UNIWERSYTET MIKOŁAJA KOPERNIKA

Wydział Fizyki, Astronomii i Informatyki Stosowanej

Jakub Przybyła

Mechanizm szeregowania zadań w jądrach z serii 2.6 systemu GNU/Linux

Toruń 2007

Spis treści

1 Wstęp

System operacyjny GNU/Linux, który jest intensywnie rozwijany od 1991 r., osiągnął taką dojrzałość i jakość, że na jego bazie powstały dwie dystrybucje klasy korporacyjnej: Red Hat Enterprise oraz Suse Linux Enterprise Server. Jądro systemu GNU/Linux jest w wysokim stopniu konfigurowalne nie tylko dzięki swojej modułowej budowie, prawie 800 konfigurowalnym parametrom, które można zmieniać w czasie pracy systemu, ale także dzięki dostępności źródeł, które także można modyfikować. Żeby jednak dostrajać jądro do konkretnych potrzeb trzeba dobrze poznać jego funkcjonowanie.

Niniejsza praca poświęcona jest budowie narzędzia, które służy do śledzenia pracy planisty, czyli jednego z kluczowych elementów jądra. Składa się ono ze specjalnego modułu jądra *kernstat* oraz programu *schedstat* działającego w przestrzeni użytkownika. Moduł *kernstat* służy do przekazywania danych sterujących generowanych przez program *schedstat* z przestrzeni użytkownika do przestrzeni jądra, odczytywaniu odpowiednich danych ze wskazanych struktur jądra oraz przekazywanie ich z powrotem do przestrzeni użytkownika. Wymiana danych odbywa się za pośrednictwem pliku urządzenia znakowego */dev/kernstat*.

Moduł *kernstat* został tak pomyślany i zaimplementowany, aby w przyszłości mógł zostać rozszerzony o dodatkowe funkcje pobierające z jądra inne parametry jego pracy. Jeśli zostanie to powiązane z tworzeniem odpowiednich narzędzi działających w przestrzeni użytkownika, narzędzi podobnych do programu *schedstat*, to uzyskamy nie tylko cenne narzędzia diagnostyczne, ale także dydaktyczne, które powinny ułatwić zrozumienie działania jądra systemu GNU/Linux.

Budowa omawianego narzędzia jest utrudniona z dwóch powodów. Po pierwsze, dostęp do zmiennych planisty z poziomu modułów jądra, a w szczególności do kolejki zadań gotowych do wykonania, jest niemożliwy, ponieważ konsolidator tworzy takie dane w specjalnej sekcji wykonywalnej *.data.percpu*, do której moduły mają zabroniony dostęp [1]. Po drugie, nie mamy możliwości wykorzystania definicji zmiennych, struktur danych oraz funkcji planisty, gdyż te definicje nie są dostępne poprzez pliki nagłówkowe.

Taka możliwość istniała jeszcze do wersji 2.6.18, gdyż prawie wszystkie potrzebne definicje znajdowały się w pliku *<include/linux/sched.h>*, a nie *<kernel/sched.c>*. Ta zmiana podyktowana została względami bezpieczeństwa, ale utrudnia budowę modułów wykorzystujących te definicje. Poprzednie podejście pozwalało na tworzenie instancji odpowiednich zmiennych. Wówczas dostęp do zmiennych uprzednio zdefiniowanych wymagał jedynie odczytania adresów z eksportowanych symboli jądra i ich rzutowanie (o tym czy jakiś symbol jest eksportowany można się przekonać przeglądając plik */boot/System.map-ver*, gdzie *ver* jest numerem wersji jądra).

Przy implementacji sterownika urządzenia znakowego *kernstat* oraz funkcji, które

są odpowiedzialne za pobieranie statystyk planisty wykorzystano jądro 2.6.18 ze specjalnie zmodyfikowaną wersją pliku *<kernel/sched.c>* oraz *<linux/sched.h>*. Zmiany polegają na dodaniu oraz wyeksportowaniu funkcji, która daje nam bezpośredni dostęp do instancji zmiennej typu *struct rq*, czyli do kolejki zadań gotowych do wykonania. Bywa tak, że kolejne wersje jądra wprowadzają zmiany w nazewnictwie symboli, tzn. zmiennych, struktur danych oraz funkcji, a także w położeniu ich definicji. W celu uniknięcia tego problemu ograniczono rozważania do jednego ściśle określonego jądra, co daje gwarancję, że tworzony moduł będzie działać poprawnie.

Dzięki zastosowaniu programu *schedstat* możliwe jest obserwowanie w jaki sposób system operacyjny realizuje operacje wielozadaniowości, jak dba o to, aby w systemie, w którym pracuje jednocześnie wiele procesów nie dochodziło do przestojów. Program *schedstat* ilustruje mechanizmy wykorzystywane przez planistę w postaci prezentacji różnego rodzaju statystyk oraz warunków determinujących odpowiednie ich traktowanie.

Układ pracy jest następujący. Po wstępie następują trzy rozdziały omawiające zagadnienia związane z działaniem i implementacją planisty oraz zasad działania i tworzenia sterowników oraz modułów w jądrach systemu GNU/Linux. W rozdziale piątym przedstawiono także szczegóły implementacji modułu *kernstat* oraz programu *schedstat*. Kolejny rozdział zawiera omówienie szeregu przykładowych zastosowań programu *schedstat*. Pracę kończy dodatek wyjaśniający znaczenie szeregu terminów przewijających się w pracy.