

Rola BIOSu w systemie operacyjnym. Specyfikacja EFI

Wstęp

Zgodnie z intencjami inżynierów firmy IBM, BIOS oprócz wykonywania procedur inicjalizujących wywoływanych po włączeniu komputera miał służyć jako warstwa pośrednicząca między systemem operacyjnym a sprzętem. Tak zwane "usługi BIOSu" - to funkcje obsługi urządzeń takich jak napędy, karta graficzna, klawiatura itp. Usługi BIOSu są zrealizowane za pomocą mechanizmu przerwań programowych - po uruchomieniu komputera BIOS wpisuje do tablicy wektorów przerwań adresy procedur obsługi odpowiednich przerwań. System operacyjny, chcąc wykonać jakąś operację na urządzeniu, generuje przerwanie poleceniem *int*. System operacyjny działał zatem w pewnym sensie na "maszynie wirtualnej" ukrywającej przed nim szczegóły sprzętowe.

Wykorzystanie BIOSu w systemie MS-DOS

W przypadku starszych systemów operacyjnych ten model rzeczywiście był zachowany - bardzo często wykorzystywały one funkcje BIOSu. Niektóre systemy operacyjne mogły być wręcz nazwane "nakładką" na BIOS, ponieważ nie oferowały dużo więcej niż obsługę systemu plików, wszystkie pozostałe operacje wykonywane były przy użyciu wywołań funkcji BIOSu. Najbardziej jaskrawym przykładem jest system DOS - pozbawiony wieloprogramowości, ciężar obsługi pamięci zrzucał na programy użytkownika (prymitywne rozwiązania w rodzaju sterownika EMM) itp. Przez wiele osób nie jest w ogóle nazywany systemem operacyjnym, na przykład w podręczniku programu Emacs paragraf o uruchamianiu Emacsa pod MS-DOS zaczyna się tak:

This section briefly describes the peculiarities of using Emacs under the MS-DOS "operating system" (also known as "MS-DOG").

System DOS sprowadzał się w gruncie rzeczy do dwóch elementów - od strony użytkownika była to garstka programów diagnostycznych i niewygodny interpreter poleceń (pozbawiony historii wpisywanych poleceń za to z możliwością zmiany znaku zachęty), od strony programisty był to przede wszystkim zestaw funkcji przerwania 21h, takich jak między innymi:

Funkcja	Opis
09H	Wypisywanie tekstu
0AH	Czytanie wiersza z klawiatury.
0FH	Otwieranie pliku
13H	Usuwanie pliku
14H	Sekwencyjne czytanie pliku
15H	Sekwencyjne pisanie w pliku
16H	Tworzenie pliku

2AH	Pytanie o datę
2CH	Pytanie o czas

Kod procedury obsługi tych przerw wywoływał przerwania BIOSu. Na przykład przy operacjach dyskowych wywoływał funkcje przerwania 13h, przy komunikacji z klawiaturą - przerwania 09h, wypisywanie na ekran - 10h, czas systemowy - 70h. Programy użytkownika również miały możliwość wywoływania funkcji BIOSu, mogły one również ustawiać własne procedury obsługi przerw.

Większość współczesnych systemów operacyjnych nie komunikuje się ze sprzętem poprzez wywołania funkcji BIOSu, ale wykorzystuje własne sterowniki. Dotyczy to szczególnie operacji dyskowych. Dzieje się tak z dwóch powodów - po pierwsze wywoływanie funkcji BIOSu podczas każdej transakcji dyskowej jest nieefektywne. Po drugie starsze wersje BIOSu mają trudności ze współpracą z dużymi dyskami twardymi. Zamiast zmuszać użytkowników do aktualizowania BIOSu (co to znaczy?) system operacyjny przejmuje na siebie obsługę urządzeń.

Na potwierdzenie przedstawiamy tabelkę prezentującą liczbę wywołań przerw BIOSu w kodzie jądra Linuxa (wersja 2.4.20).

Plik	Numer przerwania	Liczba wywołań
arch/i386/boot/bootsect.S	0x10	9
	0x13	5
arch/i386/boot/setup.S	0x10	1
	0x11	1
	0x13	5
	0x15	11
	0x16	1
	0x1a	1
arch/i386/boot/video.S	0x10	31
	0x16	4
arch/x86_64/boot/bootsect.S	0x10	9
	0x13	4
arch/x86_64/boot/setup.S	0x10	1
	0x11	1
	0x13	3
	0x15	4
	0x16	1
	0x1a	1
arch/x86_64/boot/video.S	0x10	31
	0x16	4

Jak widać Linux wykorzystuje funkcje BIOSu (takie jak wykonywanie operacji dyskowych, wypisywanie na ekran w trybie tekstowym) tylko podczas startu systemu.

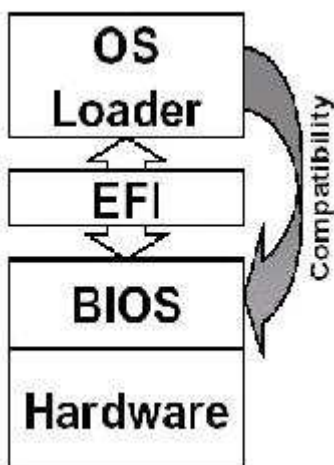
Widać zatem, że współczesne systemy operacyjne "obchodzą" BIOS i komunikują się bezpośrednio ze sprzętem. Do wykorzystywania funkcji BIOSu są natomiast zmuszeni twórcy *boot loaderów*, czyli programów ładujących

system operacyjny. Z tego powodu nie jest możliwe wystartowanie systemu operacyjnego, którego obraz znajduje się na urządzeniu, z którym nie współpracuje BIOS (lub na tej części dysku twardego, do której nie ma dostępu). Drugim problemem, który napotyka się podczas tworzenia *boot loadera* jest bardzo uboga funkcjonalność oferowana przez BIOS. Na przykład aby *boot loader* mógł działać w trybie graficznym (tak jak program GRUB) sam musi zapewniać obsługę trybu graficznego karty.

Alternatywy dla BIOSa. Specyfikacja EFI

Podobno inżynierowie z IBM, którzy opracowali BIOS, przewidywali, że trafi on do około 250 tysięcy komputerów, a następnie zostanie zastąpiony innym, lepszym rozwiązaniem. Stało się jednak inaczej - BIOS nieprzerwanie od ponad dwudziestu lat króluje na komputerach osobistych. Natomiast jego rola systematycznie się zmniejszała. Obecnie właściwie jedynymi programami wykorzystującymi BIOS są programy ładujące system operacyjny.

Z tych powodów firma Intel opracowała specyfikację EFI (Extensible Firmware Interface). EFI jest warstwą pośredniczącą między programem ładującym system operacyjny a sprzętem. Ponadto system operacyjny już po uruchomieniu może wywoływać funkcje EFI. Różnicą w stosunku do tradycyjnego modelu, w którym sterowniki urządzeń takich jak karty graficzne, sieciowe itp. umieszczane są jako *firmware* w pamięci ROM, jest fakt, że według specyfikacji EFI sterowniki mogą być zapisywane na twardym dysku (patrz dalej). Ponadto nie będą one napisane w assemblerze, lecz w C.



Dużym zyskiem jest fakt, że już w chwili uruchomienia *boot loadera* ma on do dyspozycji bogate środowisko - tryb graficzny, stos TCP/IP itp. To samo dotyczy programu do konfiguracji urządzeń (odpowiednika Setupa BIOSu) - nie będzie on już nieprzyjemną aplikacją w trybie tekstowym, tylko przyjaznym programem z graficznym interfejsem użytkownika, który na dodatek można uruchamiać zdalnie.

Wprowadzenie EFI nie oznacza całkowitej rezygnacji z BIOSu. Płyty główne nadal będą zawierać kostkę z BIOSem, zawierającą instrukcje wykonywane przez procesor tuż po uruchomieniu komputera. Podobnie jak dzieje się to dotychczas, pierwszym krokiem będzie sprawdzenie pamięci, wykrycie twardego dysku itp. Na końcu zainicjalizowany zostanie interfejs EFI. Oczywiście sterowniki urządzeń nie zostaną zamieszczone w pamięci flash BIOS (po pierwsze kości mają ograniczoną pojemność, po drugie dodanie nowego sterownika oznaczałoby konieczność uaktualnienia pamięci flash), tylko zapisane na specjalnie wydzielonym obszarze twardego dysku. Ta część specyfikacji jest przedmiotem wielu kontrowersji, ponieważ specyfikacja określa, że ten obszar dysku powinien być partycją logiczną z systemem plików FAT 32 o rozmiarze równym 1% rozmiaru całego dysku. Na partycji oprócz sterowników można zamieszczać kod *boot loaderów* oraz różnego rodzaju narzędzia diagnostyczne.