

Bios, partycje dysków, programy ładowania wstępnego**A) BIOS**

- kod 16 i 32 bitowy
- obsługa dysków twardych
 - geometria CHS i jej ograniczenia
 - limit „1024 cylindrów” cylindrów sposoby jego obejścia
- LBA

B) Partycje dysków

- cel partycjonowania (klaster, FAT 12, FAT 16, FAT 32)
- typy partycji, kody systemów plików
- narzędzia do partycjonowania
 - FDisk (DOS, Windows 9x)
 - Administrator dysków dla Windows NT (i Windows 2000)
 - FDisk (fdisk) w systemie Linux
 - Partition Magic (Power Quest)

C) Programy ładowania wstępnego

- działanie programu ładującego
- ukrywanie partycji
- kategorie wielosystemowych programów ładujących
 - dyskietkowa
 - oparte na MBR (Master Boot Record)
 - oparte na „boot sektorach”, programy ładujące w dedykowanych partycjach
 - ładowanie z poziomu pracującego SO (Loadlin)
- programy ładujące
 - Windows NT Loader
 - LILO (Linux)
 - BootMagic (Power Quest)

A) BIOS (dla komputerów x86)

Basic Input Output System – program zapisany w pamięci stałej płyty głównej wykonujący szereg czynności wstępnych (po włączeniu komputera) zwanych bootowaniem.

W oryginalnym PC mamy pamięć segmentowaną. Znaczy to, że adres jest dwuczłonowy (segment:offset). Segment zaczyna się zawsze od adresu podzielnego przez 16 (Pamięć jest zatem widziana jako 16 bajtowe paragrafy). Zatem adres jest określany jako:

$$\text{adres} = (\text{numer segmentu} * 16) + \text{przesunięcie}$$

Taki schemat adresowania nazywany jest adresowaniem w trybie rzeczywistym.

Procesory 8086, 8088, 80186 miały 20-bitową szynę adresową, więc ten tryb adresowania był jedyny. $(2^{16} - 1) * 16 + (2^{16} - 1) = 1.114.095$.

Poczynając od procesora 80286 (24-bitowa szyna adresowa) mamy do czynienia z trybem chronionym; oznacza to, że adres segmentu jest obliczany w specjalnych strukturach procesora. W 16-bitowym rejestrze, 14 bitów określa numer segmentu (mamy $2^{14} = 16.384$ segmenty). Zatem obszar adresowalny to $16.384 * 65.536 = 1.073.741.824$ bajty.

Począwszy od procesora 80386 mamy 32-bitowe określenie przesunięcia, co powiększa segment do 4 GB. Magistrala adresowa ma tu szerokość 32 bity, co pozwala adresować także

4 GB pamięci fizycznej. Zakres adresów wirtualnych jest równy 2^{14} segmentów po 4 GB czyli 64 TB.

Od Pentium II szerokość magistrali adresowej wynosi 36 bitów, co zwiększa górną granicę pamięci fizycznej do 64 GB.

Wszystkie procesory nadal dysponują klasycznym rzeczywistym (16-bitowym) trybem adresowania, który jest wykorzystywany w momencie „bootowania”.

Najważniejszą cechą BIOS-u jest zapewnienie środków do załadowania systemu operacyjnego. Każdy procesor jest zaprojektowany w ten sposób, iż po włączeniu komputera przystępuje do wykonania kodu znajdującego się w określonym z góry miejscu. W procesorach Inlet miejsce to zawiera kod BIOS-u wykonujący podstawowe funkcje (między innymi autotest (POST)). Po wykonaniu POST (Power On Self Test) poszukuje urządzenia (dysku, dyskietki), z którego można by załadować system operacyjny. Po znalezieniu, wczytuje kod inicjujący i przekazuje mu sterowanie.

Inną funkcją BIOS-u jest regulacja różnorodnych aspektów funkcjonowania (konfiguracja ustawień sprzętowych)

- geometria dysków twardych
- sprawnościowe aspekty (szybkość dostępu do pamięci,...)
- bezpieczeństwo (hasło dostępu, hasło BIOS-u)
- APM (Advanced Power Management)
- Zegar

Producenci BIOS-u

AMI – American Megatrends Inc.	(http://www.ami.com)
Award	(http://www.award.com)
Phoenix Technologies	(http://www.phoenix.com)
Microid Research	(http://www.mrbios.com)

We wczesnych stadiach rozwoju dysków ustalono, że będą używane następujące maksymalne parametry:

Ilość cylindrów:	1024	(10 bitów)
Ilość głowic:	256	(8 bitów)
Ilość sektorów:	63	(6 bitów)

Obecnie limity wynikające z IDE to:

16 bitów na numer cylindra, 4 bity na numer głowicy, 8 bitów na numer sektora.

Wybierając mniejsze wartości otrzymujemy: 1024 cylindry, 16 głowic i 63 sektory. Daje to $1024 \cdot 16 \cdot 63 \cdot 512 \text{ B} = 516\,096 \text{ kB}$ (w przybliżeniu 504 MB).

Gdyby nie ograniczenia IDE to mogłoby być (BIOS) $1024 \cdot 256 \cdot 63 \cdot 512 = 7,8 \text{ GB}$. Tryb adresowania LBA (Logical Blok Addressing).

Niech (Ct, Ht, St) określa geometrię dysku. Wówczas liniowy (logiczny) adres sektora będzie się wyrażał wzorem:

$$\text{LBA} = ((\text{Cn} \cdot \text{Ht} + \text{Hn}) \cdot \text{St}) + \text{Sn} - 1$$
 gdzie (Cn, Hn, Sn) jest adresem CHS. Ponieważ w EIDE adres dyskowy ma 28 bitów (16-C, 4-H, 8-S), więc w LBA można zaadresować dyski posiadające 2do 28 sektorów, tj. ok. 128 MB.

B) Partycje dysków

Jedną z pierwszych przyczyn podziału dysku na partycję była konieczność podziału dużych dysków na segmenty mieszczące się w limitach rozmiaru wyznaczanych przez ówczesne systemy.

Dyski dzieli się na partycje także z innych powodów (efektywne wykorzystanie miejsca, wiele systemów operacyjnych, itp.).

Cechą wspólną wszystkich systemów plików jest przydział miejsca w ustalonych kwantach zwanych klastrami.

Rozmiary klastrów

FAT-16	Klaster (kB)
Rozmiar partycji (MB)	
1 – 127	2
128 – 255	4
256 – 511	8
512 – 1023	16
1024 – 2048	32

FAT-32	Klaster(kB)
Rozmiar partycji (MB)/(GB)	
<512	nie wykorzystywane
512 – 8000 (8GB)	4
8 GB – 16 GB	8
16 GB – 32 GB	16
>32 GB	32

Im większy rozmiar klastrera tym więcej miejsca się traci, bo średnio każdy plik nie wykorzystuje połowy klastrera.

Np. na 2 GB partycji wypełnionej całkowicie plikami o rozmiarze 30 – 50 kB marnuje się około 40 % (FAT –16).

$2\text{GB} / 40\text{kB} \approx 52\,000$ plików;

$52\,000 * (32\text{kB}/2) \approx 858\,996\,000$ bajtów co stanowi około 40% z 2 GB.

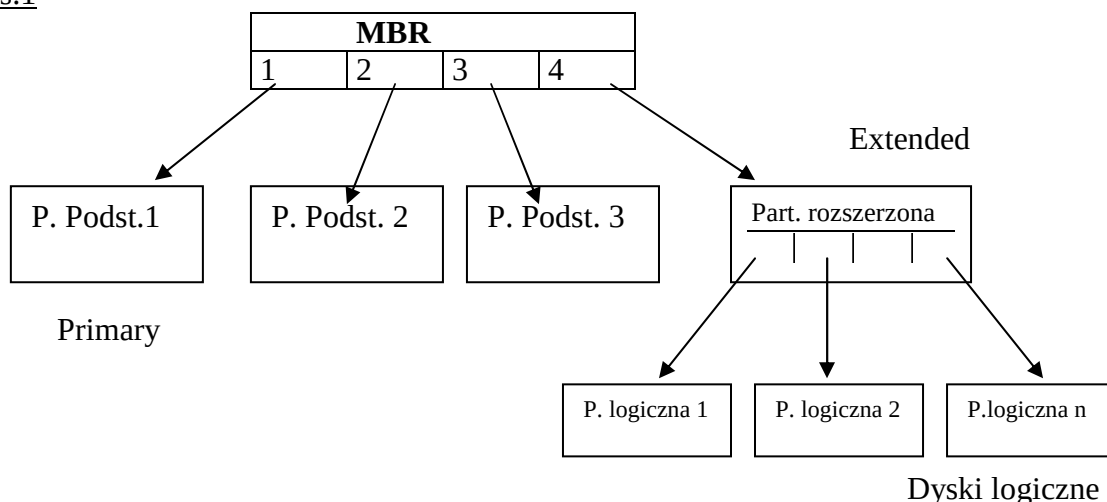
Gdyby podzielić tę partycję na $4 * 512$ MB to strata wynosiłaby ok. 10%.

Typy partycji

1. podstawowe (tylko 4 na 1 dysku – IBM)
2. rozszerzone (próba obejścia ograniczenia do 4 partycji podstawowych)
3. logiczne (w ramach partycji rozszerzonej)

Dwupoziomowy schemat podziału dysku na partycje

Rys.1



Niektóre systemy operacyjne muszą być bootowane z partycji podstawowych (Microsoft, FreeBSD). Wynika to z ich ścisłego związku z MBR. Nie jest konieczne, by partycja rozszerzona była ostatnia! Zaleca się jednak unikania takich sytuacji. Partycja rozszerzona może jednak występować jako pierwsza.

Narzędzia do partycjonowania

1. FDISK – DOS i Windows 9x (skrót od Fixed Disk)

Umożliwia tworzenie partycji podstawowych (tylko jednej na dysku), rozszerzonych i logicznych. Umożliwia też wyświetlanie informacji o partycjach i usuwanie partycji. (Wymagana ostrożność!!). Tryb tekstowy.

Ograniczenia:

- tylko jedna partycja podstawowa
- konieczność istnienia partycji podstawowej przed utworzeniem rozszerzonej
- sztywny schemat – nowa za ostatnią istniejącą
- ograniczona obsługa partycji z innymi systemami plików

2. Administrator Dysków Windows NT

- graficzny interfejs
- opóźniane wprowadzanie zmian
- możliwość tworzenia partycji logicznych bez istnienia podstawowych
- możliwość usuwania partycji innych systemów

3. FDISK (fdisk) – Linux

- tryb tekstowy
- ograniczenie do jednego dysku
- możliwość odtworzenia usuniętej partycji (w tym samym miejscu i rozmiarze)
- bardzo rozbudowany w możliwości
- istnieją też inne wersje : cfdisk, DiskDruid (RedHat, Mandrake)

4. PartitionMagic – niezależny program nie związany z żadnym systemem operacyjnym. Występuje w wielu wersjach.

Cechy:

- graficzny interfejs
- obsługuje partycje systemów: FAT-12, 16, 32, HPFS (OS/2), NTFS (4o. 5o), ext2fs (Linux) i swap Linuxa
- kolejkovanie operacji
- wiele partycji podstawowych
- tworzenie systemów plików (format, mkfs)
- zmiana rozmiaru i przemieszczanie partycji
- kopiowanie partycji między dyskami
- konwersja formatu (FAT-16 $\leftrightarrow$ FAT-32, FAT-16 $\rightarrow$ HPFS)
- w wersji 5.0 lub późniejszej konwersja partycji podstawowej na logiczną i odwrotnie).

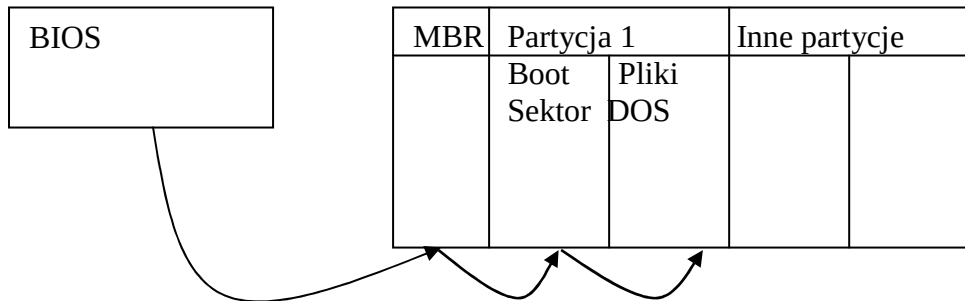
C) Programy ładowania wstępnego

Bios po odszukaniu urządzenia, z którego można załadować system wczytuje do pamięci pierwszy sektor (512 bajtów), a następnie przekazuje sterowanie wczytanemu kodowi. 512 bajtów to za mało, by pomieścić kompletny program ładujący, w dodatku wraz z informacjami o partycjach dysku, więc na ogół ten kod przekazuje sterowanie do innego kodu znajdującego się w innym (określonym) obszarze dysku.

W środowisku systemu DOS bootowanie z dysku twardego przebiega następująco:

1. BIOS wczytuje zawartość MBR i przekazuje sterowanie do wczytanego kodu
2. Wykonanie kodu z MBR powoduje załadowanie zawartości boot sektora partycji, w której rezyduje systemu DOS i przystępuje się do wykonania tego kodu
3. Wykonanie kodu załadowanego w p. 2 powoduje wczytanie i wykonanie plików systemowych DOS-u.

4. Gdy DOS przejmuje kontrolę, następuje analiza config.sys i ładowanie sterowników
5. Jeśli jest Autoexec.bat to wykonywane są zawarte w nim polecenia i c:\>.



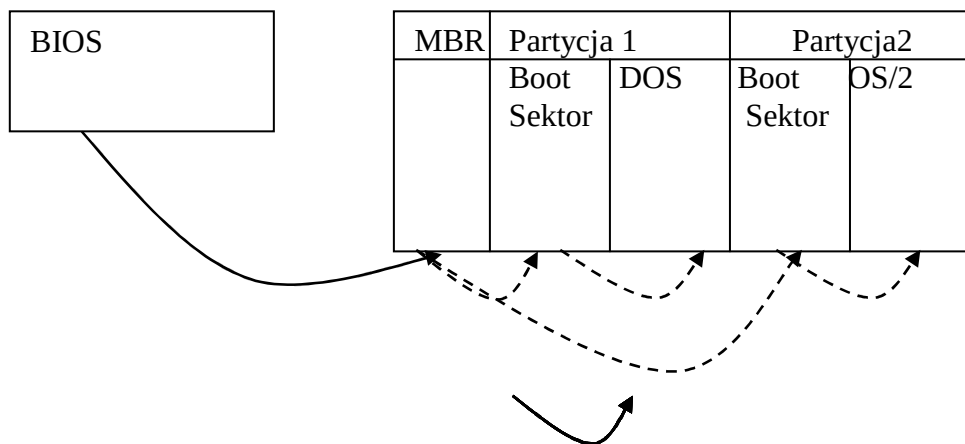
Rys. 2

Jeśli w komputerze zainstalowana więcej systemów operacyjnych to mamy różne możliwości.

- Specjalizowany MBR – sektor MBR zawiera kod wykonywany jako pierwszy w procesie bootowania. Możemy go zmodyfikować tak by dawał możliwość wyboru partycji.

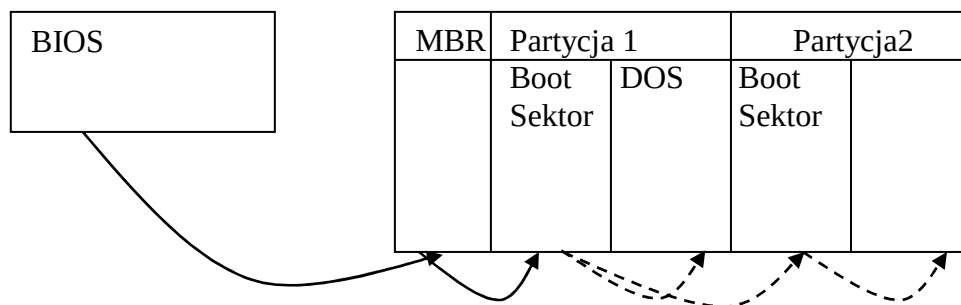
Dwie sekwencje bootowania

Rys. 2



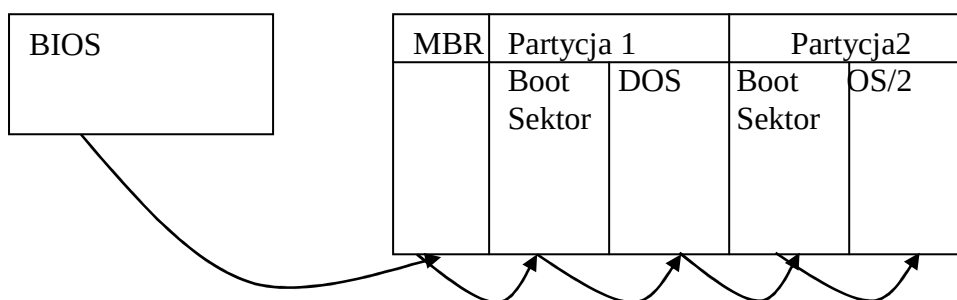
- Specjalizowany bootsektor – tu modyfikujemy kod bootsektora w jednej z partycji (np. pierwszej).

Dwie sekwencje bootowania



Rys. 3

- Łącuchowanie bootowania – nie są modyfikowane ani MBR ani bootsektory. Każdy bootsektor jest odpowiedzialny za ładowanie „swojego” systemu operacyjnego. Ten ostatni po załadowaniu swoich plików daje wybór, czy kontynuować bootowanie czy przekazać sterowanie do innej partycji.



1. WNT OS Loader
 - oparty na pliku BOOT.INI (w c:\ - obejrzeć)
 - może ładować inne systemy (w tym Linux)
 2. LILO (Linux)
 - bootsektor dyskiety, MBR lub Bootsektor partycji
 - konfiguracja przez /etc/lilo.conf (po zmianie zawartości wydać polecenie lilo)
 3. BootMagic
 - musi być na partycji FAT – nie naruszając bootsektora „podmienia” pliki systemowe DOS lub Windows na własne.
-