

Notícias do Kernel

■ Liberdade Total

Já faz algum tempo que o kernel Linux decide a quais recursos um módulo tem acesso baseado em sua licença. Se o módulo exporta uma variável dizendo estar licenciado sob a GPL, terá maior liberdade dentro do kernel.

A existência desta “variável da licença” é controversa, assim como a legalidade ou ilegalidade dos módulos binários; Mas alguns fabricantes de hardware proprietário conseguiram evitar esta variável, de propósito ou inadvertidamente, de uma forma bem interessante.

A LinuxAnt é uma destas empresas, e usa a seguinte string na variável de licença: “GPL\0 for files in the \“GPL\” directory; for others, only LICENSE file applies”. Note o ‘\0’ depois de ‘GPL’. Como na linguagem C o ‘\0’ significa o fim de uma string, isso faz com que o módulo veja apenas a sigla GPL e descarte o resto da frase.

O resultado é que o módulo da LinuxAnt obtém acesso total ao kernel, apesar do fato de que nem todos os seus componentes sejam livres. Alguns dos desenvolvedores acreditam que a LinuxAnt adicionou o ‘\0’ de propósito para burlar a verificação, mas de qualquer forma, futuras versões do kernel não serão vulneráveis a truques deste tipo. Ainda assim, é interessante observar o desenvolvimento deste “recurso”. ■

■ Contando os custos

O projeto ELSA (Enhanced Linux System Accounting) começou como uma tentativa de estender os mecanismos atuais de contabilização de processos. Uma das formas de atacar o problema é agrupar os processos em “bancos” de processos relacionados, que poderão ser analisados como um grupo. Em teoria, outras coisas além de processos podem ser incluídas nestes grupos, tornando possível chegar a conclusões sofisticadas de acordo com os dados obtidos.

O problema de rastrear o uso dos recursos do sistema é crucial no design de sistemas operacionais. Embora a maioria dos usuários não tenha necessidade de rastrear seu próprio comportamento, as

estatísticas sobre o comportamento de vários usuários são valiosas na criação de um grande número de algoritmos no kernel. O subsistema de memória virtual é um exemplo de recurso do kernel que pode se beneficiar de uma contabilidade precisa no sistema.

Questões como que aspectos de um sistema em execução precisam estar na RAM ou em SWAP, e quando uma transição deve ser feita, além do que fazer quando estes recursos se tornam escassos, são difíceis de responder, pois para cada sistema em particular há um comportamento adequado diferente. ■

■ Escondendo dados

Guillaume Lacôte começou a trabalhar em um meio de esconder dados no nível do mapeador de dispositivos, de modo que atacantes em potencial tenham dificuldades em interpretar as informações disponíveis a eles. Isso envolve um algoritmo de compressão que, na verdade, não precisa comprimir nada.

A compressão tende a fazer com que os dados pareçam aleatórios, porque padrões são identificados e removidos como parte do processo. Guillaume também pretende usar uma forma de interleaving para adicionar dados aleatórios.

Ele espera que um agressor simplesmente desista, em vez de tentar separar o “lixo”. Há algum apoio entre os desenvolvedores à idéia, mas não há certeza de que o método de Guillaume atingirá o nível de “ofuscação” desejado. ■

■ Só mais um recurso

A tentativa de Marcelo Tosatti de fechar a árvore do kernel 2.4 para a adição de novos recursos progride lentamente, enquanto ele tenta balancear as necessidades dos usuários com a necessidade do processo de desenvolvimento do Linux de estabilizar as árvores de código das versões 2.4 e 2.6 e avançar em direção à 2.7. Recentemente ele aceitou um patch para suporte à tecnologia Serial ATA (SATA) no kernel, uma decisão corajosa e de certa forma controversa, já que o kernel 2.6 também suporta SATA. A razão é que uma boa parte do novo

INFORMAÇÕES

A lista de discussão do kernel é o núcleo das atividades de desenvolvimento do kernel do Linux. O volume de tráfego é imenso e estar em dia com todo o processo de desenvolvimento é uma tarefa impossível para uma só pessoa. Uma das poucas almas corajosas o suficiente para aceitá-la é Zack Brown.



Nossa coluna mensal manterá você informado sobre as últimas discussões e decisões, selecionadas e resumidas pelo próprio Zack, que já publica um resumo semanal, sob a forma da lista Kernel Traffic, há vários anos. Só a leitura do resumo já é uma tarefa que consome um bom tempo. A LM agora lhe dá acesso à essência das atividades de desenvolvimento do kernel, direto da fonte.

hardware no mercado vem com controladoras SATA. A árvore 2.6 ainda não tomou conta da base de usuários devido a várias mudanças radicais que ocorreram durante o desenvolvimento da série 2.5, como novo código de módulos, novas ferramentas para compilação e outros recursos que tornam difícil simplesmente trocar um kernel 2.4 por um 2.6.

Marcelo começou a se repetir, mais uma vez dizendo que “este é o último recurso que será adicionado ao 2.4”

A tarefa de Marcelo é difícil, porque ele precisa balancear a necessidade de manter o kernel 2.4 estável como o kernel Linux principal, enquanto aceita alguns recursos julgados essenciais e rejeita outros que, por uma razão ou outra, não são tão essenciais assim. ■

■ Módulos, de um jeito fácil

Fico feliz ao ver nova documentação sobre o kernel, e recentemente Sam Ravnbourg escreveu um tutorial sobre como compilar módulos para um kernel já compilado. O sistema de configuração KBuild, tão controverso durante o desenvolvimento da série 2.5, fornece uma série de recursos que facilitam a tarefa.

Para compilar um módulo, o usuário precisa do kernel já compilado e seu código fonte. Sam lista uma série de comandos ‘make’ para realizar tarefas como compilar ou instalar módulos, ou remover arquivos objeto dos diretórios. Vale a pena dar uma olhada. ■