

# Notícias do Kernel

## ■ O BitKeeper será substituído?

O desenvolvimento do Kernel ainda é mantido e controlado, basicamente, com a ajuda do software proprietário BitKeeper, algo que não deve mudar tão cedo. Mas as alternativas livres estão ficando cada vez melhores. O programa *tla*, também chamado de *arch*, provavelmente possui um projeto mais bem elaborado que o do BitKeeper e recursos bem mais poderosos, mas ainda não é capaz de trabalhar na escala de produção em que o kernel do Linux é desenvolvido, cuja velocidade e quantidade são de tirar o fôlego.

Andrea Arcangeli tem sido o mais ferrenho defensor do *tla* entre os desenvolvedores do kernel, mas Linus Torvalds ultimamente tem berrado cobras e lagartos sempre que alternativas ao BitKeeper são mencionadas na lista. David Roundy, autor original do sistema *darcs* de controle de versão, evitou um derramamento de sangue ao anunciar um repositório das fontes do Linux baseado em seu “filho”. Isso é um marco para qualquer gerenciador de versões, pois o tamanho descomunal de todo o histórico de alterações do kernel, sem contar a velocidade em que novos patches são adicionados, torna a tarefa digna de Hércules. O fato do *darcs* conseguir gerenciar esse mastodonte já é um feito considerável. O problema da escalabilidade não foi resolvido, entretanto: David afirma que há alguns problemas de desempenho. Mas a existência no BitKeeper de uma “porta de conexão” (o chamado *gateway*) para o *darcs* semelhante às já existentes para CVS e Subversion permitem que David obtenha realimentação importante vinda dos desenvolvedores – e, com isso, possa melhorar o *darcs*.

Esses gateways pavimentarão o caminho pelo qual passará aquele que substituirá o BitKeeper no futuro. Só quando a vasta maioria de desenvolvedores do kernel passar a usar *tla*, *darcs* ou outra alternativa qualquer de desenvolvimento distribuído é que Linus vai, finalmente, jogar a toalha e bradar “ok, vocês venceram!”, abandonando de vez o BitKeeper. É

importante ter em mente que uma grande parte das idéias que vêm sendo introduzidas nas ferramentas livres de gestão de desenvolvimento foram motivadas e inspiradas na decisão de Linus de adotar um software proprietário para a tarefa. A teimosia do criador pode ser combustível para a ira de seus pares, mas também o é para o aperfeiçoamento das ferramentas usadas por eles. ■

## ■ Violações da GPL

Uma penca de aparentes violações à licença GPL, tentativas de encontrar brechas em sua aplicação e mal-entendidos em geral vêm pipocando por toda parte. É difícil determinar se é um fenômeno sazonal, uma tendência temporária ou a maneira como as coisas serão daqui para a frente. Entre outubro e novembro, cerca de 6 empresas foram acusadas de violação à GPL na lista de desenvolvimento do kernel, a famosa *lkml*. Algumas dessas acusações foram produto de simples mal-entendidos, como quando Adrian Bunk sugeriu que o driver *drivers/char/rocket.c* continha duas licenças conflitantes: a GPL e a licença não-livre da Control Corporation. Theodore Ts'o, num comentário sobre o autor do código, afirmou que era apenas um engano e rapidamente conseguiu um documento da Control que permitia a remoção da licença conflitante.

Os casos mais graves envolvem empresas procurando por brechas no código e na própria licença. Uma delas, por exemplo, distribui binários em uma versão e código-fonte em outra para os drivers de seus produtos, alegando que as versões mais novas são idênticas às antigas. Mentira deslavada: as versões novas contêm uma certa quantidade de código proprietário. Ela também se recusa a liberar atualizações para qualquer um que exercite o saudável direito – assegurado na GPL – de redistribuir seu software. Não são exatamente violações à GPL mas, se a veracidade das denúncias for comprovada, certamente trata-se de uma prática bastante grosseira e vulgar, para não dizer imoral e desonesta. Embora seja raro uma empresa usar código GPL e ao mesmo tempo ignorar o fato, pelo menos uma

empresa – segundo Marcus Metzler – está se negando a distribuir o código fonte de seu produto baseado em código GPL. ■

## ■ Revisitando a querela das versões

Linus Torvalds, além de rebatizar a versão 2.6.10-rc1 do kernel Linux – era Zonked Quokka, agora chama-se Woozy Numbat – está tentando aplacar os ânimos de todos os que vêm reclamando do sistema de numeração de versões nos últimos tempos. Para reduzir a doideira das variações do tipo x.y.z.w-pre-rc-final que vem provocando o não-funcionamento de scripts de usuários ao redor do mundo, Linus decidiu que a série 2.6 terá apenas lançamentos *-rc* entre as versões pontuais. Sem *-pre*, sem *-final*, sem caos.

Ou, pelo menos, até amanhã. O sistema de numeração pode até se estabilizar, mas o código do 2.6 continua mutante, com alterações ocorrendo a velocidades cada vez mais estupefacientes. A versão 2.6.9 teve 3549 patches, mais do que qualquer outra versão exceto a 2.6.0. O número de desenvolvedores também alcançou o impressionante patamar de 400 colaboradores para as últimas duas versões, tendo ficado numa média de 250 para as anteriores. Um “fork” para a versão 2.7 não desponta no horizonte, e mesmo a idéia de se dividir as versões entre *estável* e *em desenvolvimento* parece ter sido defenestrada. Para o futuro, tenho a impressão de que a estabilidade será encontrada apenas nos principados governados por desenvolvedores como Alan Cox – que não tem medo de fazer “forks” importantes em sua série *-ac* – e distribuições engajadas como Red Hat, Debian e Tinfoil Hat Linux. Na outra ponta da corda, novos recursos e características vão continuar sendo adicionados ao kernel principal a uma velocidade de quebrar pescoços e chacoalhar bochechas. Se a coisa ficar assim, uma mudança da série de 2.6 para 2.7 (ou de 2.7 para 2.8) estará relacionada apenas ao cumprimento de metas de projeto (os famosos *milestones*) e nada terá a ver com estabilidade. Para mais detalhes, consulte as notícias do kernel nos números anteriores da Linux Magazine – esse parece ser um assunto recorrente. ■