

Notícias do Kernel

❑ Série 2.4 terá uma área para “Hot Fixes”

Após consultar Marcelo Tosatti, Willy Tarreau criou uma árvore chamada “Hot Fix”, que conterá correções agendadas para a próxima versão oficial do kernel 2.4. A razão para isso é que as versões oficiais podem demorar dois meses ou mais para ser lançadas. A árvore *-hf* seria um paliativo para quem precisasse das correções mais cedo. Entre as correções, teremos atualizações de segurança, correções para falhas críticas de corrupção de dados e *kernel panic*, remendos para conter *oops* e vazamentos de memória, soluções para funcionamento errôneo em geral, problemas com o sistema de compilação e com a documentação.

Essa é uma daquelas mudanças destinadas a uma minoria muito pequena de usuários, já que compromete o objetivo de manter a árvore 2.4 o mais estável possível. Toda a empolgação do pessoal da árvore 2.6 parece não contagiar Marcelo e não o desviam do mantra “2.4 estável, 2.4 estável, 2.4 estável” – mesmo que tenha sido persuadido a aceitar mais características portadas da conturbada versão 2.6 do que gostaria. Sua resistência a qualquer influência desestabilizante tem sido bem forte, no geral, e isso é comprovado pela criação de uma árvore separada como a *-hf* para conter a sanha modificatória na árvore principal. ■

❑ FUSE põe um pé dentro do kernel

Embora ainda não esteja incluído na árvore principal do kernel 2.6, o FUSE (*Filesystem in User Space* ou sistema de arquivos no espaço do usuário) foi aceito na árvore de Andrew Morton, a famosa *-mm*. Com isso, o número de usuários do FUSE aumenta bastante e, com ele, a re-

alimentação sobre problemas, falhas e desempenho. Grande vitória para os desenvolvedores do FUSE, já que o caminho para a inclusão no kernel 2.6 pode ser comparado a uma batalha campal contra um inimigo fortificado colina acima. Seu primeiro obstáculo foi Linus Torvalds em pessoa, que considera existirem problemas inerentes à idéia de se colocar o controle dos sistemas de arquivos nas mãos do usuário. Linux põe esses problemas no mesmo balaio dos que infestam os microkernels (lembra das “brigas” com Andrew Tannenbaum?) como as intermináveis camadas de abstração e a compartimentalização das funcionalidades. Mas ele ainda pode ser convencido e a presença do FUSE na árvore *-mm* pode encurtar o caminho e responder a muitas de suas objeções. ■

❑ O pesadelo da documentação

Pelo menos dois documentos significativos foram trazidos à luz do dia recentemente. O primeiro, de Josh Aas, documenta o agendador (*scheduler*) da CPU para o kernel 2.6.8.1 – aparentemente um kernel ainda bastante popular em sistemas de produção. O empregador de Josh, a empresa SGI, detém os direitos autorais sobre o documento, lançado sob a licença GNU FDL – GNU Free Document License.

Noutra frente, Stelian Pop liberou um *Como Fazer (HOWTO)* descrevendo o rastreamento e controle de versões do kernel usando o sistema de controle de versões *Subversion*. Esse documento (ou talvez o debate sobre ele na lista de discussão do Kernel) trouxe à baila opiniões iradas e apaixonadas, já que colocou em cheque – mais uma vez – se é apropriado ou não usar o BitKeeper para um projeto livre como o kernel do Linux.

Qualquer desenvolvedor que queira contornar o BitKeeper usando o *Subversion*, o *arch* ou qualquer outra ferramenta tende a exigir que a empresa BitMover ofereça um suporte cada vez melhor à sua alternativa do coração. Mesmo que Larry McVoy, CEO da BitMover, tenha a intenção de ir até o limite do possível para auxiliar os usuários do BitKeeper, ele tem sido um verdadeiro “paredão” intransponível quando se trata de ajudar os competidores do BitKeeper. Sua postura não pode ser censurada, afinal ele construiu um negócio sólido ao redor do BitKeeper e o mundo do software livre ainda não foi capaz de criar nada tão bom que possa substituí-lo – mesmo depois de anos de luta. É fato que projetos como o *arch* tem um grande futuro, mas há um abismo gargantual entre as melhores alternativas livres e o BitKeeper. Mesmo o *arch* mostrou recentemente que, devido a falhas de projeto, está muito aquém do que um empreendimento tão gigantesco quanto o kernel Linux precisa. ■

❑ Arquivos além da imaginação

O kernel é mesmo uma criatura bizarra. Quando funciona, é a mais bela das feras. Quando entra em parafuso, pode se tornar o monstro mais horrendo do espaço sideral. Entretanto, algumas pequenas falhas são tão estranhas que nada podemos fazer a não ser gargalhar.

Wichert Akkerman experimentou uma dessas falhas recentemente. Com um sistema de arquivos *ext3* rodando sob um kernel 2.6.10-ac10 (de Alan Cox, portanto), ele notou que o comando *df* informou que a taxa de ocupação do disco estava em 101% e que o número de blocos disponíveis era de -73786976294838127736. Não parece ter havido nenhuma perda de da-

dos ou corrupção e nada mais apresentou problemas. A informação maluca foi-se embora depois de um `fsck` redentor.

Sem informações suficientes para tentar um diagnóstico, essa falha deve ser catalogada para referência futura caso algo similar seja informado. Se o problema não se deveu a alguma falha de hardware, Wichert deve ter disparado um *bug* de alguma forma. Como ele o fez, entretanto, ainda é um mistério.

Um problema menos misterioso mas não menos embaçante foi descoberto por Mitch Williams. Ele descobriu que

se abríamos um arquivo do SysFS para edição e modificarmos algo, a modificação não será inserida no ponto em que a escrevemos. Em vez disso, o arquivo será substituído por outro de mesmo nome contendo **apenas** o novo dado inserido. Da mesma forma, a tentativa de concatenar dois arquivos no SysFS resulta, em vez da combinação dos dois arquivos, na substituição do primeiro pelo segundo. Isso foi um choque para caras como Greg Kroah-Hartman, que colocou meses de trabalho e noites sem dormir no SysFS.

Aparentemente, a maneira como o SysFS é usado não é prejudicada pela falha citada. As pessoas não costumam modificar os arquivos do SysFS, mas substituem completamente seu conteúdo. Ainda assim, esse comportamento errático deve ser consentado: o SysFS precisa, obrigatoriamente, funcionar como qualquer sistema de arquivos comum. Mitch enviou uma correção para consertar esse comportamento macabúzio e Greg se ofereceu para auxiliar Mitch a tornar o patch aceitável para inclusão na árvore principal. ■

❑ Novos planos para a numeração das versões

Reviravolta na novela! O processo de desenvolvimento do kernel está trocando a pele novamente, revelando uma nova encarnação de sistemas de numeração. Há dois aspectos principais e nenhum deles foi registrado em cartório ainda (aliás, sequer foram escritos a lápis em papel de padaria...). O primeiro é que Linus Torvalds e Andrew Morton estão considerando o retorno do modelo versão par/versão ímpar para diferenciar novamente a versão de desenvolvimento da versão estável. Entretanto, haveria um novo elemento no enredo: em vez de apenas o chamado *número menor* ter seu significado ditado pela diferenciação par/ímpar, sugeriu-se que todos os três números sigam essa lógica – o *número maior*, que representa a versão principal do kernel, o *número menor*, associado às versões intermediárias ou *edições*, e o chamado *número de patch*, que indica as revisões e correções feitas numa determinada série (a *série* do kernel é indicada pelos números maior e menor, por exemplo, série 2.6).

Da forma como foi proposta por Linus, o significado de um dígito ímpar aumenta de acordo com sua posição – quanto mais à esquerda, mais importante ele é. Portanto, se todos os números forem pares (2.6.14 por exemplo), isso representaria uma semana e pouco de esforço para estabilização do kernel. Se o número de patch for ímpar (2.6.13), o objetivo principal é a estabilidade, mas esse kernel é o resultado de grandes mudanças (ou seja, novas funcionalidades foram adicionadas e modificadas) e será o produto de um mês ou dois de trabalho. Se o número de edição (o número menor) for ímpar (2.7.10), esse kernel é objeto de mudanças tão profundas que podem desestabilizar o código por muitas edições no período de um ano ou dois. E se o número principal for ímpar (3.0.0) houve quebra geral de compatibilidade e o kernel está sendo praticamente reescrito do zero. Também significa que Linus está no hospital recuperando-se de estafa ou de algum distúrbio psiquiátrico.

Não nos parece que a bênção final tenha sido dada, mas é muito raro que esse tipo de mudança tenha um ponto de partida tão claro como o que Linus e Morton ofereceram desta vez. Linus prefere uma abordagem mais iterativa, difícil para os jornalistas entenderem, mas que facilita a vida dos desenvolvedores do kernel.

O segundo aspecto novo do desenvolvimento do kernel é algo que vinha cozinhando em fogo lento já há algum tempo: a criação de um esquema de numeração de **quatro** números, *X.Y.Z.A* – chamada por Linus de “*the sucker tree*” (a árvore dos otários) pois só um otário (*sucker*) concordaria em manter uma barbaridade dessas. A razão é óbvia: depois de todos os senões e restrições discutidos na LKML (a lista de discussão do kernel) essa árvore vai ser um pesadelo de se manter. Afinal, o “salário” é baixo, as honorarias são quase inexistentes e haverá uma multidão de usuários irados vociferando contra o mantenedor e pedindo sua cabeça.

Linus quer um controle rígido sobre a árvore *X.Y.Z.A* para assegurar que apenas as alterações adoradas como deusas por todos os desenvolvedores sejam incluídas. Os patches devem ser *apenas* correções de erros com nível de pureza próximo dos 100%, o que quer dizer que os desenvolvedores “espertos” que costumam incluir novas funcionalidades (não testadas) nos patches serão escorraçados da árvore proposta. Haverá um período de dois dias para apreciação do patch pelos demais desenvolvedores. Apenas se sobreviver a esse período o patch pode ser aceito. Linus irá puxar todos os patches da árvore dos otários para a sua própria, qualquer que seja o estágio de desenvolvimento ou instabilidade que a árvore de Linus esteja experimentando. Os usuários terão muito mais confiança na estabilidade do kernel se usarem a versão da “árvore dos otários”, deixando a árvore principal para os malucos dos desenvolvedores enfiarem seus códigos cheios de doideiras barulhentas. Greg Kroah-Hartman tornou-se um dos mantenedores principais da “árvore dos otários” e liberou a versão 2.6.11.1 com três pequenas correções para falhas no 2.6.11. ■