

NOWOŚCI W JĄDRZE

NO I GIT ;-)

Prace nad narzędziami do kontroli wersji, git i Cogito, postępują w zaskakującym tempie. Na liście dyskusyjnej twórców gita co tydzień pojawiają się 2-3 megabajty maili, zaś od czasu kiedy do rozwoju gita zaczęto używać gita (ledwie dwa miesiące temu), zgłoszono już ponad tysiąc trzysta poprawek.

Autorzy wielu projektów związanych z jądrem rezygnują z BitKeepera i przesiadają się na gita. Jednym z pierwszych takich projektów był NTFS, wykorzystujący gita od początku maja, kiedy to program ten miał zaledwie miesiąc. Gita używa się już także do rozwoju sterownika libata, JFS, kodu sterowników sieciowych i sterowników ALSA. Przeniesiono drzewo 2.4, znajdujące się pod opieką Marcela Tosattiego, a także stabilne drzewo w.x.y.z, którym zajmują się Greg Kroah-Hartman i Chris Wright. Andrew Morton zdecydował, że przy drzewie -mm nie będzie się używać żadnego systemu kontroli wersji, ale Matthias Urlichs napisał skrypt, zapisujący kolejne poprawki gałęzi -mm w repozytorium gita – pełna historia dotycząca tego drzewa także będzie zatem dostępna.

Do rozwijania jądra Linuksa praktycznie nie

używa się już BitKeepera; z drzewa źródeł jądra usunięto dokumentację tego systemu i niemal wszystkie jej ślady. To niesamowite, jak szybko prace nad jądrem wróciły do normalności – jest nawet lepiej niż było. Niektórzy programiści twierdzą, że w niektórych kwestiach git i Cogito pozwalają nawet pracować wydajniej, niż było to możliwe przy użyciu BitKeepera.

Nagła utrata BitKeepera i konieczność stworzenia jego zamiennika opóźniła jednak proces rozwoju jądra, zwłaszcza wydanie wersji 2.6.12. Opóźnienie to było tak duże, że niektórzy twórcy jądra zaczęli się niepokoić; nowe wydanie powinno jednak pojawić się w ciągu tygodnia lub paru – najprawdopodobniej zanim ta informacja ukaże się drukiem. Jądro 2.6.12-rc3 było pierwszym wydaniem, w którym wykorzystano gita. Od tego czasu pojawiły się trzy dodatkowe wydania -rc, wszystkie utworzone za pomocą tego systemu. Andrew Morton zasugerował w pewnym momencie, że wydanie 2.6.12-rc6 (już opublikowane) będzie ostatnim przed pojawieniem się jądra 2.6.12; zobaczymy, czy tak będzie faktycznie.

W tym samym czasie w związku z rozwojem gita powstało mnóstwo pomocniczych narzędzi i usług. Martin J. Bligh i wielu pracowników IBM utworzyli automatyczny system testowania dla wydań jądra Linuksa, obsługujący zarówno

wydania oficjalne, jak i wersje eksportowane każdej nocy z gita. W serwisie *kernel.org*, w którym znajdują się wszystkie wersje jądra Linuksa, utworzono stronę WWW, umożliwiającą przeglądanie kilkudziesięciu aktualnie tam przechowywanych repozytoriów gita. Można je znaleźć pod adresem <http://kernel.org/git/>. Dostępnych jest także kilka opartych na WWW (i parę niezależnych) narzędzi do przeglądania repozytorium. Lista, służąca wcześniej do rozsyłania informacji o zmianach w BitKeeperze, jest teraz wykorzystywana do informowania o zgłoszeniach poprawek do gita.

Oczywiście, nad samym gitem pracuje stale wielu dobrych programistów. Linus nadal odgrywa tam aktywną rolę, choć po zrealizowaniu najważniejszych elementów jego pierwotnego pomysłu, próbuje przekazać części projektu innym, zwłaszcza Petrowi Baudisowi, opiekującemu się Cogito. Bardzo aktywnie pracuje Junio C. Hamano, który zaimplementował wiele funkcji gita. Wielu innych także wniosło duży wkład w rozwój systemu. W ciągu najbliższych paru miesięcy udział Linusa w pracach nad gitem będzie się nadal zmniejszał, a projekt zacznie żyć własnym życiem. Niewątpliwie jednak nadal będzie się to toczyć zgodnie z życzeniami Linusa.

STAN STABILNY

Kiedy po licznych rozważaniach na temat przyszłości jądra 2.6 pojawiła się nowa, stabilna seria w.x.y.z, Linus Torvalds nie dawał jej wielu szans powodzenia. Uważał, że opieka nad tym drzewem będzie zbyt uciążliwa i na dłuższą metę nikt nie sprostą jej wymaganiom. Stworzył również zestaw bardzo sztywnych wymogów, dotyczących wyboru poprawek włączanych do drzewa i okresu ich oczekiwania na akceptację. Wymyślił nawet nazwę „drzewo frajerów”.

Greg Kroah-Hartman i Chris Wright zdecydowali się podjąć tego zadania; od tego czasu pojawiło się już ponad dziesięć wydań. Na początku zdarzyło się parę potknięć – zatwierdzono na przykład parę poprawek, nad którymi – zda-

niem Linusa – należało się jeszcze zastanowić. Udało się to jednak rozwiązać i wygląda na to, że rozwój drzewa w.x.y.z przebiega w sposób już całkiem przewidywalny i systematyczny: Greg lub Chris ogłaszają nowy zestaw poprawek do rozważenia, przez kilka dni trwa nad nimi dyskusja, a potem, na jej podstawie, tworzone jest nowe wydanie.

Kilku programistów, m.in. Jeff Garzik i Alan Cox, chwali pracę Grega i Chrisa, potwierdzając, że drzewo w.x.y.z jest faktycznie stabilniejsze i lepiej nadaje się do zastosowań komercyjnych. Wcześniejsze założenia, dotyczące stabilizowania jądra dopiero w poszczególnych dystrybucjach, okazały się w zasadzie niepotrzebne – jądro zaczyna być stabilne niezależnie od tego, nawet pomimo poważnych zmian w kodzie dokonywanych podczas prac nad jądrem 2.6.

SMP W KLASTRACH

Dinakar Guniguntala pracował nad połączeniem w klastry procesorów w systemach przetwarzania równoległego (SMP) tak, aby określone procesy wykonywane były tylko na konkretnych zestawach procesorów. To jedno z wielu usprawnień szeregowania zadań, wprowadzanych w jądrze Linuksa w ostatnich latach. Praca Dinakara naprawdę robi wrażenie, programiści tacy jak Nick Piggin i Paul Jackson nadal mają jednak problemy związane z implementacją. Ciągłe istnieją problemy ze zgodnością pomiędzy tą poprawką i innymi związanymi z nią łataniami, np. z poprawką odpowiedzialną za podłączanie procesorów podczas pracy (CPU hotplug). Przez pewien czas poprawka ta będzie prawdopodobnie funkcjonować niezależnie, ponieważ lista jej funkcji różnie i pojawiają się różne kwestie dotyczące zgodności.

IDE

Sterownik IDE znów prowokuje programistów do wytaczania ciężkiej amunicji i ścierania się nawzajem na proch. Wygląda na to, że opiekunowie tego sterownika skazani są na ciągłe cierpienie. Tym razem w starciu uczestniczyli Alan Cox i Bartłomiej Żołnierkiewicz. Alan – który uratował IDE podczas wczesnych prac nad rozwojem jądra 2.6 – uważa, że cały kod warstwy IDE wygląda coraz gorzej; Bartłomiej zaś – aktualnie opiekujący się sterownikiem – twierdzi, że nic nie zostało uszkodzone, nie usunięto żadnych funkcji, a Alan miał wiele możliwości dyskusowania na temat każdej z poprawek, gdy były zgłaszane na liście dyskusyjnej.

Kłątwa IDE sięga głębiej i wiąże się nie tyle z kodem, co ze sprzętem. Ze względu na wszystkie standardy IDE i ich niekompatybilne implementacje, opiekun IDE musi w pewnym sensie przewidzieć wszystkie szczególne sytuacje, wymagania, konfliktowe wersje, przypadki trudnego do zidentyfikowania sprzętu i sprawić, że wszystko to będzie działać. Fakt, że IDE jest jednym z najważniejszych elementów jądra Linuksa, niczego nie ułatwia – jeżeli choćby niewielka część napędów IDE nie będzie funkcjonować poprawnie pod Linuksem, pojawi się mnóstwo głosów protestu. IDE po prostu musi działać.

Kilku wyjątkowo dobrych programistów próbowało sprostać temu wyzwaniu. Przez lata zajmował się tym Mark Lord, który jednak

poddał się ostatecznie po względnie niewielkiej różnicy zdań z Linusem. Andre Hedrick dawał z siebie wszystko, ale i jego projekt ten doprowadził powoli do szaleństwa. Marcin Dalecki, na życzenie Linusa, przejął prace nad IDE w jądrze 2.5; próbował naprawić sterownik za pomocą młota oburęcznego, ale został usunięty przez programistów sfrustrowanych faktem, że w każdym wydaniu jądra sterownik IDE zawierał błędy. Alan uczynił w końcu rzecz niemożliwą i sam naprawił sterownik, nie chciał jednak stale się nim opiekować; zadanie to trafiło więc ostatecznie do Bartłomieja.

Nie wiadomo, co będzie się działo dalej; jeśli jednak brać pod uwagę przeszłość, Bartłomiej ma ciężki orzech do zgryzienia.

NARUSZENIE GPL

Carlos Silva poinformował o możliwości naruszenia GPL przez firmę Panda. Jak twierdzi Silva, rozprowadzany przez tę firmę system Gatedefenders 8200 oparty jest na Linuksie, ale jego twórcy nie wspominają o tym na swojej stronie WWW ani w dokumentacji. Nie udostępniają też kodu źródłowego jądra, tłumacząc się, że zawiera ono kod chroniony prawami autorskimi. Jeżeli okaże się to prawdą, będzie ewidentnym naruszeniem Powszechnej Publicznej Licencji GNU. Różnego rodzaju przypadki złamania GPL pojawiają się dość regularnie, a czasami trwają nadal nawet wtedy, gdy firma naruszająca licencję zostaje o tym poinformowana. Miejmy nadzieję, że tym razem będzie inaczej.

ZAPISYWANIE W SYSTEMACH PLIKÓW TYLKO DO ODCZYTU

Markus Klotzbuecher napisał narzędzie zwane `mini_fo`, które, najwyraźniej, potrafi uczynić rzecz niemożliwą: pozwala zapisywać dane w plikach i systemach plików tylko do odczytu. Nie jest to jednak naruszenie bezpieczeństwa, a plikom użytkowników nic nie grozi. Działanie nowego, „wirtualnego” systemu plików stworzonego przez Marcusa polega na pozornym przekierowaniu. Program tworzy w odpowiednim systemie plików obszar, w którym użytkownik może zapisywać dane, i nakłada te dane na system plików przeznaczony tylko do odczytu. Użytkownik ma wrażenie, że edytuje pliki `read-only`, w rzeczywistości jednak tworzy tylko nakładkę. Przy próbie odczytania widać oryginalne pliki i nałożone na nie zmiany. Z punktu widzenia użytkownika wygląda to tak, jakby pliki `read-only` zostały zmienione. Narzędzie to ma wiele zastosowań, poczynając od systemów wbudowanych, w których należy zachować obszar przeznaczony tylko do odczytu, ale jednocześnie umożliwić użytkownikowi konfigurowanie tego obszaru i wprowadzanie w nim zmian.

**Jeżeli chcesz wiedzieć
jak się skutecznie
zabezpieczyć
– zgłoś się już dziś!**



Chcesz poznać:

- * praktyczne metody przelamywania zabezpieczeń
- * metody ukrywania obecności intruza w systemie
- * zacieranie śladów przez intruzów
- * zabezpieczanie się przed włamaniami
- * praktyczne aspekty analizy powłamaniowej
- * metody namierzenia intruza
- * prawdziwe przykłady z życia wzięte

Zapraszamy do udziału w trzydniowych warsztatach, które przybliżą uczestnikom metody stosowane przez komputerowych włamywaczy i w efekcie pomogą obmyślić strategię obrony.

**Warsztaty
Hacking Linux
Cała prawda
5-7 września 2005
Warszawa**

**Warsztaty
Hacking Windows
Cała prawda
28-30 września 2005
Warszawa**

**software
KONFERENCJE**

Szczegółowe informacje:
Lizbieta Rogowska
ul. Bizielka, 10, 02-097 Warszawa, e.com.pl
tel: (022) 887 14 66
www.software.com.pl/konferencje