

1 INTRODUÇÃO

Este trabalho visa apresentar um estudo detalhado sobre o Sistema de Arquivos (File System) **ReiserFS** que foi desenvolvido inicialmente por **Hans Reiser** e que atualmente está sendo patrocinado pela empresa/distribuição alemã do sistema operacional livre Linux; **SuSE** e sendo mantido pela empresa **NameSys**.

2 DESENVOLVIMENTO

2.1 OBJETIVO

Buscamos apresentar ao leitor, de forma simples e objetiva, informações como características, testes de uso, vantagens e desvantagens de se utilizar o Sistema de Arquivos **ReiserFS** nas partições de um disco rígido (winchester, HD) instalado com o software **sistema operacional livre Linux**.

2.2 SISTEMA DE ARQUIVOS (FILE SYSTEM)

Conceito: é um sistema de armazenamento de dados que tem a função/objetivo de fornecer ao sistema operacional uma estrutura necessária para **ler/gravar/executar** arquivos. Citamos também que um sistema de arquivos é a parte mais visível de um **sistema de computação**. Veja abaixo alguns exemplos de sistemas de arquivos juntamente com o sistema operacional que o mesmo é suportado e também seu criador/desenvolvedor:

SISTEMA DE ARQUIVOS	SISTEMA OPERACIONAL	DESENVOLVEDOR
FAT	Windows	Microsoft
FAT16	Windows 95	Microsoft
FAT32	Windows 95 OSR2, 98, ME	Microsoft
NTFS	Windows NT, 2000, XP	Microsoft
HPFS	OS/2	IBM
ext2	Linux	-

ext3	Linux	Stephen Tweedie -Red Hat
ReiserFS (3.5, 3.6, 4.0)	Linux	Hans Reiser
XFS	Linux	Silicon Graphics
JFS	Linux	IBM

O Sistema de Arquivos é criado durante a **formatação** de uma partição no disco rígido (HD, winchester). Após a formatação, toda a estrutura para leitura/gravação/execução de arquivos e diretórios pelo sistema operacional estará pronta para ser utilizada. No sistema operacional livre Linux, este passo é feito **durante a instalação** do mesmo, onde o usuário poderá escolher/optar qual tipo de sistema de arquivos será utilizado em cada partição do disco rígido que ele desejar criar.

É de suma importância que o **sistema de arquivos** escolhido seja **consistente**. Nem todos os sistemas de arquivos garantem isso. Neste trabalho apresentaremos o sistema de arquivos **ReiserFS**. Este **file system** trabalha com o recurso de **journal** - significa que todas as alterações dos dados são antes registradas no disco rígido para que, caso o sistema venha a falhar durante este processo, a **transação** possa ser recuperada quando o sistema voltar. Isto confere agilidade ao processo de **recuperação de falhas**, bem como aumenta a confiabilidade das informações armazenadas no disco rígido.

2.3 SISTEMAS DE ARQUIVO “JORNALADOS” (JOURNALING FILE SYSTEM)

Conceito: também conhecido como **loggin**, journaling é um mecanismo utilizado para garantir que as estruturas de dados de um Sistema de Arquivos estejam sempre gravadas/armazenadas corretamente no disco rígido. Esta técnica foi inicialmente desenvolvida/criada para banco de dados e tem como finalidade garantir que todos os passos de uma determinada alteração na sua estrutura aconteçam completamente ou, caso ocorra algum imprevisto, nada acontecerá, pois graças ao journaling, se mantêm em **LOG** todas as operações no sistema de arquivos, caso aconteça uma queda de energia elétrica (ou qualquer anormalidade que interrompa o funcionamento do sistema), o journaling verificará o sistema de arquivos no ponto em que estava quando houve a interrupção, evitando a demora

para checar todas as partições em um sistema de arquivos (que poderia levar vários minutos em sistemas de arquivos grandes).

Com o avanço da utilização do sistema operacional livre Linux em sua crescente utilização em sistemas servidores; novas características se tornaram necessárias a ele. Uma delas é o mecanismo de **journaling** que fornece uma **rápida** e **segura** recuperação do sistema de arquivos após uma falha no sistema servidor.

Caso ocorra uma situação catastrófica em um **sistema de arquivos tradicional**, ou seja, que não tenha suporte ao recurso de journaling, a próxima vez que ele for utilizado toda sua estrutura de dados deve ser cuidadosamente verificada. Isto pode ser um processo muito lento e dependendo do tamanho do sistema de arquivos. Embora a técnica de journaling não proteja contra falhas no disco e erros de programação no sistema de arquivos, as demais falhas são passíveis de correção, que consomem poucos segundos e é independente do tamanho do sistema de arquivos. Resumindo, **o recurso de journaling é considerado um modo de alcançar/atingir a integridade das informações depois de uma falha no sistema.**

2.4 VANTAGENS DOS SISTEMAS DE ARQUIVO “JORNALADOS”

Os sistemas de arquivos com recurso de **journaling** (como o **ReiserFS**, o **ext3** ou **JFS**) possuem uma série de vantagens em relação ao sistema de arquivos tradicional do sistema operacional Linux (o padrão **ext2**). Talvez a mais expressiva delas seja a possibilidade de recuperação rápida após um desligamento incorreto do sistema operacional (sem necessidade de intermináveis minutos rodando/executando o software aplicativo/utilitário **fsck**). Com isso, os dados são mais confiáveis graças à segurança que o recurso de **journaling** oferece aos sistemas de arquivos que o implementam/suportam. O sistema de journaling grava qualquer operação que será feita no disco rígido em uma área especial chamada "journal", assim se acontecer algum problema durante uma operação de disco, ele poderá voltar ao estado anterior do arquivo ou finalizar a operação. O journal **acrescenta ao sistema de arquivos o suporte a alta disponibilidade e maior tolerância a falhas**. Após uma falha de energia, por exemplo, o journal é analisado durante a montagem do sistema de arquivos e todas as operações que estavam sendo feitas no disco rígido são verificadas. Dependendo do estado da operação,

elas podem ser desfeitas ou finalizadas. O retorno do servidor é **praticamente imediato**. Outra situação que pode ser evitada é com **inconsistências no sistema de arquivos** do servidor após a situação exposta acima.

2.5 O SISTEMA DE ARQUIVOS REISERFS

O sistema de arquivos **ReiserFS** faz parte da nova geração de sistemas de arquivos do sistema operacional Linux (**já é o sistema de arquivos padrão da distribuição SuSE Linux 8.1**), sendo que seu maior benefício é o suporte a **journaling**. Além disso, o ReiserFS não utiliza **blocos de tamanho fixo**, mas ajusta o tamanho de acordo com o tipo de arquivo utilizado em cada parte do disco rígido. Arquivos com o tamanho reduzido (pequenos) resultam em **blocos menores** e em uma **economia considerável de espaço**. Como consequência do tamanho de **bloco dinâmico**, o **ReiserFS** é muito mais rápido ao ler pequenos arquivos, principalmente com o tamanho menor ou igual a **2KB**, já que com o fim dos blocos de 4KB os arquivos podem ficar muito mais próximos um do outro, agilizando assim o acesso aos dados.

O uso deste sistema de arquivos comparado ao sistema de arquivos 'padrão' do Linux, **ext2**, na maioria dos casos, melhora o desempenho do sistema de arquivos através da **gravação seqüencial dos dados na área de meta-dados**.

Os **meta-dados** são as estruturas de controle de um sistema de arquivos. Estas estruturas são formadas pelos seguintes itens/componentes:

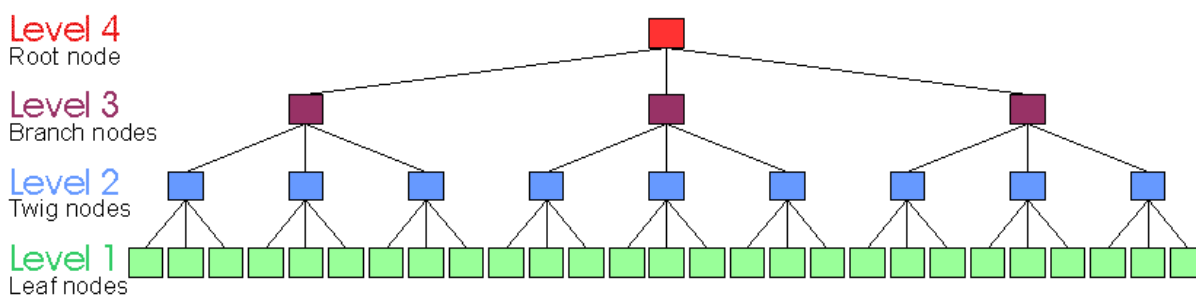
- ✓ i-nodes;
- ✓ mapa de blocos livres;
- ✓ mapa de i-nodes;
- ✓ diretórios, etc.

Os dados, que são **as informações realmente armazenadas nos arquivos**, não possuem um **LOG** (que pode ser definido como um conjunto de registros das ocorrências em um sistema). Portanto, em caso de falha no sistema, somente a **consistência do sistema de arquivos é garantida**, podendo haver alguma perda de dados em determinadas situações.

O **ReiserFS** mantém um **LOG** de todas as transações efetuadas em disco. Este **LOG** é armazenado/gravado no disco rígido e quando, inesperadamente, acontece uma falha no sistema ou uma queda de energia, o **LOG** sempre nos

fornecerá uma posição consistente do disco rígido, ao contrário do sistema de arquivos **ext2** onde esta informação reside apenas na memória RAM e se perde quando ocorrem quedas de energia, podendo ocasionar uma situação inconsistente na(s) partiç(ões) no disco rígido com blocos de dados perdidos, e outros com problemas que, infelizmente, podem não ser recuperados depois da falha no sistema.

O modo de organizar os dados no sistema de arquivos **ReiserFS** é através de **árvores (trees)**. Quando nós organizarmos informações em um computador, nós as ordenamos tipicamente em **pilhas (nodos ou nós)**, e há um nome (**ponteiro**) para cada pilha que o computador utilizará para encontrar informações. Uma árvore pode ser definida como um conjunto de **nodos/nós** organizados em um **nó raiz** (principal), e estes conjuntos de nós adicionais são chamado de **subtrees (sub-árvores)**. Veja abaixo um exemplo de uma árvore com seus **ramos (twigs)** e suas **folhas (leaves)**. As folhas são nodos/nós que não possuem filhos (**children**).



Uma procura começará com o nó raiz (root node - nível 4), atravessa 3 nós mais internos (branch nodes - nível 3), que por sua vez cria mais alguns ramos (twig nodes - nível 2) até terminar no nível 1 com nós de folhas (leaf nodes) que não possuem filhos.

O que acontece com um sistema de arquivos (filesystem) que **não implementa o recurso de journaling** quando há uma **falha (crash)** no sistema? (exemplo: **falta de energia elétrica**). Existem três cenários:

1. se você não tinha nenhum arquivo aberto, ótimo, não se preocupe;
2. se você tinha algum arquivo texto aberto, mas não estava digitando nada, as alterações serão perdidas; mas o arquivo antigo estará preservado;
3. devemos nos preocupar quando ocorrer uma falha no sistema justamente no momento em que o arquivo estava sendo digitado. Dependendo da situação, você perde este arquivo por completo e se você tiver o azar da falha no sistema

ocorrer quando o diretório (pasta) estava sendo atualizado, então poderemos perder todo este diretório juntamente como seu conteúdo.

Já o sistema de arquivos **ReiserFS** trabalha de com uma espécie de agenda (um **LOG** ou um **journal**). A escrita ocorre em duas etapas:

1. dados sobre a futura operação de escrita são agendadas (gravadas no journal)

2. a operação de escrita é realmente realizada. Assim, se ocorrer alguma falha na fase de "**agendamento**", o arquivo **não é atualizado**, mas se **mantém intacto**. Se ocorrer algum problema na fase de "**escrita**" propriamente dita, então o journal deve conter as informações necessárias para completar a operação de escrita e o disco rígido é atualizado.

As **partições são divisões existentes no disco rígido** que marcam onde se inicia e onde termina um sistema de arquivos. Graças a estas partições, poderemos utilizar mais de um sistema operacional no mesmo disco rígido (como o Linux e o Windows) ou dividir o disco rígido em uma ou mais partes para ser utilizado por um único sistema operacional. Para **particionar (dividir)** o disco rígido em uma ou mais partes é necessário utilizar um programa (software) de particionamento como o **fdisk**, **cfdisk** ou **Disk Druid** (ambos para o sistema operacional livre Linux). Após criada e formatada, a partição será identificada como um dispositivo no diretório **'/dev'** que deverá ser "montada" para permitir seu uso no Linux. Uma determinada partição não interfere em outras partições existentes.

Para utilizarmos o sistema de arquivos **ReiserFS**, devemos checar se o **kernel linux (é núcleo do sistema operacional livre Linux)** possui o suporte habilitado (na guia **File Systems**) ao ReiserFS e, logo após, instalar o software **reiserfsprogs** que contém os utilitários para gerenciar partições **ReiserFS**.

2.6 VANTAGENS E CARACTERÍSTICAS DO SISTEMA DE ARQUIVOS REISERFS

As **principais características e vantagens** do sistema de arquivos ReiserFS, além de fornecer o recurso de journaling, são:

- ✓ **BOOT** (processo de inicialização do sistema) é muito mais rápido pois verifica se no disco rígido apenas o que é apontado pelo "**journal file**".

- ✓ Implementa o tamanho de **blocos variáveis**;

- ✓ Suporta **arquivos maiores que 2GB** (Giga Bytes);
- ✓ O acesso mhash a árvore de diretórios é mais rápido que o **ext3**;
- ✓ É extremamente rápido devido ao uso de **árvores balanceadas**;
- ✓ A integridade do sistema é assegurada pelo **jornal file** e pela descrição de **meta-dados**;
- ✓ Existem opções suplementares que podem ser habilitadas no kernel linux para **checagem e controle da sincronia** entre **dados** e o **disco rígido**.
- ✓ O **ReiserFS** trata toda a partição do disco rígido como se fosse uma única tabela de banco de dados contendo diretórios, arquivos, e arquivos de meta-data. Estes são organizados em uma eficiente estrutura de dados chamada de "**árvore equilibrada - (balanced tree)**". Isto difere um pouco do modo no qual os sistemas de arquivos tradicionais operam, mas oferece grandes melhorias de velocidade para muitos aplicativos, especialmente os que utilizam vários **pequenos arquivos (small files)**. Com isso, agilizasse o processo de pesquisa de arquivos.
- ✓ A '**árvore equilibrada**' não armazena apenas os arquivos **meta-data**, mas também a própria informação do arquivo. Em um sistema de arquivos tradicional como **ext2**, o espaço no disco é alocado em blocos que variam em tamanho de **512 bytes** até **4096 bytes (4KB)** ou até maior. Se o tamanho de um arquivo exceder um múltiplo exato do tamanho do bloco, então ocorrerá desperdício de espaço no disco.
- ✓ O **ReiserFS** não aloca espaço de armazenamento em um '**k**' fixo ou quatro blocos de '**k**'. Ao invés disso, pode-se alocar o tamanho exato que o arquivo precisa. Este **FS** também inclui **otimizadores** para aumentar desempenho, de pequenos arquivos porque eles normalmente podem ser lidos com uma única **operação de I/O no disco rígido**.

✓ O **ReiserFS 4** chega a ser duas vezes mais rápido que a versão atual (**3.6**). Considerando que o **ReiserFS 3** já é conhecido pelo seu excelente desempenho porque utiliza um mecanismo de **estrutura de árvore** para manter/gerenciar arquivos. Com pequenos armazenados em disco, este mecanismo do **ReiserFS** pode economizar muito espaço de armazenamento. Além disso, desde que mais arquivos são colocados mais perto um do outro, o sistema pode abrir e ler muitos arquivos pequenos com só um acesso físico. Este processo melhora drasticamente o **desempenho** e **rendimento** do sistema de arquivos.

2.7 DESVANTAGENS E LIMITAÇÕES DO SISTEMA DE ARQUIVOS REISERFS

✓ O **ReiserFS** não trabalha perfeitamente com o sistema de arquivos de rede **NFS** (**Network File System**). Existem alguns "patches" (remendos) disponíveis para consertar o problema, mas eles não o resolvem completamente.

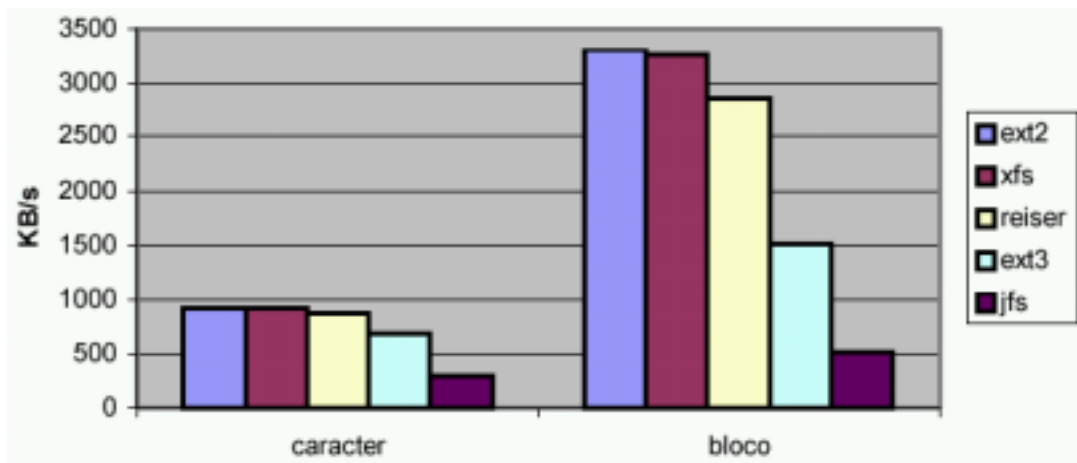
✓ Igualmente o **ReiserFS** não trabalha perfeitamente com **RAID** implementado via **software** (técnica de gerenciar discos - tolerantes à falhas). Se você for implementar **RAID** via **hardware**, fique tranquilo, pois o **ReiserFS** o suporta muito bem.

2.8 TESTES DE USO

Os testes de uso foram realizados em um computador com a seguinte configuração:

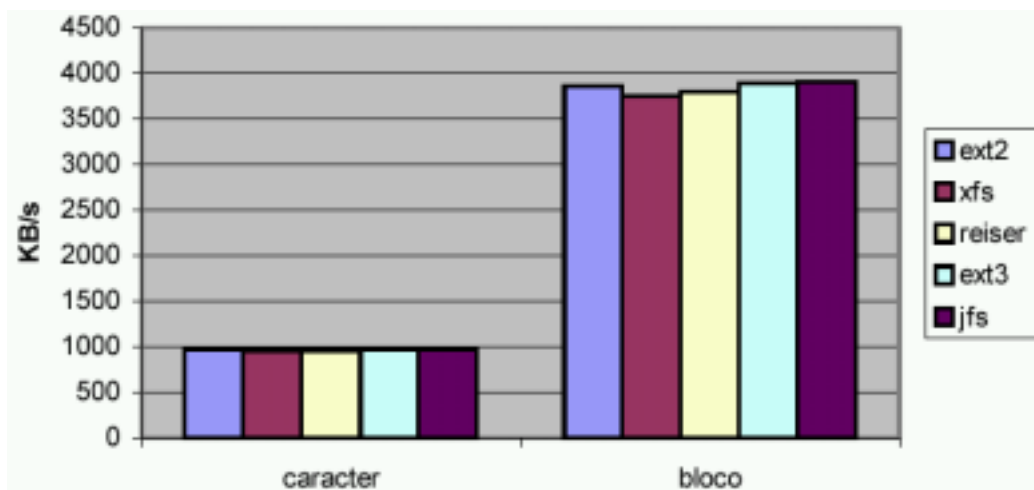
- ✓ processador Intel Pentium 100MHz;
- ✓ 64MB (Mega Bytes) de memória RAM;
- ✓ 512KB de memória cache;
- ✓ disco rígido Quantum Fireball de 2GB (Giga Bytes)
- ✓ uma partição vazia de 600MB (Mega Bytes)

ESCRITA SEQÜENCIAL:



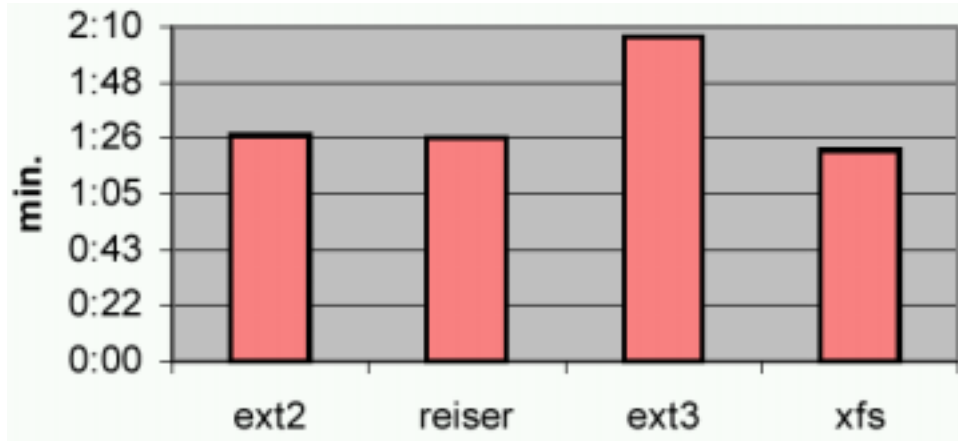
Fonte: Análise de desempenho de sistemas de arquivos com journaling para Linux. Autor: Silvano Bolfini dias

LEITURA SEQÜENCIAL:



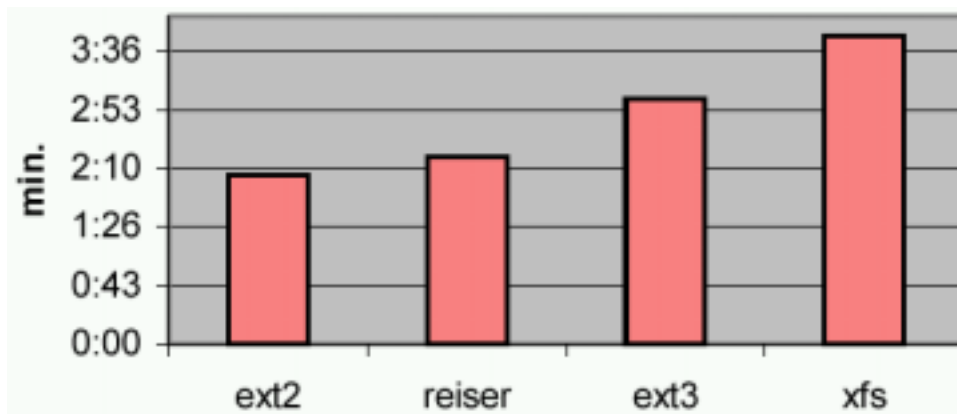
Fonte: Análise de desempenho de sistemas de arquivos com journaling para Linux. Autor: Silvano Bolfini dias

CÓPIA DE GRANDES ARQUIVOS:



Fonte: Análise de desempenho de sistemas de arquivos com journaling para Linux. Autor: Silvano Bolfini dias

CÓPIA DE GRANDES ARQUIVOS:



Fonte: Análise de desempenho de sistemas de arquivos com journaling para Linux. Autor: Silvano Bolfini dias

2.9 EXT3 X REISERFS, QUAL É O MAIS SEGURO?

Por Carlos E. Morimoto – WebMaster do site www.guiadohardware.net

A principal diferença entre esses 2 sistemas de arquivos “jornalados” é que o **ext3** tenta guardar informações tanto sobre a meta-data, ou seja, as informações sobre o espaço ocupado pelos arquivos e suas permissões quanto sobre os dados, enquanto o **ReiserFS** guarda/armazena apenas informações sobre a meta-data. No caso de um desligamento incorreto o **ReiserFS** é capaz de recuperar a consistência do sistema de arquivos em frações de segundos e a possibilidade de perda de diretórios ou partições é praticamente nula. Em compensação, os arquivos que eventualmente estiverem sendo gravados no exato momento em que acabou a energia ficarão com seus dados alterados. Você continuará tendo acesso aos arquivos normalmente, mas o conteúdo estará truncado ou incompleto.

Já o **ext3** tenta sempre preservar não só a meta-data, mas também os dados dos arquivos em si. Isto se revela ao mesmo tempo uma força e uma fraqueza. A vantagem é que existe uma possibilidade maior de recuperar os arquivos que estiverem sendo gravados no exato momento em que acabar a energia. Por outro lado o recurso de journal guarda/armazena mais informações e é acessado mais freqüentemente, causando uma certa degradação no desempenho (é justamente por isso que o **ReiserFS** costuma se sair melhor nos **benchmarks**) e ao mesmo tempo faz com que exista a possibilidade do próprio journal se corromper durante o desligamento.

Este é o grande perigo do **ext3**, pois sem o recurso de **journal** a tolerância à falhas é a mesma que no seu antecessor, **ext2**: o sistema rodará/executará o software **fsck** (imagine que este software se pareça com o aplicativo **scandisk**) que demorará vários minutos e você corre um grande risco de perder completamente os arquivos e/ou diretórios que estiverem sendo acessados no momento da pane/falha do sistema.

Concluindo, o **ReiserFS** oferece uma grande proteção contra corrompimento do sistema de arquivos, mas em compensação pouca proteção para os arquivos em si. O **ext3** por sua vez oferece uma maior proteção aos arquivos, mas em troca oferece um menor desempenho e uma proteção mais frágil para o sistema de arquivos em si.

3 CONCLUSÃO

Concluimos que, já com todas as características que o **sistema operacional Linux** oferece (estabilidade, performace, velocidade, segurança etc), um **sistema de arquivos** com recursos de **journaling** como o **ReiserFS** vem adicionar uma maior segurança e integridade aos dados para pequenos, médios e grandes servidores.

4 BIBLIOGRAFIA

SITE oficial do sistema de arquivos ReiserFS

Disponível em: <<http://www.namesys.com>>

ARBEX, Wagner. **Sistemas de Arquivos (Versão 3.2)**, Juiz de Fora.

Disponível em: <<http://www.jfnet.com.br/~arbex>>

LINUX in Brazil

Disponível em: <<http://www.br-linux.org>>

GUIA Foca Linux

Disponível em: <<http://focalinux.cipsga.org.br>>

MORIMOTO, Carlos E. **EXT3 X ReiserFS, qual é o mais seguro?**

Disponível em: <<http://www.guiadohardware.net>>

DIAS, Silvano Bolfoni. **Análise de desempenho de sistemas de arquivos com journaling para Linux**, Rio de Janeiro, Dezembro 2000.

LINUX Gazete

Disponível em: <<http://www.linuxgazette.com/issue68/>>

DEV Linux ReiserFS

Disponível em: <<http://devlinux.com/projects/reiserfs>>