

Sprzętowa implementacja SMP w systemie Linux

**Państwowa Wyższa Szkoła Zawodowa w Elblągu
Instytut Informatyki Stosowanej**

Marzec 2003

mgr inż. Mariusz Bagiński, mgr Katarzyna Wasielewska

Abstract:

This paper describes some facts about implementing and starting SMP (Symmetric MultiProcessing) – architecture on Linux platform. Describes kernel configuring and compiling, APIC chip, local APIC chips and interrupts management.

Celem niniejszej publikacji jest wprowadzenie w zagadnienie wieloprocesorowości - architektury SMP w systemie Linux. Przedstawiono sposób konfiguracji i kompilacji jądra, układ APIC i lokalne APIC oraz sterowanie przerwaniem.

W komputerze z jednym procesorem w danej chwili wykonuje się tylko jeden proces. Jeżeli system umożliwi równoczesne uruchomienie wielu aplikacji, to w rzeczywistości tylko jedna z nich korzysta w danym momencie z procesora. Pozostałe oczekują na swoją kolej. Natomiast w komputerze z wieloma procesorami oraz odpowiednim systemem operacyjnym aplikacje mogą wykonywać się równocześnie na wszystkich procesorach. W ten sposób uzyskuje się znaczną poprawę wydajności jej pracy.

Systemy wieloprocesorowe wykorzystuje się jako serwery sieciowe, złożone stacje robocze, inżynierskie systemy CAD/CAM/CAE, systemy bazodanowe i in. Dostarczają one olbrzymiej mocy obliczeniowej. Duże wpływ ma tu też wykorzystanie dysków SCSI.

Najprostszą, a zarazem najbardziej rozpowszechnioną, architekturą wieloprocesorową jest architektura SMP (*Symmetric MultiProcessing*) – symetryczne przetwarzanie na wielu procesorach, w szczególności: zasoby pamięci oraz I/O są współdzielone przez wszystkie procesory i komunikują się z nimi za pośrednictwem wspólnej magistrali lub przełącznika. Do zalet SMP należy również prosty model programowy.

W strukturze SMP przydział procesorów dla poszczególnych wątków realizowany jest przez system operacyjny. Zatem wielozadaniowy system operacyjny musi być przystosowany do dzielenia zadań (również tych, które wchodzi w skład samego systemu) na kolejne fragmenty do wykonania w wyznaczonych przedziałach czasu przez pojedynczy procesor oraz rozdzielania ich na wiele procesorów.

SMP obsługują m.in. procesory Intel 486, Pentium i szybsze, ale zgodne z Intel MP1.1/1.4 oraz UltraSparc, SparcServer, Alpha i PowerPC.

Do systemów obsługujących SMP należy między innymi Linux.

Linux może pracować praktycznie na dowolnej platformie sprzętowej, z procesorami firmy Intel, PowerPC, Alpha, Sparc i in.

Obsługa SMP została zaimplementowana w jądrze Linuxa począwszy od wersji 1.2.33. Obecnie jądro obsługuje od 2 do 32 procesorów, w zależności od wersji jądra i dystrybucji systemu.

Większość dystrybucji Linuxa nie jest zaopatrzona w gotowe jądro SMP. Aby jądro obsługiwało symetryczną wieloprocesorowość należy je przekompilować i włączyć obsługę SMP:

- w przypadku jądra wersji od 2.0 (z wyłączeniem 2.1.132) należy w pliku `/usr/src/linux/Makefile` odznaczyć linię `SMP=1`;
- w wersji 2.2 podczas konfiguracji należy odpowiedzieć twierdząco na pytanie o obsługę SMP.

W przypadku pracy na procesorach x86 należy włączyć obsługę zegara czasu rzeczywistego (*RTC Support – Real Time Clock*), wyłączyć obsługę APM (*Advanced Power Management*) oraz włączyć obsługę MTRR (*Memory Type Range Register*) Support..

Obsługa RTC jest wymagana w przypadku pracy na niektórych płytach Intela.

SMP i APM nie są kompatybilne, zatem włączenie APM może powodować zawieszanie systemu podczas uruchamiania. W wersji 2.1.x włączenie obsługi SMP automatycznie wyłącza obsługę APM.

Obsługa MTRR jest wymagana przez niektóre BIOS'y starszych płyt głównych, które wymagają ręcznego włączenia pamięci cache dla drugiego procesora.

Dla poprawnej pracy z SMP konieczna jest również kompilacja wszystkich modułów jądra (`make modules`, `make modules_install`).

W przypadku niektórych wersji 2.2.x mogą zaistnieć problemy z przejściem z SMP do pracy jednoprocessorowej. Dlatego należy zachować oryginalny plik `.config`, wykonać kompilację `make mrproper`, przywrócić plik `.config` i przebudować jądro (`make dep`). Należy pamiętać o przeładowaniu LILO po skopiowaniu nowego jądra.

Przykładowa kompilacja:

```
make config # albo menuconfig albo xconfig
make dep
make clean
make bzImage
# skopiować ręcznie obraz jądra i uruchomić LILO
# albo make lilo
make modules
make modules_install
```

Informację o trybie pracy i ilości procesorów można znaleźć w pliku `/proc/cpuinfo`.

Do zbadania wydajności systemu SMP można wykorzystać któryś z testów Cameron'a MacKinnon'a dostępnego na <http://www.phy.duke.edu/brama/benchmarks.smp>.

Systemy wieloprocesorowe sterują statyczną i dynamiczną dystrybucją przerwań pomiędzy poszczególnymi procesorami dzięki istnieniu układów APIC (*Advanced Programmable Interrupt Controller*), które znajdują się w procesorach i na płycie głównej, a obecnie w całości w chipie płyty głównej.

Jednym z pierwszych układów APIC (lub IOAPIC) był układ 82093AA.

W dużym uproszczeniu układ IOAPIC składa się z pięciu elementów:

- interfejsu magistrali systemowej (*System Bus Interface*)
- interfejsu magistrali APIC (*APIC Bus Interface*)
- kontrolera przerw (Interrupt Controller)
- wejścia zegarowo-resetującego (*Clock and Reset*)
- wejścia testującego (*Test*)

Układ APIC może być z powodzeniem stosowany również w systemach jednoprocessorowych. Przerwania mogą być kontrolowane zarówno przez standardowy kontroler ISA jak i osobną jednostkę APIC albo przez oba układy jednocześnie. Zastosowanie tego układu może nieco zmniejszyć opóźnienia, jakie wnoszą do systemu zwykłe kontrolery. Oprócz tego systemy uniprocessorowe, aby skorzystać z APIC powinny posiadać tzw. lokalne układy APIC (*Local APIC*). Znajdują się one w samym procesorze. Układ lokalny pozwala podejmować decyzję, czy dana jednostka obliczeniowa może w danej chwili współpracować z magistralą APIC.

W systemach SMP zaleca się stosowanie jednakowych lokalnych układów APIC w każdym procesorze, co w praktyce przekłada się na używanie procesorów z tej samej serii produkcyjnej.

Linux SMP obsługuje kontroler APIC standardowo – wszystkie wymagane opcje związane z lokalnymi APIC i głównym APIC podczas konfiguracji jądra są zaznaczane automatycznie.

Poniższy przykład pokazuje jak w 8-procesorowym komputerze można „przerzucić” obsługę przerw operując na pliku `proc/irq/IRQ#/smp_affinity`, który specyfikuje docelowe CPUs, poprzez ustawienie odpowiedniej bitmaski. Manipulujemy tutaj IRQ44 (eth1) pomiędzy dwoma grupami procesorów, tj. CPU0-3 i CPU4-7 (numeracja procesorów zaczyna się od 0).

Przykład.

```
[root@serwer]# cat smp_affinity
ffffffff
[root@serwer]# echo 0f > smp_affinity
[root@serwer]# cat smp_affinity
0000000f
[root@serwer]# ping -f h
PING hell (10.0.0.1): 56 data bytes
...
--- hell ping statistics ---
6029 packets transmitted, 6027 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.4 ms
[root@serwer]# cat /proc/interrupts | grep 44:
44: 0 1785 1785 1783 1783 1 1 0 IO-APIC-level eth1
[root@serwer]# echo f0 > smp_affinity
[root@serwer]# ping -f h
PING hell (10.0.0.1): 56 data bytes
...
--- hell ping statistics ---
2779 packets transmitted, 2777 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.5/585.4 ms
[root@serwer]# cat /proc/interrupts | grep 44:
```

44: 1068 1785 1785 1784 1784 1069 1070 1069 IO-APIC-level eth1

Architektura SMP jest technologią dojrzałą, stabilną i wydajną. Znajduje to potwierdzenie w implementacji tej architektury przez duże światowe koncerny komputerowe jak np.: IBM, HP, Compaq, Dell.

Literatura:

1. Linux SMP Howto, David Mentré, v1.8, 8 november 1999
2. Dokumentacja projektu Linux SMP: www.linux.org.uk/SMP/title.html
3. Testy Cameron'a MacKinnon'a: <http://www.phy.duke.edu/brahma/benchmarks.smp>
4. Źródła jądra Linuxa: www.kernel.org
5. Dokumentacja techniczna Intela: www.intel.com