

UNIwersytet w Białymstoku
Wydział Matematyki i Informatyki
Instytut Informatyki

Marcin Kulesza

Instalacja i rozruch różnych
systemów operacyjnych na
jednej maszynie

*Praca dyplomowa napisana
pod kierunkiem
dr. Mariusza Żynela*

Białystok 2018

Spis treści

1	Zagadnienia ogólne	1
1.1	Wstęp	1
1.2	BIOS - opis i charakterystyka	1
1.2.1	Czym jest BIOS	1
1.2.2	Wersje systemu Bios	2
1.2.3	Specyfikacja rozruchu w Biosie	2
1.2.4	Charakterystyka pracy	3
1.2.5	Pojęcia związane z Biosem	3
1.2.6	Indentyfikacja urządzeń IPL	4
1.2.7	Sekwencja procedury POST	5
1.2.8	Konfiguracja BIOS'u	6
1.2.9	Bezpieczeństwo BIOS'u	6
1.3	UEFI - opis i charakterystyka	7
1.3.1	Opis	7
1.3.2	Cele modelu sterownika UEFI	8
1.3.3	Historia UEFI	9
1.3.4	Charakterystyka działania	9
1.4	Różnice pomiędzy BIOS'em a UEFI	10
1.5	Sposoby bootowania systemów	10
1.5.1	Bootowanie	10
1.5.2	Elementy rozruchu BIOS	11
1.5.3	Rozruch systemu Windows XP	14
1.5.4	Rozruch systemu Windows 7	15
1.5.5	Rozruch systemu Windows 10	16
1.5.6	Rozruch systemu Linux	18
2	Rozruch systemu	19
2.1	MBR - Master Boot Record	19
2.1.1	Specyfikacja Master Boot Record	19
2.2	GPT - GUID Partition Table	22
2.2.1	Pojęcia związane z GPT	23
2.2.2	Schemat tabeli partycji GUID	23
2.2.3	Główny nagłówek GPT	24
2.3	BCD	24

2.3.1	Architektura BCD	24
2.3.2	Opis komponentów architektury BCD	26
2.4	GRUB	28
2.4.1	Historia GRUB'a	28
2.4.2	Proces uruchamiania z użyciem GRUB'a	28
3	Opis narzędzi użytych w pracy	30
3.1	Oracle VM VirtualBox	30
3.2	Bcdedit	31
3.3	Bootsect	32
3.4	Bootrec	33
3.5	Diskpart	36
	Bibliografia	38

Rozdział 1

Zagadnienia ogólne

1.1 Wstęp

Rozruch systemu, w oparciu o postrzeganie przez człowieka jest dosyć szybkim procesem. Jednak sam komputer podczas tych kilku sekund wykonuje mnóstwo operacji, których zwykły użytkownik zazwyczaj nie zauważa. Za rozruch systemu odpowiedzialny jest system wejścia - wyjścia a mianowicie BIOS lub w nowszych komputerach UEFI. Termin BIOS został utworzony w 1975 roku i po raz pierwszy pojawił się w systemie CP/M, opisywał on określoną część tego systemu załadowaną podczas rozruchu, która łączy się bezpośrednio ze sprzętem komputerowym.

1.2 BIOS - opis i charakterystyka

1.2.1 Czym jest BIOS

BIOS (*The Basic Input- Output System*) jest to podstawowy układ wejścia-wyjścia niezbędny do prawidłowego funkcjonowania płyty głównej [2]. Działa jako pośrednik między sprzętem komputerowym, a jego systemem operacyjnym. Bez Bios'u system operacyjny komputera musiałby komunikować się ze sprzętem lub przejąć nad nim kontrolę. Bios jest podstawowym elementem komputera, jeśli zostanie ustawiony niepoprawnie, BIOS może spowolnić komputer i jego pracę.

1.2.2 Wersje systemu Bios

Wersja	Zmiany
0.80	Wersja oryginalna
0.81	Gramatyczne korekty
1.00	Gotowe do publicznego wydania
1.01	Dodanie parametru BiosSelector do funkcji wykonawczych. Uporządkowano dokumentację funkcji uruchomieniowych.

Tabela 1.1: Wersje systemu BIOS.

1.2.3 Specyfikacja rozruchu w Biosie

Specyfikacja rozruchu definiuje funkcję w BIOS'ie, która tworzy i utrzymuje listę wszystkich początkowych urządzeń ładujących program, znalezionych w systemie i przechowuje tę listę w pamięci nieulotnej. Początkowe urządzenie ładujące program dostępne jest w trzech wersjach: BAID, PnP Card i Legacy Card. Legacy Card czyli karty ze starszymi wersjami nie są wspierane, gdyż mają tendencję do przejęcia kontroli nad procesem uruchamiania tworząc ten proces nieprzyjaznym dla użytkownika oraz nie zapewniają możliwości zidentyfikowania się jako urządzenie IPL. Specyfikacja rozruchu BIOS udostępnia jedną kluczową funkcję, a jest to **priorytet IPL**. Ten priorytet jest określony przez użytkownika oraz jest ustawiony w instalatorze. Priorytet IPL powoduje że za każdym razem, gdy użytkownik włącza system, wszystkie urządzenia IPL w systemie są wyliczane i numerowane. Specyfikacja BIOS definiuje również priorytet BCV (Boot Connection Vector, wektor połączenia rozruchowego) priorytet ten to określona przez użytkownika lista priorytetów kontrolerów urządzeń INT 13h, które są ustawione w rozruchu systemu. Ta lista określa kolejność, w której kontrolery będą wywoływane w celu ich instalacji.

Obsługa napędu INT 13h podczas procedury POST

Jeśli urządzenie IPL nie załaduje systemu operacyjnego, BIOS w takiej sytuacji odzyskuje kontrolę i próbuje się uruchomić z następnego urządzenia z listy IPL. Procedura ta będzie kontynuowana, do momentu gdy wszystkie możliwe urządzenia z listy IPL nie będą reagować na start systemu. Tylko w takiej sytuacji BIOS wyświetli na ekranie komunikat o tym iż system operacyjny, nie może zostać znaleziony i będzie oczekiwać na wciśnięcie klawisza. Po wciśnięciu klawisza ponownie zostanie wywołane metoda INT 19h. Metoda ta zapewnia nas o tym że BIOS inteligentnie wykonał każdą próbę rozruchu systemu.

Specyfikacja rozruchu BIOS

Specyfikacja rozruchu BIOS obejmuje zarówno proces uruchamiania systemu PnP jak i non-PnP. Standard zgodny z systemem AT jest prostszy niż ten z BIOS'em PnP, ponieważ ten standard obsługuje tylko BAID. Legacy System nie musi zapewniać żadnego dynamicznego urządzenia IPL, nie obsługuje on również kart PnP w ich macierzystym trybie.

1.2.4 Charakterystyka pracy

Każdy komputer codziennego użytku zawiera mikroprocesor, który jest jednostką centralną całego układu. Aby mikroprocesor wykonał swoją pracę musi on wykonać zestaw instrukcji, znanych jako oprogramowanie. Do oprogramowania należą: system operacyjny, aplikacje zas trzecim typem oprogramowania jest właśnie BIOS. Oprogramowanie BIOS'u ma wiele różnych ról, ale jego najważniejszą funkcją jest ładowanie systemu operacyjnego.

Innymi ważnymi funkcjami, które spełnia BIOS są:

1. Test POST następuje po włączeniu zasilania, przeprowadzany dla wszystkich komponentów sprzętowych w systemie, celem tego testu jest sprawdzenie czy wszystko działa poprawnie.
2. Aktywacja innych układów BIOS na różnych kartach, które są zainstalowane na komputerze.
3. Udostępnienie zestawu procedur niskopoziomowych używanych przez system operacyjny do łączenia z różnymi urządzeniami sprzętowymi.
4. Zarządzanie zbiorem ustawień dla dysków twardych i zegara.

1.2.5 Pojęcia związane z Biosem

BAID (BIOS Aware IPL Device) to urządzenie, które może uruchomić system operacyjny, urządzenie to wymaga systemu BIOS, do tego by posiadać specyficzny kod do wsparcia BIOS'u. Przykładowo są to:

- pierwsza stacja dyskietek,
- pierwszy dysk twardy,
- pierwsza, wbudowana karta sieciowa.

BEV (Bootstrap Entry Vector) jest to wskaźnik wskazujący na kod wewnątrz opcjonalnej pamięci ROM, która będzie bezpośrednio ładować system operacyjny. BEV znajduje się w rozszerzeniu nagłówkowym PnP ROM. Przykładem opcji ROM z BEV jest kontroler ethernetowy PnP ISA.

BCV (Boot Connection Vector) jest to wektor połączenia rozruchowego, który wskazuje na kod wewnątrz opcji ROM, która wykona inicjalizację urządzenia oraz wykryje czy urządzenie peryferyjne jest dołączone. BCV znajduje się w opcji ROM rozszerzenia nagłówkowego PnP. Przykładem opcji ROM z BCV jest kontroler PnP ISA SCSI.

Legacy Card - (karta ze starszymi wersjami) to standardowa karta ISA, która nie zawiera konfiguracji zgodnych z PnP i nie zawiera nagłówka rozszerzenia PnP.

Karty PNP składają się z kart zawierających opcję ROM z rozszerzeniem nagłówkowym PnP.

Urządzenie IPL Początkowe urządzenie ładujące program to dowolne urządzenie, które może się uruchomić i zainicjować ładowanie Systemu operacyjnego. W zwykłych komputerach jest to stacja dyskietek (komputery starsze) lub dysk twardy.

NV jest to *pamięć nieulotna*, która pozostaje zachowana nawet po wyłączeniu zasilania. Najbardziej znaną pamięcią nieulotną jest pamięć CMOS RAM, która służy do przechowywania informacji o konfiguracji systemu.

POST Jest to część systemu BIOS, która przejmuje kontrolę natychmiast po włączeniu komputera. Procedura POST inicjalizuje sprzęt komputerowy, dzięki czemu system operacyjny może zostać załadowany.

BIOS Setup jest to program konfiguracji systemu. Jest on wykonywany po naciśnięciu przez użytkownika specjalnego klawisza, podczas inicjalizacji systemu BIOS. Setup pozwala użytkownikowi na konfigurację systemu, a także wybranie priorytetu IPL danego systemu.

1.2.6 Indentyfikacja urządzeń IPL

Różnorodność możliwych urządzeń IPL decyduje o sposobie implementacji takim, w którym wszystkie istniejące i przyszłe urządzenia można było uruchomić szybko oraz by nie było to zbyt skomplikowane.

Wszystkie urządzenia IPL w systemie muszą być wyliczone podczas procedury POST, ale przed INT 19h.

Urządzenia IPL	Typ
Pierwszy napęd dyskietek	BAID
Pierwszy dysk twardy ATA	BAID
Karta PCI ATA z napędem	BAID
Kontroler PCMCIA z kartą startową	BAID
Kontroler Ethernet z kodem osadzonym w BIOS'ie	BAID
Karta Token Ring PnP	PnP (BEV)
Karta sieciowa PnP	PnP (BEV)
Karta Non-PnP z rozszerzeniem nagłówkowym PnP	PnP (BEV lub BCV)
Karta z napędem PNS SCSI	PnP (BCV)

Tabela 1.2: Urządzenia IPL.

1.2.7 Sekwencja procedury POST

Działanie procedury POST w głównej mierze zależy od BIOS'u. POST zaczyna się od przejścia do adresu 0xe05b. Sam program rozpoczyna testy kilku rejestrów procesora. Dla AMI BIOS, kolejność testów wygląda następująco:

1. Testowanie rejestrów procesora.
2. Analiza sumy kontrolnej pamięci ROM BIOS'u.
3. Analiza kontrolera klawiatury.
4. Kontrola rejestru zamykania CMOS.
5. Test zegara systemowego.
6. Sprawdzenie odświeżania pamięci.
7. Testowanie początkowych 64kB pamięci.
8. Sprawdzenie pamięci cache.
9. Testowanie baterii CMOS.
10. Sprawdzenie karty graficznej dla różnych trybów.
11. Wejście i wyjście z obszaru chronionego - test.
12. Analizowanie szyny adresowej.
13. Testowanie zapisu i odczytu pamięci.

14. Testowanie DMA.
15. Sprawdzenie poprawności działania klawiatury.
16. Test konfiguracji SETUP'u.

Zaprezentowane testy i analizy mogą również zakończyć się niepowodzeniem. Większość błędów w testach możemy zobaczyć za pomocą komunikatu na ekranie, w przypadku błędnej konfiguracji karty graficznej, informację o błędach wydaje głośniczek systemowy.

1.2.8 Konfiguracja BIOS'u

W większości systemów zbudowanych od lat 80- tych program konfiguracyjny BIOS jest przechowywany w samym układzie BIOS. Program konfiguracyjny zazwyczaj uruchamiany jest za pomocą naciśnięcia jednego lub więcej klawiszy w ciągu kilku sekund od włączenia komputera, są to np. klawisze (**Del**, **Esc**, **F10**) dla komputerów wyprodukowanych po 1995, zaś dla komputerów starszych sekwencja klawiszy może być następująca **Ctrl+Alt+Esc**; **Ctrl+Alt+Ins** ; **Ctrl+Alt+Enter**.

Ustawienia wprowadzone w programie konfiguracji BIOS są przechowywane w nieulotnej pamięci RAM układu CMOS, stąd nazwa dla tych ustawień: *ustawienia CMOS*.

CMOS (Complementary Metal Oxide Semiconductor) program, który przechowuje dane o konfiguracji podzespołów komputera, posiada również kalendarz oraz zegar rzeczywistego czasu. Dane o zainstalowanych fragmentach systemu są odczytywane w momencie rozruchu systemu w pamięci ROM, podczas procedury POST i są porównywane z obecnym stanem układów. Nazwa CMOS wywodzi się od zastosowanej technologii, która wykorzystuje mało energii lecz jej dostęp do poszczególnych rzeczy jest dosyć wolny.

1.2.9 Bezpieczeństwo BIOS'u

BIOS jest podatny na ataki i skażenie wirusami. Szczególnie ważne jest bezpieczeństwo tego układu, gdyż jest on odpowiedzialny za całkowity rozruch systemu i jego prawidłową konfigurację. Producenci sprzętu zwykle wydają aktualizację BIOS'u w celu ulepszenia swoich produktów, poprawienia kompatybilności oraz usunięcia występujących błędów. Jednak aktualizacje wiążą się z niewłaściwym wykonaniem lub przerwaniem, co może uczynić komputer bezużytecznym. Sposobem na to jest użycie bloku rozruchowego, czyli części systemu BIOS, która działa jako pierwsza i musi być aktualizowana osobno. Kod ten sprawdza czy reszta systemu BIOS jest nietknięta przed przeniesieniem do niej kontroli. Jeśli blok rozruchowy wykryje jakieś uszkodzenie w głównym BIOS'ie, ostrzega on użytkownika o tym iż proces inicjalizacji musi

być dokonany z nośnika wymiennego. Niektóre płyty główne są dodatkowo zabezpieczone, gdyż posiadają zapasowy system BIOS (tak zwany DualBIOS) mogący odzyskać uszkodzoną część przez system BIOS.

Istnieją obecnie co najmniej cztery znane wirusy atakujące system BIOS, z których dwa służą do celów demonstracyjnych.

1. **CIH virus (Chernobyl)** jego twórcą był Chen Ing Hau, i był w stanie wymazać zawartość pamięci flash ROM BIOS. Zainfekowane w ten sposób komputery nie mogły się już normalnie uruchomić, zaś użytkownicy musieli usunąć pamięć flash ROM IC z płyty głównej i ją przeprogramować. Obecne systemy nie są już podatne na CIH gdyż posiadają zróżnicowane chipsety, które nie są kompatybilne z chipsetem Intel i430TX. Dodatkową ochroną przed wirusem CIH są bloki rozruchowe, które są chronione przed przypadkowym nadpisaniem, zaś systemy wyposażone w podwójne lub poczwórne systemy BIOS mogą obecnie skorzystać z zapasowego systemu BIOS. Wirus ten stał się aktywny w 1999 roku i atakował głównie chińskich użytkowników.
2. **Wirus zaprezentowany przez Johna Heasmana** W 2006 roku pokazał on jak podnosić uprawnienia i odczytywać pamięć fizyczną, wykorzystując złośliwe procedury, które zastąpiły normalne funkcje ACPI przechowywane w pamięci flash.
3. **Infectivity BIOS** Pojawił się w 2009 roku na konferencji bezpieczeństwa w Vancouver oraz na konferencji w Singapurze. Badacze Anibal Sacco i Alfredo Ortega z firmy Core Security Technologies zaprezentowali, jak wstawić złośliwy kod do procedur dekompresyjnych w BIOS'ie, umożliwiając niemal pełną kontrolę nad komputerem podczas uruchamiania, nawet przed startem systemu operacyjnego.
4. **Membromi** czwarty wirus, jest to trojan atakujący komputery z zainstalowanym systemem AwardBIOS, Microsoft Windows oraz oprogramowaniem Rising Antivirus i Jiangmin KV Antivirus. Wirus ten instaluje rootkita, który w późniejszej fazie infekuje główny rekord rozruchowy.

1.3 UEFI - opis i charakterystyka

1.3.1 Opis

UEFI (Unified Extensible Firmware Interface) jest to specyfikacja dla oprogramowania która łączy oprogramowanie układowe komputera z systemem operacyjnym [3].

Obecne **UEFI** jest rozszerzeniem prekursora EFI (Extensible Firmware Interface) wyprodukowanego przez firmę Intel. UEFI powstało między innymi wskutek znacznych ograniczeń w funkcjonowaniu BIOS'u, planowano wydać

oprogramowanie, które pozwalałoby na większą elastyczność w pracy. Finalnie UEFI ma zastąpić BIOS w jego pracy.

Projekt UEFI opiera się na następujących podstawowych elementach:

- Ponownym użyciu istniejących interfejsów opartych na tabelach. UEFI posiada szereg wymogów, które wybrane platformy muszą spełnić by na tym oprogramowaniu pracować.
- Partycji systemowej. Definiuje ona partycję i system plików dla których została zaprojektowana. UEFI udostępnia możliwość zwiększenia wartości dodanej platformy, bez znacznego wzrostu zapotrzebowania na nieulotną pamięć.
- Usługach rozruchowych.
- Usługach uruchomieniowych.

1.3.2 Cele modelu sterownika UEFI

- **Zgodność** Sterowniki muszą zachowywać zgodność ze specyfikacją EFI 1.10 oraz specyfikacją UEFI.
- **Prostota** Sterowniki, które są zgodne ze specyfikacją, muszą być proste w implementacji oraz w obsłudze. Model sterownika UEFI musi umożliwiać użytkownikowi, skoncentrowanie się na konkretnym urządzeniu, dla którego opracowywany jest sterownik.
- **Skalowalność** Model sterownika UEFI musi być w stanie dostosować się do wszystkich typów platform. Platformy te obejmują obszar systemów wbudowanych, systemów stacjonarnych oraz mobilnych, stacji roboczych i serwerów.
- **Elastyczność** Model UEFI musi obsługiwać możliwość wyliczenia wszystkich urządzeń lub możliwość wyliczenia urządzeń niezbędnych do uruchomienia systemu operacyjnego.
- **Rozszerzalność** Model sterownika UEFI musi być w stanie usprawniać i rozszerzać technologię, w oparciu o przyszłe typy magistrali.
- **Przenośność** Sterowniki napisane do modelu UEFI muszą być przenośne między poszczególnymi platformami oraz pomiędzy obsługiwanymi architekturami procesorów.
- **Współdziałanie** Sterowniki muszą współpracować z innymi sterownikami i oprogramowaniami układowymi systemu. Muszą współdziałać bez generowania konfliktów zasobów.

- **Opis hierarchii szyn magistrali** Model sterownika UEFI musi opisywać wiele różnych topologii magistrali, począwszy od bardzo prostych-pojedynczych platform do bardzo złożonych zawierających wiele szyn magistrali różnych typów.
- **Rozwiązywanie problemów ze starszymi wersjami ROM** W modelu UEFI musi istnieć możliwość zbudowania dodatku karty, który będzie obsługiwał zarówno sterowniki UEFI jak i starsze opcje ROM. Rozwiązanie powinno zapewnić ewolucyjną ścieżkę do migracji ze starszej wersji sterownika ROM do sterowników UEFI.

1.3.3 Historia UEFI

Motywacją do pracy nad EFI było opracowywanie pierwszych systemów Intel-HP Itanium w latach 90'tych. Ograniczenia BIOS'u były zbyt restrykcyjne w przypadku większych platform serwerowych. W roku 1998 prekursorem było oprogramowanie, które nazywało się Intel Boot Initiative, dopiero później zmienione na Extensible Firmware Interface (EFI). W roku 2005 Intel zaprzestał opracowywania specyfikacji EFI w wersji 1.10 i wniósł ją na forum Unified EFI Forum, które opracowało specyfikację znaną pod nazwą UEFI (Unified Extensible Firmware Interface). Oryginalna specyfikacja EFI pozostaje własnością Intela. W roku 2007 wydano wersję 2.1 specyfikacji UEFI. Dodano w niej kryptografię, uwierzytelnianie sieci, architekturę interfejsu użytkownika. Ostatnia aktualna wersja specyfikacji UEFI 2.7 została zatwierdzona i wydana w maju 2017 roku.

1.3.4 Charakterystyka działania

Specyfikacja UEFI obejmuje dwa aspekty procesu, które znane są jako usługi rozruchowe **boot service** i usługi uruchomieniowe **runtime services**.

- Usługi rozruchowe określają sposób w jaki sprzęt zainicjuje system operacyjny do załadowania.
- Usługi uruchomieniowe pomijają procesor rozruchu i ładują aplikację bezpośrednio z UEFI.

1.4 Różnice pomiędzy BIOS'em a UEFI

BIOS i UEFI to dwa interfejsy oprogramowania układowego dla komputerów, które działają jako interpreter pomiędzy systemem operacyjnym a oprogramowaniem komputerowym. Zarówno BIOS jak i UEFI są używane przy uruchamianiu komputera w celu inicjalizacji podzespołów sprzętowych oraz uruchomienia systemu operacyjnego zapisanego na dysku twardym. Poniżej przedstawione są główne różnice dotyczące konkretnych zagadnień pomiędzy BIOS'em a UEFI.

	BIOS	UEFI
<i>Tryb pracy</i>	16-bit	32/64-bit
<i>Interfejs</i>	tekstowy	graficzny
<i>Pamięć operacyjna</i>	1MB	Maksymalna dostępna
<i>Maksymalny rozmiar partycji</i>	2 TB (terabajty)	9 ZB (Zettabajt)
<i>Dozwolona ilość partycji podstawowych</i>	4	128
<i>Wykorzystywane rekordy rozruchowe</i>	MBR	GPT
<i>Bezpieczeństwo uruchamiania</i>	Wysokie (Wykorzystuje cyfrowy podpis programów ładujących)	Niski poziom bezpieczeństwa (Podatność na instalacje złośliwego oprogramowania)
<i>Ilość przechowywanych bootloaderów</i>	Wyłącznie jeden	Wiele (za pomocą ESP - EFI System Partition)
<i>Dostęp do sieci</i>	Nie	Tak

Tabela 1.3: Różnice pomiędzy BIOSem a UEFI.

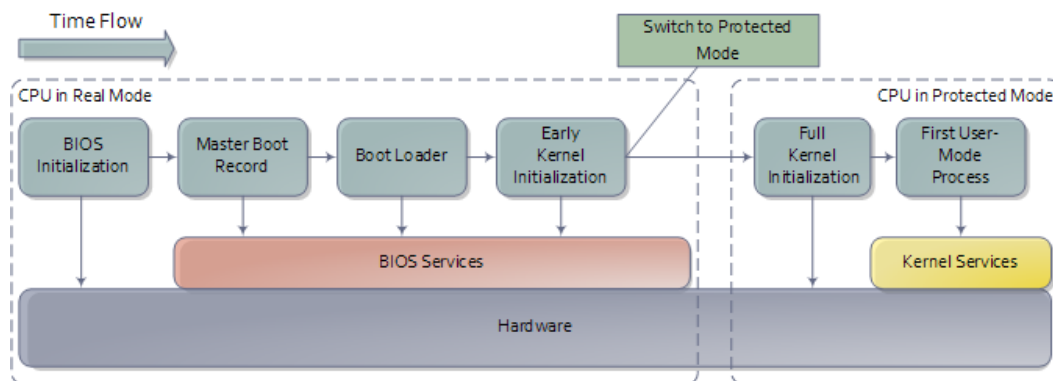
1.5 Sposoby bootowania systemów

1.5.1 Bootowanie

Bootowanie oznacza rozruch komputera. Komputer w tym procesie ma załadować system operacyjny do głównej pamięci komputera lub pamięci RAM. Po załadowaniu nasz komputer jest gotowy do korzystania z aplikacji.

Proces startu komputera może odbywać się na dwa sposoby.

1. *Cold boot* oznacza zimny rozruch jest on zwykle w momencie naciśnięcia przycisku zasilania komputera. W większości przypadków uruchamiany



Rysunek 1.1: Proces rozruchu [45].

jest zimny rozruch, aby komputer mógł wykonać standardowe zadania obliczeniowe. Jest to proces uruchamiania od włączenia i ustawienie go w normalnym stanie roboczym.

2. *Hot boot* dotyczy ponownego uruchamiania komputera czyli restartu. Ciepły rozruch jest konieczny w momencie gdy program napotka błąd, z którego nie można odzyskać kontroli nad systemem.

1.5.2 Elementy rozruchu BIOS

Składnik	Wykonywanie procesora	Obowiązki	Lokalizacja
Master Boot Record (MBR)	16-bitowy tryb rzeczywisty	Czyta i ładuje rekord rozruchowy woluminu	Na urządzeniu pamięci masowej
Sektor startowy (VBR)	16-bitowy tryb rzeczywisty	Rozumie system plików na partycji i lokalizuje bootmgr po nazwie ładując go do pamięci	Na każdej aktywnej partycji
bootmgr	16-bitowy tryb rzeczywisty i 32-bitowy bez stronicowania	Czyta bazę danych konfiguracji rozruchu (BCD) prezentuje menu startowe i pozwala na wykonywanie programów przed startem.	Na systemie operacyjny

Składnik	Wykonywanie procesora	Obowiązki	Lokalizacja
<code>winload.exe</code>	32-bitowy tryb chroniony ze stronicowaniem i 64-bitowy tryb chroniony podczas uruchamiania instalacji Win64	Ładuje plik <code>ntoskrnl.exe</code> i jego zależności oraz bootstart sterownika urządzeń	Przy instalacji Windows
<code>winresume.exe</code>	32 bitowy tryb chroniony i 64-bitowy tryb chroniony po wznowieniu instalacji Win64	Uruchamia się jeśli wznowienie po stanie hibernacji zostanie rozpoczęte z pliku hibernacji <code>hiberfill.sys</code> zamiast typowego ładowania systemu windows.	Przy instalacji Windows
<code>memttest.exe</code>	32-bitowy tryb chroniony	Przy wybraniu z Boot Manager uruchamia się i zapewnia graficzny interfejs dla skanowania pamięci i wykrywania uszkodzeń RAM.	Na systemie operacyjnym
<code>ntoskrnl.exe</code>	Tryb chroniony ze stronicowaniem	Inicjuje podsystemy wykonawcze, rozruch i sterowniki systemu startowego, przygotowuje system do uruchamiania aplikacji natywnych	Przy instalacji Windows
<code>hal.dll</code>	Tryb chroniony ze stronicowaniem	Biblioteka DLL trybu jądra, łącząca się z <code>ntoskrnl</code> i sterownikami do sprzętu. Działa ona również jako sterownik dla samej płyty głównej.	Przy instalacji Windows

Składnik	Wykonywanie procesora	Obowiązki	Lokalizacja
smss.exe	Aplikacja natywna	Inicjał początkowy uruchamiania kopię, by zainicjować każdą sesję. W tym momencie załadowuje się się instancja sesji 0, następnie sterownik podsystemu Windows uruchamia proces podsystemu Windows i proces inicjalizacji systemu Windows	Przy instalacji Windows
wininit.exe	Aplikacja Windows	Uruchamia menedżera sterowania usługą, proces lokalnego urzędu do spraw bezpieczeństwa oraz lokalny menedżer sesji. Dodatkowo inicjuje resztę rejestru i wykonuje zadanie inicjalizujące w trybie użytkownika.	Przy instalacji Windows
winlogon.exe	Aplikacja Windows	Koordinuje logowanie i bezpieczeństwo użytkownika, uruchamia LogonUI	Przy instalacji Windows
logonui.exe	Aplikacja Windows	Przedstawia interaktywne okno dialogowe logowania	Przy instalacji Windows
services.exe	Aplikacja Windows	Ładuje i inicjuje automatyczne uruchamianie sterowników urządzeń i usługi Windows	Przy instalacji Windows

Tabela 1.4: Elementy rozruchu BIOS.

1.5.3 Rozruch systemu Windows XP

Niezbędne informacje o plikach potrzebnych do uruchomienia Windows:

- **ntldr.exe** program znajdujący się na głównej partycji dysku systemowego. Jego zadaniem jest ładowanie systemu operacyjnego opartego na jądrze NT.
- **boot.ini** znajduje się on w głównej partycji dysku systemowego. Plik ten zawiera w sobie opisy zainstalowanych systemów operacyjnych wraz z partycjami.
- **ntdetect.com** plik ten zlokalizowany jest w głównej partycji dysku systemowego. Jego zadaniem jest sprawdzenie działania zainstalowanego sprzętu.
- **ntoskrnl.exe** jest to jądro systemu operacyjnego.

Proces uruchamiania systemu dla Windows XP wygląda następująco:

1. Włączenie zasilania komputera.
2. Procedura POST dokonuje badania niektórych urządzeń sprzętowych komputera, te zaś informacje przekazywane są do BIOSu.
3. BIOS sprawdza resztę urządzeń podłączonych do komputera i przekazuje do pamięci zawartość głównego sektora rozruchowego.
4. MBR ładuje do pamięci systemu program o nazwie NT Loader (**ntldr.exe**), który w tym miejscu przejmuje sterowanie.
5. **ntldr.exe** czytuje zawartość pliku **boot.ini** i w wyniku tego generuje proces rozruchowy, jeśli jest to konieczne. W tym momencie użytkownik może wybrać system operacyjny który ma być uruchomiony na maszynie.
6. **ntldr.exe** ładuje do pamięci zawartość pliku **ntdetect.com**
7. **ntdetect.com** przeprowadza szereg prób i testów sprzętu komputerowego i dane te przekazuje do **ntldr.exe**.
8. **ntldr.exe** ładuje system **ntoskrnl.exe** do pamięci.
9. **ntoskrnl.exe** ładuje do pamięci resztę plików systemowych niezbędnych do uruchomienia systemu, sterowniki poszczególnych urządzeń oraz uruchamia usługi automatyczne.

1.5.4 Rozruch systemu Windows 7

1. Po włączeniu komputera jego procesor zaczyna wykonywać instrukcje zawarte w BIOS lub EFI.
2. Pierwszym zestawem instrukcji rozruchowych jest POST.
3. Po zakończeniu POST płyty głównej adaptery dodatkowe, które mają własne oprogramowanie układowe przeprowadzają wewnętrzne testy diagnostyczne, jeśli uruchamianie zakończy się niepowodzeniem przed lub w trakcie procedury POST oznaczać to będzie awarię sprzętu.
4. W następnej kolejności ładowany jest Windows Boot Manager czyli menedżer rozruchu systemu Windows. Starsze komputery BIOS i nowsze komputery EFI robią to nieco inaczej.

BIOS:

Urządzenia które są zgromadzone w pamięci nieulotnej, takie jak kolejność rozruchu określają urządzenia z których może korzystać komputer w celu uruchomienia systemu operacyjnego. Oprócz dyskietek lub dysków twardej podłączonych do kontrolerów ATA, Serial ATA i kontrolerów SCSI, komputery mogą uruchamiać system operacyjny z innych urządzeń takich jak płyty CD lub DVD, karty sieciowe, napędy flash USB, wymienne dyski. Występuje tu możliwość niestandardowej kolejności rozruchu. Jeśli uruchomienie nie powiedzie się podczas początkowej fazy uruchamiania, występuje problem z konfiguracją systemu BIOS, podsystemem dysku lub systemem plików, pojawia się komunikat o błędzie czyli oznacza to wtedy fakt że żaden ze skonfigurowanych typów nośników startowych nie był dostępny. Jeśli aktywna partycja nie istnieje lub brakuje informacji o sektorze rozruchowym lub jest ona uszkodzona, może pojawić się jeden z komunikatów: nieprawidłowa tabela partycji, błąd ładowania systemu operacyjnego, brak systemu operacyjnego.

EFI:

Komputery EFI mają wbudowany menedżer rozruchowy, który umożliwia sprzętowi komputerowemu wybór spośród wielu systemów operacyjnych opartych na danych wprowadzanych przez użytkownika. Podczas instalowania systemu Windows 7 na komputerze z interfejsem EFI system Windows dodaje pojedynczy wpis do menedżera rozruchu EFI z tytułem menedżer uruchamiania systemu Windows. Ten wpis wskazuje na program wykonywalny EFI - Windows Boot Manager. Jest to ten sam menedżer który ostatecznie jest ładowany na komputerach z BIOS'em. System Windows dokonuje konfiguracji menedżera rozruchu EFI, tak aby wyświetlał menu startowe EFI tylko przez 2 sekundy, a następnie załadował domyślnie menedżera rozruchu systemu Windows, aby zminimalizować złożoność i czas uruchamiania.

5. Następnie następuje faza rozruchu systemu Windows. Menedżer rozruchowy systemu Windows jest zdolny do samodzielnego czytania obsługiwanych systemów plików i wykorzystuje tę możliwość do analizy rejestru BCD bez pełnego wczytywania systemu plików. Jeśli użytkownik nie nacisnie przycisku w ciągu kilku sekund po zakończeniu POST, menedżer rozruchu systemu uruchomi program rozruchowy Windows, który z kolei uruchomi system Windows 7.
6. Kolejno Windows Boot Manager uruchamia Windows Boot Loader. Program ładujący rozruch systemu Windows w tym momencie wykonuje następujące czynności: ładuje jądro systemu operacyjnego, ładuje warstwę abstrakcji sprzętu `hal.dll`, warstwa ta nie będzie używana dopóki jądro nie zostanie uruchomione. Ładuje gałąź rejestru systemu do pamięci, skanuje klucz dla sterowników urządzeń i ładuje do pamięci wszystkie sterowniki skonfigurowane dla klasy rozruchowej. Windows Boot Loader nie inicjuje sterowników, są one uruchomione dopiero w fazie ładowania jądra.
7. Następną fazę rozpoczyna jądro systemu operacyjnego. Jądro systemu oraz HAL inicjują grupę funkcji oprogramowania, które nazywa się wykonawczym Windows. Menedżer Windows przetwarza informacje o konfiguracji i uruchamia usługi i sterowniki.
8. Ostatnim etapem jest faza logowania, która jest obsługiwana przez program `winlogon.exe` czyli usługę systemową, która umożliwia zalogowanie i wylogowanie się. `winlogon.exe` wykonuje między innymi następujące czynności:
 - uruchamia podsystem usługi `services.exe`,
 - uruchamia proces lokalnego urzędu zabezpieczeń `lsass.exe`,
 - inicjuje zabezpieczenia i funkcje uwierzytelniania.
9. Uruchamianie systemu Windows 7 jest zakończone dopiero wtedy gdy użytkownik pomyślnie się zaloguje.

1.5.5 Rozruch systemu Windows 10

Urządzenie z systemem Windows 10 ma kilka wymagań dotyczących uruchamiania systemu operacyjnego. Po tym jak oprogramowanie układowe urządzenia zainicjuje sprzęt, urządzenie musi zapewnić wystarczającą moc do rozruchu. Następnie urządzenie musi upewnić się, że urządzenie uruchamia się w odpowiednim systemie operacyjnym, w zależności od tego, czy użytkownik chce przeprowadzić aktualizację lub przywrócić urządzenie, czy też chce uruchomić urządzenie w głównym systemie operacyjnym.

Proces uruchamiania systemu używa następujących składników: ładowanie rozruchowe oprogramowania sprzętowego dostarczone przez dostawcę SoC, UEFI oraz menedżer rozruchu systemu Windows (Windows Boot Manager).

1. Urządzenie jest zasilane i uruchamia ładowarkę rozruchową, które inicjują dysk w urządzeniu i zapewniają funkcję awaryjnego flashowania.
2. Programy ładujące układ rozruchowy ładują środowisko UEFI i przekazują kontrolę do aplikacji UEFI napisanych przez dostawcę SoC. Aplikacje te mogą wykorzystywać sterowniki i usługi UEFI.
3. Środowisko UEFI uruchamia menedżer rozruchu systemu (Windows Boot Manager), który określa czy uruchomić się z miganiem FFU, czy w trybie resetowania urządzenia, czy do aktualizacji systemu operacyjnego lub do głównego systemu operacyjnego.
4. Po tym jak środowisko UEFI uruchomi Boot Manager'a czyli menedżera rozruchu systemu, inicjuje on biblioteki rozruchowe, odczytuje bazę danych konfiguracji rozruchu, w celu określenia, które aplikacje rozruchowe mają zostać uruchomione i w jakiej kolejności je uruchamiać. Menedżer rozruchu systemu uruchamia kolejno aplikacje rozruchowe, a każda z aplikacji powraca do menedżera rozruchu po zakończeniu.
5. Menedżer rozruchu przechwytuje najpierw wszystkie zastrzeżone kombinacje przycisków sprzętowych które są naciskane przez użytkownika.
6. Kolejno menedżer rozruchu systemu uruchamia aplikację rozruchową awaryjnego zrzutu, która pozwala urządzeniu przechwytywać migawkę pamięci fizycznej z poprzedniej sesji systemu operacyjnego. Jeśli urządzenie nie zostało nieprawidłowo zresetowane w poprzedniej sesji systemu operacyjnego, aplikacja zrzutu awaryjnego znika natychmiast.
7. Menedżer rozruchów uruchamia następnie `mobilestartup.efi`. Aplikacja ta uruchamia kilka bibliotek rozruchowych, z których niektóre działają tylko przy pierwszym uruchomieniu lub tylko w obrazach innych niż detaliczne.
8. `mobilestartup.efi` uruchamia bibliotekę, która wdraża ładowanie baterii UEFI. Biblioteka ta pozwala użytkownikowi ładować swoje urządzenie, gdy urządzenie znajduje się w środowisku rozruchowym. Jest ona uruchamiana jako pierwsza, aby upewnić się że urządzenie ma wystarczającą moc do pełnego rozruchu.
9. Kolejno `mobilestartup.efi` uruchamia biblioteki, które realizują flashowanie, resetowanie urządzeń i aktualizacje. Biblioteki te określają,

czy urządzenie powinno uruchomić się w trybie flashowania lub resetowania urządzenia, czy też urządzenie powinno kontynuować aktualizację systemu operacyjnego lub głównego systemu operacyjnego. Jeśli `mobilestartup.efi` nie uruchomi się w trybie flashowania lub resetowania urządzenia to menedżer urządzeń uruchomi główny system operacyjny lub aktualizację systemu operacyjnego.

1.5.6 Rozruch systemu Linux

W systemie Linux proces bootowania przebiega następująco:

1. W pierwszej kolejności odpalany jest program GRUB, dane i informacje się tam znajdujące pozwalają procesorowi na ustawienie ekranu, uruchomienie właściwego Kernela (jądra systemu).
2. Proces **init** zostaje odpalony, ma od identyfikator PID o wartości jeden. Proces ten odpala kolejny proces a mianowicie getty. Getty jest to proces który ustawia terminal, zaś terminal to interpreter, który pobiera dane wejściowe od użytkownika i wyświetla dane wyjściowe dla polecenia `command`.
3. Proces `init` montuje system plików zgodnie z danymi w pliku `/etc/fstab` po zamontowaniu wyniki zapisuje w pliku `/etc/mntab`.
4. W następnej kolejności czytane są skrypty startowe.
5. Administrator decyduje w kolejnym kroku o tym jakie programy mogą zostać odpalone za pomocą skryptów w katalogu `/etc/xinetd/`. Praca programu odbywa się nie w osobnym procesie a pod kontrolą nadserwera `inetd` (lub `xinetd`).
6. Proces działania rozpoczyna skrypt `/etc/rc.d/rc.local`. W nim możemy umieszczać odwołania do własnych skryptów.
7. System po zakończonym ładowaniu automatycznie włącza program *login* umożliwiający zalogowanie się do systemu operacyjnego. Gdy nastąpi zalogowanie, proces rozpoczyna skrypty konfiguracyjne shella.
8. Gdy system uruchomi się w 3 levelu (tzn. w powłoce tekstowej), zaś użytkownik postanowi pracować w okienkach to finał wykona skrypt `/usr/bin/X11/startx`.

Rozdział 2

Rozruch systemu

2.1 MBR - Master Boot Record

MBR, czyli Master Boot Record, jest to sektor rozruchowy dysku twardego, znajduje się w pierwszym sektorze tego dysku. Rejestruje on informacje o wszystkich partycjach. Dane te obejmują systemy plików, rozmiar i organizację partycji. MBR zawiera w sobie program rozruchowy oraz tablicę partycji. Master Boot Record nie może istnieć na niepartycjonowanych mediach takich jak dyskietki. MBR musi zmieścić się w 512 bajtach. Z ograniczonego miejsca wynika, iż dysk twardy może mieć maksymalnie cztery partycje podstawowe.

Rozruch podczas uruchomienia komputera wygląda następująco. System BIOS szuka urządzenia docelowego do rozruchu, które zawiera główny rekord rozruchowy MBR. Gdy go znajdzie, uruchamia kod rozruchowy znajdujący się w MBR. Następnie ładowany jest pierwszy sektor z partycji oznaczonej jako aktywna, zwykle jest to partycja systemowa. Ten sektor rozruchowy partycji systemowej jest używany do załadowania i uruchamiania systemu operacyjnego.

2.1.1 Specyfikacja Master Boot Record

Współrzędne pierwszego sektora dysku twardego czyli MBR to: Cylinder 0, Głowica 0, Sektor 1. Jeden sektor zajmuje 512 bajtów, zaś struktura klasycznego MBR wygląda jak w tabeli 2.1.

Adres (offset)		Opis	Pojemność (bajty)
Hex	Dec		
+0x000	+0	Obszar kodu bootstrap	446
+0x1BE	+446	Partycja 1	16
+0x1CE	+462	Partycja 2	16
+0x1DE	+478	Partycja 3	16
+0x1EE	+494	Partycja 4	16
+0x1FE	+510	0x55	2
+0x1FF	+511	0xAA	
Całkowity rozmiar			512

Tabela 2.1: Specyfikacja MBR.

Jeśli na dysku znajduje się partycja rozszerzona, to tabela partycji głównej zawiera łącznie do tabeli rozszerzonej partycji, która opisuje pierwszy wolumin logiczny partycji. Tabela ta zawiera informacje o tym pierwszym woluminie logicznym i łącznie do następnej rozszerzonej tabeli partycji, która opisuje drugi wolumin logiczny na tej partycji. Dzięki czemu użytkownik może posiadać nieskończoną (ograniczoną rozmiarem dysku) liczbę woluminów logicznych.

Specyfikacja partycji Master Boot Record

W latach 90 - XX wieku kilku ekspertów branżowych m.in Hale Landis, Ralf Brown, Matthias Paul zaczęło prace nad badaniem i publikacją typów partycji. Służyło to do dokumentacji standardu branży oraz zmniejszeniu ryzyka konfliktów. Konflikty te powstawały dlatego iż Microsoft nie dokumentował wszystkich typów partycji w takim wypadku osoby trzecie zaczęły przypisywać typy partycji we własnym imieniu w sposób nieskoordynowany, próbny i błędny. Prace nad dokumentacją zaowocowały utrzymaniem oprogramowania obsługującego partycje do pracy z uaktualnionymi listami. Specyfikacja partycji w MBR podana jest w tabeli 2.2.

Adres (offset)		Rozmiar (bajty)	Zawartość
Hex	Dec		
+0x00	+0	1	Flaga aktywności
+0x01	+1	1	Głowica startowa
+0x02	+2	2	10 bitów na cylinder startowy i 6 bitów na sektor startowy
+0x04	+4	1	ID partycji
+0x05	+5	1	Głowica kończąca
+0x06	+6	2	10 bitów na cylinder końcowy i 6 bitów na sektor końcowy
+0x08	+8	4	Numer pierwszego sektora partycji w stosunku do całego dysku
+0x0C	+12	4	Rozmiar partycji mierzony w sektorach

Tabela 2.2: Specyfikacja pozycji w tabeli partycji MBR.

Identyfikatory partycji

Identyfikator partycji określa typ partycji lub do jakiego systemu operacyjnego należy. W tabeli 2.3 znajdują się najczęściej używane identyfikatory partycji.

Identyfikator partycji	Opis
0x01	DOS12 (12-bitowy FAT)
0x04	DOS16 (16-bitowy FAT)
0x05	Rozszerzony
0x06	BIGDOS FAT-16
0x07	Windows NT (NTFS), OS/2 (HPFS)
0x0B	FAT-32
0x0C	FAT-32 LBA
0x0E	FAT 16 LBA
0x0F	Rozszerzone LBA
0x63	Unix SysV/386
0x81	Linux/Minix
0x82	Solaris x86 UFS, Linux swap
0x83	Linux ext2fs
0x85	Rozszerzony Linux
0xA5	FreeBSD, NetBSD, BSD/386
0xBE	x86 Boot
0xEB	BeOS

Tabela 2.3: Przykładowe identyfikatory partycji.

Do załadowania systemu, BIOS musi wiedzieć gdzie ten system się znajduje. Korzysta wtedy z Master Boot Code czyli małego początkowego programu startowego w MBR. Program ten ostatecznie przenosi kontrolę do programu rozruchowego partycji, która jest przechowywana w pierwszym sektorze dysku.

2.2 GPT - GUID Partition Table

GPT jest to część standardu EFI, definiuje on układ tablicy partycji na dysku twardym. GPT zostało zaprojektowane jako ulepszenie systemu partycjonowania Master Boot Record, który ma ograniczenie rozmiaru partycji o wartości 2.2 TB. GPT może być również używane w starszych systemach BIOS. Przy użyciu Guid Partition Table dysk może obsłużyć od 8 do 9.4 ZB w zależności od wielkości sektora. Są to na obecne standardy wielkości pamięci bardzo ogromne. Schemat GPT został wprowadzony przez Intela pod koniec lat 90'tych i od tego czasu stał się standardowym układem tablicy partycji na fizycznym dysku twardym. Jest on następcą wielu tablic partycji, takich

jak MBR i APM, przewyżając ograniczenia wykorzystania 32 bitów dla logicznych adresów bloków i standardowego rozmiaru bloku 512 bajtów.

GPT jest również używany w niektórych systemach BIOS z powodu ograniczeń tabel partycji MBR. Wszystkie nowoczesne systemy operacyjne dla komputerów PC obsługują GPT. MacOS obsługuje uruchamianie z partycji GPT tylko w systemach z oprogramowaniem EFI, zaś większość dystrybucji Linuksa może uruchamiać się z partycji GPT na systemach ze starszym interfejsem oprogramowania czyli BIOS jak i na nowszym czyli EFI.

2.2.1 Pojęcia związane z GPT

CRC (cyclic redundancy check) czyli cykliczne sprawdzanie nadmiarowości jest to kod wykrywający błędy, ma zastosowanie w sieciach cyfrowych i urządzeniach pamięci masowej do wykrywania przypadkowych zmian w blokach danych. CRC może zostać użyty do korekcji błędów.

LBA (logical block addressing) czyli adresowanie bloków logicznych jest to powszechny schemat używany do określania położenia bloków danych przechowywanych na komputerowych urządzeniach pamięci masowej, np. dyskach twardych. LBA jest schematem liniowego adresowania w którym bloki są zlokalizowane według liczby całkowitej, w takim wypadku pierwszy blok to LBA 0, drugi LBA 1 i tak dalej.

2.2.2 Schemat tabeli partycji GUID

Schemat ilustruje układ tabeli partycji GPT. Każdy blok logiczny ma rozmiar 512 bajtów, zaś każdy wpis partycji ma 128 bajtów. Adresy LBA, które mają wartości ujemne wskazują na pozycję od końca woluminu, przy czym LBA -1 oznacza ostatni adresowalny blok. Adresy od LBA 0 do LBA 3 wchodzą w skład głównej części GPT. Adresy od LBA -34 do LBA -1 wchodzą w skład pobocznej części GPT.

- LBA 0 : Ochronny MBR.
- LBA 1 : Główny nagłówek GPT.
- LBA 2 : Wejście 1,2,3,4.
- LBA 3 : Wejścia od 5 do 128.
- od LBA 34 do LBA -34 : Partycja 1, Partycja 2 oraz wymagane partycje.
- LBA -33: Wejścia 1,2,3,4.
- LBA -2: Wejścia od 5 do 128.
- LBA -1: Poboczny nagłówek GPT

2.2.3 Główny nagłówek GPT

Główny nagłówek GPT wskazuje użyteczne bloki logiczne na dysku oraz liczbę i rozmiar rekordów partycji. Specyfikacja tego nagłówka znajduje się w tabeli 2.4 na stronie 25.

W nagłówku GPT 128 rekordów o partycji jest zarezerwowanych, przy czym każdy rekord ma 128 bajtów, daje to możliwość utworzenia aż 128 partycji na dysku. Wewnątrz nagłówka znajduje się informacja o jego rozmiarze, jego lokalizacji oraz o rozmiarze i lokalizacji drugorzędnej nagłówka i tablicy partycji, te informacje znajdują się w ostatnich sektorach dysku twardego. Zabezpieczeniem w tym przypadku jest fakt, w którym wszelkie modyfikacje GPT zmieniają sumę kontrolną, a EFI zmieni przepis główny GPT na dodatkowy. W przypadku gdy oba GPT zawierają niepoprawną sumę kontrolną, dostęp do przestrzeni dyskowej stanie się niemożliwy.

2.3 BCD

BCD (Boot Configuration Data) jest to magazyn, w którym systemy od Windowsa Vista i nowsze gromadzą pliki oraz ustawienia programów, które dotyczą procesu rozruchu. Zapewnia on niezależny od oprogramowania układowego mechanizm manipulowania danymi środowiska rozruchowego dla dowolnego typu systemu Windows.

Kluczowe cechy BCD to:

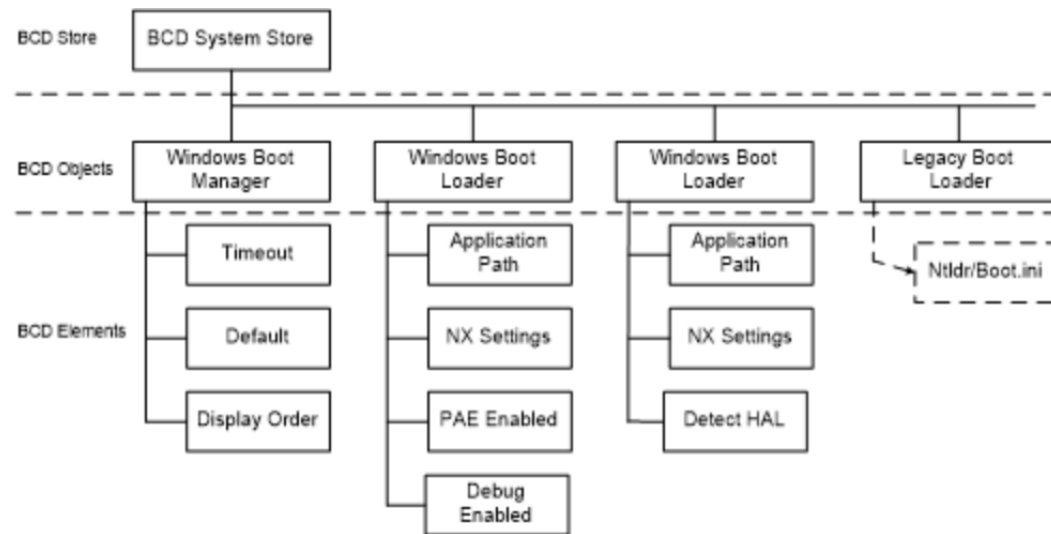
- BCD usuwa bazowe oprogramowanie układowe.
- Interfejsy BCD wykonują wszystkie niezbędne interakcje z oprogramowaniem układowym. W systemach EFI BCD, BCD tworzy i obsługuje wpisy NVI EFI.
- BCD zapewnia przejrzystą i intuicyjną strukturę pamięci dla ustawień rozruchowych.
- BCD jest dostępny w czasie jego pracy oraz podczas fazy rozruchu.
- By manipulować BCD, użytkownik powinien posiadać zwiększone uprawnienia.
- Przeznaczeniem BCD jest obsługa systemów z wieloma wersjami i konfiguracjami systemu Windows, również starsze wersje systemów z rodziny Windows.

2.3.1 Architektura BCD

Architektura BCD jest hierarchią złożoną z trzech podstawowych komponentów. Są to magazyny, obiekty i elementy.

Adres (offset)		Rozmiar (bajty)	Zawartość
Hex	Dec		
+0x00	+0	8	Podpis EFI PART 0x45 0x46 0x49 0x20 0x50 0x41 0x52 0x54
+0x08	+8	4	Wersja (dla GPT wersji 1.0 wartość to 0x00 0x00 0x01 0x00)
+0x0C	+12	4	Rozmiar nagłówka w małym endianie (w baj- tach rozmiar wynosi 0x5C 0x00 0x00 0x00 lub 92 bajty)
+0x10	+16	4	CRC32 (Cykliczne sprawdzanie nadmiarowo- ści nagłówka) w małym endianie, z polem wyzerowanym podczas obliczeń
+0x14	+20	4	Zarezerwowane, muszą być zera
+0x18	+24	8	Bieżący LBA (lokalizacja tej kopii nagłówka)
+0x20	+32	8	Zapasowy LBA (lokalizacja kopii nagłówka zapasowego)
+0x28	+40	8	Pierwszy użyteczny LBA dla partycji (pierw- sza tablica partycji LBA +1).
+0x30	+48	8	Ostatni użyteczny LBA dla partycji (druga tablica partycji LBA -1).
+0x38	+56	16	Dysk GUID.
+0x48	+72	8	Rozruch tablicy wpisów partycji LBA (2 jed- nostki w kopii pierwotnej).
+0x50	+80	4	Liczba wpisów partycji w tablicy.
+0x54	+84	4	Wielkość pojedynczego wejścia partycji (za- zwyczaj 80h lub 128 bajtów).
+0x58	+88	4	CRC macierzy w mały endianie.
+0x5C	+92	-	Jest to obszar zarezerwowany, który musi być wypełniony zerami dla reszty bloku.

Tabela 2.4: Główny nagłówek GPT



Rysunek 2.1: Architektura BCD [46].

- **Magazyn BCD** jest to komponent najwyższego poziomu w hierarchii. Służy jako pojemnik przestrzeni nazw dla obiektów BCD i elementów, które składają się na zawartość zasobów.
- **Obiekt BCD** jest to kontener elementów BCD. Najpopularniejszy typ obiektu BCD opisuje aplikację środowiska rozruchowego, przykładowo instancję programu ładującego system Windows.
- **Element BCD** jest to pojedynczy element danych, taki jak np. ustawienia debuggera, nazwa aplikacji rozruchowej czy urządzenie systemu operacyjnego.

2.3.2 Opis komponentów architektury BCD

1. Magazyn BCD jest kontenerem przestrzeni nazw dla obiektów BCD i elementów przechowujących informacje wymagane do załadowania systemu Windows lub do uruchamiania innych aplikacji rozruchowych. Fizycznie magazyn BCD jest plikiem binarnym w formacie gałęzi rejestru. Komputer ma systemowy magazyn BCD, który opisuje wszystkie zainstalowane systemy operacyjne Windows Vista i zainstalowane aplikacje rozruchowe systemu Windows. Magazyn systemowy to gałąź rejestru, której plik nosi nazwę BCD. W systemach PC/AT BIOS plik znajduje się w folderze boot folder, zaś w systemach EFI(ESP) w katalogu \EFI\MICROSOFT-BOOT. Magazyn systemu jest używany przez menedżera rozruchu systemu Windows do kontrolowania przepływu rozruchowe-

go. Lokalizację magazynu systemowego opisują dwa interfejsy: dostawca BMI WMI i BCDedit.exe.

2. Obiekty BCD dzielą się na trzy kategorie to znaczy:

obiekty aplikacji, obiekty dziedziczne i obiekty urządzeń.

Najpopularniejszym rodzajem obiektu BCD jest obiekt aplikacji, który opisuje aplikację środowiska startowego. Każdy obiekt jest reprezentowany przez 128-bitowy unikatowy identyfikator GUID i zawiera 32-bitowe opracowanie, które opisuje typ obiektu.

- Obiekt aplikacji BCD reprezentuje plik wykonywalny środowiska rozruchowego taki jak program ładujący system Windows. Standardowe obiekty aplikacji to: obiekt menadżera rozruchu systemu Windows, obiekt ładujący system Windows, obiekt Windows Ntldr, obiekt wznawiania systemu Windows, obiekt testera pamięci Windows. Obiekty aplikacji BCD mają dwie charakterystyczne cechy: typ obrazu i typ aplikacji. Typ obrazu określa sposób ładowania pliku wykonywalnego, zaś plik wykonywalny można załadować za pomocą oprogramowania układowego lub menedżera rozruchu systemu Windows.
 - Obiekty dziedziczne BCD to elementy BCD, które można zastosować do więcej niż jednego obiektu aplikacji BCD, niektóre nawet globalnie dla całego magazynu BCD. Obiekt dziedziczny to kontener dla elementów współużytkowanych w wielu instancjach obiektów BCD. Dla przykładu istnieją obiekty dziedziczne, które określają, czy debugger jądra używa połączenia COM, USB lub 1394.
 - Obiekty urządzeń BCD to kontenery dla elementów BCD dla złożonego urządzenia. Obiekt urządzenia BCD jest używany na przykład podczas uruchamiania z dysku RAM utworzonego z pliku obrazu systemu Windows (WIM). Obiekt urządzenia zawiera lokalizację pliku WIM, a po uruchomieniu w sieci informację o porcie sieciowym.
3. Elementy BCD to element danych konfiguracyjnych dla aplikacji środowiska startowego lub części procesu uruchamiania systemu Windows. W BCD każda pozycja danych wejściowych jest zawarta w oddzielnym elemencie BCD. Elementy BCD są zawarte w wyższej strukturze czyli obiekcie BCD. Niektóre elementy można powiązać tylko z określonymi obiektami, podczas gdy inne można zastosować do dowolnego typu aplikacji środowiska rozruchowego.

2.4 GRUB

(GNU GRand Unified Bootloader) jest to pakiet programu rozruchowego z projektu GNU. GRUB zapewnia użytkownikowi możliwość uruchomienia jednego z wielu systemów operacyjnych zainstalowanych na komputerze lub wybrania konkretnej konfiguracji jądra dostępnej na partycjach konkretnego systemu operacyjnego. GNU GRUB jest stosowany głównie w systemach typu Unix. GRUB został zaprojektowany w celu rozwiązania złożoności uruchamiania komputera osobistego. W momencie bootowania za pomocą GRUB'a można skorzystać z dwóch opcji:

- Opcja interfejsu wiersza poleceń. Przy użyciu tej opcji należy ręcznie wpisać specyfikację dysku i nazwę jądra systemu.
- Opcja interfejsu menu. Gdy skorzystamy z opcji menu, system można wybrać za pomocą klawiszy strzałek. Menu jest w tym przypadku oparte na pliku konfiguracyjnym.

2.4.1 Historia GRUB'a

GRUB powstał w 1995 roku, kiedy Erich Boleyn próbował uruchomić system GNU Hurd z mikrojądrem Mach 4. Erich i Brian Ford zaprojektowali specyfikację Multiboot, ponieważ byli zdecydowani nie dodawać do dużej liczby wzajemnie niekompatybilnych metod uruchamiania komputera. Erich zaczął pracę nad modyfikacją programu rozruchowego FreeBSD, aby mógł zrozumieć Multiboot'a. Wkrótce zdał sobie sprawę, że o łatwiej napisać program od zera, stąd pomysł utworzenia GRUB'a. W 1999 GRUB został przyjęty jako oficjalny pakiet GNU, w ciągu kilku następnych lat GRUB został mocno rozszerzony by spełnić wiele potrzeb, jednak projekt nadal nie nadążał za wprowadzanymi rozszerzeniami dlatego w 2002 roku Yoshinori Okujii rozpoczął pracę nad PUPA (Preliminary Universal Programming Architecture for GNU GRUB), mając na celu przepisanie jądra GRUB, aby był czystszy i bezpieczniejszy. PUPA została przemianowana na GRUB 2, a oryginalna wersja została przemianowana na GRUB Legacy. Od 2007 roku dystrybucje GNU LINUX zaczęły używać GRUB 2 w ograniczonym zakresie, a do końca 2009 roku wiele głównych dystrybucji instalowało je domyślnie.

2.4.2 Proces uruchamiania z użyciem GRUB'a

GRUB posiada dwie odrębne metody uruchamiania. Jedna jest to załadowanie systemu operacyjnego bezpośrednio, zaś druga to załadowanie do łańcucha innego programu ładującego, który finalnie załaduje system operacyjny.

Systemy operacyjne które nie obsługują Multiboot'a oraz nie mają wsparcia w GRUB są obciążone łańcuchem, co wiąże się z ładowaniem innego programu rozruchowego i przejściem do niego w trybie rzeczywistym.

Polecenie **chainloader** służy do ustawienia tego. Zwykle konieczne jest załadowanie niektórych modułów GRUB i ustawienie urządzenia root. Ładowanie łańcucha jest obsługiwane tylko na platformach BIOS komputera i platformach EFI.

Rozdział 3

Opis narzędzi użytych w pracy

3.1 Oracle VM VirtualBox

Jest to darmowy i otwarty hiperwizor dla komputerów x86 rozwijany obecnie przez Oracle Corporation. VirtualBox może zostać zainstalowany na wielu systemach operacyjnych np.: Linux, macOS, Windows, Solaris, OpenSolaris. Jego główną rolą jest tworzenie gościnnych maszyn wirtualnych z wersjami pochodnymi systemów Windows, Linux, BSD, Solaris, Haiku, OSx86 i innych oraz ograniczoną wirtualizację gości MacOS na urządzeniach Apple. Użytkownicy VirtualBox'a mogą łączyć wiele systemów- gości w jednym systemie operacyjnym hosta. Każdy gość może być uruchamiany, wstrzymywany, zatrzymywany niezależnie. Użytkownik może konfigurować dowolną maszynę wirtualną i uruchamiać ją na w ramach opartej na oprogramowaniu wirtualizacji lub wirtualizacji sprzętowej. Systemy hosta i systemy wirtualne mogą komunikować się ze sobą za pomocą między innymi wspólnego schowka lub zwirtualizowanego obiektu sieciowego. By maszyny wirtualne gości mogły komunikować się ze sobą należy je odpowiednio skonfigurować.

- **Wirtualizacja oparta na oprogramowaniu**

W przypadku braku wirtualizacji wspomaganej sprzętowo VM VirtualBox stosuje standardowe podejście wirtualizacyjne oparte na oprogramowaniu. Tryb ten obsługuje 32-bitowe systemy - gości, które działają w pierścieniach 0 i 3 w architekturze pierścienia firmy Intel. W założeniu system w tej wirtualizacji rekonfiguruje kod gościa systemu operacyjnego, który standardowo działa w pierścieniu 0, aby został on wykonany w pierścieniu 1 na sprzęcie hosta. Z uwagi na to iż kod ten zawiera instrukcje uprzywilejowane, które nie mogą zostać uruchomione w pierścieniu pierwszym, VM VirtualBox wykorzystuje program **CSAM** do skanowania rekurencyjnie kodu pierścienia 0 przed jego pierwszym wykonaniem. Jest to konieczne by zidentyfikować problematyczne instrukcje, oraz je w późniejszej fazie naprawić. Co do pierścienia 3 to kod trybu użytkownika-

gościa działający w tym pierścieniu, działa również na sprzęcie hosta w tym samym pierścieniu.

- **Wirtualizacja urządzenia** System emuluje dyski twarde w jednym z trzech formatów obrazu dysku:
 1. VDI: Format ten jest obrazem dysku wirtualnego i przechowuje dane w plikach zawierających rozszerzenie '.vdi'.
 2. VDMK': format ten jest używany przez produkty VMware. Przechowuje on dane w jednym lub większej liczbie plików z rozszerzeniem '.vmdk'
 3. VHD: jest to format używany przez Windows Virtual PC i jest natywnym formatem dysku wirtualnego systemu operacyjnego Microsoft Windows Server 2008. Dane są przechowywane w jednym pliku z rozszerzeniem '.vhd'. Maszyna wirtualna wykorzystuje zatem dyski uprzednio utworzone w VMware lub Microsoft Virtual PC oraz własny format natywny.

3.2 Bcdedit

Bcdedit jest to narzędzie wiersza poleceń do zarządzania zasobami BCD, jest to podstawowe narzędzie do edycji konfiguracji rozruchowej dla systemu Windows Vista i nowszych. Może on być użyty do następujących celów:

1. Tworzenia nowych zasobów
2. Modyfikacji istniejących w systemie zasobów
3. Dodawania parametrów do menu startowego

Bcdedit w poszczególnych systemach:

- **Windows XP** tutaj oraz w poprzednich wersjach systemu windows nie ma bcdedit.exe jest za to bootcfg.exe który jest częścią konsoli odzyskiwania systemu Windows XP.
- **Windows 7**

W systemie Windows 7 do uruchomienia bcdedit.exe mamy następujące opcje:

- Gdy posiadamy dysk instalacyjny należy wykonać następujące kroki:
 1. Włożenie dysku instalacyjnego do napędu optycznego.
 2. Ponowne uruchomienie komputera i naciśnięcie dowolnego przycisku po wyświetleniu komunikatu na ekranie monitora.

3. Wybór języka, czasu i klawiatury.
 4. Wybór opcji napraw mój komputer.
 5. Wybór odpowiedniego systemu operacyjnego z listy oraz przejście do kolejnego kroku.
 6. Należy w tym kroku wpisać komendę `bcdedit.exe` i nacisnąć klawisz `enter`.
- Jeśli posiadamy opcję napraw mój komputer w zaawansowanych opcjach rozruchu należy postępować według następujących kroków:
1. Należy zrestartować komputer.
 2. Nacisnąć przycisk `F8` do otwarcia zaawansowanej opcji rozruchu.
 3. Wybrać z listy opcję napraw mój komputer i wcisnąć klawisz `enter`.
 4. W opcjach przywracania systemu należy uruchomić wiersz poleceń i wpisać `bcdedit.exe` oraz nacisnąć klawisz `enter`.

- **Windows 10**

W systemie Windows 10 proces uruchomienia `bcdedit.exe` przebiega następująco:

1. Włożenie dysku instalacyjnego systemu Windows 10.
2. Ponowne uruchomienie komputera oraz rozruch z płyty.
3. Wybór opcji napraw mój komputer.
4. Wybór opcji rozwiązywania problemów.
5. Uruchomienie wiersza poleceń oraz wpisanie komendy `bcdedit.exe` oraz naciśnięcie przycisku `enter`.

3.3 Bootsect

Bootsect jest to narzędzie które aktualizuje główny kod rozruchowy dla partycji dysku twardego użytkownika, w celu przełączania między `Bootmgr` i `NT Loader`. Przy pomocy `Bootsect` można przywrócić sektor rozruchowy na komputerze. Narzędzie `bootsect.exe` jest jedynie dostępne dla systemów Windows XP, Vista, Windows 7, Windows 8, Windows 8.1 oraz Windows 10, wcześniejsze systemy z rodziny Windows nie posiadają tego narzędzia.

Opcje uruchomienia Bootsect w poszczególnych systemach:

- **Windows 7**

By uruchomić `bootsect.exe` możemy skorzystać z następujących opcji:

- Gdy posiadamy dysk instalacyjny należy wykonać następujące kroki:
 1. Włożenie dysku oraz ponowne uruchomienie komputera.
 2. Rozruch z dysku systemu Windows 7.
 3. Wybór opcji napraw mój komputer lub kliknięcie przycisku R.
 4. Wybór systemu operacyjnego z listy oraz przejście do kolejnego kroku.
 5. Uruchomienie wiersza poleceń oraz wpisanie komendy `bootsect /nt60 sys`, następnie należy wcisnąć przycisk enter.
- Jeśli nie posiadamy dysku instalacyjnego należy wykonać następujące kroki:
 1. Należy uruchomić ponownie komputer.
 2. Wcisnąć przycisk F8.
 3. W zaawansowanych opcjach rozruchu wybrać napraw mój komputer.
 4. Należy uruchomić wiersz poleceń z opcji przywracania systemu.
 5. W wierszu poleceń należy wpisać komendę `bootsect /nt60 sys` i nacisnąć przycisk enter.

- **Windows 10**

W systemie Windows 10 uruchomienie `bootsect.exe` przebiega według następujących kroków:

1. Włożenie dysku instalacyjnego systemu Windows 10 do napędu.
2. Ponowne uruchomienie komputera oraz rozruch z płyty.
3. Kliknięcie opcji napraw mój komputer.
4. Wybór polecenia rozwiązywania problemów
5. Uruchomienie wiersza poleceń oraz wpisanie komendy `bootsect /nt60 sys` oraz naciśnięcie przycisku enter.

3.4 Bootrec

Jest to narzędzie dostarczane przez Microsoft'a w środowisku Recovery Environment. Przydatne jest do rozwiązywania problemów oraz ewentualnego naprawiania błędów, które uniemożliwiają normalne uruchomienie komputera. Narzędzie `bootrec` jest używane w połączeniu między innymi z głównym rekordem rozruchowym (MBR), sektorem rozruchowym, BCD oraz plikiem `BOOT.INI`.

Narzędzie to obsługuje następujące opcje:

- **/FixMbr** Opcja ta zapisuje MBR zgodny z systemem Windows 7 na partycji systemowej. Wykorzystywana jest w momencie problemów z MBR lub gdy należy usunąć niestandardowy kod z głównego rekordu rozruchowego.
- **/FixBoot** Opcja `/FixBoot` zapisuje nowy sektor rozruchowy na partycji systemowej za pomocą sektora rozruchowego, który jest zgodny z systemem Windows 7. Wykorzystywana jest gdy:
 - Sektor rozruchowy został zastąpiony innym niestandardowym systemem rozruchowym.
 - Sektor rozruchowy uległ uszkodzeniu.
 - Gdy po instalacji systemu nowszego typu Windows 7 lub Windows Vista został zainstalowany system Windows XP lub starszy.
- **/ScanOs** Opcja ta skanuje wszystkie dyski użytkownika. Wyświetla ona również wpisy, które obecnie nie znajdują się w magazynie BCD. Wykorzystywana jest ta opcja gdy, są instalacje systemu Windows 7 których nie ma na liście menedżera rozruchów.
- **/RebuildBcd** Opcja `/RebuildBcd` skanuje wszystkie dyski w przypadku instalacji zgodnych z systemem Windows 7. Pozwala również wybrać instalację, które chcemy dodać do magazynu BCD. Opcja ta jest wykorzystywana gdy należy całkowicie odbudować magazyn BCD.

Opcje uruchomienia Bootrec w wybranych systemach operacyjnych:

- **Windows XP** Windows XP nie posiada narzędzia bootrec.exe zaś posiada zamiast tego narzędzie bootcfg.exe które może być używane do modyfikacji plików boot.ini.
- **Windows 7**

By uruchomić bootrec.exe możemy skorzystać z następujących opcji:
- Gdy posiadamy dysk z Windowsem 7 należy postępować według następujących kroków:
 1. Włożenie dysku instalacyjnego
 2. Ponowne uruchomienie komputera i naciśnięcie przycisku do rozruchu z płyty.
 3. Wybór języka, czasu oraz klawiatury.
 4. Wybranie opcji napraw mój komputer.
 5. Wybór systemu operacyjnego oraz przejście do kolejnego kroku.

6. W okienku opcji odzyskiwania systemu należy uruchomić wiersz poleceń.
 7. W wierszu poleceń wpisać należy bootrec.exe i wcisnąć enter.
- Jeśli nie posiadamy dysku instalacyjnego wykonujemy następujące kroki:
 1. Należy uruchomić ponownie komputer.
 2. Wcisnąć przycisk F8.
 3. Należy wybrać opcję napraw mój komputer i następnie wcisnąć przycisk enter
 4. Uruchomić należy wiersz poleceń i w wierszu poleceń wpisać następującą komendę bootrec /rebuildbcd i wcisnąć przycisk enter.

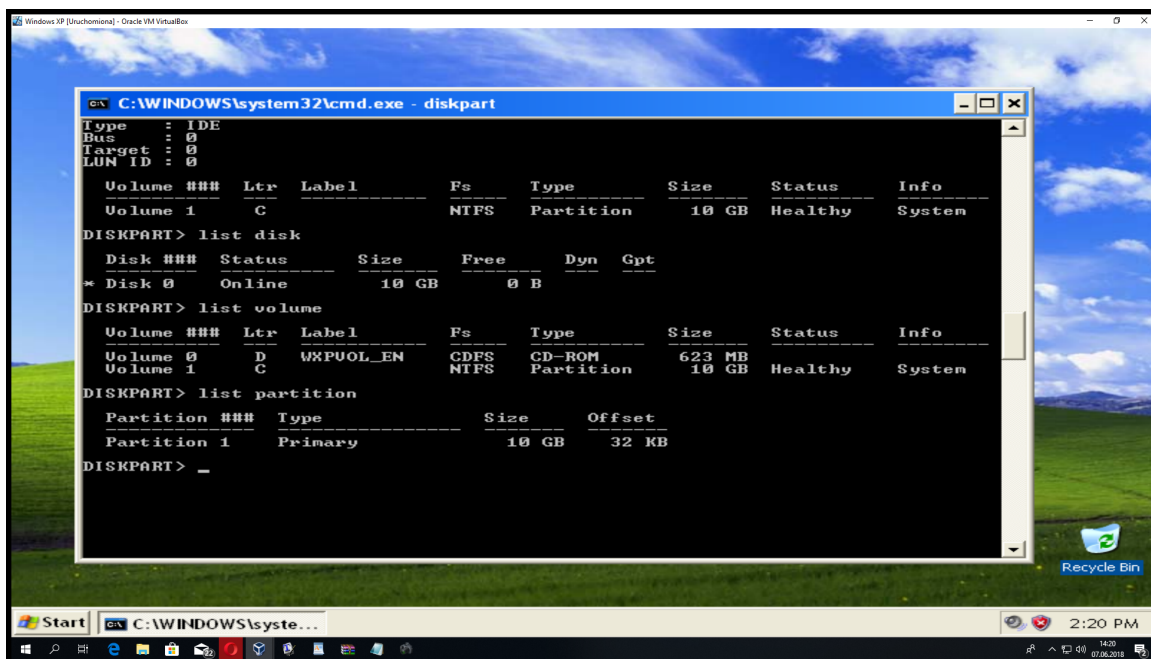
Windows 10

- Jeśli posiadamy dysk instalacyjny powinniśmy wykonywać następujące kroki:
 1. Włożenie dysku instalacyjnego systemu Windows 10
 2. Ponowne uruchomienie komputera.
 3. Naciśnięcie dowolnego klawisza po uruchomieniu komputera.
 4. Wybranie opcji napraw mój komputer.
 5. Wybór opcji rozwiązywania problemów oraz uruchomienie wiersza poleceń
 6. Po uruchomieniu wiersza poleceń konieczne jest wpisanie polecenia bootrec /FixMbr i naciśnięcie przycisku enter.
 7. Gdy zakończony zostanie proces, wpisujemy exit oraz wciskamy przycisk enter i wysuwamy dysk z napędu.
- Jeśli nie posiadamy dysku to powinniśmy wykonywać następujące kroki:
 1. Ponowne uruchomienie komputera.
 2. Szybkie wciśnięcie przycisku Shift oraz przycisku F8.
 3. Wybór opcji rozwiązywania problemów.
 4. Uruchomienie wiersza poleceń.
 5. Po uruchomieniu wiersza poleceń, należy wpisać komendę bootrec /FixMbr i wcisnąć enter.

3.5 Diskpart

Narzędzie to może być użyte do tworzenia oraz usuwania partycji dyskowych na komputerze użytkownika. Narzędzie to pozwala na następujące rzeczy:

- Porcjowanie.
- Usuwanie partycji dyskowych.
- Usuwanie formatowań.
- Przypisywanie i usuwanie liter dyskowych oraz punktów instalacyjnych.
- Konwertowanie dysku z podstawowego na dysk dynamiczny.
- Tworzenie i rozszerzanie woluminium dyskowego.



```
Windows XP [Uruchomiona] - Oracle VM VirtualBox
C:\WINDOWS\system32\cmd.exe - diskpart
Type : IDE
Bus : 0
Target : 0
LUN ID : 0

Volume ### Ltr Label Fs Type Size Status Info
-----
Volume 1 C NTFS Partition 10 GB Healthy System

DISKPART> list disk

Disk ### Status Size Free Dyn Gpt
---
* Disk 0 Online 10 GB 0 B

DISKPART> list volume

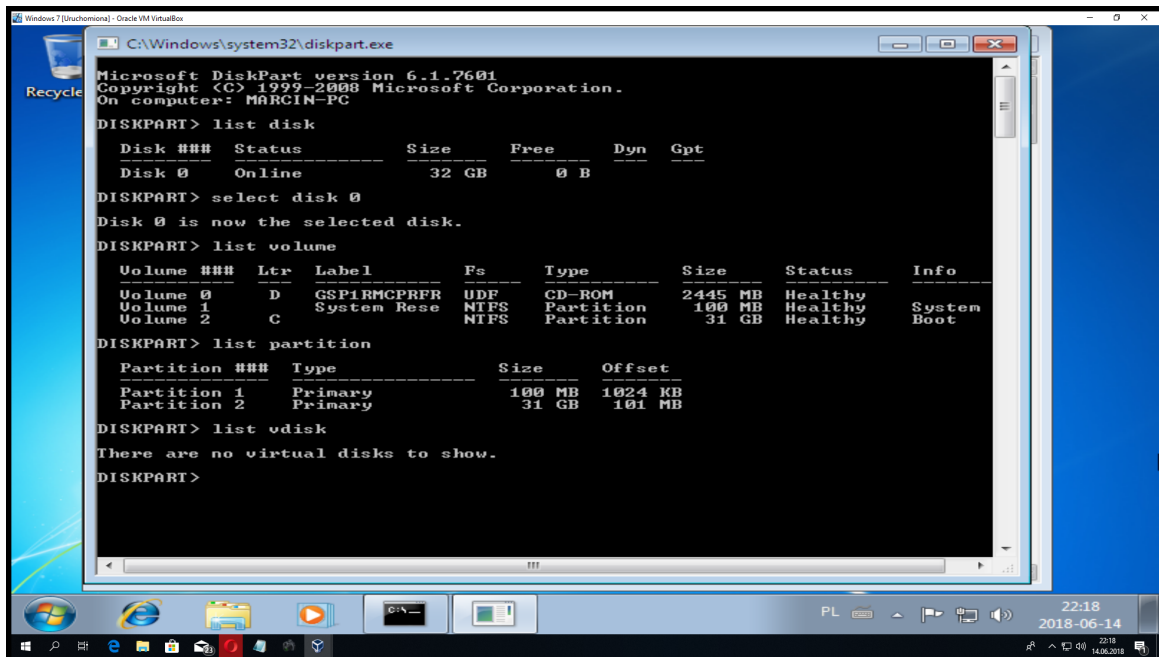
Volume ### Ltr Label Fs Type Size Status Info
-----
Volume 0 D WXPVOL_EN CD-ROM 623 MB Healthy System
Volume 1 C NTFS Partition 10 GB Healthy System

DISKPART> list partition

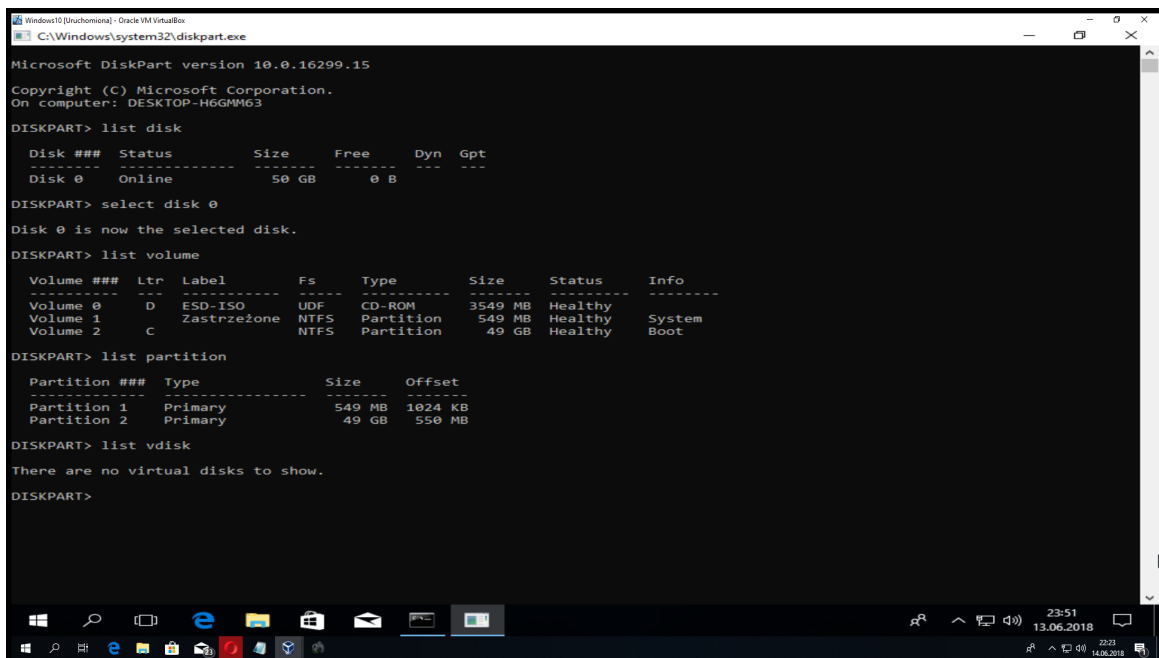
Partition ### Type Size Offset
-----
Partition 1 Primary 10 GB 32 KB

DISKPART> -
```

Rysunek 3.1: „Partycje w Windows XP”



Rysunek 3.2: „Partycje w Windows 7”



Rysunek 3.3: „Partycje w Windows 10”

Bibliografia

- [1] <http://www.tomshardware.com/reviews/bios-beginners,1126.html>
- [2] <https://computer.howstuffworks.com/bios.html>
- [3] <https://www.howtogeek.com/56958/htg-explains-how-uefi-will-replace-the-bios/>
- [4] <https://www.chip.pl/2009/10/kolebka-windowsa-czyli-o-microsofcie-od-samego-poczatku-czesc-1/>
- [5] <https://computer.howstuffworks.com/bios1.htm>
- [6] <https://www.lifewire.com/unified-extensible-firmware-interface-833069>
- [7] <http://searchwindowserver.techtarget.com/definition/Windows>
- [8] <https://www.linux.com/what-is-linux>
- [9] <http://forum.komputerswiat.pl/topic/14697-etapy-uruchamiania-systemu-windows/>
- [10] <http://linuxmdv.cba.pl/jakdzialalinux1.php>
- [11] <http://en.community.dell.com/techcenter/extras/w/wiki/2838>
- [12] <https://www.minitool.com/lib/mbr-master-boot-record.html>
- [13] <https://www.techopedia.com/definition/3332/cold-boot>
- [14] https://www.webopedia.com/TERM/W/warm_boot.html
- [15] <http://multiboot.solaris-x86.org/ii/2.html>
- [16] <http://multiboot.solaris-x86.org/ii/4.html>
- [17] <https://neosmart.net/wiki/bcdedit/>
- [18] <https://neosmart.net/wiki/bootsect/>

-
- [19] <https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/bootsect-command-line-options>
- [20] <https://neosmart.net/wiki/bootrec/>
- [21] <https://support.microsoft.com/en-sg/help/927392/use-bootrec-exe-in-the-windows-re-to-troubleshoot-startup-issues>
- [22] <https://neosmart.net/wiki/diskpart/>
- [23] https://soisk.info/index.php/Zarzdzanie_dyskami_DISKPART
- [24] <http://www.geoffchappell.com/notes/windows/boot/bcd/index.htm>
- [25] <https://www.computerhope.com/issues/ch000192.htm>
- [26] <http://students.mimuw.edu.pl/S0/Projekt03-04/temat3-g4/cmos.html>
- [27] <http://www.informit.com/articles/article.aspx?p=130913&seqNum=5>
- [28] <http://students.mimuw.edu.pl/S0/Projekt03-04/temat3-g4/post.html>
- [29] <https://www.maketecheasier.com/differences-between-uefi-and-bios/>
- [30] <https://www.quora.com/What-is-the-difference-between-UEFI-and-Legacy-Mode-which-we-need-to-choose-while-installing-the-OS>
- [31] https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface
- [32] https://en.wikipedia.org/wiki/GNU_GRUB
- [33] <https://www.gnu.org/software/grub/manual/grub/grub.html#Introduction>
- [34] <https://en.wikipedia.org/wiki/VirtualBox>
- [35] https://en.wikipedia.org/wiki/Partition_type
- [36] https://en.wikipedia.org/wiki/Master_boot_record
- [37] http://www.centrumxp.pl/Expert/1491,1,BCDEdit_bez_tajemnic.aspx

-
- [38] https://en.wikipedia.org/wiki/GUID_Partition_Table
- [39] https://en.wikipedia.org/wiki/Logical_block_addressing
- [40] https://en.wikipedia.org/wiki/Cyclic_redundancy_check
- [41] <https://www.quora.com/What-is-the-step-by-step-booting-process-of-Windows>
- [42] <http://math.uwb.edu.pl/~mariusz/w/Russinovich\%20-\%20Windows\%20Internals\%20-\%20Part\%202.pdf>
- [43] https://www.gegeek.com/tech_reference/home_page/Windows7\%20Boot\%20Process.pdf
- [44] <https://docs.microsoft.com/en-us/previous-versions/windows/hardware/bringup/dn756626\%28v=vs.85\%29>
- [45] <https://www.quora.com/What-is-the-step-by-step-booting-process-of-Windows>
- [46] Boot Configuration Data in Windows Vista