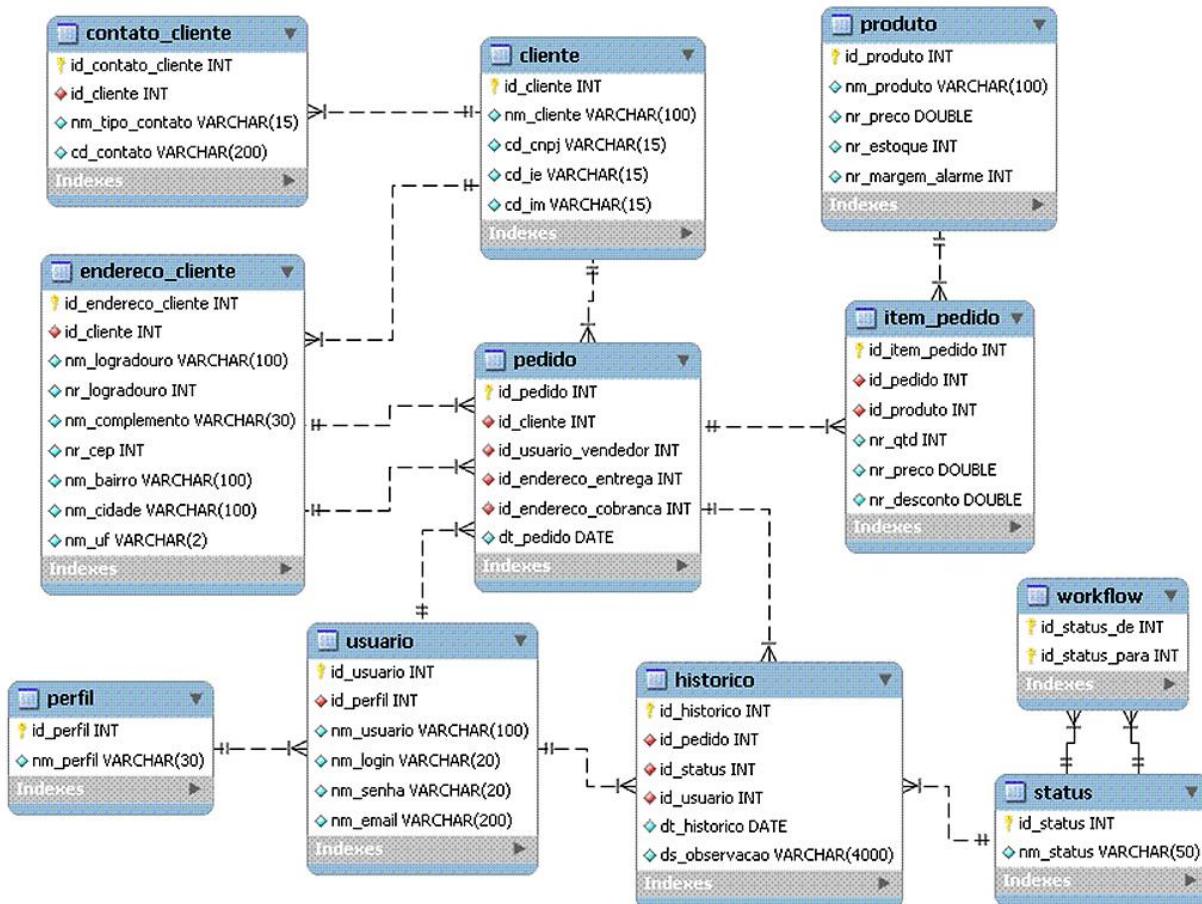
- Esse modelo de dados só permite um único produto por venda, ou seja, caso o cliente quisesse comprar dois produtos, seriam necessárias duas vendas. O correto seria possuir outra tabela para associar os produtos de cada venda (veja como resolver no próximo diagrama abaixo).
- O campo de valorProd (em Produtos), para identificar o preço do produto, utilizou um campo do tipo inteiro. Dessa forma, não é possível representar centavos, somente números inteiros. O correto seria que esse campo fosse do tipo fracionado.

O campo dataComp (em Venda) foi definido como tipo inteiro, mas uma data não é do tipo inteiro.
Para ser possível registrar a data da venda, o correto seria que esse campo fosse do tipo data/hora.

18

b) Exemplo 2

Vamos agora ver um modelo um pouco mais complexo, ele representa um sistema de vendas de uma empresa:



Exemplo de diagrama de relacionamentos e modelo de dados.

Vamos ver se você fixou bem os conceitos:

- Contato_cliente, cliente, endereco_cliente, pedido são algumas das tabelas presentes nesse modelo.
- Contato_cliente é uma tabela auxiliar que foi criada para permitir que um cliente possua várias formas de contato.
- Da mesma forma, endereco_cliente é uma tabela auxiliar que foi criada para permitir que um cliente possa ter vários endereços.
- A tabela cliente tem como campos: id_cliente (representa o código do cliente e também é a chave primária, veja o símbolo de chave ao lado de “id_cliente”), nm_cliente (que representa o nome do cliente), cd_cnpj (representa o CNPJ do cliente), cd_ie (representa o código da Inscrição Estadual), e cd_IM (Inscrição Municipal).
- A tabela contato_cliente possui os seguintes campos: id_contato_cliente (código e chave primária do cadastro do contato do cliente), id_cliente (código do cliente, utilizado para o relacionamento com a tabela de clientes), nm_tipo_contato (forma de contato) e cd_contato (descrição do contato).
- Observe que a tabela de cliente está relacionada também com a tabela de pedido, ou seja, um cliente pode efetuar um ou vários pedidos, por isso, dentro da tabela de pedido há também o código do cliente (id_cliente), utilizado para referência. Veja outros códigos nessa mesma tabela.

- A tabela de pedido, por sua vez, está relacionada com item_pedido, onde ali constam os produtos a que cada pedido está associado. E a tabela item_pedido está associada à tabela de produtos, para identificar as informações relacionadas aos produtos da empresa.
- A palavra INT, presente no modelo, refere-se ao tipo “número inteiro”, que é o tipo de informação relacionada àquele campo. Veja outros tipos de dados que aparecem no modelo.

Veja outros códigos

Nessa mesma tabela há vários outros códigos de outras tabelas utilizadas para referência, como o código do usuário vendedor, o código do endereço de entrega e o código do endereço de cobrança. Todas as referências externas têm o sinal de losango vermelho ao lado do campo correspondente.

Veja outros tipos de dados

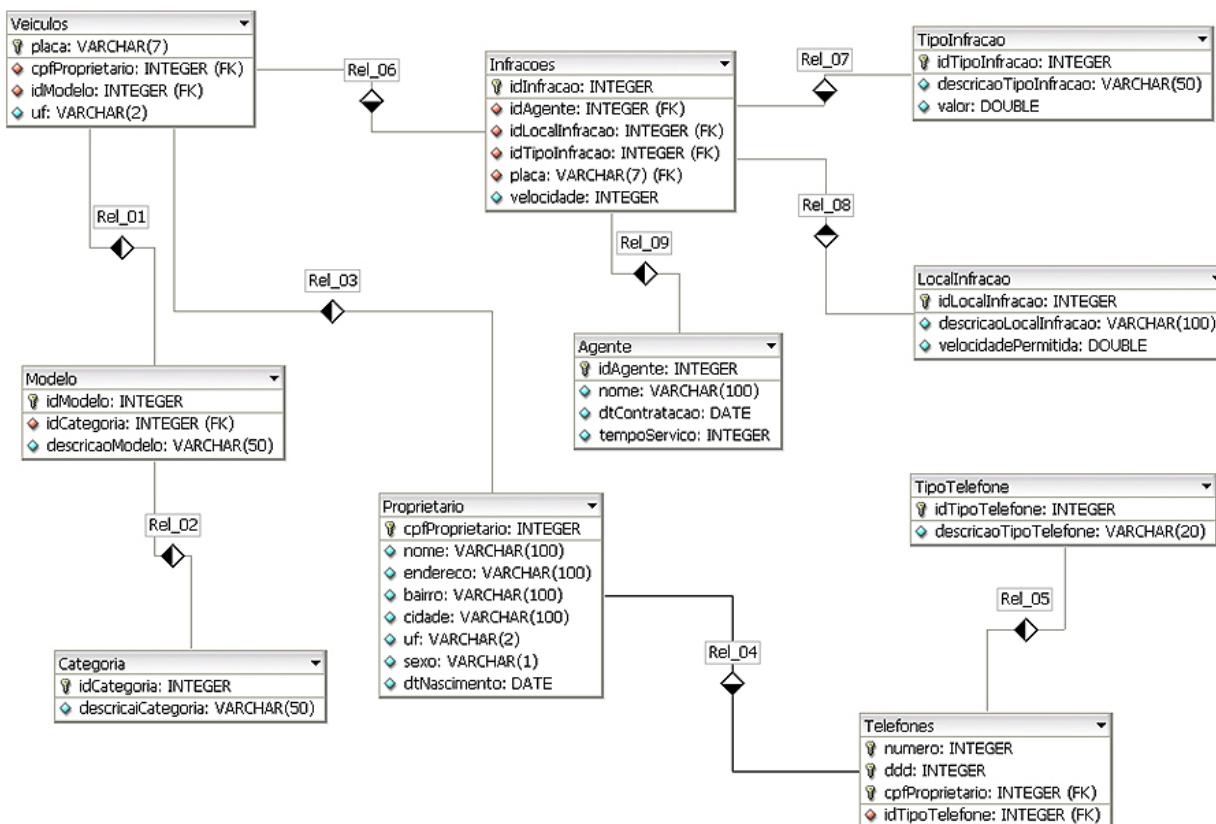
Outros tipos de dados que aparecem no modelo são:

- a) VARCHAR, para indicar um texto com até X caracteres (o número entre parêntesis expressa o limite do campo, portanto “varchar (18)” permite até 18 caracteres de texto);
- b) DATE, que indica uma data/hora;
- c) DOUBLE, que indica um número fracionado, como dinheiro, por exemplo.

19

c) Exemplo 3

Cada ferramenta de modelagem apresenta um *layout* diferente na apresentação dos diagramas de relacionamento, veja o exemplo abaixo: apesar da diferença do *layout*, você é capaz de identificar as tabelas, os campos, as chaves primárias, os relacionamentos e os tipos de dados? Faça esse exercício.

**Exemplo de diagrama de relacionamentos e modelo de dados.**

Ao analisar os nomes dos campos, conseguimos até mesmo identificar que tipo de sistema está relacionado ao modelo de dados. Esse exemplo acima, por exemplo, relaciona-se a um sistema de cadastro de multas de condutores e veículos. Você percebeu isso?

20

RESUMO

Neste módulo, aprendemos:

- Que arquivos de dados são sistemas e tecnologias que permitem o armazenamento de informações.
- Que todas as empresas atualmente utilizam sistemas de informação.
- Que os sistemas que gerenciam bancos de dados possuem a sigla SGBD.
- Que os primeiros sistemas de bancos de dados eram constituídos por simples arquivos nos quais a informação era registrada sequencialmente e de modo exclusivo.
- Que tecnologias de segunda geração permitiram a busca rápida por meio de índices e o acesso simultâneo de vários usuários.
- Que tecnologias de terceira geração juntaram o conteúdo de vários arquivos em um único arquivo de dados.

- g. Que tecnologias de quarta geração gerenciam como os usuários e aplicações acessam e manipulam os dados.
- h. Que tabela é o local onde os dados semelhantes são armazenados.
- i. Que campo refere-se a cada tipo de informação que é cadastrada em uma tabela.
- j. Que chave primária é um código que identifica exclusivamente cada registro dentro da tabela.
- k. Que tipos de dado referem-se ao tipo de informação cadastrada na tabela.
- l. Que registro refere-se a cada item de informação cadastrado em uma tabela.
- m. Os tipos de dados mais comuns são: inteiro, fracionado, texto, data, booleano e binário.
- n. Que normalização é uma técnica utilizada para eliminar redundâncias do banco de dados.
- o. Que o relacionamento é uma forma de organizar melhor os dados relacionados entre as tabelas.
- p. Que o modelo entidade relacionamento (MER) é um diagrama que representa a relação entre as tabelas do banco. Pode ainda incluir outras informações como nome e tipo dos campos, informações de chaves primárias e índices.

UNIDADE 4 – ARQUIVOS DE DADOS, EXECUTÁVEIS E BIBLIOTECAS, BAT E SCRIPTS.

MÓDULO 2 – VERSIONAMENTO DE ARQUIVOS

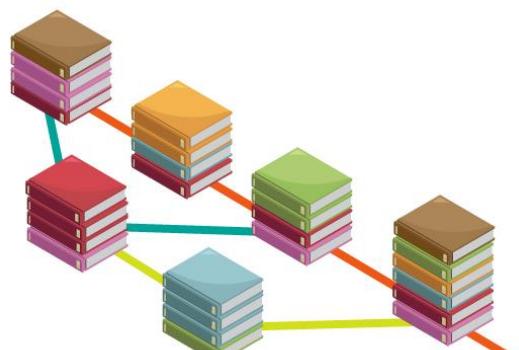
01

1 - VERSIONAMENTO DE ARQUIVOS

A partir de agora, iremos aprender características da estrutura dos arquivos.

Versionamento de arquivos e documentos refere-se ao processo de armazenar cópias dos arquivos e documentos à medida que o conteúdo deles vai evoluindo.

O controle de versionamento é um recurso interessantíssimo que permite que várias pessoas trabalhem ao mesmo tempo alterando um arquivo ou documento de modo à posteriormente ser possível juntar (mesclar) as alterações feitas por cada pessoa.

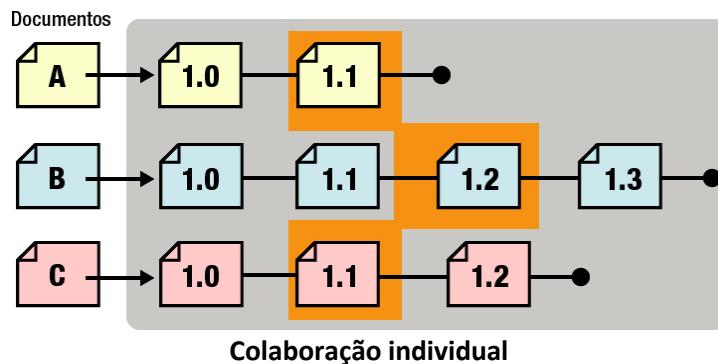


Evolução de um trabalho colaborativo

02

O trabalho colaborativo é uma realidade comum nas empresas atualmente. Sem a gestão de versionamento, só há duas formas de se trabalhar:

- **Uma pessoa por vez edita o documento.**
Cada pessoa precisa trabalhar em um documento em momentos distintos, não é possível duas pessoas trabalharem ao mesmo tempo.
- **Cada pessoa cria um documento à parte, e depois as partes são juntadas em uma versão final; ou seja, é necessário um trabalho final de mesclar cada parte do trabalho feito pela equipe.**



Essas duas formas de trabalhar são consideradas ultrapassadas, além do que gastam muito tempo e levam a erros frequentes de mesclagem de conteúdo. Muitas vezes se perde conteúdo nas mesclagens manuais.

Um sistema de controle de versão (ou versionamento) é um *software* com a finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer. Esses sistemas são comumente utilizados no desenvolvimento de *software* e também de documentação.

03

Uma **ferramenta de controle de versionamento**, comumente, possui as seguintes funcionalidades:

- Permite que todos os colaboradores recebam o conteúdo do projeto (que pode ser apenas um arquivo);
- Permite controlar quem pode ver, quem pode editar documentos e arquivos e quem pode adicionar novos documentos e arquivos;
- Permite gerenciar o status de cada documento/arquivo (exemplo: em desenvolvimento, versão para teste, testado, aprovado, etc).
- Permite gerenciar versões do documento/arquivo (exemplo: versão 1.0, 1.01, 1.02, ...).
- Permite que duas ou mais pessoas editem o mesmo documento, criando versões distintas para cada pessoa.
- Permite mesclar as diferentes versões, criando uma outra versão resultante.
- Permite, em caso de conflito, apontar opções para ajuste (há um exemplo logo a seguir neste módulo, no subitem “mesclagem”).
- Permite “desfazer” alterações feitas e consolidadas, “voltando no tempo”, em versões anteriores do documento.

- Permite comparar quaisquer versões do documento (exemplo: comparando a 12^a versão do documento com a 67^a versão).

04

1.1 - Principais focos de versionamento

Os principais usos do versionamento são para o desenvolvimento de *softwares* e para a edição de documentos.

Para **desenvolvimento de software**, as ferramentas mais comuns de versionamento são:

- *Softwares* livres/gratuitos: CVS, Mercurial, Git e SVN;
- *Softwares* comerciais/pagos: Github, SourceSafe, TFS, PVCS (Serena) e ClearCase.

O desenvolvimento de *software* livre prefere o SVN, que vem substituindo o clássico CVS. Muitas empresas também adotam o SVN, embora algumas empresas prefiram uma solução comercial, optando pelo Git Hub, ClearCase (da IBM) ou Team Foundation Server (da Microsoft). Optar por uma solução comercial geralmente está relacionada à garantia, pois as soluções livres não se responsabilizam por erros no *software* e perdas de informações, apesar das soluções livres poderem ter melhor desempenho e segurança que as comerciais. As soluções comerciais, apesar de supostas garantias adicionais não garantem o sucesso da implementação nem indenizam por qualquer tipo de erro, mesmo que comprovadamente advindo do *software*.

Já para a **edição de documentos** (de texto, apresentações, planilhas etc.), as principais ferramentas de versionamento são:

- Sistemas de GED e sistemas SIGAD (ambos devem implementar a gestão de versionamento);
- Sistemas de gestão de conteúdo como o Ms SharePoint (pago) e o Alfresco (gratuito);
- Sistemas de colaboração on-line como o Wike, Google Docs;
- Sistemas de gestão de disco e storages;
- Ou mesmo utilizando os mesmos aplicativos utilizados com foco em desenvolvimento de *software*.

04

2 - CONCEITOS RELACIONADOS

2.1 - Repositório

Normalmente, salvamos nossos arquivos em pastas do computador. Em um ambiente empresarial é comum que os documentos produzidos fiquem em um servidor de arquivos na rede da empresa.

Acessar um compartilhamento é o meio mais simples de se manter arquivos. Entretanto, a maior parte das ferramentas de gerenciamento de versão não trabalha com compartilhamentos de rede.

Elas possuem locais específicos para guardar as informações e a partir desses locais, permitem que estruturas de diretório sejam copiadas nos computadores dos colaboradores da rede ou acessadas diretamente. Independentemente da técnica aplicada (veremos mais detalhes logo adiante), o local onde ficam armazenadas as informações é chamado de repositório.

Esse repositório pode ser tecnicamente implementado como:

- Um compartilhamento de rede;
- Um banco de dados;
- Um endereço de rede;
- Um repositório na nuvem (internet).

05

Independentemente da tecnologia utilizada, o conceito de repositório pode ser implementado de diversas maneiras:

- Um repositório único para a organização;
- Um repositório por departamento;
- Um repositório por projeto;
- Um repositório por cliente;
- Ou qualquer outra organização que a empresa julgue ideal.

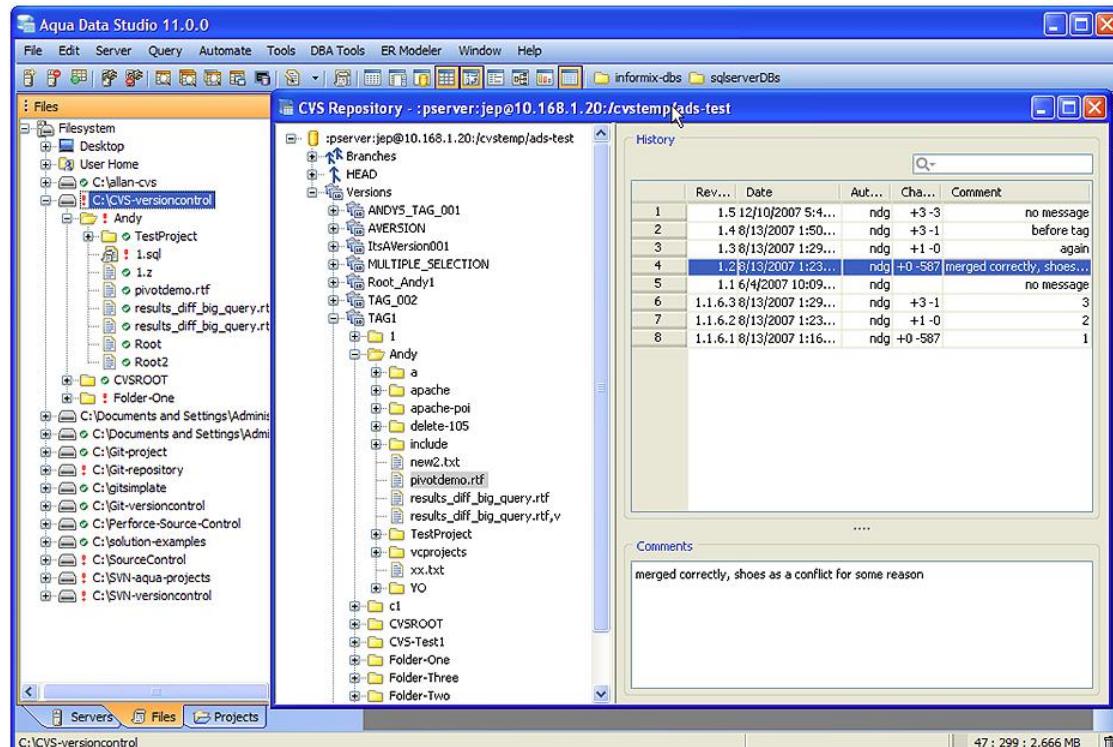


Ilustração de ambiente com vários repositórios utilizando o CVS.

Há basicamente três tipos de papéis em relação a um repositório:

- Administrador;
- Gerente de configuração;
- Usuário.

Administrador

Pessoa(s) habilitada(s) em instalar e configurar a ferramenta. Gerencia, por vezes, os backups do repositório, bem como cadastra e habilita outros usuários. Também é ela que cria novos repositórios.

Gerente de configuração

Pessoa(s) habilitada(s) em, principalmente, realizar as operações de mesclagem, acessar versões anteriores dos documentos e, quando necessário, substituir a versão atual por versões mais antigas. Por vezes também é a responsável por alocar pessoas e permissões de acesso ao(s) repositório(s) que ele gerencia. Possui também a competência de criar pastas no mais alto nível daquele(s) repositório(s) que ele gerencia. Essas pastas são utilizadas para classificar (separar) conjuntos independentes de informações, como departamentos menores ou subprojetos.

Usuário

Pessoas habilitadas a criar novos arquivos e/ou ler arquivos e/ou alterar arquivos e/ou excluir arquivos. Pode haver níveis diferentes de usuários (exemplo: usuários que só consultam ou que só podem criar arquivos em uma determinada subpasta do repositório).

06

2.2 - Métodos de controle de acesso

Veja essa situação hipotética: suponha dois usuários, João e Maria. Cada um deles decide editar o mesmo arquivo ao mesmo tempo. Se o João salvar suas alterações primeiro, é possível que a Maria poderia acidentalmente sobreescriver as alterações feitas pelo João com sua versão do arquivo. Enquanto a versão do arquivo do João se perderia para sempre, qualquer alteração que João fez não estariam presentes na nova versão do arquivo de Maria, porque ela nunca viu as modificações do João. O trabalho do João foi perdido e esta é definitivamente uma situação que nós queremos evitar!

Há duas soluções para esta questão: **travamento de arquivo** e **mesclagem**.

a) Travamento de arquivo

O travamento de arquivo é a solução mais simples para controle de acesso, ela funciona a partir do controle de bloqueio de edição de arquivo, que funciona com a sistemática: alocar-modificar-desalocar.

Esse método funciona assim: a princípio, todos os arquivos do repositório são editáveis. Mas, quando um usuário quer realmente editar um arquivo, ele efetua o bloqueio do arquivo. Nesse momento, o arquivo do repositório fica bloqueado para edição (fica marcado “apenas para leitura”), somente esse usuário pode editar o arquivo. Dessa forma, nenhum outro usuário poderá modificar a versão do arquivo que está no repositório, apenas ler seu conteúdo. Quando o primeiro usuário terminar de editar o arquivo, ele faz o envio da versão que ele tem do arquivo para o repositório e desbloqueia-o. Dessa forma, outro usuário pode agora bloquear a nova versão do arquivo para completar com outras informações. Assim, um mesmo arquivo nunca estará sendo editada por duas pessoas ao mesmo tempo.

Clique aqui para ver quais são os principais problemas do travamento de arquivo.

Clique aqui

Os principais problemas do travamento de arquivo são:

- Ao bloquear o arquivo, somente o administrador do repositório poderá desbloqueá-lo, se, por exemplo, o usuário que tiver bloqueado o arquivo viajar, o sistema manteria o arquivo travado até que o usuário volte e desbloqueie o arquivo (ou que o administrador do repositório cancele o bloqueio).
- Trabalho em série: não é possível duas pessoas trabalharem no mesmo arquivo ao mesmo tempo. Enquanto o arquivo estiver bloqueado, outras pessoas não podem trabalhar nele.

Quebra de integridade entre arquivos: suponha que João aloque e altere o arquivo A, enquanto ao mesmo tempo Maria aloca e edita o arquivo B. Suponha que os arquivos A e B dependam um do outro, e que as modificações feitas em cada um são incompatíveis. Imprevisivelmente A e B não funcionam mais juntos. O sistema de alocação não tem como prever este problema, ainda que esse sistema passe uma falta sensação de segurança. É fácil para João e Maria imaginar que alocando arquivos, cada um está seguro, numa tarefa isola, entretanto, ao gerar versões incompatíveis entre si, o sistema acabou criando um erro que precisará ser corrigido.

07

b) Mesclagem (merge)

A mesclagem (“merge” em inglês) é uma alternativa mais avançada para o controle de versionamento.

Por meio da técnica da mesclagem, os arquivos nunca ficam bloqueados: é possível, a qualquer hora, que dois ou mais usuários trabalhem no mesmo arquivo ao mesmo tempo.

Essa tecnologia funciona assim:

- Imagine um arquivo qualquer de nome X1.doc.
- Suponha que os usuários João e Pedro comecem a editar o mesmo arquivo.
- Ao salvar as edições, o sistema criará para o João uma nova versão do arquivo, por exemplo, de nome X1-João.doc, e o para as edições do Pedro, outra versão, de nome X1-Pedro.doc).

- Ao salvar ambas versões, o sistema começa a comparar os arquivos, se não houver nenhum dado conflitante, o sistema automaticamente mescla os dois conteúdos gerando outra nova versão do documento, por exemplo, de nome X2.doc.
- Entretanto, pode ser que exista algum conflito na mesclagem. Suponha que no documento X1.doc estava escrito “fruta=uva”, que no documento X1-João.doc, João tenha alterado para “fruta=banana”, e que no documento X2-Pedro.doc, Pedro tenha alterado para “fruta=maçã”. Observando essa divergência, o sistema identificará o conflito e perguntará para o Pedro qual das duas opções é a correta: “maçã” ou “banana”.
- Nesse momento, Pedro deve escolher a opção correta (ou mesmo voltar para a opção inicial: “fruta=uva”) após a escolha da informação correta, o sistema irá salvar o novo arquivo X2.doc com o conteúdo correto.
- Feito isso, a nova versão X2.doc será colocada no repositório, no computador do João e no computador da Maria.
- Ainda, nenhuma das três versões será perdida, a qualquer momento pode-se optar pelas versões: “fruta=uva”, “fruta=banana”, ou “fruta=maçã”.

The screenshot shows a GitHub commit interface. On the left, there's a sidebar with icons for History, Changes (selected), Branches, and Settings. The main area has a title "Committing to homepage_update" and a sub-section "Uncommitted Changes". It lists several files with status indicators (NEW, DELETED, SELECTED). On the right, a code diff view for "app/assets/stylesheets/github/header.scss" is shown, comparing two versions of the file. The diff highlights changes in red and green, with line numbers on the left.

```

diff --git a/app/assets/stylesheets/github/header.scss b/app/assets/stylesheets/github/header.scss
index 1234567..8901234 100644
--- a/app/assets/stylesheets/github/header.scss
+++ b/app/assets/stylesheets/github/header.scss
@@ -30,30 +30,34 @@ .header-logo-invertcat {
@@ -33,34 +33,38 @@ .header-logo-invertcat {
@@ -35,35 +35,39 @@ .header-logo-invertcat {
@@ -36,36 +36,40 @@ .header-logo-invertcat {
@@ -85,85 +85,89 @@ .header-logo-invertcat {
@@ -87,87 +87,91 @@ .header-logo-invertcat {
@@ -88,88 +88,92 @@ .header-logo-invertcat {
@@ -89,89 +89,93 @@ .header-logo-invertcat {
@@ -90,90 +90,94 @@ .header-logo-invertcat {
@@ -91,91 +91,95 @@ .header-logo-invertcat {
@@ -100,100 +100,104 @@ .notification-indicator {
@@ -102,102 +102,106 @@ .notification-indicator {
@@ -103,103 +103,107 @@ .notification-indicator {
@@ -104,104 +104,108 @@ .notification-indicator {
@@ -105,105 +105,109 @@ .notification-indicator {
@@ -106,106 +106,110 @@ .notification-indicator {
@@ -107,107 +107,111 @@ .notification-indicator {
@@ -131,131 +131,135 @@ .notification-indicator.contextually-unread {
@@ -132,132 +132,136 @@ .notification-indicator.contextually-unread {
@@ -133,133 +133,137 @@ .notification-indicator.contextually-unread {

```

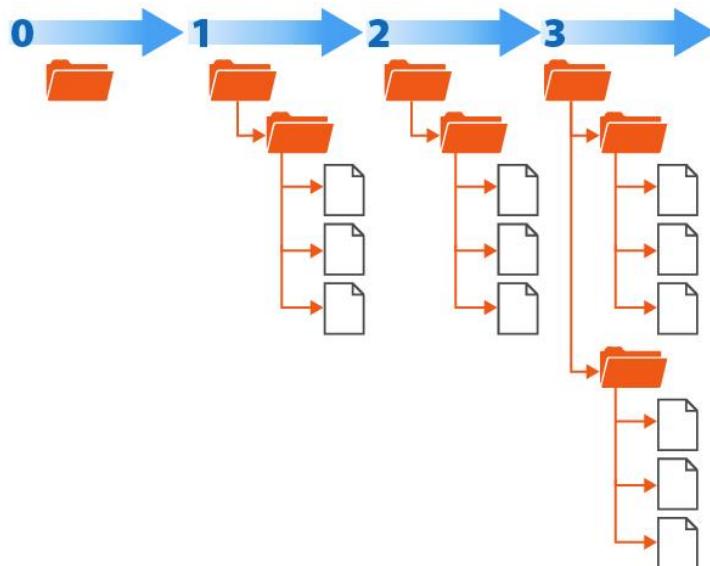
Tela de resultados após uma sincronização (merge) no GIT.

08

2.3 – Versão

Versão refere-se ao conjunto de informações que vai evoluindo dentro do repositório.

Novos arquivos e pastas são sinalizados em versões, dessa forma, é possível identificar diversos momentos do trabalho.



Exemplo de evolução da estrutura de pastas e arquivos representada em versões de 0 a 3.

As ferramentas de controle de versão gerenciam não só arquivos isolados, mas pode sim controlar várias pastas e arquivos ao mesmo tempo.

Exemplo: suponha que você está em um ambiente de trabalho desenvolvendo um projeto. Pense que nesse momento estamos na versão “1 – em desenvolvimento”. Por uma questão técnica, suponha que o gerente do projeto crie duas variantes do projeto uma de nome “2 – teste de carga” e outra “3 – teste de segurança”. Cada conjunto desses tem várias pastas e arquivos, com várias características distintas. É possível que cada usuário trabalhe com um conjunto específico e que, ao término do projeto, sejam escolhidos de cada uma das variantes, pastas e arquivos específicos que vão compor outra versão, de nome “4 – versão final”.

Dessa forma, o conceito de “backup” e “restauração” é largamente ampliado. É possível ir e vir de versões e para versões diferentes, tanto em nível de conjunto de arquivos como em nível de um arquivo específico. Isso pode parecer confuso num primeiro momento, mas é amplamente utilizado em projetos de software e controlados pelo Gerente de Configuração.

09

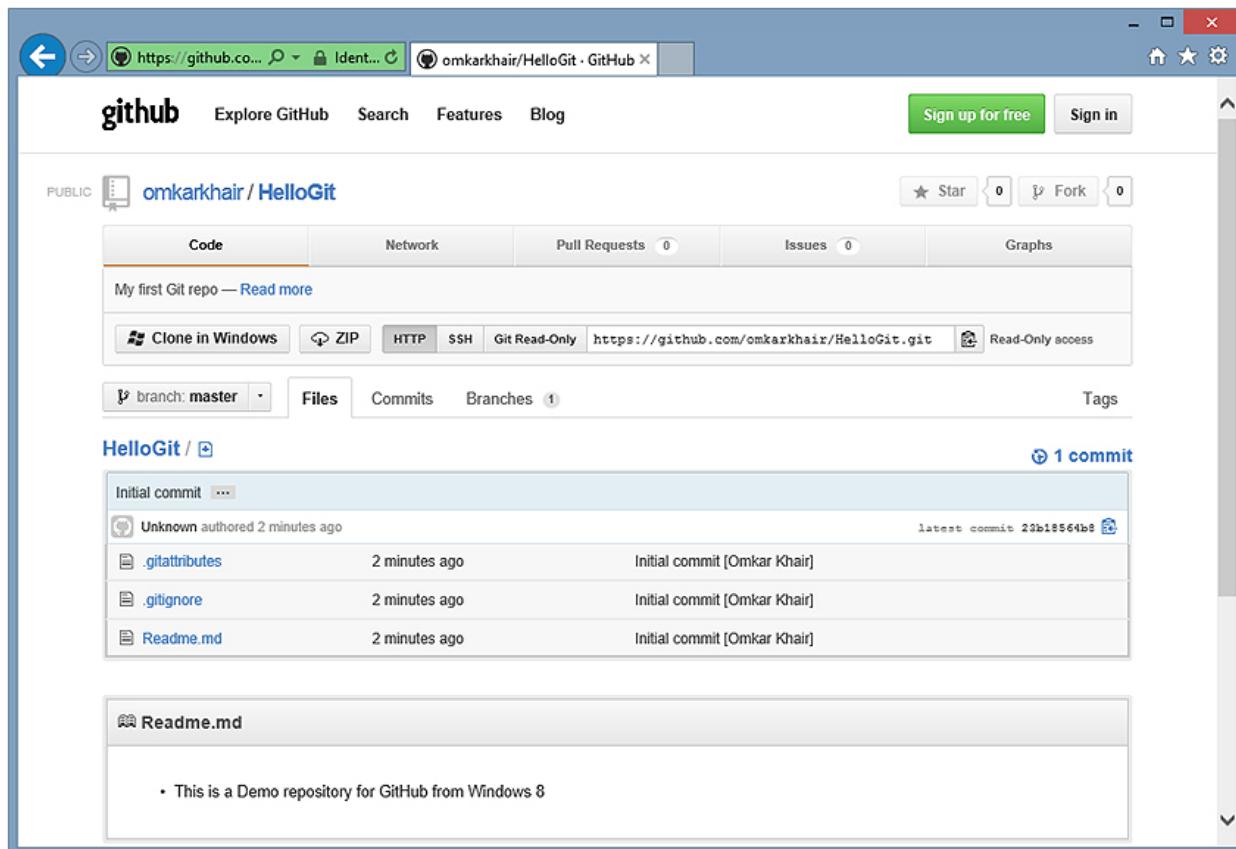
2.4 - Tipos de repositório

Há dois tipos de tecnologia de repositórios:

- **Repositórios com edição interna:** permitem a edição de arquivos diretamente dentro do repositório.
- **Repositórios com edição externa:** só permitem a edição de arquivos fora do repositório.

O repositório com edição interna é utilizado principalmente para edição de documentos de texto, planilhas, formulários e apresentações por meio de páginas Web utilizando seu navegador. Tecnologias como o Google Docs e o Wiki permitem tal tarefa. Saiba+

Já o repositório com edição externa é o mais comum de vermos, alguns *softwares* permitem a edição de documentos de texto, planilhas e outros (como o SharePoint), enquanto outros permitem a edição de documentos também, mas são voltados para ambientes de programação (como o Git, o SVN, o TFS e outros).



Tela do repositório no GitHub.

Nesse tipo de repositório, o usuário precisa efetuar uma cópia local do arquivo. Depois de editar, localmente, é necessário enviar a cópia atualizada para o repositório. Dessa forma, em um projeto, os usuários ficam fazendo várias operações de recebimento e envio de arquivos. Os usuários que acessam o repositório não acessam as alterações em tempo real, é necessário que um usuário envie suas alterações para o repositório para que outros possam consultar as novas versões.

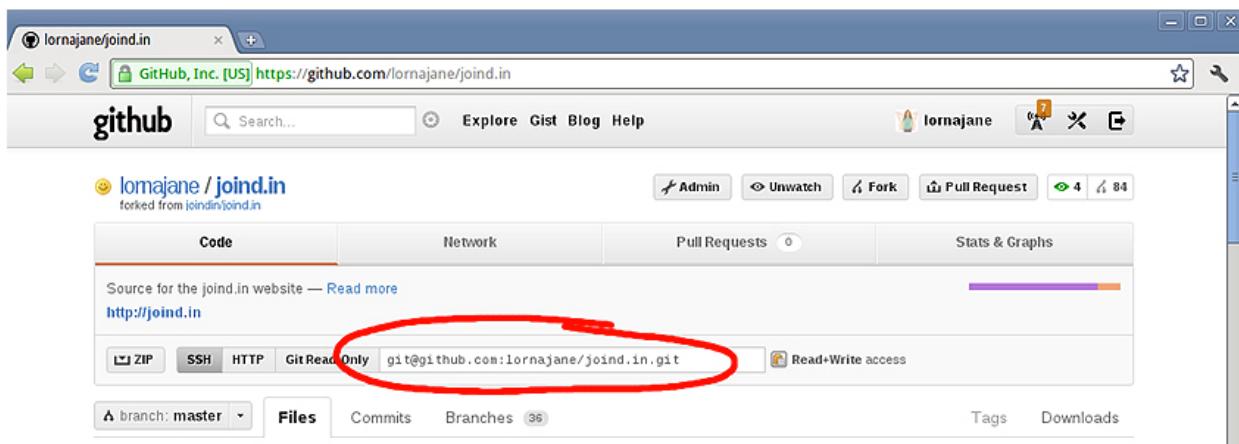
Saiba+

Acessando um determinado site, os usuários podem editar documentos simultaneamente, e, à medida que as alterações vão sendo feitas, elas vão sendo atualizadas nos navegadores de todos os usuários que estão acessando (em tempo real). Dessa forma, não há o conflito de informações.

2.5 - Conectar ao repositório

A conexão ao repositório se dá de duas formas distintas, de acordo com o tipo de repositório:

- Em um repositório de **edição interna**, ao se conectar ao repositório, todos acessam o mesmo conteúdo, e lá poderão efetuar suas atividades. Esse tipo de repositório geralmente é acessado utilizando o seu navegador, apontando-o para um determinado site.
- Em um repositório de **edição externa**, ao se conectar ao repositório, o sistema obrigatoriamente perguntará ao usuário onde ele deseja salvar os dados na sua máquina (localmente). Nesse momento, ao escolher uma pasta, o sistema irá efetuar uma cópia de todos os arquivos do repositório na máquina do usuário. Ao acessar um arquivo, o usuário acessará localmente, e não no repositório. Esse tipo de repositório é acessado por meio da ferramenta de gestão de configuração, que normalmente é instalada no computador do usuário ou por meio de uma ferramenta Web de gestão de configuração.



Configuração do acesso ao repositório no GitHub.

11

2.6 - Receber (*Checkout*)

Receber é o ato que solicita ao repositório uma cópia local do arquivo para edição.

Esse conceito só é aplicável em um repositório de edição externa (em um repositório de edição interna o que fazemos é chamado de “download” do documento).

Efetuar um “*checkout*” é sinalizar que você pretende editar o arquivo. Isso indica para o repositório que uma nova versão será criada. Futuramente, essa indicação será utilizada para efetuar uma operação de atualização ou de mesclagem no repositório.

2.7 - Enviar (*Checkin*)

Enviar (*checkin*) é o ato oposto ao receber (*checkout*). O “*checkout*” significa enviar para o repositório a versão alterada por você do documento inicial.

Ao enviar esse novo documento para o repositório, uma nova versão dele ficará lá armazenada, podendo então ser compartilhada por outras pessoas do projeto.

Caso duas pessoas tenham alterado a versão inicial, a segunda pessoa ao realizar o “checkin”, terá seu arquivo analisado pelo sistema. No caso de encontrar alguma divergência, o sistema efetuará o merge apenas depois de solucionar as informações conflitantes.

12

2.8 - Status de arquivos e pastas

Durante a utilização do sistema de controle de versão de edição externa, os arquivos locais podem assumir diferentes status e significados, sendo os principais:

- Não modificado, e atualizado;
- Localmente alterado, e atualizado;
- Não modificado, e desatualizado;
- Localmente modificado, e desatualizado.



Status dos arquivos e ícones de representação.

Não modificado, e atualizado

O arquivo local é idêntico ao do repositório, ou seja, o arquivo não foi modificado na cópia de trabalho, e nenhuma nova modificação foi submetida ao repositório.

Localmente alterado, e atualizado

O arquivo local possui modificações, e nenhuma modificação para o arquivo foi enviada para o repositório considerando a revisão atual. Existem alterações que não foram submetidas para o repositório, deste modo um “enviar” do arquivo vai obter sucesso ao publicar suas alterações.

Não modificado, e desatualizado

O arquivo local não sofreu alterações, entretanto, há modificações no arquivo do repositório; ou seja, algum outro usuário modificou e atualizou a versão do arquivo no repositório. O arquivo local deverá ser atualizado, para sincronizar com a revisão publicada no repositório. Um comando de “atualizar” do arquivo trará as últimas alterações para a cópia local.

Localmente modificado, e desatualizado

Este é caso onde a mesclagem ocorrerá. O arquivo local possui algumas modificações, e o do repositório possui outras alterações. Um “enviar” do arquivo pode falhar caso o mesclar encontre divergências: se o sistema não conseguir mesclar o conteúdo automaticamente, ele solicitará ao usuário para resolver o conflito.

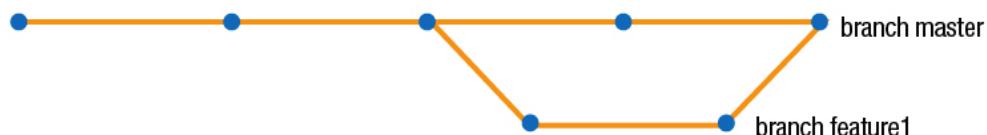
13

2.9 - Ramificação (Branch e Trunk)

Quase todos os Sistemas de Controle de Versão têm alguma forma de suporte à ramificação (branching).

Criar um branch significa que você vai clonar a última versão da linha principal de desenvolvimento e continuar a trabalhar em possíveis bugs e novas funcionalidades sem bagunçar a linha principal de produção que provavelmente está no ar e não pode receber falhas.

A última revisão, a mais atual, normalmente é chamada de “head” (cabeça). Existem momentos onde há vários “heads”, pois o projeto tem vários branches.



Exemplo de branch com dois heads (branch master e branch feature1).

Exemplo: imagine que você esteja trabalhando em um novo recurso em um branch específico. O cliente liga e diz que encontrou um bug, originado numa modificação feita na semana passada. Você sai do branch do novo recurso, entra no branch que tem o código que está em produção (que é o que o cliente está vendendo) e cria um novo branch. Você resolve o bug neste novo branch. Feito isso, você pode mover as modificações deste branch para o branch de produção e aí entra o processo de entrega (deploy) que pode mudar de empresa para empresa.

Quando tudo estiver finalizado, você volta para o branch onde estava desenvolvendo o novo recurso e pronto. Sem misturar códigos, sem modificações manuais, tudo simples e transparente para o desenvolvedor.

14

3 - PASSO A PASSO PARA O GERENCIAMENTO DE SISTEMAS DE CONTROLE DE VERSÃO BASEADOS EM EDIÇÃO EXTERNA

Independentemente do *software* que você escolha para o gerenciamento de arquivos com edição externa, é muito provável que os passos a seguir sejam aplicáveis ao seu ambiente.

3.1 - Instalação da ferramenta

O primeiro passo é escolher a ferramenta e instalá-la no seu ambiente. É bem provável que exista uma instalação específica para a versão “servidor” (que ficará no servidor) e outra versão para os usuários, que deverá ser instalada em todos os computadores dos usuários.

Algumas ferramentas modernas possuem apenas a versão “servidor”, os usuários acessam o servidor por meio de uma determinada página Web utilizando o navegador. Nesse contexto, solicite ajuda do pessoal responsável pela rede de dados, pois são eles que normalmente fazem esse tipo de atividade.

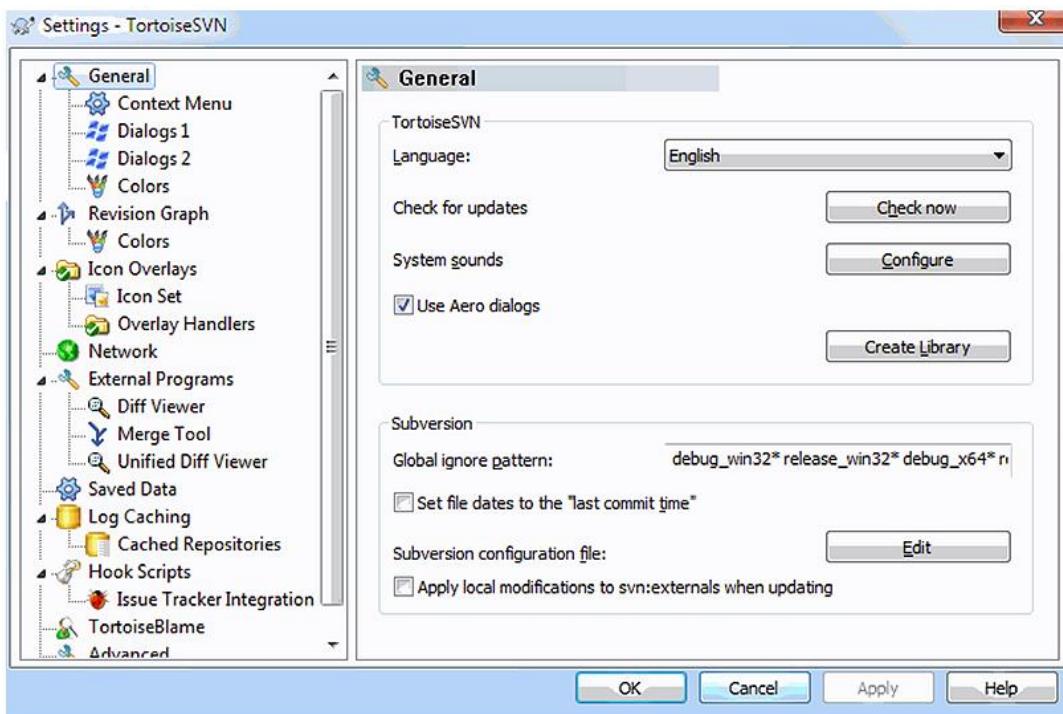
15

3.2 - Configuração básica

Após a instalação da ferramenta é bem provável que uma configuração básica da ferramenta seja necessária. Algumas ferramentas, por exemplo, solicitam:

- Criação dos perfis de acesso.
- Cadastro dos usuários do sistema.
- Definição de algumas regras de trabalho.
- Criação de espaços de trabalho.

Veja, no manual do *software* escolhido, quais são as operações de configuração inicial necessárias.



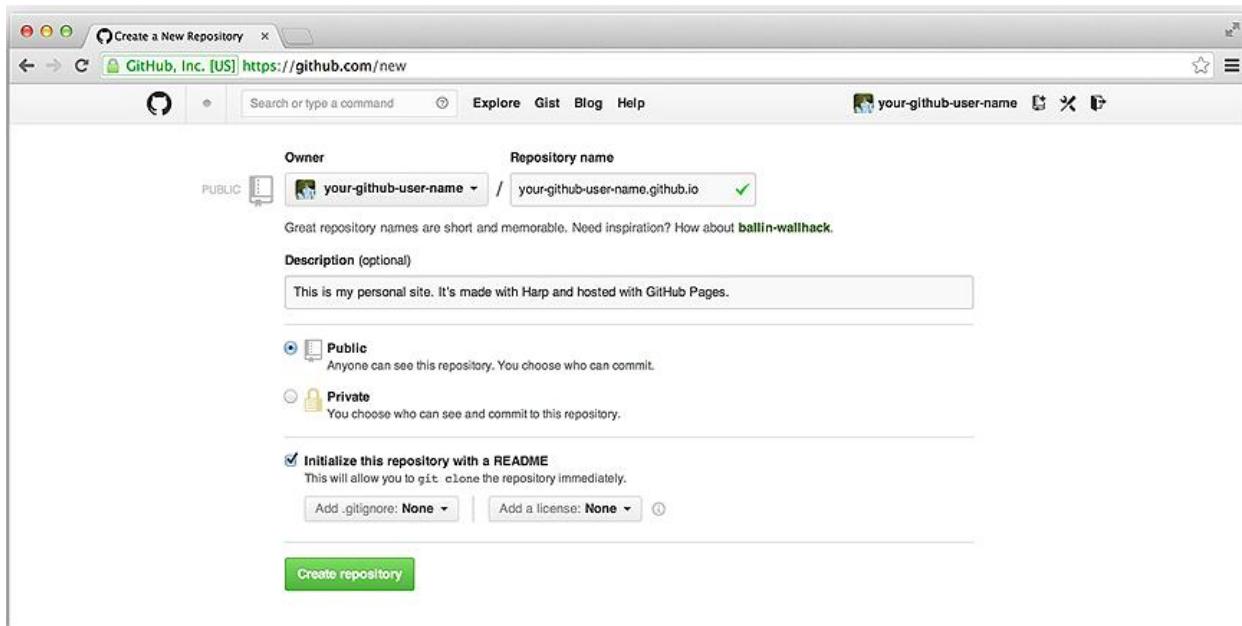
Tela de configuração do SVN.

16

3.3 - Criação do repositório

Instalada e configurada a ferramenta, ela está pronta para ser utilizada. Entretanto, é necessário haver um planejamento prévio de como a empresa utilizará o conceito de repositórios.

O conceito mais simples é a criação de um repositório único para a empresa. Essa ideia é a melhor estratégia quando a empresa possui apenas um *software* para desenvolvimento ou quando a empresa é bem pequena (algo como 5 a 10 funcionários).



Criação de um repositório do GitHub.

Entretanto, as ferramentas permitem a criação de vários repositórios. Nesse caso, é importante definir a estratégia de criação de repositórios. As alternativas mais comuns são:

- Um repositório para cada tecnologia utilizada;
- Um repositório para cada equipe;
- Um repositório para cada cliente da empresa;
- Um repositório para cada *software* desenvolvido.

Um repositório para cada tecnologia utilizada

Essa é uma opção interessante quando se precisa de ajustes diferentes da ferramenta para cada tecnologia utilizada. Exemplo: suponha que a empresa desenvolva *software*s nas tecnologias Microsoft .Net e Java. Para cada ambiente de desenvolvimento pode haver configurações específicas da ferramenta de controle de versões e do próprio ambiente de programação. Esse tipo de configuração leva à criação de poucos repositórios, geralmente dois a cinco no máximo.

Um repositório para cada equipe

Essa é uma opção interessante quando há equipes bem distintas trabalhando, em locais diferentes ou sob regimes diferentes de trabalho. Para facilitar o gerenciamento das pessoas das equipes, organizar repositórios por times de trabalho pode ser uma boa alternativa. Esse tipo de configuração leva à criação de uma quantidade pequena de repositórios, cerca de dois a cinco.

Um repositório para cada cliente da empresa

Essa é uma opção interessante quando se tem clientes com variações diferentes do *software* principal,

assim, além de um repositório principal, surge também repositórios específicos para cada cliente da empresa. Essa opção, embora bastante usada nos anos 2.000, atualmente encontra-se em desuso, pois ferramentas modernas de versionamento implementam a funcionalidade de versões variantes dentro de um mesmo repositório. Quando utilizada, há uma quantidade média de repositórios, normalmente algo entre 5 a 50.

Um repositório para cada *software* desenvolvido

Essa é a alternativa que permite a maior flexibilização e complexidade das ferramentas. É também a de maior custo operacional e de gestão dos repositórios. Poucas empresas utilizam esse método devido à grande complexidade de gestão. Nessa configuração há necessidade de várias pessoas realizarem o papel de administrador da ferramenta. Uma característica desse tipo de repositório é que cada repositório normalmente é arquivado quando se encerra o projeto. Nesse tipo de configuração, pode haver centenas de repositórios criados.

17

Dependendo da tecnologia utilizada, criar um repositório pode significar:

- A criação de um novo banco de dados em um sistema gerenciador de banco de dados ou apenas criar novos registros dentro de um banco de dados.
- Criação de uma nova pasta no local escolhido como repositório de arquivos.

Após a criação de um repositório, normalmente é necessário realizar as seguintes operações:

- Configurar as regras de acesso e utilização do repositório.
- Criar perfis de acesso.
- Cadastrar usuários e alocá-los a perfis de acesso.
- Permitir o acesso de usuários ao(s) repositório(s) criado(s).
- Autorizar o uso geral do repositório (um repositório pode ser bloqueado temporariamente para acesso).
- Fazer cargas iniciais do repositório ou criar estruturas de arquivo padrão a serem utilizadas (normalmente cria-se uma estrutura padrão para apontar os locais onde os desenvolvedores irão colocar seus arquivos).
- Caso seja utilizado um repositório para vários projetos, criar a estrutura de diretórios dos projetos.
- Configurar, se necessário, as regras de acesso para cada pasta e/ou cada projeto.
- Configurar como os usuários irão acessar o(s) repositório(s). Há várias formas de acesso e você precisa configurar quais formas estarão autorizadas. Por exemplo, você pode autorizar apenas o acesso via protocolo HTTPS, proibindo as demais formas.
- Comunicar aos usuários como eles devem acessar o repositório.
- Verificar se os usuários estão conseguindo acessar o repositório.
- Verificar se as ferramentas de auditoria e controle estão funcionando.
- Testar todas as configurações e funcionalidades a serem utilizadas, principalmente aquelas voltadas ao controle de segurança.

- Criar rotinas de backup do repositório. Se o servidor falhar, você terá condições de acessar uma versão recente de seus arquivos, mas sem o repositório, todo o histórico estará perdido para sempre. Se possível, teste a restauração dos backups em um ambiente de teste.
- Formalizar e comunicar à chefia ou pessoa competente que o ambiente está pronto para uso.

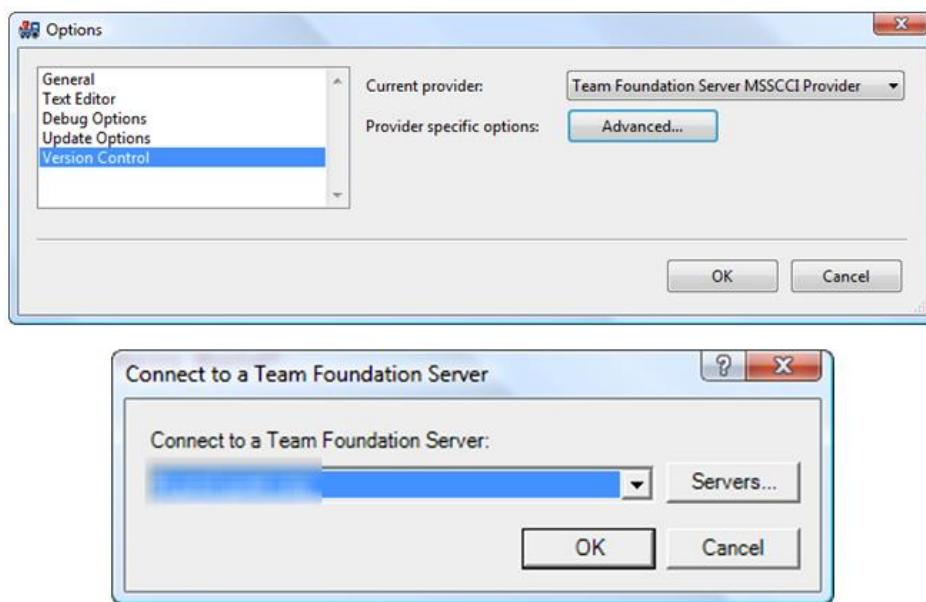
17

3.4 - Verificação periódica do funcionamento dos repositórios

Como normalmente diversas pessoas acessam os repositórios, é importante que você monitore os repositórios para garantir que eles estão acessíveis. Há várias maneiras de implementar esse monitoramento: desde uma rotina manual, onde regularmente o administrador acessa e verifica alguns arquivos dos repositórios, ou até mesmo implementando rotinas automatizadas de monitoramento e informação. A equipe de redes da empresa poderá lhe ajudar a decidir a melhor estratégia para realizar essa tarefa.

3.5 - Configuração local e carga inicial

A configuração local geralmente é bastante simples. De posse dos dados de acesso, o usuário define onde localmente os arquivos serão copiados e onde sua ferramenta de desenvolvimento irá acessar. De posse dessa informação, o usuário configura o acesso ao repositório.

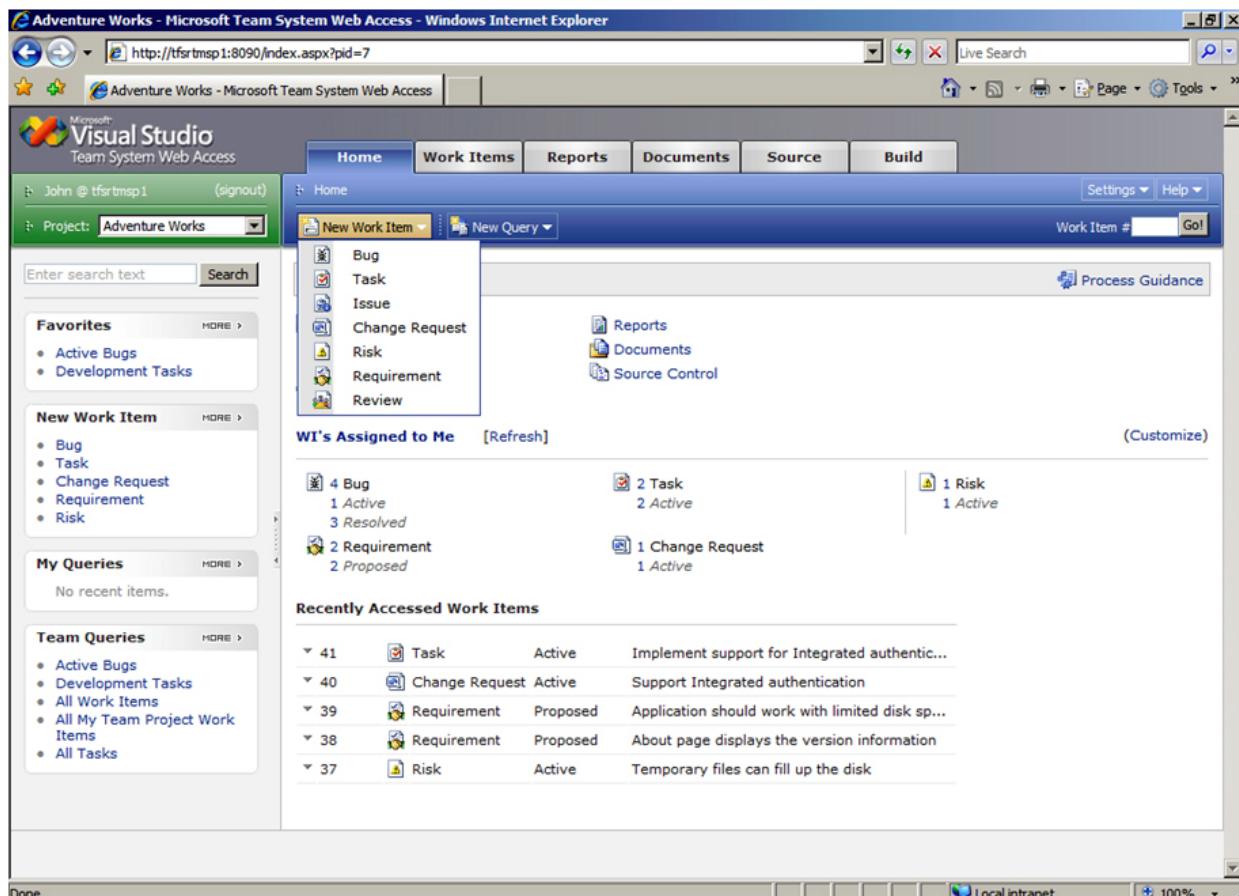


Conectando ao TFS

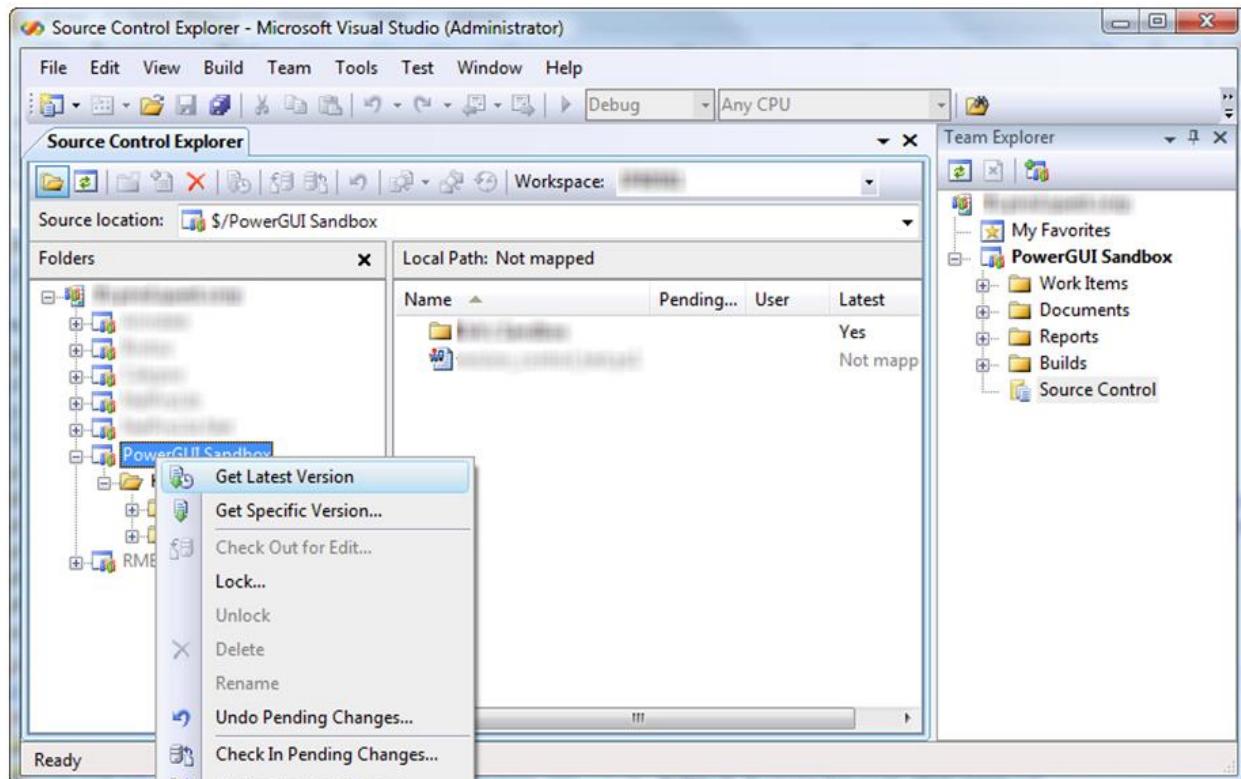
18

Ao se conectar pela primeira vez ao repositório, as ferramentas normalmente perguntam onde elas devem copiar a estrutura inicial de pastas e arquivos. Após o usuário informar o local, o sistema irá

descarregar toda a estrutura do projeto. Dependendo do método de acesso, banda de rede e tamanho do projeto, essa atividade pode levar vários minutos (em certos casos até mesmo várias horas).



Tela principal do TFS.

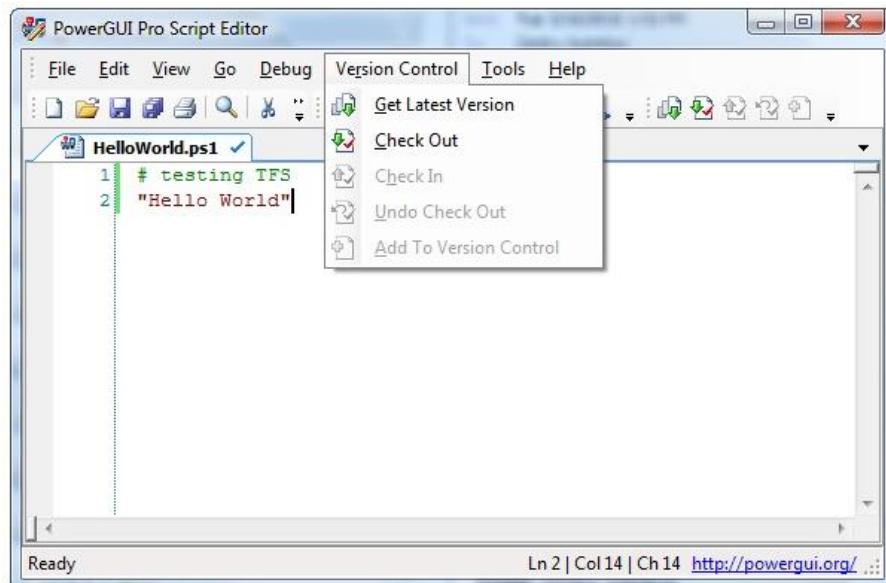


Obtendo os arquivos do repositório do TFS dentro do Visual Studio

20

3.6 - Uso do sistema de gerenciamento e controle de versões

Após o término da carga inicial, o sistema encontra-se pronto para operar. Normalmente, um teste simples de sincronização é bem interessante de se fazer nesse momento.



Teste inicial de validação do controle de versão.

Concluído o teste, o ambiente encontra-se pronto para o uso. Cabe agora ao desenvolvedor realizar as atividades de recebimento, envio e sincronização conforme estipulado no padrão de trabalho da empresa.

21

Para algumas atividades, como geração de versão, pode ser que o administrador da ferramenta ou o gerente de configuração seja a pessoa com autoridade para fazer essas configurações. Nesses casos, peça orientações e ajuda a eles sobre como você deverá trabalhar com o sistema.

Resta agora a realização das rotinas periódicas em relação ao ambiente (já descritas anteriormente), que se resumem em:

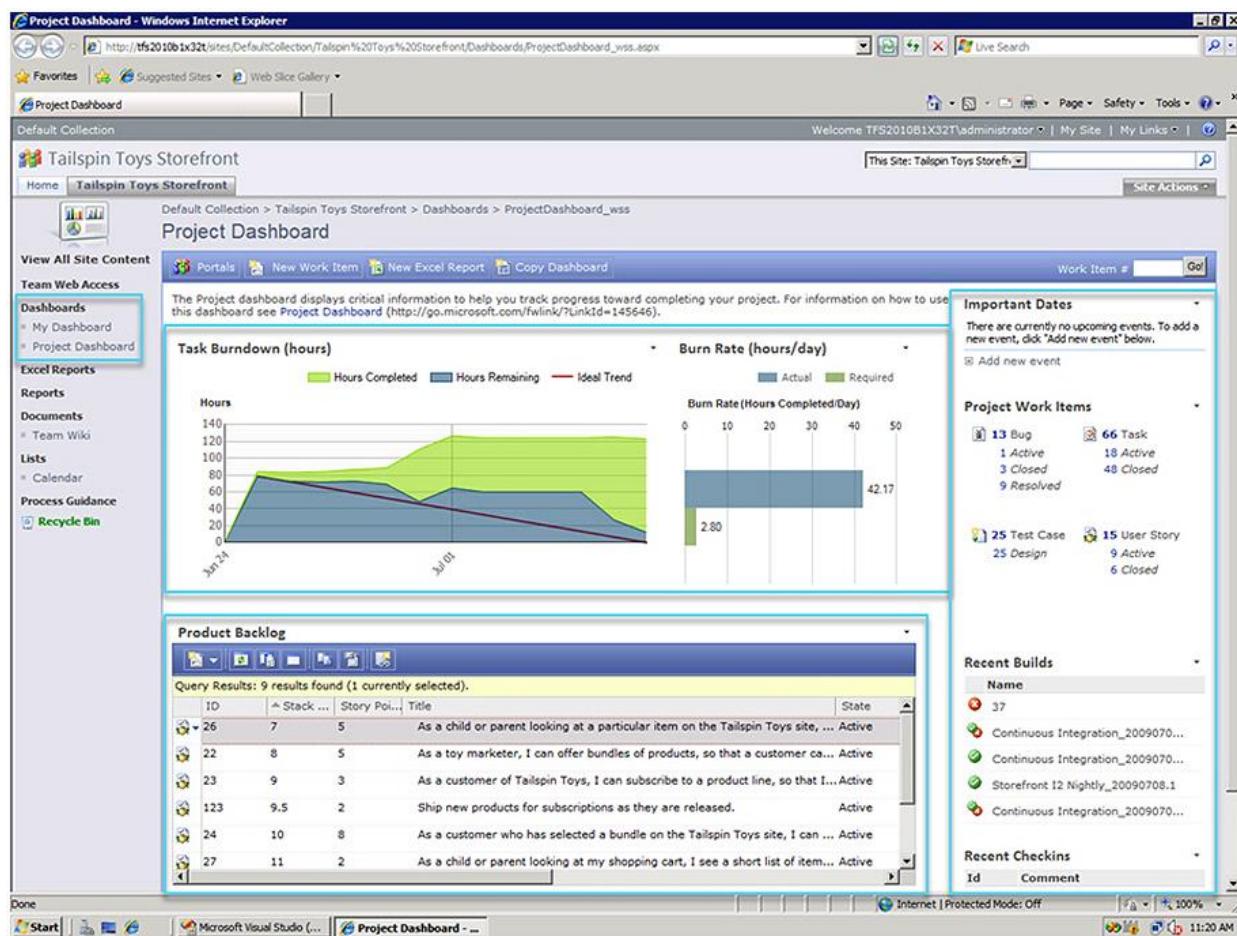
- Importar dados para um repositório (normalmente vazio ou conceitualmente vazio).
- Obter cópia local dos arquivos do repositório.
- Atualizar cópia local com arquivos alterados por outros usuários.
- Atualizar a cópia local com a última versão do repositório.
- Realizar um checkout com bloqueio do arquivo (somente em ferramentas antigas de versionamento).
- Submeter alterações ao repositório (checkin).
- Sincronizar o arquivo junto ao repositório, efetuando mesclagem e ajustando conflitos.
- Obter informações de status dos arquivos e pastas.
- Criar novos arquivos e adicioná-los ao controle de versão.
- Visualizar diferenças entre arquivos.
- Desfazer alterações.
- Criar versões.
- Voltar versões.
- Listar mudanças e versões.

- Auditando mudanças.
- E demais operações.

22

3.7 - Gerenciando projetos e repositórios

Gerentes de projetos, clientes, patrocinadores, administradores da ferramenta e outras pessoas interessadas podem requerer acesso ao projeto no sistema de versionamento de arquivos. Algumas ferramentas implementam relatórios e painéis que apresentam muitas informações gerenciais dos repositórios ou mesmo dos projetos que lá estão. A tela a seguir mostra um exemplo de um painel de informações do projeto no TFS.



Painel de projeto no TFS.

23

RESUMO

Neste módulo, aprendemos que:

- a. O versionamento é uma funcionalidade que permite o controle de versões dos arquivos à medida que vão evoluindo, permitindo assim o controle de cada momento durante seu desenvolvimento.
- b. O versionamento permite o compartilhamento do mesmo arquivo para que duas ou mais pessoas trabalhem nele ao mesmo tempo, sem gerar perda de informação.
- c. As funcionalidades básicas do gerenciamento de versões são: compartilhamento, controle de alteração, mesclagem, resolução de conflitos e auditoria.
- d. Há basicamente dois tipos diferentes de sistemas de gerenciamento de versões, um aplicável principalmente para código fonte de programação e outros para documentos de escritório.
- e. Repositório é o local onde são armazenadas todas as informações pertinentes ao controle de versão.
- f. Independentemente da tecnologia utilizada, o conceito de repositório pode ser implementada de diversas maneiras.
- g. O travamento de arquivo é a técnica mais simples para o versionamento, mas ela apresenta uma série de desvantagens como, por exemplo, a impossibilidade de edição simultânea.
- h. A mesclagem é a técnica mais moderna de versionamento e permite a edição simultânea.
- i. Há dois tipos de edição em repositórios: edição interna e edição externa.
- j. Existem ferramentas de versionamento web e desktop.
- k. Os arquivos locais referentes a um repositório normalmente possuem um dos seguintes estados: não modificado e atualizado; localmente alterado e atualizado; não modificado e desatualizado; e localmente modificado e desatualizado.
- l. Ramificações são formas de clonar uma versão do repositório para uma tarefa pontual e a parte.

UNIDADE 4 – ARQUIVOS DE DADOS, EXECUTÁVEIS E BIBLIOTECAS, BAT E SCRIPTS. MÓDULO 3 – SISTEMAS DE ARQUIVOS WINDOWS, LINUX E APPLE

01

1 - SISTEMAS DE ARQUIVOS

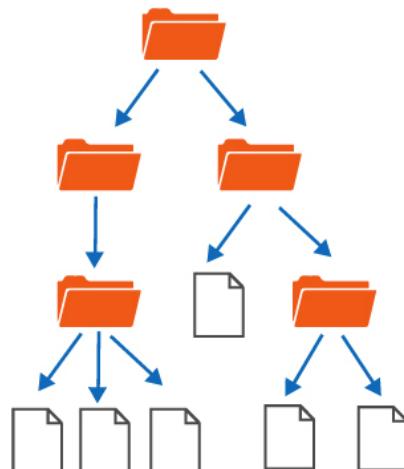
Olá, seja bem-vindo a mais uma etapa do nosso estudo. Aprenderemos, a partir de agora, as características dos sistemas de arquivos do Windows, Linux e Apple.

Já vimos que todas as aplicações precisam armazenar e recuperar informações. Além disso, as informações mantidas na memória RAM são perdidas com o encerramento do programa. Ao desligar o computador, todos os dados em memória RAM também são perdidos. A solução para armazenar a informação em discos é criar arquivos. A informação deve ser persistente, ou seja, um arquivo deverá ficar armazenado em disco até que o usuário o descarte. O sistema de arquivos (parte do sistema operacional) é responsável pelo modo como os dados são estruturados, nomeados, acessados, usados, protegidos e implementados.

Um **sistema de arquivos** é a técnica (regras e característica) pela qual um sistema operacional utiliza para administrar arquivos em uma unidade lógica (disco, disquete, pendrive, fita de *backup*, CD, DVD etc.), ou seja, a forma pela qual os arquivos estão organizados.

Sabendo interpretar o sistema de arquivos de um determinado disco, o sistema operacional pode decodificar os dados armazenados e lê-los ou gravá-los.

Podemos fazer uma analogia do sistema de arquivos com uma biblioteca: o bibliotecário organiza os livros conforme um padrão, cuja busca, convenientemente, procura deixar mais fácil, sem ocupar muitas prateleiras. Ele ainda organiza os livros segundo suas características (assunto, tema, autor etc.). Depois de organizados, ou durante a organização, o bibliotecário cria uma lista com todos os livros da biblioteca, com seus assuntos, localizações e códigos respectivos.



02

Aplicando a analogia à informática, o sistema operacional seria o bibliotecário da "biblioteca de arquivos e diretórios" do computador. Exatamente igual à organização de uma biblioteca, o sistema operacional guarda os dados nos espaços vazios do disco. Sabendo a posição do arquivo a ser aberto/gravado, o sistema operacional solicita a leitura desta, decodifica/codifica e realiza a abertura/gravação do dado.

O sistema de arquivos consiste em duas partes distintas: uma coleção de arquivos, cada um deles armazenando dados relacionados, e uma estrutura de diretórios, que organiza e fornece informação sobre todos os arquivos do sistema.

Dentro de qualquer tipo de sistema de arquivos há somente dois tipos de objetos: ou é diretório ou é arquivo.

Contudo, armazenar arquivos organizados em diretórios já não satisfaz as necessidades de empresas e organizações quando falamos sobre necessidades de gestão de arquivos. Por conta disso, novas tecnologias de armazenamento de arquivos trazem recursos importantes e muito utilizados na grande maioria das empresas:

- Possibilidade de compressão,
- Controle de acesso em nível de pasta e arquivos,
- Controle de cotas: outro recurso muito utilizado,
- Criptografia.

Possibilidade de compressão

Alguns sistemas de arquivos possuem recursos de compressão de dados, com isso, podemos armazenar mais dados no mesmo espaço lógico. Esse recurso funciona como um sistema de compactação e descompactação transparente e automático. Na prática consegue-se um ganho nominal na ordem de 50% de compactação, ou seja, dobramos o espaço disponível. Entretanto, esse recurso consome um tempo de processamento (para compactar e descompactar), o que pode levar o sistema em geral a ficar mais lento. Devido ao barateamento do geométrico do custo por MegaByte dos discos rígidos,

alinhado à duplicação periódica do tamanho padrão de discos rígidos (hoje o tamanho padrão é de 1 TeraByte), essa tecnologia torna-se cada vez mais subutilizada.

Controle de acesso em nível de pasta e arquivos

Esse sim é um recurso altamente utilizável, controlar o que cada funcionário (ou perfil de funcionário) pode ou não acessar, pode ou não alterar, pode ou não criar. Esse controle de acesso não só permite a criação de documentos com certo grau de sigilo, mas protege-nos também de eventuais erros de usuários que apagam informações importantes. Todos os sistemas operacionais modernos permitem o controle de acesso a pastas e arquivos.

Controle de cotas: outro recurso muito utilizado.

Refere-se à capacidade de separarmos cotas de utilização do espaço em rede (storage ou discos rígidos) de forma a garantir um mínimo de espaço a todos os usuários. Sem controle de cotas, um usuário, por exemplo, poderia utilizar todo o espaço de disco da rede inadvertidamente, fazendo com que outros usuários não tivessem mais espaço para criar seus próprios arquivos.

Criptografia

Este é um recurso interessante e pouco utilizado, refere-se a um sistema de criptografia que impede o acesso Externo. Em outras palavras, somente os usuários autenticados pela rede tem como acessar o conteúdo dos arquivos da rede. Se um disco rígido foi roubado do servidor a fim de se obter acesso Externo às informações, por estarem criptografadas, esse usuário Externo não conseguiria abrir os documentos.

03

1.1 - Implementação de arquivos

Para criar arquivos é fundamental que o sistema operacional tenha controle de quais áreas ou blocos no disco estão livres. Também é importante saber que blocos de discos estão relacionados a que arquivos. Este gerenciamento pode ser feito de várias maneiras: alocação contígua, alocação por lista encadeada, alocação indexada e alocação combinada. Vejamos cada uma a seguir.

a) Alocação contígua

É o esquema mais simples de alocar e armazenar os arquivos no disco. Consiste em armazenar um arquivo em blocos sequencialmente dispostos.

Neste tipo, o sistema localiza um arquivo através do endereço do primeiro bloco e da sua extensão em blocos.

Vantagens	Desvantagens
<p>Este tipo de alocação apresenta duas vantagens significativas:</p> <ul style="list-style-type: none"> • É bastante simples de implementar e de realizar o controle sobre onde os blocos estão. • O desempenho de leitura é excelente, pois todo o arquivo pode ser lido em uma única operação a partir do primeiro bloco de dados. 	<p>No entanto este tipo de implementação apresenta uma grande problema: fragmentação do disco.</p> <p>Como os arquivos podem ser criados e eliminados repetidamente, os segmentos livres vão fragmentando-se em pequenos pedaços por todo o disco. O problema pode tornar-se crítico quando um disco possui blocos livres disponíveis, porém não existe um segmento contíguo em que o arquivo possa ser alocado. Clique aqui para saber como resolver esse problema.</p>

Clique aqui

O problema da fragmentação pode ser contornado através de rotinas que reorganizem todos os arquivos no disco de maneira que só exista um único segmento de blocos livres. Este procedimento, denominado desfragmentação, geralmente utiliza uma área de trabalho no próprio disco ou em fita magnética.

04

b) Alocação por lista encadeada

Esse tipo de alocação consiste em organizar os arquivos, cada um, como uma lista encadeada de blocos de disco. Dessa forma, uma parte de cada bloco é usada como ponteiro para o próximo bloco. O restante do bloco é usado para dados.

Vantagens	Desvantagens
<p>Uma vantagem desse tipo de alocação é que o tamanho do arquivo não necessita ser conhecido antes de sua criação, já que cada bloco terá um ponteiro para o próximo bloco. Assim o fato de o disco estar fragmentado não acarreta problemas para a criação do arquivo.</p>	<p>O problema deste tipo de alocação é o tempo de leitura extremamente lento. O acesso deverá ser sempre sequencial. Assim, para acessar um bloco intermediário será necessário percorrer o arquivo desde o início.</p>

05

c) Alocação indexada

A alocação indexada é uma maneira de resolver o problema do tempo de leitura da alocação por lista encadeada, o que pode ser feito por meio de uma tabela de endereços dos blocos ocupados pelo arquivo. Para cada novo bloco alocado, inclui-se mais um novo item na tabela.

Vantagens	Desvantagens
Uma grande vantagem da alocação indexada é que o acesso aleatório fica mais fácil, pois não é necessário percorrer todos os blocos de forma sequencial, o endereço de cada bloco fica armazenado na tabela de alocação.	A desvantagem desse método é que a tabela alocação deverá ser mantida na memória principal e dependendo do tamanho dos blocos do disco ocupará muito espaço de memória o tempo todo.

d) Alocação combinada

A alocação combinada abrange o uso de blocos de índices e de encadeamento dos mesmos. Essa técnica combina a baixa ocupação de espaço em memória da técnica de lista encadeada e o bom desempenho da técnica de tabela de índices.

Na alocação combinada é usada uma técnica chamada de **níveis de interação** na indexação. Nesta técnica, parte de um bloco é utilizado para apontar diretamente blocos de dados, e a outra parte é utilizada para apontadores de blocos. Se usarmos outros níveis de apontadores, podemos estender essa mesma técnica para apontadores duplamente indiretos e apontadores triplamente indiretos.

Saiba o que é um apontador direto e um apontador indireto.

Esta técnica é comumente utilizada na implementação do sistema de arquivos Unix, chamada de I-nodes. Um I-node (index-node) constitui os metadados, armazenados em estrutura de dados própria, que relaciona atributos e os endereços dos blocos de um arquivo.

Apontador direto

É aquele que aponta para o local no disco onde cada bloco do arquivo está armazenado. É o método mais simples de índice, também é o que suporta arquivos de menor tamanho. Como são 13 entradas de índice (0 a 12), cada um apontando para blocos de dados de 4K, o tamanho máximo de um arquivo mantido por esta estrutura é de 40KBytes.

Apontador indireto

Como a tabela de índices tem um tamanho finito (13 posições), muitas vezes o tamanho desta tabela não é suficiente para criar índices para todos os blocos de um arquivo maior do que 40Kbytes, dessa forma, um apontador indireto, ao invés de apontar para um pedaço do arquivo, aponta para um local que na verdade é um índice para outros pedaços do arquivo. O que você faz é transformar um pedaço do disco em uma extensão do índice. Um apontador indireto suporta 1024 apontadores, o que permite indexar um arquivo de até 4MBytes.

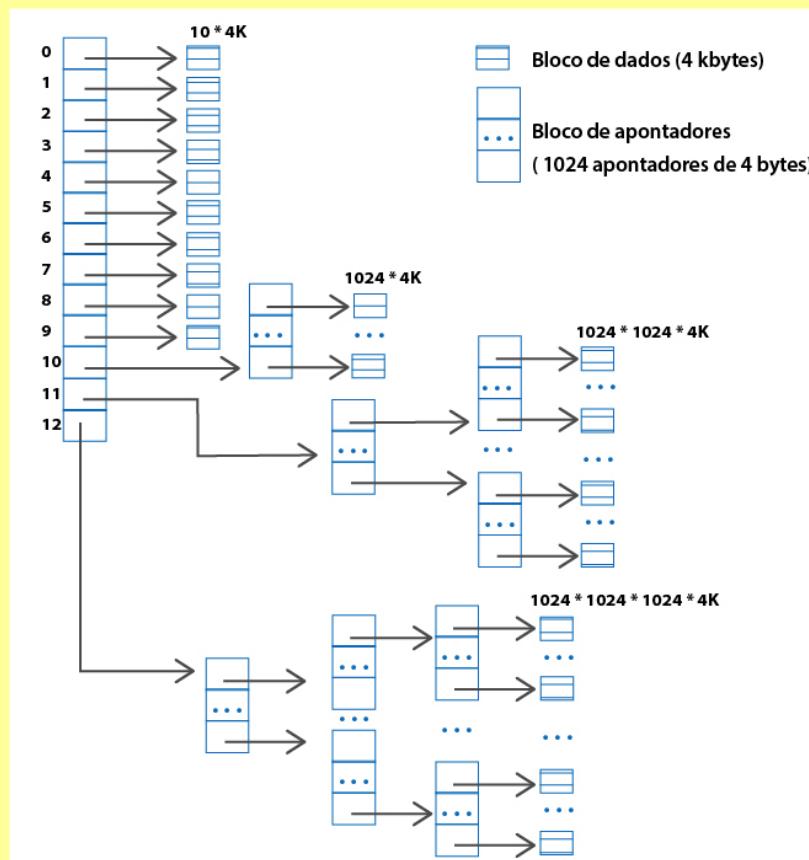
Apontadores duplamente indiretos

Possui um apontador que aponta para outro arquivo apontador, ou seja, dois níveis hierárquicos, com isso é possível ter 1024 índices em primeiro nível que cada um aponta para outros 1024 endereços.

Dessa forma é possível referenciar arquivos de até 4Gbytes de tamanho.

Apontadores triplamente indiretos: neste tipo de apontador, aparece um terceiro nível de hierarquia, ou seja os 1024 índices de segundo nível tem cada um mais 1024 apontadores. Com isso, podemos mapear arquivos de até 4 Tera bytes de tamanho. Hoje em dia, esse é o tamanho máximo previsto para um arquivo.

A figura abaixo apresenta uma tabela de índices de 13 posições (0 a 12), note que os apontadores 0 a 9 são apontadores diretos. O apontador 10 é um apontador indireto. O apontador 11 é um apontador duplamente indireto. O apontador 12 é um apontador triplamente indireto.



06

1.2 – Diretórios

O diretório é uma estrutura de dados que contém entradas associadas aos arquivos onde são armazenadas informações como localização física, nome, organização e demais atributos; ou seja, um diretório (ou subdiretório) é simplesmente outro arquivo, mas é tratado de modo especial.

Todos os diretórios têm um bit em cada entrada de diretório que define a entrada como um arquivo (0) ou um subdiretório (1). O sistema operacional, ao criar um arquivo ou diretório, configura o bit de identificação como 0 (zero) para informar que o item é um arquivo ou seta o bit como 1 (um) para

informar que é um diretório. Quando um arquivo é aberto, o sistema operacional procura a sua entrada na estrutura de diretórios, armazenando as informações sobre atributos e localização do arquivo em uma tabela mantida na memória principal.

Há, basicamente, duas organizações possíveis para as entradas:

- Cada entrada contém o nome do arquivo, seus atributos e os endereços no disco onde ele está armazenado;
- Nome do arquivo e um ponteiro para outra estrutura de dados, onde podem ser encontrados os atributos dos arquivos.

Independente da forma adotada para a implementação de arquivos, a principal função do diretório é mapear o nome ASCII do arquivo na informação necessária à localização do dado.

07

1.3 - Partição

É possível dividir um disco rígido em várias partes ou partições. Uma partição, portanto, é uma divisão do espaço de um disco rígido.

Cada partição independente das outras, ou seja, cada uma pode ter o seu próprio sistema de arquivo, sem interferir nas outras partições. Podemos, por exemplo, instalar o Linux em uma partição e o Windows em outra partição.

Um disco pode ser dividido em até quatro partições. Uma partição pode ser primária ou estendida. Sendo que, no máximo, apenas uma partição pode ser do tipo estendida. Isto significa que você pode ter 4 partições primárias ou 3 partições primárias e uma partição estendida. É possível dividir uma partição estendida em partições menores chamadas de partições lógicas (a partição estendida não armazena dados e sim, outras partições lógicas). Não é possível, entretanto, dividir uma partição primária. A tabela onde são armazenadas as informações sobre as partições fica no primeiro setor do disco e chama-se **MBR** (*Master Boot Record*). Por questões históricas, esta tabela possui apenas 4 entradas onde cada entrada descreve uma única partição.

Partição primária

É a primeira partição criada, aquela que a BIOS do computador procura o sistema operacional a ser carregado. Um disco precisa ter ao menos a sua partição primária. Num computador baseado em Windows, essa partição normalmente leva o nome de “C”.

Partição estendida

Opcionalmente, um disco pode possuir outras partições. Criamos partições para organizar melhor as informações que queremos armazenar. Por exemplo, em um computador com o sistema operacional Windows, você poderia deixar a partição “C” para o sistema operacional, a partição “D” para programas e a partição “E” para filmes e músicas. Todas as partições a mais que você cria são

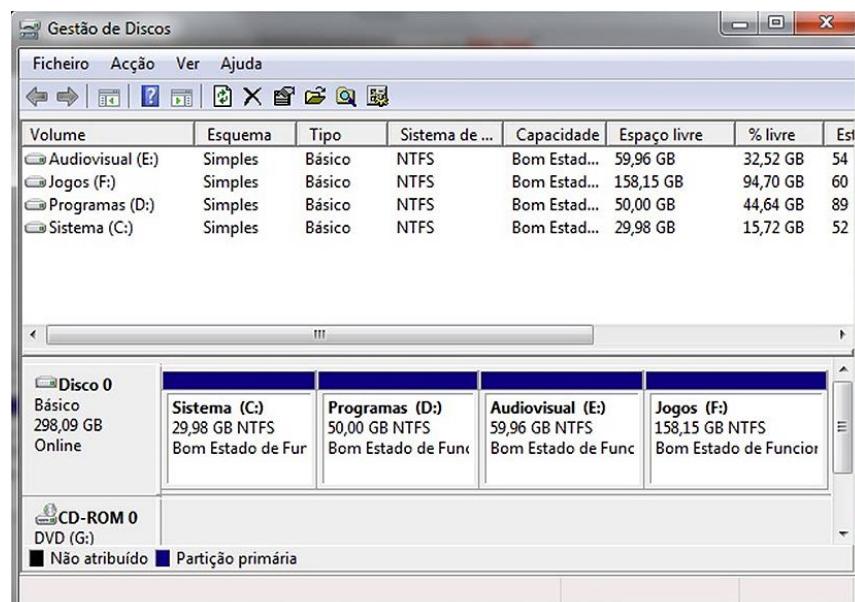
chamadas de partições estendidas.

MBR (Master Boot Record)

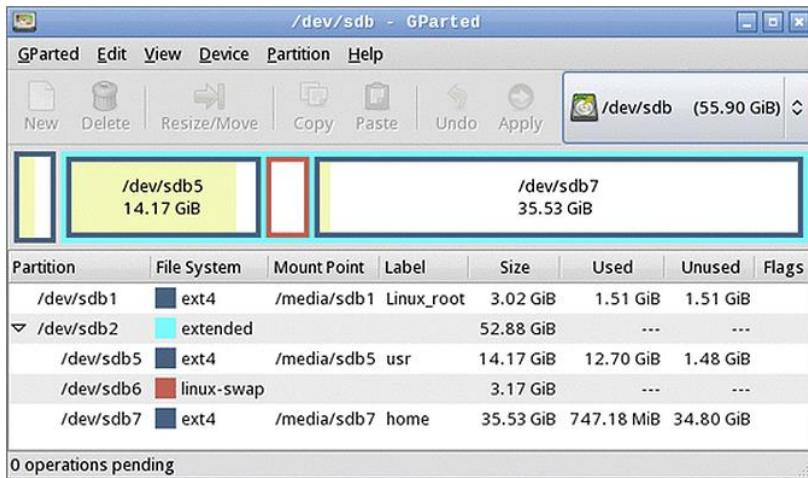
É uma área no início do disco rígido de 512 bytes de tamanho. Essa área contém informações de onde está o sistema operacional para poder ser carregado. No MBR também contém a informação de quais partições existem no disco e onde elas estão localizadas.

08

O Windows deve ser colocado em uma partição primária, pois este sistema operacional não consegue inicializar numa partição lógica, enquanto o Linux não possui qualquer restrição. Por outro lado, o Linux requer na sua instalação a criação de pelo menos duas partições, uma para instalar o próprio Linux (partição Linux nativo) e a outra para servir de memória auxiliar para o Linux (partição de swap ou troca). A partição Linux nativa é chamada de “diretório raiz do Linux” e é representada por “/”.



Exemplo de disco particionado em várias unidades no Windows.



Exemplo de disco particionado em várias unidades no Linux.

09

2 - O SISTEMA DE ARQUIVOS FAT

FAT é a sigla para **File Allocation Table** (ou **tabela de alocação de arquivos**). O primeiro FAT surgiu em 1977, criado pela Microsoft, para funcionar com a primeira versão do DOS.

Esse sistema de arquivos funciona como uma espécie de tabela que contém indicações de onde estão as informações de cada arquivo.

Quando um arquivo é salvo num disquete, por exemplo, o FAT divide a área do disco em pequenos blocos. Assim, um arquivo pode ocupar vários blocos, mas eles não precisam estar em sequência. Os blocos de determinados arquivos podem estar em várias posições diferentes, daí a necessidade de uma tabela para indicar cada bloco.

O FAT é o sistema de arquivo mais simples e comum que existe, por isso é utilizado na maioria dos dispositivos móveis (MP3 player, pen drivers, cartões de memória de smartphones e máquinas fotográficas). Também por isso, é único compatível com todos os sistemas operacionais que comumente existem (Linux, Windows, iOS, Android etc.).

10

Podemos dizer que o FAT é uma espécie de sumário que contém as informações de cada arquivo. Cada arquivo pode ser contínuo ou ser gravado em pedaços. Veja a imagem abaixo que ilustra um disco rígido. Suponha que a área em amarelo é a área reservada para a FAT, suponha que neste disco temos apenas dois arquivos, o verde e o vermelho. Os demais blocos (quadradinhos da imagem) em branco são espaços não utilizados (disponíveis).

Então, na FAT, estaria escrito algo como:

- **Arquivo verde:** começa no bloco 1 e vai até o bloco 6, continua do bloco 9 ao bloco 9, continua do bloco 18 até o bloco 21, continua do bloco 26 até o bloco 30 e termina, totalizando 19 blocos.
 - **Arquivo vermelho:** começa no bloco 7 e vai até o bloco 7, continua do bloco 10 ao bloco 13, continua do bloco 23 ao bloco 24 e termina, totalizando 7 blocos.

Área da FAT												1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	
43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	

Exemplo hipotético de FAT

Com o surgimento de dispositivos de armazenamento com mais capacidade e mais sofisticados, o sistema FAT foi ganhando alterações (identificadas pelos nomes FAT12 e FAT16). Isso foi necessário porque o FAT era limitado a determinada capacidade de armazenamento. Por exemplo, ele só operava com tamanho máximo de 2 GB. Assim, num disco de 5 GB, seria necessário dividi-lo em 3 partições. Fora o fato de que o FAT apresentava problemas com informações acima de 512 MB. Diante de tantos problemas, em 1996, a Microsoft lançou um novo FAT: o FAT32, que é compatível com os Windows 9x/Me/2000 e XP (apesar destes dois últimos terem um sistema de arquivos mais avançado, o NTFS).

11

2.1 - Diferenças entre FAT e FAT32

O sistema FAT (ou FAT16) consegue trabalhar com 65536 clusters. Esse número é obtido elevando o número 2 a 16 (daí a terminologia FAT16). Mas, na verdade, o sistema FAT16 usa apenas 65525 clusters por disco (ou partição). É importante frisar que o tamanho do cluster deve obedecer também uma potência de 2: 2 KB, 4 KB, 8 KB, 16 KB e 32 KB, ou seja, não é possível ter cluster de 5 KB, 7 KB, etc. O tamanho dos clusters no sistema FAT também é uma potência de 2. O limite máximo de tamanho para uma partição em FAT16 é de 2 GB (correspondente a 2 elevado a 16).

Já no caso do sistema de arquivos FAT32 (seu nome se deve ao mesmo motivo que no FAT32), o tamanho dos clusters é determinado através da relação entre os comandos FDISK e FORMAT, apesar de que é possível determinar o tamanho do cluster do FAT32 também por programas de terceiros. O tamanho máximo da partição em FAT32 é de 2 TB. Mas se você fizer as contas notará que 2^{32} é equivalente a 128 TB. Então porque o FAT32 usa somente 2 TB? Clique aqui para descobrir.

As diferenças entre FAT (ou FAT16) e FAT32 não param por aí. O FAT32 também é mais confiável, além disso, este sistema também consegue posicionar o diretório principal em qualquer lugar do disco. Fora o fato de que no sistema FAT havia uma limitação no número de entradas que podiam ser alocadas no diretório principal (512 arquivos e/ou pastas). Não há essa limitação no FAT32.

Algo curioso de ser citado, é que o FAT32 pode mudar o tamanho da partição sem perder dados. Apesar desta capacidade, a Microsoft, por alguma razão misteriosa, não implementou esta característica no FAT32. Hoje em dia, programas particionadores, como o Partition Magic ou então particionadores de disco de distribuições Linux, conseguem redimensionar uma partição FAT32 "inserido" este poder ao sistema de arquivos.

Clique aqui

Pode parecer confuso, mas o número máximo de clusters no caso do FAT32 não é de 2 elevado a 32. Apesar de seu endereçamento ser de 32 bits, na verdade são usados apenas 28 bits. Com isso, a quantidade máxima de clusters seria 2 elevado a 28, que corresponde a 8 TB. Não está errado, é 8 TB mesmo! Então, qual a razão do FAT32 ter tamanho máximo de espaço de 2 TB? Segundo a Microsoft, o número máximo de setores (setores, não clusters!) que um disco pode ter é de 2 elevado a 32. Como cada setor tem 512 bytes, o tamanho máximo de um disco no FAT32 acaba sendo de 2 TB.

12

3 - COMO O NTFS SURGIU

O sistema de arquivos FAT é aceitável e perfeitamente funcional para a maioria dos usuários domésticos. Trata-se um sistema antigo, que mesmo com novas versões, herdou a simplicidade da primeira versão.

As limitações, principalmente quanto à segurança, capacidade e confiança, fizeram do FAT um sistema de arquivos inadequado para uso em servidores e aplicações críticas. A Microsoft, estando ciente disso, decidiu desenvolver um sistema de arquivos que se adequasse aos princípios de funcionamento do Windows NT e lançou o **New Technology File System**, conhecido pela sigla **NTFS**. Entre os objetivos da idealização do NTFS estavam o fornecimento de um sistema de arquivos flexível, adaptável, altamente seguro e confiável. Sem dúvida, tais características fizeram do Windows NT um sistema operacional aceitável para as aplicações cujo seu desenvolvimento foi planejado.

A ideia de lançar o Windows NT surgiu em 1990, quando a Microsoft sentiu a necessidade de ter um sistema operacional com as qualidades citadas acima e com funcionalidades típicas de servidor. Nesta época, a Microsoft não tinha nenhum sistema que pudesse se equiparar ao poderoso Unix (tinha somente o MS-DOS e o Windows 3.x).

Decidida a ter uma fatia do mercado que pertencia ao Unix, a Microsoft deu início ao desenvolvimento do Windows NT. Esta sigla - **NT** - significa **New Technology**. A Microsoft logo percebeu que este novo Windows não teria sucesso se utilizasse o FAT, pelas razões já citadas. Era preciso criar um sistema de arquivos novo.



13

Muitas pessoas até hoje pensam que o NTFS é um sistema de arquivos inteiramente desenvolvido pela Microsoft, o que não é verdade. Seu projeto teve por base as análises das necessidades do novo sistema operacional, mas seus conceitos funcionais foram legados do sistema de arquivos HPFS (High Performance File System). Um sistema operacional muito conhecido nesta época era o OS/2, um projeto realizado em conjunto entre a Microsoft e a IBM.

Tanto a Microsoft quanto a IBM buscavam criar um sistema operacional de grande sucesso, cujo apelo principal seria a capacidade gráfica (naquela época, era muito maior o uso de sistemas operacionais baseados em linha de comando, como o DOS).

De todos os atributos do OS/2, os mais importantes estão em sua proteção, impedindo que um aplicativo derrube o sistema todo, pois o acesso à memória é exclusivo, e totalmente gerenciado pelo OS/2. O OS/2 também foi o primeiro sistema operacional a implementar TCP/IP, ter uma versão em 32 bits e máquina Java.

O OS/2 de fato continha inovações tecnológicas, mas esbarrava nos quesitos suporte e *marketing*. Fora isso, a IBM e a Microsoft começaram a se desentender e a empresa de Bill Gates decidiu abandonar o projeto e se dedicar ao desenvolvimento do Windows NT. No entanto, a Microsoft acabou levando consigo muitos conceitos funcionais do sistema de arquivos do OS/2, o HPFS. É claro que tais conceitos foram essenciais para a criação do NTFS.

OS/2

A sigla significa "*Operating System/2*". É um sistema operacional criado pela IBM e desenvolvido em conjunto com a Microsoft. O OS/2 e o Windows possuem uma origem comum: eram um único sistema nos anos 80 (mas com versões distintas) feitas simultaneamente pela IBM e Microsoft. A Microsoft resolveu abandonar o projeto OS/2 em 1990 e dedicar-se ao Windows. Os sistemas passaram então a competir pelo mercado corporativo e doméstico adentrando os anos 90.

14

3.1 - Características do NTFS

O NTFS possui características importantes, que o fez ser considerado um bom sistema de arquivos. Entre essas qualidades estão:

- Confiança

Pois permite que o sistema operacional se recupere de problemas sem perder informações, fazendo-o ser tolerante a falhas.

- Segurança

Onde é possível ter um controle de acesso preciso e ter aplicações que rodem em rede, fazendo com que seja possível o gerenciamento de usuários, incluindo suas permissões de acesso e escrita de dados.

- Armazenamento

Onde é possível trabalhar com uma grande quantidade de dados, permitindo inclusive o uso de arrays RAID.

- Rede

Fazendo do sistema plenamente funcional para o trabalho e o fluxo de dados em rede.

Há muitas outras características, que ficam mais ainda visíveis se comparadas ao FAT. A Microsoft vem trabalhando bastante para aperfeiçoar o NTFS, por isso, é de se esperar que novas características sejam implementadas no sistema de arquivos, de acordo com o lançamento de novas versões do Windows.

15

3.2 - Versões do NTFS

O NTFS tem versões, que foram lançadas principalmente no surgimento de novos Windows. A cada versão, são feitas correções de falhas, são implementados suportes a *hardware* e novas características são dadas ao NTFS. A princípio houve o NTFS 1.0, usado no Windows NT 3.1 (por isso, esta versão do NTFS também ficou conhecida por NTFS 3.1).

Após o lançamento do Windows NT 4, o NTFS ganhou a versão 1.1 (ou versão 4). Esta versão também foi usada no Windows NT 3.51. O sucesso do Windows NT foi tão grande que sua versão do NTFS virou referência em sistemas de arquivos.

A Microsoft resolveu então lançar a versão conhecida como NTFS 5.0 com o lançamento do Windows 2000, que substituiu o Windows NT. Apesar da nova versão, o NTFS 4 foi tão difundido que seu suporte a outros sistemas operacionais não acabará tão cedo.

A nova versão do NTFS apresenta novas características, além daquelas herdadas da versão anterior. Essas alterações foram essenciais para fazer do Windows 2000 um sistema que fosse realmente adequado para substituir o Windows NT. Para se ter uma ideia, o serviço Active Directory é um dos chamativos do Windows 2000 e foi implementado graças a alterações no NTFS. Entre os novos recursos do NTFS 5 estão:

- Reparse Points;
- Novas características de segurança;
- Quotas de discos;
- Diários de alterações;
- Criptografia;
- Suporte a arquivos esparsos.

Com o lançamento do Windows XP, Windows 2003 Server e futuras versões, o NTFS vai ganhando melhoramentos e novas características, mas certamente a versão 4 ainda será uma referência, o que nos faz concluir que o NTFS não será abandonado tão cedo pela Microsoft.

Points

Onde arquivos e pastas dentro do sistema de arquivos podem ter ações associadas a eles, de forma que operações particulares a estes arquivos possam ser executadas.

Novas características de segurança

Onde o mecanismo para gerenciamento da segurança e de usuários, principalmente em relação a acesso e arquivos foram melhorados.

Quotas de discos

Onde o administrador do sistema pode determinar o espaço em disco disponível a um usuário ou a um grupo de usuários.

Diários de alterações

Onde volumes podem ser ajustados para rastrear as operações efetuadas nos arquivos e pastas.

Criptografia

Onde o sistema permite que arquivos sejam codificados/decodificados automaticamente.

Suporte a arquivos esparsos

Onde é possível armazenar de forma eficiente arquivos esparsos (que são arquivos grandes, mas que possuem algumas estruturas vazias, desperdiçando espaço em disco).

16

3.3 - Funcionamento do NTFS

Seguindo as características legadas do HPFS, o NTFS trabalha de uma forma eficiente no gerenciamento do espaço de disco, pois as informações são armazenadas em uma base por setor do disco, em vez de utilizar clusters de múltiplos setores. Essa disposição apresenta várias vantagens, como menor necessidade de desfragmentação de disco e maior consistência de dados. Isso porque essa arquitetura de dados por base em setor possibilita manter os dados próximos, ao invés de espalhados pelo disco.

O gerenciamento de grandes quantidades de dados também é beneficiado por esta característica, já que trabalhar com clusters por setor, como acontecia com o FAT, tornava o sistema de arquivos dependente de um número predeterminado de setores.

17

4 - FUNCIONAMENTO PADRÃO DOS SISTEMAS DE ARQUIVOS EM LINUX

Do mesmo modo que praticamente todos os sistemas operacionais baseados em Unix, o Linux usa um sistema de arquivos que possui uma hierarquia, composta de arquivos e diretórios, que podem conter outros diretórios ou arquivos.

As unidades lógicas no Linux são organizadas para manipulação por meio do comando **mount**, geralmente acionado no processo de inicialização (*startup*), que ocorre quando o computador é ligado e o sistema operacional inicia o carregamento. O Linux consegue trabalhar com vários sistemas de arquivos em um mesmo disco (o que ocorre com usuários que possuem Windows e Linux em suas máquinas, por exemplo) e para visualizá-los, armazena a lista de sistemas de arquivos disponíveis no arquivo /etc/fstab (repare que /etc/ indica um caminho de diretório). Porém, há uma lista de sistemas de arquivos que estão efetivamente em uso, disponível no arquivo /etc/mtab, também conhecido como **tabela mount**. Esta lista é atualizada no processo de inicialização e indica ao sistema operacional quais sistemas de arquivos ele poderá acessar.

A cada sistema de arquivos montado na inicialização, um bit no cabeçalho do sistema de arquivos é zerado para indicar que o sistema de arquivos está em uso a partir daquele momento e que as estruturas de dados usadas para alocação e organização de arquivos/diretórios podem sofrer atualizações.

Quando o usuário decide desligar o computador e usa comandos para encerrar o Linux, os sistemas de arquivos são desmontados, fazendo com que o bit citado acima seja modificado para indicar que o sistema de arquivos está consistente, ou seja, não pode mais sofrer mudanças.

18

4.1 - Como tratar erros e corrompimento do sistema de arquivos

Embora os sistemas de arquivos para Linux tenham sofrido muitos tipos de tratamentos e reescrita de código para eliminar o corrompimento causado por aplicações ou pelo próprio Kernel, eliminar o corrompimento de dados em arquivos causados, por exemplo, pela queda de energia ou pelo desligamento incorreto por parte do usuário, sempre foi uma tarefa quase impossível.

Caso o sistema seja desligado incorretamente, o bit do cabeçalho do sistema de arquivos não é ajustado. A solução é fazer que, no próximo processo de carregamento do Linux, seja verificado se o cabeçalho está com o bit de cabeçalho configurado para indicar que o sistema de arquivos está consistente e não manipulável. Caso não esteja, a ferramenta "fsck" verifica o sistema na busca de erros.

4.2. Journaling

O journaling é um sistema de arquivo que possibilita ao sistema operacional manter um log (*journal*) de todas as mudanças no sistema de arquivos antes de escrever os dados no disco.

No caso de o sistema travar ou faltar energia, o journaling oferece maior probabilidade de não sofrer corrupção de dados, bem como uma recuperação mais rápida, pois dispensa a necessidade de verificar todo o disco, concentrando os esforços somente naqueles que pertenciam a um log que não tenha sido fechado devidamente.

Há vários sistemas de arquivos com a tecnologia journaling, conheça alguns.

Conheça alguns

Alguns sistemas de arquivos com a tecnologia Journaling:

- XFS, desenvolvido originalmente pela Silicon Graphics e posteriormente disponibilizado com código aberto;
- ReiserFS, desenvolvido especialmente para Linux;
- JFS, desenvolvido originalmente pela IBM, mas também liberado com código aberto,

Ext3 (o mais conhecido de todos), desenvolvido pelo Dr. Stephen Tweedie juntamente com outros colaboradores, na Red Hat.

19

4.2.1 - Técnicas de Journaling

Como vimos, o journaling armazena todas as ações que serão feitas nos arquivos, como gravação e/ou alteração de dados, por exemplo. Há ao menos duas técnicas usadas para o journaling:

Journal Físico

Registra uma cópia antecipada de todos os blocos de arquivos que serão posteriormente gravados no sistema de arquivo principal. Caso ocorra uma falha durante a gravação pelo sistema de arquivos, esta pode simplesmente ser repetida até a conclusão, quando o sistema de arquivo for montado da próxima vez. Se houver uma falha enquanto a gravação está sendo registrada no Journal, a gravação parcial terá um checksum ausente, ou incompatível, e pode ser ignorada na próxima montagem do sistema de arquivos.

Journal Lógico

Consiste em gravar apenas os metadados que sofrerão ações, tais como leitura/gravação e alteração dos arquivos. Um sistema de arquivos, com um “jornal” lógico, ainda recupera-se rapidamente após um acidente, mas pode permitir que os dados não sejam recuperados devido a não gravação antecipada dos blocos do arquivo alterado ou em uso, causando corrupção de dados. Contudo, isso aumenta o desempenho, o que pode ser notado na inicialização do sistema.

Checksum

É uma validação semelhante ao cálculo de um dígito verificador. Sempre ao criar um arquivo, o valor do checksum é calculado e armazenado junto ao arquivo. Ao ler o arquivo, novamente o checksum é calculado e comparado ao valor gravado, se o valor calculado for diferente do valor armazenado, então o arquivo é considerado inválido.

20

4.1 - Sistema de arquivos Ext2

O Ext2 é o sistema de arquivos Linux mais antigo dentre os que estão atualmente em uso. O Ext2 é sucessor do Ext (Extended). É muito robusto, rápido e confiável. Mas não possui o recurso de Journal. Infelizmente o Ext2 não possui este importante recurso e é a primeira partição a dar problema quando o sistema não é desligado de modo adequado. O Ext2 ainda é utilizado em alguns casos, saiba por que.

4.2. Sistema de arquivos Ext3

O sistema de arquivos Ext3 é basicamente o sistema de arquivos Ext2 com recursos de Journaling e por essa razão o seu uso é amplo.

O Ext3 passou a ser efetivamente suportado pelo Kernel do Linux a partir da versão 2.4. Consequentemente, todas as distribuições Linux lançadas com esse Kernel ou superior, tem suporte padrão para Ext3.

No Ext3, o código de Journaling usa uma camada chamada "Journaling Block Device" (JBD). A JBD foi criada com o propósito de implementar Journal em qualquer tipo de dispositivo com base em blocos de dados. Por exemplo, o código Ext3 informa e "pede autorização" à JDB para efetuar as mudanças, antes de modificar/adicionar qualquer dado no disco. Sendo assim, é o JDB que verdadeiramente "gerencia" o Journal. O fato mais interessante disso é que, a JDB funciona como uma entidade independente, permitindo que não só o Ext3 a use, mas também outros sistemas de arquivos. Saiba+

4.2.1. Níveis de Journaling

Há três níveis de operação do Journaling disponíveis na implementação do Ext3:

- Modo de Operação Ordered (modo padrão);
- Modo de Operação Writeback;
- Modo de Operação Journal.

Apesar de todos os recursos apresentados, cabe lembrar que o Ext3 também apresenta desvantagens.

Saiba por que

O Ext2 ainda é utilizado em alguns casos, pois seu sistema de quota de disco é perfeito. O do Ext3 não é confiável, o ReiserFS nem dispõe deste recurso, o xfs possui um quota só dele que funciona muito bem.

Saiba+

A JDB utiliza um método diferente de outros Journalings para recuperação de informações. Ao invés de armazenar as informações em bytes que depois devem ser implementados, a JDB grava os próprios blocos modificados do sistema de arquivos. Assim, o Ext3 também armazena "réplicas" completas dos blocos modificados em memória para rastrear as operações que ficaram pendentes. A desvantagem desta forma de trabalho é que o Journal acaba sendo maior. No entanto, o Ext3 não precisa lidar com a complexidade dos Journalings que trabalham gravando bytes.

Modo de Operação Ordered (modo padrão)

Somente os metadados dos arquivos são escritos na área de Journal. O conteúdo de cada arquivo não é armazenado no Journal, mas é迫使 a ser escrito no sistema de arquivos, logo após o Journal receber os metadados. Este nível de operação disponibiliza a melhor relação entre desempenho e confiabilidade. Este comportamento faz com que o Journal consiga (nem sempre) restaurar o conteúdo do arquivo que está sendo criado ou lido no momento de desligamento repentino, por exemplo.

Modo de Operação Writeback

O Journal armazena apenas informações referentes à estrutura do sistema de arquivos (metadados) e não em relação aos arquivos propriamente ditos. Os blocos de qualquer dado é gravado no sistema de arquivos de forma ocasional, pois é o kernel quem define quando será feita a gravação. Assim, o conteúdo de um arquivo pode ser escrito em disco antes ou depois que os metadados são atualizados no Journal. Este modo é o mais rápido, mas em compensação oferece uma segurança muito menor contra perda e corrompimento de arquivos causados pelos desligamentos incorretos.

Modo de Operação Journal

É o mais seguro, porém, mais lento. Nele, o Journal armazena não apenas informações sobre as ações sofridas, mas também uma cópia de segurança de todos os arquivos modificados, que ainda não foram gravados no disco. A cada alteração, o sistema grava uma cópia do arquivo (no Journal), atualiza as informações referentes à estrutura do sistema de arquivos, grava o arquivo e atualiza novamente o Journal, marcando a operação como concluída. Como disse, isso garante uma segurança muito grande contra perda de dados, mas em compensação, reduz o desempenho drasticamente. Justamente por causa disso, este é o modo menos usado.

Desvantagens

Veja algumas desvantagens do Ext3:

- Poucas funcionalidades: como o Ext3 visa uma grande compatibilidade com o Ext2, muitas das estruturas on-disk são similares àquelas do Ext2. Por causa disso o Ext3 não possui muitas das funções mais recentes, como alocação dinâmica de inodes e tamanhos de blocos variáveis (fragmentos ou caudas).
- Risco de corrompimento de arquivos durante a montagem: os sistemas de arquivos Ext3 não podem ser checados enquanto são montados para escrita. Um dump do sistema de arquivos feito enquanto ele está sendo montado para leitura e escrita pode resultar em dados corrompidos dentro do arquivo de dump.
- Ausência de ferramenta para desfragmentação: apesar de não fragmentar muito o disco na estrutura do Ext3, não há uma ferramenta para desfragmentação on-line funcional em nível de sistema de arquivos.
- Impossibilidade de recuperação de dados deletados: diferentemente do Ext2, o Ext3 zera os ponteiros de blocos nos inodes de arquivos apagados. Ele faz isso para simplificar o acesso de leitura e escrita quando o Journal está sendo utilizado após uma montagem. Isso, no entanto, previne efetivamente que os arquivos sejam recuperados. Assim, provê uma remoção de arquivos um pouco mais segura que em sistemas Ext2, o que pode ser tanto uma vantagem quanto uma desvantagem.
- Baixa performance para verificação do sistema de arquivos: o Ext3 é considerado lento na verificação do sistema de arquivos.
- Baixa performance na criação do sistema de arquivos: O Ext3 é lento quando se aplica um sistema de arquivos, e se o disco tiver que marcar badblocks, demora mais ainda.
- Baixa eficiência do Journaling: o sistema de arquivos Ext3, apesar de possuir Journaling, não foi desenvolvido desde o início para suportar o Journal, e sim, foi integrado a partir da versão 3 do mesmo, por isso, o Journal não é tão eficaz no Ext3. E quando o Journal falha, ele faz uso do fsck para verificar a integridade do sistema de arquivos, e se o fsck não conseguir recuperar a integridade do mesmo, seu sistema de arquivos fica corrompido.

Para muitos, este sistema de arquivos é obsoleto e para outros ele é considerado estável, e tem um bom desempenho.

21

4.1 - Sistema de arquivos Ext4

O Ext4 é um sistema de arquivos do Linux desenvolvido para ser o sucessor do Ext3 a partir de 2006. Conheça as novas funcionalidades do Ext4:

- Alocação tardia;
- Jornal checksumming;
- Suporte para tamanhos maiores de volumes e arquivos;
- Extensões;
- Compatibilidade com versões anteriores;

- Pré-alocação;
- Sistema de arquivos de verificação rápido;
- Alocador multibloco;
- Melhor timestamps.

4.1. Sistema de arquivos ReiserFS

Este sistema de arquivos é muito confiável quanto aos quesitos confiabilidade e rapidez. Possui um excelente Journal que raramente apresenta algum tipo de problema. O que este sistema de arquivos peca é justamente não dispor de sistema quota de disco.

4.2. Sistema de arquivos XFS

O XFS é o mais recente sistema de arquivos disponível para Linux. É um código que veio da Silicom Grafics, rápido, robusto, com suporte melhorado a arquivos muito grandes e diretórios com muitos arquivos, tem um excelente Journal que funciona muito bem. Também possui um sistema de quota que é próprio dele, mas que tem um funcionamento bem adequado. Por fim, o XFS possui um extenso conjunto de ferramentas para manutenção, otimização e recuperação de dados e metadados. Não possui ferramenta para recuperar arquivos apagados.

O Ext4 usa uma técnica de execução do sistema de arquivos chamado atribuir-on-flush, também conhecida como a atribuição de atraso. Isso melhora o desempenho e reduz a fragmentação, melhorando a alocação de blocos decisões com base no tamanho do arquivo.

Jornal checksumming

O Ext4 usa checksums no jornal para melhorar a confiabilidade, já que o jornal é um dos arquivos mais utilizados do disco. Esta característica tem um lado benéfico, que pode evitar com segurança um disco I/O esperar durante o processo diário, melhorando o desempenho ligeiramente.

Suporte para tamanhos maiores de volumes e arquivos

O sistema de arquivos Ext4 pode suportar volumes com tamanho até 1 Exabyte e arquivos com tamanho até 16 Terabytes.

Extensões

As extensões são introduzidas para substituir o tradicional bloco de mapeamento de esquema usado por arquivos Ext2/3. Uma extensão é um conjunto de blocos contíguos físicos, melhorando o desempenho de muitos arquivos e redução de fragmentação. Uma única extensão em Ext4 pode mapear até 128MB de espaço contíguo com um bloco de 4 KB de tamanho.

Compatibilidade com versões anteriores

O sistema de arquivos Ext4 é compatível com o Ext3 e o Ext2. Isto irá melhorar o desempenho ligeiramente, porque alguns dos novos recursos do Ext4 também podem ser usados com Ext3 e Ext2, tal como o novo algoritmo de alocação de blocos.

Pré alocação

O sistema de arquivos Ext4 permite pré-alocação de espaço em disco para um arquivo. Um novo `fallocate` (chamada de sistema) foi adicionado ao Linux para uso de sistemas de arquivos, incluindo Ext4 e XFS, que têm essa capacidade.

Sistema de arquivos de verificação rápido

No Ext4, blocos alocados por grupos e seções da tabela de inode são marcados como tal. Isso permite ignorá-los completamente em uma verificação e reduz o tempo necessário para verificar o sistema de arquivos. Esse recurso é implementado na versão 2.6.24 do Linux.

Alocador multibloco

O alocador multiblock é usado quando a atribuição atrasada é ativado por um sistema de arquivos, ou quando os arquivos são abertos no modo `O_DIRECT`.

Melhor timestamps

O Ext4 também adiciona suporte para a data de criação (timestamps). Mas, ao mesmo tempo em que é fácil adicionar um campo data de criação extra no inode, é mais difícil de modificar ou adicionar o necessário sistema de chamadas, como `stat()` (que provavelmente exigiria uma nova versão), e as várias bibliotecas que dependem deles (como glibc). Estas alterações exigem a coordenação de vários projetos. Portanto, mesmo se os desenvolvedores do Ext4 implementarem o suporte inicial para a data de criação, esse recurso não estará disponível para programas de usuário por agora.

22

5 - FUNCIONAMENTO PADRÃO DOS SISTEMAS DE ARQUIVOS APPLE

5.1 - Sistema de arquivo HFS+

O HFS+ ou HFS Plus é o sistema de arquivos desenvolvido pela Apple para substituir o antigo Hierarchical File System (HFS) como sistema de arquivos primário usado em computadores OS X e iOS, é um dos formatos, por exemplo, do iPod.

O HFS+ oferece ao usuário a possibilidade de formatar a unidade desejada com um registro cronológico. Com ele, toda vez que seu Mac OS X ou outro sistema for reiniciado indevidamente, ao invés de procurar em todos os arquivos e pastas as informações, basta apenas verificar os últimos eventos que ocorreram no disco rígido através desta tabela.

O HFS+ supera o seu antecessor HFS em vários quesitos:

- Pode trabalhar com endereços de bloco de arquivos de até 32bits, contra 16bits do sistema anterior;
- Aceita nomes de arquivos com até 255 caracteres, em vez de apenas 31;
- Tem formato de nome de arquivo Unicode, em vez do já antiquado MacRoman;
- Possibilidade de lidar com arquivos de até 8EB (Exabytes). Um Exabyte equivale a um bilhão de Gigabytes. Essa possibilidade é o que faz os editores de vídeo gostarem tanto de usar um produto Apple.

23

RESUMO

Neste módulo, aprendemos que:

- a. Que os sistemas de arquivos implementam as tecnologias que o sistema operacional utiliza para armazenar arquivos, bem como as demais funcionalidades e ferramentas disponibilizadas pelo sistema.
- b. Que há, basicamente, quatro tipos de implementação de arquivos: alocação contígua, alocação por lista encadeada, alocação indexada e alocação combinada.
- c. Que diretórios existem para organizar arquivos.
- d. Que o particionamento é a técnica de dividir uma unidade de disco rígido em partes independentes.
- e. Que o FAT é o sistema de arquivo mais simples e comum que existe, por isso é utilizado na maioria dos dispositivos móveis (MP3 player, pen drivers, cartões de memória de smartphones e máquinas fotográficas). Também por isso, é único compatível com todos os sistemas operacionais que comumente existem.
- f. Que o NTFS é a melhor opção Microsoft para sistema de arquivos, principalmente se considerarmos o uso empresarial.
- g. Que as unidades lógicas são definidas de modo diferente entre os sistemas operacionais, enquanto no Windows cada unidade é associada a uma letra, no Linux elas são relacionadas a diretórios.
- h. Que o Ext3 é um dos modos mais comuns de sistema de arquivos no Linux.
- i. Que a Apple utiliza o HFS+ como principal sistema de arquivos.

UNIDADE 4 – ARQUIVOS DE DADOS, EXECUTÁVEIS E BIBLIOTECAS, BAT E SCRIPTS.

MÓDULO 4 – PRINCIPAIS PASTAS E ARQUIVOS DOS SISTEMAS OPERACIONAIS WINDOWS, LINUX E ANDROID

01

1 - MICROSOFT WINDOWS

Olá, seja bem-vindo a mais uma etapa do nosso estudo. A partir de agora iremos aprender características das principais pastas e arquivos dos sistemas operacionais Windows, Linux e Android.

O Microsoft Windows é de longe o sistema operacional mais comum e conhecido que existe. Sua popularidade é disseminada em todos os países do mundo, e é praticamente certo que todos os usuários de computador utilizam ou já utilizaram alguma versão do MS Windows.

Apesar de haver significativas melhorias desde o Windows XP até o recentíssimo Windows 10, a estrutura interna se mantém praticamente inalterada.

O sistema de gerenciamento de memória RAM (com arquivo de Swap), hibernação, estrutura de diretórios (especialmente da pasta c:\windows) são algumas das características presentes em todas as versões desse sistema. Vamos conhecer algumas delas.

02

1.1 - Pasta Raiz “c:\”

Geralmente, por padrão, o disco rígido onde o sistema operacional está instalado é o disco “C”. Na ordem de classificação de letras do sistema operacional, o “C” representa a primeira unidade física do primeiro disco rígido, ou seja, a primeira partição do disco rígido onde é possível instalar um sistema operacional do tipo Microsoft Windows. Outros sistemas operacionais classificam as unidades de outras maneiras, como números (1, 2, 3 etc.) ou pastas (\fdd, \hdd0, \hdd1).

É nesse primeiro disco que vamos encontrar todas as informações relacionadas ao ambiente, que, normalmente, por padrão são:

- Arquivos relacionados à inicialização (boot) do sistema operacional;
- Todos os programas relativos ao sistema operacional;
- Todos os programas instalados pelo usuário;
- Todos os arquivos e documentos criados e copiados pelo usuário.

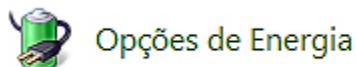
Na pasta raiz, “c:” encontramos principalmente, por padrão, os arquivos que veremos a seguir.

03

1.2 - Arquivo “c:\hiberfil.sys”

O MS Windows (todas as versões a partir do XP) possui um recurso denominado hibernação. Esse recurso permite ao usuário desligar o computador e, ao religá-lo, todos os programas e arquivos que estavam abertos no momento do desligamento reaparecerão exatamente conforme estavam.

A opção de habilitar esse recurso localiza-se no painel de controle, “opções de energia”.



Ícone de opções de energia no Painel de Controle do Windows 7

Este recurso é bastante interessante, pois permite uma espécie de “pausa”, em que o usuário pode desligar o computador em uma determinada situação que ele estava trabalhando e, ao retornar, poderá continuar o trabalho de onde parou.

Para fazer funcionar esse recurso, o sistema operacional Windows cria um arquivo na raiz do disco “c” com o nome de hiberfil.sys. Esse arquivo, quando é criado, possui o exato tamanho da memória RAM do computador, ou seja, em um computador com 16 GBytes de memória RAM, esse arquivo será criado com esse exato tamanho.

Após ser criado, ele é protegido pelo sistema operacional, ou seja, não é possível apagar (excluir) esse arquivo por comandos do explorador de arquivos (Explorer). Para apagar o arquivo é necessário desativar o recurso de hibernação. Veja como funciona o recurso de hibernação.

Veja como funciona o recurso de hibernação.

Quando o usuário deseja utilizar esse recurso, independentemente de quais programas e documentos estejam abertos, ao solicitar a hibernação do computador, o sistema operacional copiará todo o conteúdo da memória RAM para esse arquivo (leva de alguns segundos até um ou dois minutos), feito isso, ele desligará.

Ao ligar o computador, o sistema operacional identificará que ele estava em uma situação de hibernação; e, ao ligar, o sistema operacional, automaticamente, ao invés de proceder com a inicialização normal do sistema, lendo os diversos arquivos correspondentes à carga do sistema operacional, ele carregará todo (e somente) o conteúdo do arquivo de hibernação para a memória RAM. Esse processo é muitas vezes mais rápido do que o processo normal de inicialização do sistema operacional.

Com isso, todos os programas e arquivos que estavam anteriormente abertos e carregados em memória, novamente reaparecerão para o usuário.

04

1.3 - Arquivo “c:\pagefile.sys”

O Windows possui outro recurso também bastante interessante: é a capacidade de abrir mais programas do que suporta a memória disponível pelo computador.

Se o sistema operacional tivesse a limitação da memória RAM para carga de aplicativos e documentos, nós, usuários, teríamos um problema constantemente: não haveria espaço disponível na memória RAM para tantas informações. Sem esse recurso, para abrirmos um certo número de documentos e arquivos, precisaríamos sempre estar fechando outros programas e arquivos. Isso pode ser um grande problema se precisássemos de, ao mesmo tempo, manter abertos vários programas, como o Word, o Internet Explorer (abrindo várias páginas Web), por exemplo, um tocador de músicas e o explorador de arquivos.

Para resolver esse problema, o Windows cria um arquivo que simula o aumento da sua memória RAM, seria algo como simular que um computador de 4 GBytes de memória RAM tivesse 8 GBytes de memória.

Para simular esse aumento de memória, o sistema operacional cria o arquivo **c:\pagefile.sys** e, quando o usuário requisita mais espaço de memória RAM para carregar novos programas e documentos, o sistema operacional escolhe um programa carregado em memória, que no momento você não esteja utilizando, e grava-o no arquivo “pagefile.sys”, feito isso, ele apaga o programa da memória RAM e libera o espaço da memória RAM para que outro programa a ocupe.

Caso o usuário precise retornar àquele programa que está atualmente no arquivo “pagefile.sys”, o sistema operacional pode selecionar outro programa sem uso, gravá-lo no arquivo “pagefile.sys”, liberar o espaço da RAM, ler o programa anterior do arquivo “pagefile.sys” e assim permitir ao usuário voltar a acessar o seu programa ou informação.

05

Um computador pode ter um ou vários arquivos pagefile.sys, um em cada partição do HD, por exemplo. Geralmente o tamanho e a quantidade de arquivos de paginação são automaticamente controlados pelo sistema operacional. Entretanto, o usuário pode modificar as configurações de paginação. Para isso, ele acessa as configurações de memória virtual no painel de controle. Esse processo é comumente chamado **depaginação ou swap**.

Entretanto, esse trabalho todo tem uma grande desvantagem: ele consome muito tempo, quanto mais operações de leitura e gravação do arquivo “pagefile.sys” o sistema operacional tiver que fazer, mais lento ficará seu computador (você ficará vendo aquela ampulheta na tela por bastante tempo).

Experimente fazer uma análise dos programas que você utiliza e, caso seu computador esteja lento, um dos motivos podem ser as muitas atividades de leitura e gravação do arquivo “pagefile.sys”. Para resolver esse problema e aumentar a velocidade do computador, simplesmente adicione mais memória RAM.

Atualmente, 4 GBytes de memória RAM é muito pouco para a maioria das necessidades dos usuários. Esse valor seria o mínimo para o computador funcionar. Abaixo seguem algumas sugestões de quantitativo de memória RAM desejável para alguns usos do computador:

Quantidade de memória RAM	Uso do computador
4 GB	Apenas navegação na Internet (poucas páginas ao mesmo tempo) ou uso de apenas um aplicativo por vez.
8 GB	Acesso de várias páginas WEB ao mesmo tempo e uso de vários aplicativos ao mesmo tempo.
16 GB	Edição de fotos digitais e jogos de bastante performance
32 GB	Edição de vídeos digitais.

Para a maioria das empresas atualmente a recomendação é que os computadores possuam 8 GBytes de memória RAM.

06

1.4 - Pasta “c:\Arquivos de Programa”

É na pasta “c:\Arquivos de Programa” que os programas geralmente são instalados. Alguns exemplos de programas que você provavelmente encontrará nessa pasta são:

- Office,
- Internet Explorer,
- o antivírus que você utilizar,
- outros navegadores (como o Mozilla ou o Chrome).

Quando você cria um programa de computador do tipo que é instalado no computador do usuário, geralmente todos os arquivos do seu programa ficarão em uma subpasta dentro da pasta **c:\Arquivos de Programa**. É lá que provavelmente estarão os arquivos executáveis e de configuração do seu sistema.

Geralmente, os programas são gravados nesta pasta observando o seguinte padrão: “c:\Arquivos de programa \nome da empresa\ nome do aplicativo”.

07

1.5 - Pasta “c:\Arquivos de Programa (x86)”

Há sistemas operacionais de 32 bits e de 64 bits. Sistemas de 32 bits só executam programas feitos em 32 bits; já os sistemas operacionais de 64 bits conseguem executar programas de 32 ou de 64 bits.

Quando você instala um sistema operacional de 64 bits ele separa o local onde são instalados os programas de um ou de outro tipo. Dessa forma:

- Em um sistema operacional de 32 bits só existe uma única pasta para a instalação dos programas, que é a pasta “c:\Arquivos de Programa”;
- Em um sistema operacional de 64 bits, os programas feitos em 64 bits serão instalados na “c:\Arquivos de Programa”;

- Ainda, no sistema operacional de 64 bits, os programas feitos em 32 bits serão instalados na pasta “c:\Arquivos de Programa (x86)”.



A pasta “c:\Arquivos de Programa (x86)” só existe em computadores com sistema operacional Windows de 64 bits e lá são instalados os programas de 32 bits.

08

1.6 - Pasta “c:\ProgramData”

Muitos programas possuem informações e dados que são (ou podem ser) compartilhados com outros programas ou outros usuários. Também, para fins de diferentes controles de acesso de usuários, pois existem sistemas que possuem diferentes controles de acesso de usuários, podendo haver perfis de usuário que não têm permissão para instalar programas. Suponha que seja necessário que um determinado programa precise que um usuário (sem muitas permissões na pasta “c:\Arquivos de Programa”) realize alterações mais significativas de arquivos.

Por fim, configurações que são compartilhadas por todos os usuários que acessam o computador também podem ser salvas nesta pasta pelos aplicativos.

Para resolver essa questão, o sistema operacional proporciona um local especial para colocar essas informações, ditas assim “compartilháveis” e “menos protegidas”. Como isso, os programas podem acessar e alterar informações especiais que não afetam a instalação original dos sistemas.

Outro exemplo clássico de uso desta pasta é para guardar arquivos de configuração e modelos de documentos.

09

1.7 - Pasta “c:\Usuários”

Cada usuário do computador pode utilizar um login próprio de acesso (e não um login compartilhado). Quando um usuário possui um login específico (geralmente é seu próprio nome, um apelido ou uma parte do seu nome) ele cria no computador um espaço para que ele mantenha alguns documentos os quais somente interessam a ele.

Esses documentos e arquivos ficam normalmente protegidos de acesso por outros usuários, assim, os documentos do usuário A não são acessados pelos documentos do usuário B.

Dentro da pasta “c:\Usuários” estão as subpastas referentes a cada usuário que já logou no computador (além de usuários específicos como o administrador do sistema e outras que veremos a seguir). Assim, num exemplo hipotético onde João e Maria acessam o mesmo computador, dentro da pasta “c:\Usuários” haverá também as subpastas “c:\Usuários\Joao” e “c:\Usuários\Maria”.

Dentro da pasta “c:\Usuários”, além das pastas dos usuários há outras pastas padrão com os seguintes nomes e características:

Nome da subpasta	Característica
\All Users ou \Todos os usuários	Contém documentos, atalhos e configurações que são comuns a todos os usuários. Exemplo, um programa pode criar um atalho dentro da pasta “C:\Users\All Users\Desktop”, assim, todos os usuários que logarem nesse computador (mesmo nos novos usuários que ainda nunca logaram) poderão ter acesso a este atalho criado.
\Default, \Default User ou \Usuário Padrão	Refere-se ao modelo padrão de configuração de um novo usuário. Quando um novo usuário logar naquele computador pela primeira vez, o sistema operacional fará uma cópia dessa pasta e todo o seu conteúdo para a pasta “c:\Usuários\Nome_do_novo_usuário”, assim, todas as configurações e características serão copiadas para o novo usuário.
\Público	É uma pasta para uso compartilhado, quando você quer que um documento possa ser acessado por todos os usuários daquele computador, você pode colocar o documento nesta pasta. Assim qualquer pessoa que logar no computador (mesmo os novos usuários) poderá acessar o mesmo arquivo.

10

Há ainda outras subpastas importantes dentro do perfil de cara usuário. As pastas mais comuns que existem dentro da pasta dos usuários são:

Nome da subpasta	Característica
\AppData	Informações e configurações específicas que os programas fazem para cada usuário. Exemplo, se você alterar a configuração padrão que o Word traz para o tamanho do papel, tipo de fonte, etc, será dentro desta pasta que o Word salvará a sua configuração de uso da ferramenta.
\Contatos	Contatos que você criou com o Ms Outlook. Dados como nome, e-mail, telefone e endereço ficam gravados dentro desta pasta.
\Cookies	Informações que sites criam a respeito do seu acesso, como seu login em um site, ou informações que você tenha acessado anteriormente.
\Desktop	Todos os atalhos e arquivos que aparecem no seu desktop estão guardados dentro desta pasta. Seu desktop é reflexo do conteúdo desta pasta.

\Documentos e Meus documentos	Por padrão, esse é o local onde seus documentos são salvos. Word, Excel e outros tantos softwares utilizam esta pasta como padrão para você gravar seus documentos.
\Downloads	Local padrão onde ficam salvos os documentos, arquivos e programas que você baixa da Internet.
\Favoritos	Links de sites de Internet que você salva como favorito no Internet Explorer. Outros navegadores como o Mozilla e o Chrome salvam os sites favoritos em outros lugares dentro dos respectivos navegadores.
\Imagens	Local padrão para guardar fotos digitais ou imagens de Internet.
\Jogos Salvos	Local para o Windows salvar informações sobre Jogos da Microsoft. Algumas outras empresas de software também utilizam esta pasta para salvar informações sobre outros jogos de computador.
\Links e \SendTo	Atalhos que aparecem no Explorer, quando você clica em um arquivo com o botão direito do mouse e escolhe a opção “enviar para”.
\Menu Iniciar	Atalhos para os programas instalados no seu computador. São aqueles programas que aparecem quando você clica em Windows → todos os programas.
\Músicas	Local onde ficam guardadas músicas, como arquivos MP3 ou arquivos de música do Apple Store.
\Vídeos	Local onde ficam guardados arquivos de vídeo.

11

1.8 - Pasta “c:\Windows”

A pasta “c:\Windows” é a principal pasta do sistema operacional, praticamente todas as informações e arquivos relacionados à instalação do sistema operacional, drivers de periféricos, fontes e carga do sistema estão dentro dessa pasta.

Essa pasta é criada pelo software instalador do sistema operacional. Não é muito usual que usuários comuns precisem acessar ou alterar algum arquivo ou informação dentro desta pasta.



Os vírus de computador na sua grande maioria tem essa pasta como principal objetivo de infecção e alteração, pois lá estão os programas que controlam o próprio sistema operacional..

Centenas e centenas de subpastas estão dentro da pasta Windows, vamos ver algumas delas:

- a) Pasta “c:\Windows\Fonts”
- b) Pasta “c:\Windows\Prefetch”
- c) Pasta “c:\Windows\system”

- d) Pasta “c:\Windows\System32”
- e) Pasta “c:\Windows\System32\Drivers”

a) Pasta “c:\Windows\Fonts”

Contém todas as fontes (letras) instaladas no seu computador. Arquivos de fontes como Arial, Times e Verdana estão lá. Quando você instala uma nova fonte é para lá que ela é copiada. Da mesma forma, pasta se desinstalar uma determinada fonte do computador, basta exclui-la desta pasta.

b) Pasta “c:\Windows\Prefetch”

São registradas nesta pasta as informações dos aplicativos e documentos mais utilizados pelo usuário. Dessa forma o sistema controla melhor como as aplicações devem ser carregadas para a memória, paginadas no arquivo de swap e otimizadas durante a iniciação do sistema.

c) Pasta “c:\Windows\system”

É uma pasta do sistema operacional atualmente em desuso para a guarda de bibliotecas de programas feitos em 16 bits. É bem provável que no seu computador esta pasta esteja vazia.

d) Pasta “c:\Windows\System32”

É uma pasta do sistema operacional utilizada para guarda de bibliotecas (DLLs) de diversos programas, configurações e arquivos do sistema operacional.

e) Pasta “c:\Windows\System32\Drivers”

É a pasta onde estão copiados todos os drivers necessários para acessar e operar todos os recursos do seu computador. Exemplos: se seu micro é capaz de reproduzir uma música, nesta pasta está o driver da placa de áudio (provavelmente dentro da placa mãe do computador). Se seu micro é capaz de trocar informações com seu celular, nesta pasta estará o driver do seu celular para o sistema operacional saber identificá-lo e operá-lo remotamente para troca de arquivos. Há centenas de drivers no computador.

12

f) Pasta “c:\Windows\System32\drivers\etc”

Nesta pasta está localizado o arquivo “hosts”. Esse arquivo possui a configuração de acesso de alguns sites. Softwares de antivírus e antispywares modificam esse arquivo para que os navegadores do seu computador não acessem determinados sites conhecidos por serem maliciosos ou infectados.

Veja abaixo um possível conteúdo para o arquivo “hosts”, o sistema antispyware “Spybot” cadastrou alguns endereços de sites maliciosos redirecionando-os para o próprio computador. Desta forma, se o usuário digita o endereço de um site malicioso, ao invés de acessar o site, o próprio sistema operacional irá bloquear o acesso ao site (nada acontecerá).

```
# Start of entries inserted by Spybot - Search & Destroy
127.0.0.1      www.007guard.com
127.0.0.1      007guard.com
127.0.0.1      008i.com
127.0.0.1      www.008k.com
127.0.0.1      008k.com
```

Nesta mesma pasta, há também outros arquivos importantes:

- Services;
- Protocol;
- Networks;
- Imhosts.

Services

Mostra as portas abertas no seu computador para acesso via Internet.

Protocol

Lista os protocolos de comunicação Internet habilitados no seu computador.

Networks

Permite o redirecionamento de sites em uma rede local.

Imhosts

Permite a definição de alguns servidores específicos da sua rede local. Normalmente não é necessário fazer configurações manuais visto que o sistema operacional é capaz de localizar esses servidores automaticamente.

13

2 - LINUX

O Linux surgiu como um trabalho voluntário, gratuito e aberto para ser uma alternativa aos sistemas operacionais da Microsoft.

Vantagens

Desvantagens

As maiores vantagens do Linux são:

- Geralmente, as versões do Linux para usuário final são gratuitas. Para computadores servidores há opções gratuitas e pagas.
- São em geral bem mais leves que o Windows, consomem menos recursos e por isso deixam o computador mais rápido.
- Como possuem o código aberto, há uma garantia maior que não existe nenhum procedimento malicioso no sistema operacional capaz, por exemplo, de roubar informações sigilosas. No Windows, por exemplo, já houve relatos de versões do sistema que enviavam dados particulares dos usuários para a Microsoft.
- Podem ser customizados para conterem apenas as informações necessárias para funcionar um determinado sistema, por isso é uma ótima alternativa para funcionar como “servidores” em uma empresa.
- Permitem o uso concomitante com linguagens de programação gratuitas, como o Java e o PHP, com isso, uma empresa pode, por exemplo, montar um ambiente de programação totalmente gratuito quando falamos de softwares utilizados.

As maiores desvantagens do Linux são:

- As versões gratuitas não possuem garantia. Então, não há onde uma empresa procurar “assistência técnica autorizada” para resolver um problema com essas versões de sistema operacional.
- Não são compatíveis com programas feitos para Windows. Um programa Windows não funciona em um computador Linux (e vice-versa).
- Há poucos programas disponíveis para o Linux, se comparados com o Windows.
- Poucas pessoas sabem operar o Linux, muito embora não seja difícil de aprender.

O Linux, diferentemente do Windows, não utiliza letras para designar seus dispositivos. No Linux não existe “c:” para o disco rígido, “d:” comumente usado para o CDROM ou o “e:” para Pen drives. No Linux só existem pastas. Cada unidade, dispositivo, ou função do sistema operacional é representada por uma subpasta.

14

A seguir iremos conhecer as principais pastas de um computador com o sistema operacional Linux (independente de qual marca ou versão do Linux esteja sendo usado).

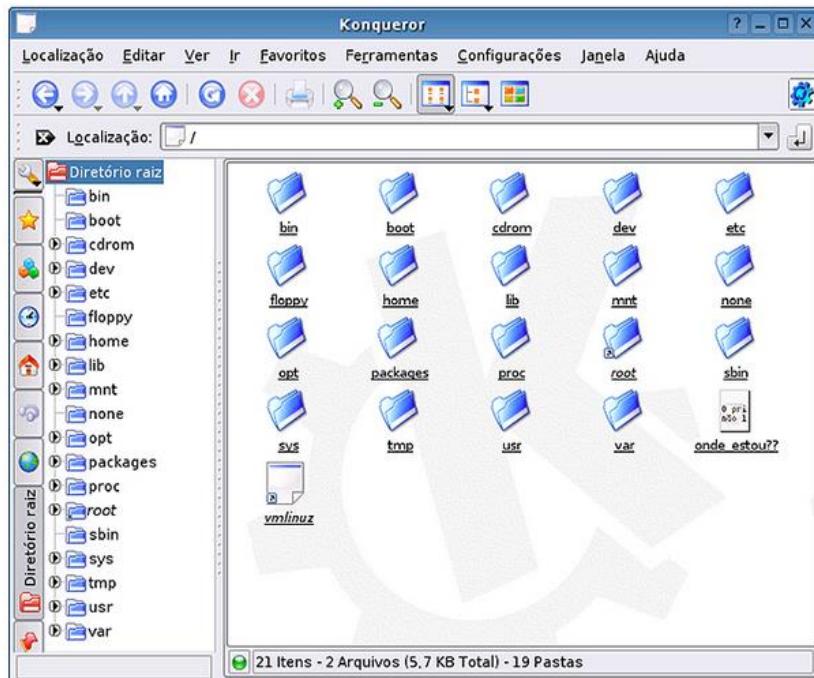
2.1 - Arquivo “config-2.6.11-kanotix-7”

O arquivo "config-2.6.11-kanotix-7" (o número de versão muda de acordo com a distribuição) é o arquivo compactado com o Kernel do sistema, que é carregado durante o boot. O gerenciador de boot do sistema (chamado de LILO) descompacta e executa os comandos dentro deste arquivo em memória.

2.2 - Diretório raiz/

O diretório raiz geralmente não contém nenhum arquivo, exceto, em algumas distribuições, pela imagem de inicialização (boot) do sistema chamada /vmlinuz. Todos os outros arquivos estão em subdiretórios dentro da raiz:

- /bin
- /sbin
- /boot
- /etc
- /lib
- /lib/modules
- /dev
- /tmp
- /boot
- /mnt
- /proc, /usr, /var, /home



Árvore de diretórios padrão de uma distribuição Linux

/bin

Comandos necessários durante a inicialização do sistema que podem ser utilizados pelos usuários (provavelmente após a ativação).

/sbin

Similar ao /bin, porém os comandos não são destinados aos usuários comuns, apesar de poderem ser utilizados por estes se necessário.

/boot

Armazena o Kernel e alguns arquivos usados pelo LILO, que são carregados na fase inicial do boot. Estes arquivos são pequenos, geralmente ocupam menos de 5 MB.

/etc

Arquivos de configuração específicos da máquina.

/lib

Bibliotecas compartilhadas necessárias aos programas no sistema de arquivos raiz.

/lib/modules

Módulo dinâmico e carregáveis pelo kernel, especialmente aqueles necessários para inicializar o sistema em caso de acidentes (por exemplo, programas de controle de sistemas de arquivos de redes).

/dev

Arquivos de dispositivos. O diretório "/dev", que é de longe o exemplo mais exótico de estrutura de diretório no Linux. Todos os arquivos contidos aqui, como, por exemplo, "/dev/hda", "/dev/dsp", "/dev/modem", etc., não são arquivos armazenados no HD, mas sim ponteiros para dispositivos de hardware. Por exemplo, todos os arquivos gravados no "arquivo" "/dev/dsp" serão reproduzidos pela placa de som, enquanto o "arquivo" "/dev/mouse" contém as informações enviadas pelo mouse. Esta organização visa facilitar a vida dos programadores, que podem acessar o hardware do micro simplesmente fazendo seus programas lerem e gravarem em arquivos. Não é preciso nenhum comando complexo para tocar um arquivo em .wav, basta "copiá-lo" para a pasta "/dev/dsp", o resto do trabalho é feito pelo próprio sistema. O mesmo se aplica ao enviar um arquivo pela rede, ler as teclas do teclado ou os clicks do mouse e assim por diante. O Kernel se encarrega da parte complicada.

/tmp

Arquivos temporários. Programas que são executados após a ativação do sistema devem usar o /var/tmp e não o /tmp, uma vez que provavelmente encontrarão mais espaço disponível neste sistema de arquivos.

/boot

Arquivos utilizados pelo gerenciador de inicialização do sistema. Imagens do kernel são normalmente mantidas aqui ao invés de no diretório raiz. Caso existam inúmeras imagens, o diretório pode facilmente crescer demasiadamente, sendo aconselhável mantê-lo em um sistema de arquivos à parte.

/mnt

Ponto para montagens temporárias realizadas pelo administrador do sistema. Programas usualmente não preveem a montagem automática no /mnt. Este deve estar dividido em subdiretórios, como por exemplo, /mnt/dosa sendo uma unidade de disquetes utilizando um sistema de arquivos MS-DOS, e /mnt/exta pode ser a mesma unidade, porém usam extensão ext2.

/proc, /usr, /var, /home

Pontos de montagem para outros sistemas de arquivos.

2.2.1 - Diretório /etc

O diretório /etc contém inúmeros arquivos. Alguns deles estão descritos resumidamente a seguir. Para outras descrições é necessário determinar a que programas eles pertencem e ler as páginas de manual daqueles programas.

Para sistemas que utilizam a tecnologia RPM, basta executar “rpm -qf arquivo” que o sistema lhe dirá a que pacote o arquivo pertence. Muitos arquivos de configuração de programas de rede estão no etc.

- /etc/rc ou /etc/rc.d ou /etc/rc?.d
- /etc/passwd
- /etc/fdprm
- /etc/fstab
- /etc/group
- /etc/inittab
- /etc/issue
- /etc/motd
- /etc/mtab
- /etc/shadow/
- /etc/longin.defs
- /etc/termcap
- /etc/printcap
- /etc/profile e /etc/bashrc
- /etc/securetty
- /etc/shells/

/etc/rc ou /etc/rc.d ou /etc/rc?.d

São scripts (arquivos de lote) ou os diretórios de scripts que são executados no início do sistema ou na mudança de nível do sistema.

/etc/passwd

Refere-se à base de dados dos usuários, cujos campos definem nome do usuário, nome real, diretório pessoal, senha criptografada e outras informações específicas de cada usuário.

/etc/fdprm

Contempla a tabela de parâmetros de disquetes. Descreve os diferentes formatos disponíveis. É utilizada pelo programa setfdprm.

/etc/fstab

Lista os sistemas de arquivos montados automaticamente, durante a inicialização do sistema, pelo comando “mount –a” (no script de inicialização /etc/rc.d/rc.sysinit ou similar). No Linux, contém ainda informações sobre as áreas de troca (swap) usadas automaticamente pelo comando “swapon –a”.

/etc/group

Similar ao /etc/passwd, porém descreve grupos ao invés de usuários.

/etc/inittab

Arquivo de configuração do init.

/etc/issue

Saída do programa getty antes do prompt de acesso ao sistema. Normalmente contém uma breve mensagem de boas-vindas. O conteúdo fica a critério do administrador do sistema.

/etc/motd

Escreve uma “mensagem do dia” automaticamente após um acesso bem sucedido. Conteúdos ficam a critério do administrador do sistema. Normalmente são usadas para enviar informações para os usuários, como por exemplo, avisos de desligamentos planejados.

/etc/mtab

Lista dos sistemas de arquivos montados. Inicialmente configurado por scripts e posteriormente atualizado pelo comando “mount”. Usado quando uma listagem dos sistemas de arquivos é necessária, como, por exemplo, pelo comando “df”.

/etc/shadow/

Arquivo de senhas em sistemas onde o shadow esteja instalado. Esta opção move senhas criptografadas do /etc/passwd para o arquivo /etc/shadow, o qual somente pode ser lido pelo superusuário. O shadow dá uma maior segurança às senhas dos usuários.

/etc/login.defs

Arquivo de configuração do programa login.

/etc/termcap

Arquivo de configuração de terminal. Descreve as sequências de escape para os diversos tipos de terminais. Os programas ao invés de escreverem diretamente uma sequência que funcione em determinado tipo de terminal devem buscar a sequência correta no /etc/termcap. Desta maneira os programas funcionam na maioria dos terminais.

/etc/printcap

Similar ao /etc/termcap, mas direcionado à impressora e com sintaxe diferente.

/etc/profile e /etc/bashrc

Arquivos executados pelo Bourne Shell e Bash, no momento do login do usuário. Estes arquivos permitem ao administrador manipular o ambiente de trabalho de todos os usuários do sistema. Veja as páginas de manual dos interpretadores de comandos para detalhes específicos.

/etc/securetty

Identifica terminais seguros, ou seja, aqueles nos quais é permitido que o root acesse o sistema. Normalmente somente quando os consoles virtuais são listados, tornando muito mais difícil obter os privilégios de super usuário do sistema através da rede ou de uma conexão via modem.

/etc/shells/

Lista as shells (interfaces) válidas. O comando “chsh” permite que os usuários mudem sua shell de login, porém, somente para aquelas listadas neste arquivo. O servidor de transferência de arquivos, “ftpd”, checará se a shell do usuário está listada no /etc/shells e não aceitará acessos, a menos que ela esteja presente neste arquivo.

16

2.2.2 - Diretório /dev

O diretório /dev contém arquivos especiais (drivers) de controle de todos os dispositivos. Estes arquivos utilizam uma denominação especial descrita na lista de dispositivos. Os controladores são criados durante a instalação, e posteriormente através do script.

2.2.3 - Diretório /usr

O diretório de arquivos /usr é grande, uma vez que a maioria dos programas está ali instalada. Os arquivos /usr são normalmente instalados pela sua distribuição do Linux. Programas locais são instalados no /usr/local. Isso torna a atualização do sistema possível para uma nova versão da distribuição ou mesmo para uma distribuição completamente nova, sem que seja necessário instalar todos os programas novamente. Alguns dos subdiretórios do /usr estão listados a seguir (os mais importantes):

- /usr/X11r6
- /usr/X386
- /usr/bin
- /usr/sbin
- /usr/man, /usr/info e /usr/doc
- /usr/include
- /usr/lib
- /usr/local
- /usr/share/magic

/usr/X11r6

Arquivos do X Window (interface gráfica do Linux). Para simplificar o desenvolvimento e a instalação do X Window, seus arquivos não foram integrados ao restante do sistema. Há uma árvore de diretórios sob /usr/X11R6 similar ao /usr.

/usr/X386

Similar ao /usr/X11R6, mas para o X11 Release 5.

/usr/bin

Praticamente todos os comandos de usuários. Alguns outros podem ser encontrados em /bin ou /usr/local/bin.

/usr/sbin

Comandos de administração do sistema que não necessitem estar no sistema de arquivos raiz, como servidores.

/usr/man, /usr/info e /usr/doc

Página de manual, documentos de GUN/Info e diversos outros documentos, respectivamente.

/usr/include

Arquivos header para a linguagem de programação C. Este diretório está sob o /usr/lib, porém por tradição tem sido mantido neste local.

/usr/lib

Arquivos estáticos de dados para programas e subsistemas, incluindo alguns arquivos de configuração globais. O “lib” vem de library (biblioteca), originalmente as bibliotecas de programação eram armazenadas neste subdiretório.

/usr/local

Local para programas instalados localmente e outros arquivos.

/usr/share/magic

Arquivo de configuração para o comando file. Contém as descrições de vários formatos de arquivos nas quais o “file” pode descobrir o tipo do arquivo. Veja as páginas de manual do magic e do file para maiores informações.

17

2.2.4 - Diretório /var

O sistema de arquivos /var contém dados que são alterados quando o sistema está sendo executado. É específico de cada sistema, ou seja, não compartilhado através da rede com outros equipamentos.

- /var/catman
- /var/lib
- /var/lock
- /var/log
- /var/run
- /var/spool
- /var/tmp

/var/catman

Um cache para páginas de manual que são formatadas. Fontes para as páginas são armazenadas em /usr/man/man*, sendo que algumas podem vir pré-formadas, estando armazenadas em /usr/man/cat*. Outras páginas necessitam ser formatadas quando acessadas pela primeira vez. A versão formatada fica armazenada em /var/catman, fazendo com que outro usuário que acesse a página, tenha esta pronta e disponível. O diretório /var/catman é frequentemente esvaziado, da mesma forma que os arquivos temporários.

/var/lib

Arquivos que mudam enquanto o sistema está ativo.

/var/lock

Arquivos de bloqueio. Muitos programas seguem uma convenção de se criar arquivos no /var/lock para indicar que eles estão utilizando um dispositivo em particular ou um determinado arquivo de bloqueio e não tentarão utilizar o mesmo dispositivo ou arquivo.

/var/log

Arquivos de históricos de vários programas, especialmente o login (/var/log/wtmp, o qual registra todas as entradas e saídas do sistema) e o syslog (/var/log/messages, o qual contém as mensagens do kernel e programas do sistema). Os arquivos no /var/log podem crescer indefinidamente com frequência e podem requerer limpezas periódicas.

/var/run

Arquivos que contêm informações sobre o sistema e que são válidas até a próxima inicialização. Por exemplo, o /var/run/utmp contém informações sobre os usuários atualmente conectados.

/var/spool

Diretórios para mail, news, filas de impressão e outros trabalhos em fila. Cada fila tem o seu próprio subdiretório sob o /var/spool. As mensagens de correio eletrônico, por exemplo, estão armazenadas em /var/spool/mail.

/var/tmp

Arquivos temporários que sejam muito grandes ou que necessitem existir por um período maior que o definido para o /tmp. Note que o administrador do sistema pode não permitir arquivos muito antigos no /var/tmp.

2.2.5 - Diretório /proc

O /proc é um sistema de arquivos ilusório. Na verdade ele não existe em um disco rígido. É criado em memória pelo kernel, sendo usado para disponibilizar informações sobre o sistema (originalmente sobre os processos, daí a origem de seu nome).

A menos que você copie este conteúdo para outro lugar, nada abaixo do /proc ocupa espaço em disco e será “apagado” quando o sistema operacional for desligado. O /proc é descrito mais detalhadamente na página de manual proc.

- /proc/1
- /proc/cpuinfo
- /proc/devices
- /proc/dma
- /proc/filesystems
- /proc/interrupts
- /proc/ioports
- /proc/kcore
- /proc/ksyms
- /proc/loadavg
- /proc/meminfo
- /proc/modules
- /proc/net
- /proc/self
- /proc/stat
- /proc/uptime
- /proc/version

Enquanto esses arquivos são de fácil leitura, por estarem em formato texto, não quer dizer que eles sejam facilmente lidos. Há muitos comandos que fazem mais que somente ler os arquivos acima e formatar o seu conteúdo para um melhor entendimento. Por exemplo, o programa “free” lê o conteúdo do /proc/meminfo e converte as informações dadas em bytes, para kilobytes, com algumas pequenas informações adicionais.

/proc/1

Diretório com as informações do processo 1 (init). Cada processo tem um diretório embaixo do /proc, cujo nome equivale ao seu número de identificação.

/proc/cpuinfo

Informações sobre o processador, tais como tipo, fabricante, modelo e performance.

/proc/devices

Lista dos controladores de dispositivos configurados no kernel atualmente em execução.

/proc/dma

Mostra quais canais DMA estão sendo utilizados no momento.

/proc/filesystems

Sistemas de arquivos configurados no kernel.

/proc/interrupts

Mostra quais interrupções estão em uso e quantas vezes foram chamadas.

/proc/ioports

Quais portas de entrada e saída estão em uso no momento.

/proc/kcore

Imagen da memória física do sistema. Tem exatamente o mesmo tamanho da memória física, mas não ocupa toda aquela área. Ela é gerada em tempo de execução, através do acesso dos programas.

/proc/ksyms

Mensagens de saída do kernel. São direcionadas também para o syslog.

/proc/loadavg

Apresenta a “carga média” do sistema, contendo três indicadores de quanto trabalho o sistema está executando no momento.

/proc/meminfo

Informações sobre o uso de memória, tanto física como de swap.

/proc/modules

Descreve quais módulos estão carregados no momento.

/proc/net

Informações sobre a situação dos protocolos de rede.

/proc/self

Um link simbólico para o diretório dos processos do programa que está olhando para o /proc no momento. Quando dois processos acessam o /proc, estes recebem diferentes links. Isto é feito para facilitar a vida dos programas quando estes olham o seu diretório de processos.

/proc/stat

Diversas estatísticas sobre o sistema, como o número de paginações desde seu início, E/S, processos executados etc.

/proc/uptime

O tempo que o sistema operacional está ativo.

/proc/version

Versão do kernel do sistema.

19

3 - SMARTPHONES E TABLETS ANDROID

Os *smartphones* e *tablets* que utilizam o sistema operacional Android possuem uma relação estrita com o sistema operacional Linux, afinal, o Android é uma adaptação do Linux para *smartphones* e *tablets*. Dessa forma, ao acessar a diretamente o cartão de memória do seu *smartphone* Android você poderá encontrar as mesmas pastas (ou boa parte delas) do Linux.

O “Padrão para sistema de arquivos hierárquicos” (FHS, na sigla em inglês) é que define a estrutura de sistema de arquivos no Linux e em outros sistemas operacionais baseados nele, como o Android. Contudo, o sistema de arquivos de ambos contém alguns diretórios que ainda não estão definidos por normas.



Pastas do Android semelhantes às do Linux para computadores.

20

Alguns diretórios exclusivos do sistema operacional Android são:

• /cache	O cache é um dispositivo de acesso rápido que serve de intermédio entre um processo e o dispositivo Android. Ele armazena informações usadas com frequência. Neste diretório, ficam arquivados esses arquivos temporários que podem ser lidos com rapidez pelo sistema.
• /Data	O diretório /Data contém dados do usuário armazenados em uma partição separada de MTD (Dispositivo de Tecnologia de Memória, em tradução livre da sigla em inglês).
• /Default.prop	Nesta pasta estão armazenadas as definições de propriedade-padrão e valores restaurados a partir dos arquivos em cada reinicialização.
• /Init	Representa um componente-chave da sequência de inicialização Android (o init), que é um programa especializado em inicializar os elementos do sistema Android. Ao contrário de outros sistemas Linux, o Android usa o seu próprio programa de inicialização.
• /Sdcard	Indica o diretório do cartão SD não removível do Android. Geralmente, esse é o único diretório que pode ser acessado dentro do Android. Os demais diretórios podem

	ser acessados através do root feito no aparelho.
• /Extsdcard	Representa um cartão de memória removível, colocado em um aparelho Android.
• /System	Armazena o sistema operacional em uma partição separada de MTD, montada na inicialização (somente leitura).

21

RESUMO

Neste módulo, aprendemos:

- a. no Windows, cada unidade é separada em letras. Usualmente, “c:” representa o disco rígido principal.
- b. o Windows possui a tecnologia de hibernação, utilizando para isso o arquivo hiberfil.sys.
- c. o Windows utiliza o arquivo pagefile.sys para simular um aumento da memória RAM, permitindo assim que mais programas possam ser executados ao mesmo tempo.
- d. que certas pastas do sistema Windows possui características específicas.
- e. que no Linux, os dispositivos como CDROM, pen drive e disco rígido são acessados por pastas.
- f. que no Linux, cada diretório possui uma função específica.
- g. que o sistema operacional Android é muito semelhante ao Linux.