kapitola

5

Spouštění a vypínání

Obsah kapitoly:

V této kapitole si popíšeme kroky potřebné ke spuštění systému Microsoft Windows a volby, které mohou spouštění systému ovlivnit. Budete-li rozumět podrobnostem spouštěcího procesu, snáze se vám budou diagnostikovat potíže, jež se v této fázi mohou projevit. Pak si vysvětlíme věci, které se mohou během procesu spouštění pokazit, a seznámíme se s jejich řešením. Nakonec si ukážeme, k čemu dochází při řádném vypínání systému.

5.1 Proces spouštění

V rámci popisu startování (bootování) Windows začneme instalací systému a budeme pokračovat vykonáváním podpůrných souborů spouštění. Zásadní součástí procesu spouštění jsou ovladače zařízení, takže si vysvětlíme způsob, jakým se řídí místo jejich zavádění a inicializace v procesu spouštění. Dále si popíšeme, jak subsystémy výkonné části inicializují a jádro spouští tu část Windows, která pracuje v uživatelském režimu, nastartováním procesu správce relací (`SMSS.exe`), subsystému Windows a přihlašovacího procesu (`Winlogon`). Přitom si zdůrazníme místa, v nichž se na obrazovce objevuje různý text pomáhající během spouštění Windows rozpoznat určitý interní proces podle toho, co vidíte na obrazovce.

Počáteční fáze procesu spouštění se na systémech x86 a x64 výrazně odlišují od systémů IA64. Následující oddíl popisuje části spouštěcího procesu, které jsou specifické pro architektury x86 a x64. Dále je oddíl popisující ty části spouštěcího procesu, jež jsou specifické pro IA64.

Počáteční fáze spouštění na x86 a x64

Proces spouštění Windows nezačíná až v okamžiku, když zapnete počítač nebo stisknete tlačítko resetu. Začíná již při instalaci Windows na počítač. V určitém okamžiku vykonávání instalačního programu Windows se primární pevný disk systému vybaví kódem, který je součástí spouštěcího procesu. Ještě než se ale dostaneme k tomu, co takový kód dělá, podívejme se na to, jak a kam tento kód Windows umisťují na disk. Již od dávných dob systému MS-DOS existoval na systémech x86 standard určující, jak se fyzické jednotky dělí na svazky. Operační systémy společnosti Microsoft dělí pevné disky na diskrétní oblasti označované za *oddíly* (partitions) a k naformátování každého takového oddílu na svazek využívají souborové systémy (např. FAT nebo NTFS). Pevný disk může obsahovat až čtyři primární oddíly. Protože toto schéma rozdělení by omezovalo disk na čtyři svazky, dokáže speciální typ oddílu, označovaný za *rozšířený oddíl*, alokovat další až čtyři oddíly v rámci každého z primárních oddílů. Rozšířené oddíly mohou obsahovat rozšířené oddíly, které mohou obsahovat další rozšířené oddíly atd., takže počet svazků, které může operační systém umístit na disk, je v zásadě nekonečný. Obrázek 5.1 zachycuje příklad rozložení pevného disku a tabulka 5.1 shrnuje soubory účastnící se procesu spouštění na architekturách x86 a x64. (O tom, jak Windows pracují s oddíly, se více dozvíte v kapitole 10, která se zabývá správou úložišť.)

Fyzické disky se adresují v jednotkách označovaných za *sektory*. Sektor pevného disku na PC kompatibilním se specifikací IBM má většinou 512 bajtů. Nástroje, které připravují pevné disky na definice svazků, včetně utility Fdisk systému MS-DOS nebo instalačního programu Windows, zapisují na první sektor pevného disku urči-

tá data označovaná za hlavní spouštěcí záznam (Master Boot Record – MBR). (Oddíly MBR si vysvětlíme v kapitole 10.) Záznam MBR zahrnuje pevně daný prostor obsahující vykonatelné instrukce (označované za *spouštěcí kód*) a tabulku (nazývanou *tabulka oddílů*) se čtyřmi položkami definujícími umístění primárních oddílů na disku. Když se spouští počítač kompatibilní se specifikací IBM, tak jako první vykoná kód označovaný za BIOS – ten je zakódován do paměti ROM počítače. BIOS vybere spouštěcí zařízení, načte MBR tohoto zařízení do paměti a přenese řízení na kód obsažený v daném záznamu MBR.

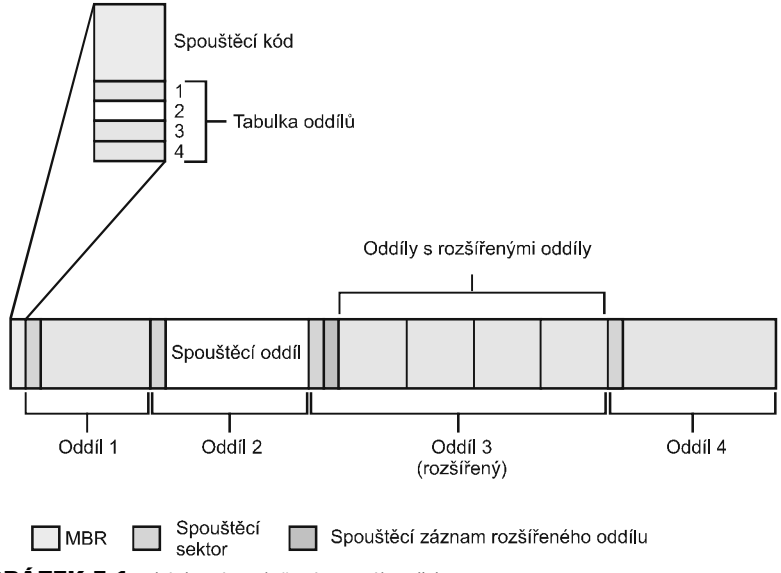
Záznamy MBR zapisované nástroji vytváření oddílů od společnosti Microsoft, jako jsou ty integrované do instalačního programu Windows a modulu snap-in správy disků konzoly MMC, procházejí podobným procesem načítání a přenášení řízení. Kód MBR nejprve prochází tabulkou primárního oddílu, dokud nenajde oddíl obsahující příznak signalizující, že je spustitelný. Jakmile MBR nalezne alespoň jeden takový příznak, načte první sektor označeného oddílu do paměti a předá řízení kódu daného oddílu. Tento typ oddílu se označuje za *spouštěcí oddíl* (boot partition) a první sektor takového oddílu se nazývá *spouštěcí sektor* (boot sector). Svazek definovaný pro tento spouštěcí oddíl se nazývá *systémový svazek* (system volume).

TABULKA 5.1: Komponenty spouštěcího procesu systémů x86 a x64

Komponenta	Vykonávání procesoru	Zodpovědnosti
Kód hlavního spouštěcího záznamu (Master Boot Record – MBR)	16bitový reálný režim	Načítá a zavádí spouštěcí sektory oddílu.
Spouštěcí (boot) sektor	16bitový reálný režim	Načítá kořenový adresář, aby bylo možné zavést `Ntldr`.
`Ntldr`	16bitový reálný režim a 32bitový nebo 64bitový chráněný režim; zapíná stránkování	Načítá `Boot.ini`, nabízí spouštěcí nabídku a nahrává `Ntoskrnl.exe`, `Bootvid.dll`, `Hal.dll` a ovladače zařízení zaváděné při startování. Spouští-li se 32bitová instalace, přepne do 32bitového chráněného režimu; je-li spouštěna 64bitová instalace, přepne do 64bitového dlouhého režimu.
`Ntdetect.com`	16bitový reálný režim	Zajišťuje detekci hardwaru pro `Ntldr`.
`Ntbootdd.sys`	Chráněný režim	Ovladač zařízení používaný pro operace I/O na systémech SCSI a ATA (Advanced Technology Attachment), kde se nepoužívá BIOS.
`Ntoskrnl.exe`	Chráněný režim se stránkováním	Inicializuje subsystémy výkonné části a ovladače zařízení zaváděné při prvotním spouštění a startu systému, připravuje systém na provozování nativních aplikací a spouští `SMSS.exe`.
`Hal.dll`	Chráněný režim se stránkováním	Knihovna DLL režimu jádra, která zajišťuje rozhraní mezi Ntoksnrl a ovladači hardwaru.

Komponenta	Vykonávání procesoru	Zodpovědnosti
`SMSS`	Nativní aplikace	Zavádí subsystém Windows včetně `Win32k.sys` a `Csrss.exe` a spouští proces `Winlogon`.
`Winlogon`	Nativní aplikace	Spouští správce řízení služeb (SCM), spouští subsystém lokálního zabezpečení (Local Security Subsystem – LSASS) a zobrazuje dialogové okno interaktivního přihlášení.
Správce řízení služeb (Service Control Manager – SCM)	Nativní aplikace	Zavádí a inicializuje automaticky spouštěné ovladače zařízení a služby Windows.

Operační systémy většinou zapisují spouštěcí sektory na disk bez akce uživatele. Když kupříkladu instalační program Windows zapisuje na pevný disk MBR, zapisuje také spouštěcí sektor na první spustitelný oddíl disku. Během instalace systému MS-DOS, Windows ME, Windows 98 nebo Windows 95 mohlo dojít k vytvoření spouštěcího sektoru systému MS-DOS. Instalační program Windows prověřuje, zda je spouštěcí sektor přepisovaný spouštěcím sektorem Windows platným spouštěcím sektorem systému MS-DOS. Pokud ano, instalační program Windows zkopíruje obsah tohoto spouštěcího sektoru do souboru nazvaného `Bootsect.dos` v kořenovém adresáři příslušného oddílu.



OBRÁZEK 5.1: Ukázkové rozložení pevného disku

Před zápisem do spouštěcího sektoru oddílu si instalační program Windows zajistí, aby byl daný oddíl naformátován systémem souborů, který Windows podporují (FAT, FAT32 nebo NTFS) – naformátuje spouštěcí oddíl (a jakýkoli jiný oddíl) vámi zadaným systémem souborů. Jsou-li oddíly již naformátované, můžete instalačnímu programu říci, aby tento krok přeskočil. Jakmile instalační program naformátuje

spouštěcí oddíl, zkopíruje na něj (tedy na systémový svazek) soubory používané Windows. To zahrnuje i dva soubory, které jsou součástí spouštěcí sekvence, `Ntldr` a `Ntdetect.com`.

Další úlohou instalačního programu je vytvořit v kořenovém adresáři systémového svazku soubor spouštěcí nabídky, `Boot.ini`. Ten obsahuje volby spuštění té verze systému Windows, kterou právě instalujete, a také již dříve instalovaných Windows. Jestliže `Bootsect.dos` obsahuje platný spouštěcí sektor systému MS-DOS, jednou z položek vytvořených v `Boot.ini` bude spuštění systému MS-DOS. Následující výstup ukazuje příklad souboru `Boot.ini` z počítače s duálním spouštěním, kde byl systém MS-DOS instalován před Windows XP:

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
[operating systems]
multi(0)disk(0)rdisk(0)partition(1) \WINDOWS="Microsoft Windows XP
Professional"
fastdetect
C:\= "Microsoft Windows"
```


Všimněte si, že je v ukázkovém souboru cesta k adresáři Windows zadána pomocí speciální syntaxe odpovídající konvenci Advanced RISC Computing (ARC) pro vytváření názvů. Systém Windows používá tři varianty této syntaxe. První, uvedená v předchozí ukázce kódu, je syntaxe `multi()`. Ta říká Windows, že mají pomocí funkcí BIOS INT 13 zavést systémové soubory. Syntaxe `multi()` je tu tedy uvedena, pokud má disk, na němž se daný spouštěcí svazek nachází, řadič podporující INT-13. Syntaxe `multi()` má následující formát:

```
multi(W)disk(X)rdisk(Y)partition(Z)
```

`W` je číslo řadiče disku (označované také za pořadové číslo) a většinou má hodnotu 0. `X` je v syntaxi `multi()` vždy 0. `Y` specifikuje fyzický pevný disk připojený k řadiči W. V případě řadičů ATA je toto číslo většinou v rozsahu od 0 do 3. V případě řadičů SCSI se většinou jedná o hodnoty mezi 0 a 15. `Z` představuje číslo toho oddílu na fyzickém disku, jenž odpovídá spouštěcímu svazku. Prvnímu oddílu se přiřazuje číslo 1.

Syntaxe scsi() ARC informuje Windows, že má pro přístup k souborům na daném spouštěcím svazku využívat služby diskových operací I/O zajišťovaných `Ntbootdd.sys` (jak si za okamžik popíšeme). Formát syntaxe scsi() vypadá takto:

```
scsi(W)disk(X)rdisk(Y)partition(Z)
```

V této syntaxi představuje `W` číslo řadiče a `X` je fyzický pevný disk připojený k řadiči, což je většinou hodnota mezi 0 a 15. `Y` specifikuje číslo logické jednotky (Logical Unit Number – LUN) SCSI disku, který obsahuje spouštěcí svazek a má většinou hodnotu 0. Konečně `Z` je oddíl odpovídající spouštěcímu svazku s číslováním začínajícím od 1.

Poslední syntaxí používanou systémem Windows je syntaxe `signature()`. Ta říká Windows, že mají nalézt disk s podpisem odpovídajícím první hodnotě v závorkách, bez ohledu na číslo řadiče přidružené tomuto disku, a pro přístup k tomuto spouštěcímu svazku použít `Ntbootdd.sys`. Podpis (signatura) disku je globálně jednoznačný

identifikátor (Globally Unique Identifier – GUID), který instalační program Windows získává z údajů v MBR a zapisuje na disk. Syntaxe signature() vypadá takto:

```
signature(V)disk(X)rdisk(Y)partition(Z)
```

`V` je 32bitový hexadecimální podpis disku, který jej identifikuje. `X` je fyzický pevný disk s určitým podpisem, který může být připojen k libovolnému řadiči v systému. `Y` je vždy 0 a `Z` je číslo oddílu, na němž se nachází spouštěcí jednotka.

Windows používají syntaxi `signature()` v následujících případech:

- Spouštěcí svazek je větší než 7,8 GB a rozšířené funkce INT-13 systému BIOS (jež se používají pro přístup k disku v částech za hranicí 7,8 GB) nemohou přistupovat k celému disku.
- BIOS nepodporuje rozšířené instrukce INT-13.

Spouštěcí sektor a Ntldr na systémech x86/x64

Před zápisem spouštěcího sektoru musí instalační program znát formát oddílu, protože obsah spouštěcího sektoru na něm závisí. Pokud má spouštěcí oddíl kupříkladu formát FAT, Windows zapíší do spouštěcího sektoru kód, který systému souborů FAT rozumí. Jestliže je ale daný oddíl ve formátu NTFS, Windows zapíší kód podporující NTFS. Kód spouštěcího sektoru má za úkol předat systému Windows informace o struktuře a formátu daného svazku a načíst soubor `Ntldr` z kořenového adresáře svého svazku. To znamená, že kód spouštěcího sektoru obsahuje právě tolik kódu systému souborů pouze pro čtení, kolik je zapotřebí ke splnění této úlohy. Jakmile kód spouštěcího sektoru zavede do paměti `Ntldr`, předá řízení vstupnímu bodu `Ntldr`. Nedokáže-li kód spouštěcího sektoru nalézt v kořenovém adresáři svého svazku `Ntldr`, zobrazí zprávu „BOOT: Couldn't find NTLDRP“ (nelze nalézt NTLDRP), jedná-li se o souborový systém FAT, nebo „NTLDR is missing“ (chybí NTLDR), je-li souborovým systémem NTFS.

Zavaděč `Ntldr` začíná existovat v okamžiku, kdy se systém vykonává v určitém operačním režimu x86 označovaném za *reálný režim* (real mode). V reálném režimu nedochází k žádnému překladu paměťových adres z virtuálních na fyzické. To znamená, že programy využívající nějaké paměťové adresy je interpretují jako fyzické adresy a že je přístupný pouze první megabajt fyzické paměti počítače. V prostředí reálného režimu se vykonávají jednoduché programy systému MS-DOS. První akcí `Ntldr` je ale přepnutí systému do *chráněného režimu* (protected mode). V tomto okamžiku stále nedochází k překladu virtuálních adres na fyzické adresy, zpřístupní se však celá 32bitově adresovatelná paměť. Jakmile je systém v chráněném režimu, může `Ntldr` přistupovat k celé fyzické paměti. Po vytvoření dostatečného počtu tabulek stránek, aby bylo možné přistupovat k paměti pod 16 MB se zapnutým stránkováním, `Ntldr` stránkování zapne (aktivuje). Právě v chráněném režimu s aktivním stránkováním běží za normálního stavu i Windows.

Jakmile zavaděč `Ntldr` zapne stránkování, je plně provozuschopný. Při přístupu k systémovým a spouštěcím diskům IDE a k displeji však stále využívá funkce dodávané spouštěcím kódem. Funkce spouštěcího kódu krátce vypnou stránkování a přepnou procesor zpět do režimu, v němž lze vykonávat služby poskytované systémem BIOS. Pokud je disk, obsahující spouštěcí svazek, formátu SCSI a nepřístupný prostřednictvím podpůrného firmwaru v BIOSu, zavede `Ntldr` soubor nazvaný

Ntbootdd.sys a při přístupu k disku jej využívá místo funkcí spouštěcího kódu. Ntbootdd.sys je kopií ovladače miniportu SCSI, který Windows používají pro přístup ke spouštěcímu disku při své normální činnosti. (Více údajů o ovladačích disků najdete v kapitole 10.) Ntldr dále načítá pomocí vestavěného kódu systému souborů soubor Boot.ini v kořenovém adresáři. Podobně jako kód spouštěcího sektoru obsahuje Ntldr kód přístupu pouze pro čtení k systémům NTFS a FAT; na rozdíl od kódu spouštěcího sektoru však kód systému souborů Ntldr dokáže číst i podadresáře.

Ntldr dále vymaže obrazovku. Je-li v kořenu systémového svazku platný soubor Hiberfil.sys, zkrátí proces spouštění načtením obsahu tohoto souboru do paměti a předáním řízení kódu v jádru, který spustí chod uspaného systému. Tento kód zodpovídá za restartování ovladačů, jež byly aktivní v okamžiku vypínání systému. Soubor Hiberfil.sys bude platný, pouze pokud byl počítač při posledním vypínání uspán (více se o režimu spánku dozvíte v oddílu „Správce napájení" v kapitole 11).

Nachází-li se v souboru Boot.ini více než jedna položka volby spuštění, nabídne zavaděč uživateli menu volby spuštění. (Je-li tu jen jedna položka, tak Ntldr tuto nabídku přeskočí a postoupí k zobrazení lišty postupu spouštění.) Položky voleb v souboru Boot.ini nasměrují Ntldr na oddíl, na němž se nachází systémový adresář Windows (většinou \Windows) vybrané instalace. Tento oddíl může být stejný jako spouštěcí oddíl, nebo se může jednat o jiný primární oddíl.

Odkazuje-li se daná položka Boot.ini na instalaci systému MS-DOS (když tedy za systémový oddíl označuje C:\), načte Ntldr do paměti obsah souboru Bootsect.dos, přepne se zpět do 16bitového reálného režimu a zavolá kód MBR v Bootsect.dos. Tato akce způsobí vykonání kódu v Bootsect.dos, jako by byl MBR načten z disku. Kód v Bootsect.dos pokračuje ve spouštění specifickém pro systém MS-DOS, jak se používá při spouštění Microsoft Windows ME, Windows 98 nebo Windows 95 na počítači, kde jsou tyto operační systémy instalovány společně s jinými Windows.

Položky v souboru Boot.ini mohou zahrnovat volitelné argumenty, které interpretuje Ntldr a další komponenty účastnící se procesu spouštění. Tabulka 5.2 obsahuje úplný seznam těchto voleb a jejich vlivu. Nástroj Bootcfg.exe, poprvé uvedený ve Windows XP, nabízí pohodlné prostředí umožňující nastavit řadu těchto přepínačů. Všechny volby obsažené v Boot.ini se ukládají do hodnoty registru HKLM\System\CurrentControlSet\Control\SystemStartOptions.

TABULKA 5.2: Možnosti spouštění

Kvalifikátor spouštění	Význam
/3GB	Zvyšuje velikost adresového prostoru pro uživatelské procesy ze 2 GB na 3 GB (a omezuje tedy velikost systémového prostoru ze 2 GB na 1 GB). Poskytnutí většího adresového prostoru může zvýšit výkonnost aplikací intenzivně využívajících virtuální paměť, jako jsou např. databázové servery. Aby mohla aplikace využít tohoto prvku, musejí být ale naplněny dvě dodatečné podmínky: Na systému musí běžet Windows XP, Windows Server 2003, Windows 2000 Advanced Server nebo Datacenter Server a soubor EXE dané aplikace musí být označen příznakem podpory 3GB prostoru (platí pouze pro 32bitové systémy). (Další informace najdete v oddílu „Rozvržení adresového prostoru" v kapitole 7.)

Kvalifikátor spouštění	Význam
/BASEVIDEO	Způsobuje, že systém Windows používá pro operace v režimu GUI (grafickém) standardní ovladač displeje VGA.
/BAUDRATE=	Aktivuje ladění v režimu jádra a specifikuje překrytí výchozí rychlosti komunikace (19 200), na které se vzdálený hostitel debuggeru jádra připojí. Příklad: /BAUDRATE=115200.
/BOOTLOG	Systém Windows zapíše protokol spouštění do souboru %SystemRoot%\Ntbtlog.txt.
/BOOTLOGO	Po zadání tohoto přepínače zobrazí Windows XP nebo Windows Server 2003 instalovatelnou úvodní obrazovku místo standardní úvodní obrazovky. Nejprve vytvořte 16barevnou (obsahující 16 libovolných barev) bitovou mapu 640 × 480 a uložte ji do adresáře Windows pod názvem Boot.bmp. Pak doplňte do boot.ini volbu „/bootlogo /noguiboot“.
/BREAK	Způsobí, že se vrstva abstrakce hardwaru (HAL) zastaví při své inicializaci v bodu přerušení. První věcí, kterou jádro Windows dělá při inicializaci, je právě inicializace vrstvy HAL, takže tento bod přerušení je nejdřívější možný. HAL bude donekonečna čekat v daném bodu přerušení na vytvoření připojení debuggeru jádra. Je-li tento přepínač použit bez přepínače /DEBUG, vyvolá systém modrou obrazovku s kódem STOP hodnoty 0x00000078 (PHASE0_EXCEPTION).
/BURNMEMORY=	Specifikuje množství paměti, kterou Windows nemohou používat (podobá se přepínači /MAXMEM). Tento údaj se zadává v megabajtech. Příklad: /BURNMEMORY=128 bude znamenat, že Windows nemohou použít 128 MB z celkového množství fyzické paměti na počítači.
/CHANNEL=	Používá se ve spojení s /DEBUGPORT=1394 ke specifikaci kanálu IEEE 1394, kterým poteče komunikace ladění jádra. Může se jednat o libovolné číslo od 0 do 62 a není-li specificky nastaveno, využije se výchozí hodnota 0.
/CLKLVL	Způsobí, že se standardní víceprocesorová vrstva HAL architektury x86 (Halmps.dll) nakonfiguruje pro systémové hodiny spouštěné úrovní (level-sensitive) a nikoli hranou (edge-triggered) hodiny. *Level-sensitive* a *edge-triggered* jsou pojmy používané k popisu typů hardwarových přerušení.
/CMDCONS	Předává se při náběhu do konzoly pro zotavení (Recovery Console – popsáno dále v této kapitole).
/CRASHDEBUG	Debugger jádra se zavede při spouštění systému, zůstane ale neaktivní, pokud nedojde ke zhroucení systému. Sériový port využívaný debuggerem jádra tak bude k dispozici systému až do jeho zhroucení (to je rozdíl oproti /DEBUG, kdy debugger jádra využívá zadaný sériový port po celou dobu systémové relace).
/DEBUG	Aktivuje ladění v režimu jádra.

Kvalifikátor spouštění	Význam
/DEBUGPORT=	Aktivuje ladění v režimu jádra a přepisuje výchozí sériový port (obvykle COM2 na systémech s přinejmenším dvěma sériovými porty), k němuž je připojen vzdálený hostitel debuggeru jádra. Windows XP a Windows Server 2003 podporují také ladění prostřednictvím portů IEEE 1394. Příklady: /DEBUGPORT=COM2, /DEBUGPORT=1394.
/EXECUTE	Deaktivuje ochranu před vykonáváním. Další informace najdete u přepínače /NOEXECUTE.
/FASTDETECT	Výchozí spouštěcí volba Windows. Nahrazuje přepínač /NOSERIALMICE systému Windows NT 4. Důvod existence tohoto kvalifikátoru (místo toho, aby NTDETECT prostě tuto operaci prováděl standardně) je dán tím, že NTDETECT musí podporovat spouštění Windows NT 4. Ovladače zařízení Plug and Play systému Windows vykonávají detekci paralelních a sériových zařízení, systém Windows NT 4 se s tím ale spoléhá na NTDETECT. Proto zadání /FASTDETECT způsobí, že NTDETECT přeskočí výčet paralelních a sériových zařízení (akce, které nejsou pro spouštění Windows nezbytné). Když přepínač není zadán, NTDETECT zmíněný výčet zajistí (což je nezbytné pro spouštění Windows NT 4).
/INTAFFINITY	Říká standardní víceprocesorové vrstvě HAL architektury x86 (Halmps.dll), že má nastavit takové afinity přerušení, aby přerušení přijímal pouze procesor s nejvyšším číslem. Bez tohoto přepínače využije HAL své výchozí chování spočívající v tom, že přerušení přijímají všechny procesory.
/LASTKNOWNGOOD	Systém se spustí, jako by byla vybrána volba LastKnownGood (Poslední známá funkční konfigurace).
/MAXMEM=	Omezuje systém Windows tím způsobem, že bude ignorovat (nebude používat) fyzickou paměť za zadaným množstvím. Toto číslo představuje megabajty. Příklad: /MAXMEM=64 omezí systém na využívání prvních 64 MB fyzické paměti, i když je jí k dispozici více.
/MAXPROCSPERCLUSTER=	V případě standardní víceprocesorové vrstvy HAL architektury x86 (Halmps.dll) si vynucuje adresování řadiče APIC (Advanced Programmable Interrupt Controller) v režimu clusterů (což není podporováno na systémech s externím řadičem přerušení APIC 82489DX).
/MININT	Tuto volbu používá předinstalační prostředí (Preinstallation Environment – PE) Windows a způsobuje, že správce konfigurace zavádí podregistr SYSTEM jako nestálý, takže změny v něm prováděné v paměti se neukládají do obrazu podregistru.
/NODEBUG	Zabraňuje inicializaci ladění v režimu jádra. Překrývá specifikaci všech ostatních tří přepínačů souvisejících s laděním, /DEBUG, /DEBUGPORT a /BAUDRATE.

Kvalifikátor spouštění	Význam
/NOLOWMEM	Vyžaduje přítomnost přepínače /PAE a systém musí mít navíc více než 4 GB fyzické paměti. Jsou-li tyto podmínky naplněny, pak nebude ta verze jádra Windows, která podporuje PAE (Ntkrnlpa.exe), používat první 4 GB fyzické paměti. Místo toho bude zavádět všechny aplikace a ovladače zařízení a alokovat všechny paměťové fondy až nad touto hranicí. Tento přepínač je užitečný pouze k testování kompatibility ovladačů zařízení se systémy vybavenými velkou pamětí.
/KERNEL= /HAL=	Dovoluje vám překrýt pro Ntldr výchozí název obrazu jádra (Ntoskrnl.exe) a/nebo vrstvy HAL (Hal.dll). Tyto volby jsou užitečné pro alternování mezi ověřovacím prostředím jádra a volným (běžně dodávaným) prostředím jádra nebo i k manuální volbě jiné vrstvy HAL. Chcete-li spustit ověřovací prostředí, jež sestává čistě z ověřovaného jádra a HAL, což většinou plně postačuje k testování ovladačů, postupujte na systému s instalovaným volným sestavením následovně: 1. Zkopírujte ověřované verze obrazů jádra z CD ověřovaného sestavení do adresáře \Windows\System32 a obrazům zadejte jiné názvy, než jsou výchozí. Jste-li kupříkladu na jednoprocesorovém systému, zkopírujte Ntoskrnl.exe jako Ntoschk.exe a Ntkrnlpa.exe jako Ntoschkpa.exe. Jste-li na víceprocesorovém systému, zkopírujte Ntkrnlmp.exe jako Ntoschk.exe a Ntkrpamp.exe jako Ntoschkpa.exe. Název souboru jádra musí být krátký, tedy ve stylu 8.3. 2. Zkopírujte ověřované verze příslušné vrstvy HAL vyžadované vaším systémem z \I386\Driver.cab na CD ověřovaného sestavení do adresáře \Windows\System32 a soubor přejmenujte na Halchk.dll. Chcete-li zjistit, jakou vrstvu HAL máte zkopírovat, otevřete si \Windows\Repair\Setup.log a vyhledejte Hal.dll; najdete tu řádek jako \WINDOWS\system32\hal.dll="halacpi.dll", "1d8a1". Název hned za rovnítkem představuje soubor HAL, který musíte zkopírovat. Název souboru vrstvy HAL musí být krátký, tedy ve stylu 8.3. 3. Vytvořte kopii výchozího řádku v systémovém souboru Boot.ini. 4. Do řetězce popisu volby spouštění zadejte text indikující nový výběr odpovídající prostředí ověřovaného sestavení (kupříkladu „Windows XP Professional Checked“). 5. Na konec řádku nové volby zadejte následující: /KERNEL=NTOSCHK.EXE /HAL=HALCHK.DLL Když se nyní v procesu spouštění zobrazí nabídka spuštění, můžete zvolit nově vytvořenou položku a spustit ověřované prostředí nebo zůstat u položky zajišťující spuštění volného sestavení.

Kvalifikátor spouštění	Význam
/NOGUIBOOT	Instruuje systém Windows v tom smyslu, že nemá inicializovat ovladač VGA videa zodpovědný za představení bitmapové grafiky během procesu spouštění. Tento ovladač se používá k zobrazování informací o postupu spouštění, takže po jeho zákazu nebude moci systém Windows tyto informace uvádět.
/NOPAE	Nutí Ntldr zavádět takovou verzi jádra Windows, která nepodporuje PAE (Physical Address Extension – rozšíření fyzické adresy), i když je u daného systému detekována podpora instrukcí PAE platformy x86 a má více než 4 GB fyzické paměti.
/NOEXECUTE	Tato volba je k dispozici pouze na 32bitové verzi Windows při běhu na procesorech AMD64 a za situace, kdy je zároveň aktivní PAE (vysvětleno dále v popisu přepínače /PAE). Aktivuje ochranu před vykonáváním, kdy správce paměti označuje za nevykonatelné stránky obsahující data, takže je nelze vykonat jako kód. To může být užitečné k zabránění zneužití chyb přetečení bufferu zákeřným softwarem. Ochrana před vykonáváním je na 64bitových verzích Windows běžících na procesorech AMD64 vždy aktivní. Existují čtyři modifikátory, jež lze přiřadit přepínači /NOEXECUTE: =OPTIN, =OPTOUT, =ALWAYSON, =ALWAYSOFF. Popis jejich chování najdete v kapitole 7.
/NOSERIALMICE= [COMx | COMx,y,z...]	Zastaralý kvalifikátor Windows NT 4 – nahrazen neexistencí přepínače /FASTDETECT. Deaktivuje detekci sériové myši na zadaných portech COM. Tento přepínač se používal, když jste měli během spouštění systému k některému ze sériových portů připojené jiné zařízení než myš. Použití /NOSERIALMICE bez zadání portu COM deaktivuje detekci sériové myši na všech portech COM. Další informace najdete v článku Microsoft Knowledge Base s číslem Q131976.
/NUMPROC=	Specifikuje počet procesorů (CPU), které lze využívat na víceprocesorových systémech. Příklad: /NUMPROC=2 na čtyřprocesorovém systému zabrání systému Windows ve využívání dvou ze čtyř instalovaných procesorů.
/ONECPU	Způsobí, že systém Windows používá jen jeden CPU na víceprocesorovém systému.
/PAE	Ntldr zavede Ntkrnlpa.exe, což je určitá verze jádra pro architekturu x86, která dokáže využívat instrukce PAE platformy x86. Tato verze PAE jádra nabídne ovladačům zařízení 64bitové fyzické adresy, což je užitečné k testování podpory systémů s velkou pamětí ze strany ovladačů zařízení.
/PCILOCK	Zabrání Windows v dynamickém přiřazování prostředků IO/IRQ zařízením PCI a nechá konfiguraci zařízení na systému BIOS. Další informace najdete v článku Microsoft Knowledge Base číslo Q148501.
/RDPATH=	Zadává cestu k souboru obrazu systémového disku (System Disk Image – SDI), který se může nacházet na síti a používá se ke spuštění. Často se aplikuje ve spojení s příznakem /RDIMAGEOFFSET=. Ten zavaděči NTLDR určuje, kde v zadaném souboru začíná obraz systému.

Kvalifikátor spouštění	Význam
/REDIRECT	Uvedeno prvně v systému Windows Server 2003. Windows pak aktivují Emergency Management Services (EMS), které hlásí údaje o spouštění a přijímají příkazy správy přes sériový port. Řádky redirect= a redirectbaudrate= v oddílu [boot loader] souboru Boot.ini specifikujete sériový port a rychlost přenosu používanou ve spojení s EMS.
/SAFEBOOT:	Specifikuje volby pro spouštění v nouzovém režimu. Tuto volbu asi nikdy nebudete muset zadávat manuálně, protože Ntldr ji zadá za vás, když k zadání spuštění v nouzovém režimu použijete nabídku zobrazenou klávesou F8. (Spuštění v nouzovém režimu je takové, kdy systém Windows zavádí pouze ovladače a služby specifikované názvy nebo skupinami v klíčích registru Minimal nebo Network pod HKLM\SYSTEM\ CurrentControlSet\Control\SafeBoot.) Za dvojtečkou musíte zadat jeden ze tří dodatečných přepínačů: MINIMAL, NETWORK nebo DSREPAIR. Příznak MINIMAL a NETWORK odpovídají spouštění v bezpečném režimu bez podpory sítě resp. s podporou sítě. Přepínač DSREPAIR (Directory Services Repair – oprava adresářových služeb) způsobí spuštění Windows v takovém režimu, kdy je adresářová služba Active Directory offline (vypnutá) a její databáze není otevřená. To umožní správci provádět na databázi diagnostické, opravné nebo obnovující funkce. Další volbou, kterou můžete doplnit, je (ALTERNATESHELL). Řeknete tím systému Windows, že má jako grafické prostředí použít program zadaný hodnotou HKLM\SYSTEM\CurrentControlSet\SafeBoot\AlternateShell a nikoli implicitní program, jímž je Windows Explorer.
/SCSIORDINAL:	Směruje systém Windows na identifikátor SCSI daného řadiče. (Když do systému doplníte nové zařízení SCSI s vestavěným řadičem SCSI, může dojít ke změně identifikátoru SCSI řadiče.) Další informace získáte ze článku Microsoft Knowledge Base číslo Q103625.
/SDIBOOT=	Používá se v systémech Windows XP Embedded k tomu, aby se systém spouštěl z obrazu na disku RAM (paměťovém) uloženého v zadaném souboru SDI (System Disk Image).
/SOS	Systém Windows bude vypisovat ovladače zařízení označené k zavádění během spouštění systému a následně zobrazí číslo verze systému (včetně čísla sestavení), množství fyzické paměti a počet procesorů.

Kvalifikátor spouštění	Význam
/TIMERES=	Nastavuje rozlišení systémového časovače ve standardní víceprocesorové vrstvě HAL architektury x86 (Halmps.dll). Argumentem je číslo interpretované jako stovky nanosekund, skutečná rychlost se ale nastaví na nejbližší rozlišení podporované vrstvou HAL, které není vyšší než to požadované. Vrstva HAL podporuje následující rozlišení: Stovky nanosekund / Milisekundy (ms) 97660,98 / 195322,00 390633,90 / 781257,80 Výchozím rozlišením je 7,8 ms. Rozlišení systémového časovače ovlivňuje rozlišení časovačů určeného k čekání na určitou událost. Příklad: /TIMERES=21000 nastaví časovač na rozlišení 2,0 ms.
/USERVA=	Tento přepínač je podporován pouze na Windows XP a Windows Server 2003. Podobně jako přepínač /3GB nabízí tento přepínač aplikacím větší adresový prostor. Zadejte množství v MB mezi 2048 a 3072. Tento přepínač klade na aplikace stejné požadavky jako přepínač /3GB a vyžaduje také přítomnost přepínače /3GB (platí pouze pro 32bitové systémy).
/WIN95	Přikazuje zavaděči Ntldr spustit sektor Consumer Windows uložený v souboru Bootsect.w40. Tento přepínač je využitelný pouze na systému s trojím spouštěním, na němž jsou instalované systémy MS-DOS, Consumer Windows a Windows. Další informace najdete v článku databáze Microsoft Knowledge Base číslo Q157992.
/WIN95DOS	Nařizuje zavaděči Ntldr spustit sektor Consumer Windows uložený v souboru Bootsect.w40. Tento přepínač je využitelný pouze na systému s trojím spouštěním, na němž jsou instalované systémy MS-DOS, Consumer Windows a Windows. Další informace najdete v článku databáze Microsoft Knowledge Base číslo Q157992.
/YEAR=	Říká základní časové funkci systému Windows, že má ignorovat rok hlášený hodinami reálného času a použít místo něj ten zde zadaný. To znamená, že rok stanovený v tomto přepínači ovlivní veškerý software na systému, a to včetně jádra Windows. Příklad: /YEAR=2001. (Tento přepínač byl vytvořen proto, aby pomáhal v testování Y2K, tedy problému přechodu do roku 2000.)

Nevybere-li uživatel žádnou položku ze spouštěcí nabídky v určitém čase zadaném v souboru Boot.ini, zavaděč Ntldr zvolí výchozí možnost, kterou je první položka v souboru boot.ini s cestou odpovídající cestě zadané na řádku "default=". Jakmile je zadána volba spouštění, Ntldr zavede a vykoná Ntdetect.com, což je 16bitový program pracující v reálném režimu, který se s využitím BIOSu systému dotazuje počítače na základní informace o zařízení a konfiguraci. Mezi tyto údaje spadají následující:

- Údaje o času a datu uložené v paměti CMOS (stálé) systému.

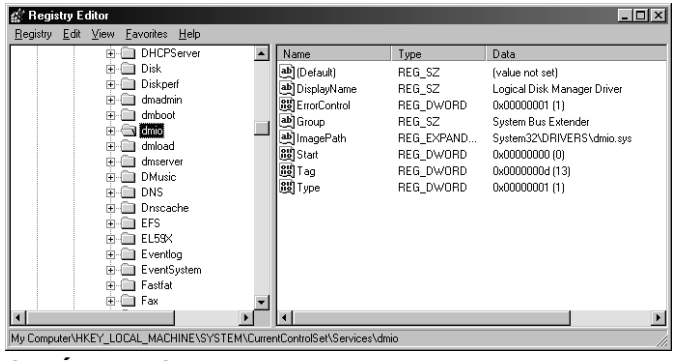
- Typy sběrnic (kupříkladu ISA, PCI, EISA, Micro Channel Architecture [MCA]) na systému a identifikátory jednotlivých zařízení připojených k těmto sběrnicím.
- Počet, velikost a typ diskových jednotek v systému.
- Typy vstupních zařízení připojených k systému.
- Počet a typ paralelních portů konfigurovaných na systému.
- Typy grafických adaptérů v systému.

Tyto údaje se převezmou do interních struktur a později v rámci spouštění se uloží do klíče registru `HKLM\HARDWARE\DESCRIPTION`.

Na systému Windows 2000 zavaděč `Ntldr` dále vymaže obrazovku a ukáže lištu postupu Starting Windows (Spouštění Windows). Tato lišta zůstane prázdná, dokud nezačne `Ntldr` zavádět základní ovladače. (Viz krok 5 v následujícím seznamu.) Pod lištou postupu je zpráva „For troubleshooting and advanced startup options for Windows, press F8" (Řešení potíží nebo upřesnění možností spuštění – stiskněte klávesu F8). Stiskne-li uživatel klávesu F8, nabídne mu systém rozšířenou nabídku spuštění, jež mu umožní vybrat volby jako spuštění poslední známé funkční konfigurace, přechod do nouzového režimu, režimu ladění atd. Na Windows XP a Windows Server 2003 zobrazí zavaděč `Ntldr` obrazovku loga nahrazující lištu postupu.

Jestliže `Ntldr` běží na systému x64 a jádro vybrané volbou příslušné položky z nabídky spuštění je pro x64, přepne `Ntldr` procesor do dlouhého režimu, v němž má slovo nativní délku 64 bitů. Dále začne `Ntldr` zavádět soubory ze spouštěcího svazku, které potřebuje ke spuštění inicializace jádra. Spouštěcí svazek je ten, který odpovídá oddílu, na němž se nachází systémový adresář (většinou `\Windows`) právě spouštěné instalace. Kroky `Ntldr` zahrnují tyto:

1. Zavedení příslušných obrazů jádra a HAL (standardně `Ntoskrnl.exe` a `Hal.dll`). Pokud se `Ntldr` nepodaří některý z těchto souborů zavést, vytiskne zprávu „Windows could not start because the following file was missing or corrupt" (systém Windows nemohl být spuštěn, protože následující soubor chybí nebo je poškozený) a dále pak název souboru.
2. Načtení podregistru `SYSTEM`, `\Windows\System32\Config\System`, aby bylo možné stanovit ovladače zařízení, které je v rámci startování zapotřebí zavést. (Podregistr je soubor obsahující nějaký podstrom registru. Více se o registru dozvíte v kapitole 4).
3. Skenování podregistru `SYSTEM` v paměti a vyhledání všech ovladačů zařízení zaváděných při startování. To jsou ovladače nezbytné ke spuštění systému a v registru jsou označeny hodnotou spuštění `SERVICE_BOOT_START (0)`. Každý ovladač zařízení má pod `HKLM\SYSTEM\CurrentControlSet\Services` určitý podklíč registru. Kupříkladu Services má podklíč nazvaný `Dmio` pro ovladač správce logických disků (Logical Disk Manager), jak jej vidíte na obrázku 5.2. (Podrobný popis položek Services registru najdete v oddílu „Služby" v kapitole 4).
4. Přidání takového ovladače systému souborů, který zodpovídá za implementaci kódu pro typ oddílu (FAT, FAT32 nebo NTFS), na němž se nachází instalační adresář, na seznam ovladačů zaváděných při spouštění. `Ntldr` musí tento ovladač zavést již nyní; pokud by to neučinil, jádro by po ovladačích vyžadovalo, aby se zavedly samy, což je požadavek vytvářející kruhovou závislost.



OBRÁZEK 5.2: Nastavení služby ovladače správce logických disků

5. Zavedení ovladačů při startování, což by měly být pouze ty ovladače, které by podobně jako ovladač systému souborů vytvářely kruhové závislosti, kdyby bylo jejich zavedení požadováno po jádru. `Ntldr` aktualizuje lištu postupu zobrazenou pod textem Starting Windows (Spouštění Windows), čímž indikuje postup zavádění. Tato lišta se posunuje s každým zavedeným ovladačem. (Předpokládá se 80 ovladačů zaváděných při startování – každé úspěšné zavedení posune lištu postupu o 1,25 %.) Je-li ve volbě `Boot.ini` specifikován přepínač `/SOS`, `Ntldr` nezobrazuje lištu postupu, ale názvy souborů jednotlivých startovaných ovladačů. Pamatujte, že v tomto okamžiku se jednotlivé ovladače zavádějí, ale neinicializují – k tomu dochází až později ve spouštěcí sekvenci.
6. Příprava registrů CPU na vykonávání `Ntoskrnl.exe`.

Tato akce ukončuje roli zavaděče `Ntldr` v procesu spouštění. Nyní `Ntldr` zavolá hlavní funkci v `Ntoskrnl.exe`, která zajistí zbytek inicializace systému.

Proces spouštění na architektuře IA64

Tabulka 5.3 uvádí soubory účastnící se procesu spouštění na architektuře IA64. Systémy IA64 odpovídají specifikaci Extensible Firmware Interface (EFI – rozhraní rozšiřitelného firmwaru) definované společnosti Intel. Systém kompatibilní s EFI má firmware, jenž vykonává kód zavaděče spouštění naprogramovaný do stálé paměti RAM (NVRAM) systému instalačním programem Windows. Tento spouštěcí kód načítá obsah také uložený v NVRAM, jenž na platformě IA64 odpovídá obsahu souboru `Boot.ini` na systémech x86 a x64. Úpravu voleb spuštění a přepínačů v paměti NVRAM umožňují jak nástroje EFI společnosti Microsoft vykonatelné v konzole EFI, tak i `Bootcfg.exe`, což je nástroj obsažený ve Windows.

Dále dojde k detekci hardwaru, kdy zavaděč spouštění stanoví pomocí rozhraní EFI počet a typ následujících zařízení:

- Síťové adaptéry,
- grafické adaptéry,
- klávesnice,
- řadiče disků,
- zařízení úložišť.

Podobně jako Ntldr na systémech x86 a x64 následně zavaděč zobrazí nabídku voleb spouštění s volitelnou dobou vypršení. Jakmile je vybráno nějaké spuštění, zavaděč přejde do toho adresáře systémového oddílu EFI, který odpovídá výběru, a zavádí několik dalších souborů nezbytných k pokračování ve startování: Fpswa.efi a Ia64ldr.efi. Specifikace EFI vyžaduje, aby měl systém oddíl určený jako systémový oddíl EFI, který je naformátován systémem souborů FAT a má velikost mezi 100 MB a 1 GB nebo až jedno procento velikosti disku. Každá instalace Windows má podadresář na systémovém oddílu EFI pod EFI\Microsoft. První instalaci se přiřadí složka Winnt50, druhé Winnt50.1 a každá následující instalace má jedinečné číslo indexu, jež je v názvu složky uvedeno za tečkou. Soubor Ia64ldr.efi zodpovídá za zavedení Ntoskrnl.exe, Hal.dll a ovladačů pro spouštění. Poté spouštění pokračuje stejnými kroky jako na architekturách x86 a x64.

TABULKA 5.3: Komponenty procesu spouštění na architektuře IA64

Komponenta	Umístění	Zodpovědnost
Fpswa.efi	EFI\Microsoft\Winnt50.x na systémovém oddílu EFI	Soubor obsahující podporu operací s pohyblivou desetinnou čárkou pro EFI
Ia64ldr.efi	EFI\Microsoft\Winnt50.x na systémovém oddílu EFI	Zavádí Ntoskrnl.exe, Hal.dll a ovladače pro spuštění
Ntoskrnl.exe	\Windows\System32	Inicializuje subsystémy výkonné části a ovladače zařízení pro spouštění a start systému, připravuje systém na běh nativních aplikací a vykonává SMSS.exe
Hal.dll	\Windows\System32	Knihovna DLL v režimu jádra zajišťující pro Ntoksnrl a ovladače potřebné rozhraní přístupu k hardwaru
Správce řízení služeb (Service Control Manager – SCM)	\Windows\System32	Zavádí a inicializuje ovladače zařízení automatického zavádění a služby Windows
SMSS	\Windows\System32	Zavádí subsystém Windows včetně Win32k.sys a Csrss.exe a spouští proces Winlogon
Winlogon	\Windows\System32	Spouští správce řízení služeb (SCM), subsystém místního úřadu zabezpečení (Local Security Authority Subsystem – LSASS) a nabídne dialogové okno interaktivního přihlášení

Inicializace jádra a subsystémů výkonné části

Když Ntldr volá Ntoskrnl, předává datovou strukturu obsahující kopii toho řádku v souboru Boot.ini, jenž představuje vybranou položku nabídky odpovídající danému spouštění, ukazatel na paměťové tabulky, které vygeneroval zavaděč Ntldr k popisu fyzické paměti v systému, ukazatel na paměťovou kopii podregistrů HARDWARE a SYSTEM a ukazatel na seznam ovladačů pro spouštění, které Ntldr zavedl.

Ntoskrnl následně začíná první fázi ze svého dvoufázového inicializačního procesu, označované za *fáze 0* a *fáze 1*. Většina subsystémů výkonné části má nějakou inicializační funkci přebírající parametr identifikující právě probíhající fázi.

Během fáze 0 jsou zakázána přerušení. Smyslem této fáze je vybudovat základní struktury nezbytné pro volání služeb potřebných ve fázi 1. Hlavní funkce Ntoskrnl volá KiSystemStartup, která zase volá HalInitializeProcessor a KiInitializeKernel u jednotlivých CPU. Funkce KiInitializeKernel, pokud běží na spouštěcím procesoru, zajistí celosystémovou inicializaci jádra, jako je inicializace různých datových struktur sdílených všemi procesory. Každá instance KiInitializeKernel pak volá funkci zodpovědnou za řízení fáze 0, ExpInitializeExecutive.

Funkce ExpInitializeExecutive začíná voláním funkce HalInitSystem vrstvy HAL, jež umožní HAL získat řízení systému, než Windows vykonají významnou další inicializaci. Jednou ze zodpovědností HalInitSystem je připravit systémový řadič přerušení jednotlivých procesorů na přerušení a nakonfigurovat přerušení intervalového časovače, jež se používá k účtování času procesorů. (Více se o účtování času CPU dozvíte v oddílu „Účtování kvant“ v kapitole 6.)

Pouze na spouštěcím procesoru vykoná funkce ExpInitializeExecutive ještě jinou inicializaci než volání HalInitSystem. Když HalInitSystem vrátí řízení, funkce ExpInitializeExecutive na spouštěcím procesoru pokračuje zpracováním přepínače /BURNMEMORY v souboru Boot.ini (pokud se tedy nachází na tom řádku souboru Boot.ini, který odpovídá volbě položky nabídky, jak ji zadal uživatel při volbě spouštěné instalace) a vypustí množství paměti zadané tímto příznakem.

Následně funkce ExpInitializeExecutive zavolá fázi 0 inicializačních rutin správce paměti, správce objektů, monitoru bezpečnostních referencí, správce procesů a správce Plug and Play. Tyto komponenty vykonají následující inicializační kroky:

1. Správce paměti zkonstruuje tabulky stránek a interní datové struktury potřebné k zajišťování základních paměťových služeb. Správce paměti také sestaví a rezervuje určitou oblast pro systémovou mezipaměť souborů a vytvoří paměťové oblasti pro stránkovaný a nestránkovaný fond. Ostatní subsystémy výkonné části, jádro a ovladače zařízení využívají tyto dva paměťové fondy k alokování svých datových struktur.
2. Během inicializace správce objektů se definují objekty nezbytné pro konstrukci oboru názvů správce objektů, aby do něj mohly své objekty vkládat ostatní subsystémy. Vytvoří se také tabulka manipulátorů, aby se mohlo začít se sledováním objektů.
3. Monitor bezpečnostních referencí inicializuje objekt typu tokenu a ten pak použije k vytvoření a přípravě prvního tokenu účtu lokálního systému pro přiřazení prvotnímu procesu. (Popis lokálního systémového účtu najdete v kapitole 8.)
4. Správce procesů zajistí většinu své inicializace ve fázi 0, kdy definuje typy objektů procesu a vlákna a vytváří seznam pro sledování aktivních procesů a vláken. Správce procesů rovněž vytváří objekt procesu pro prvotní proces a nazývá jej Idle. Jako svůj poslední krok vytváří správce procesů proces System a systémové vlákno k vykonání rutiny Phase1Initialization. Toto vlákno ale nezačne běžet ihned, protože přerušení jsou stále zakázána.

5. Dojde k fázi 0 inicializace správce Plug and Play, která zahrnuje prostou inicializaci určitého prostředku výkonné části používané k synchronizaci prostředků sběrnic.

Když se řízení vrátí funkci `KiInitializeKernel` na jednotlivých procesorech, pokračuje řízení do cyklu `Idle` (nečinného), který pak způsobí, že systémové vlákno vytvořené v kroku 4 popisu předchozího procesu začne vykonávat fázi 1. (Druhotné procesory čekají s prací na své inicializaci až do kroku 5 fáze 1, jak ji vysvětluje následující seznam.) Fáze 1 se skládá z dále uvedených kroků. Úvodní obrazovka na systémech Windows 2000 zahrnuje lištu postupu a kroky, při nichž se zobrazení lišty aktualizuje, jsou zahrnuty v tomto seznamu:

1. Zavolá se funkce `HalInitSystem`, aby připravila systém na příjem přerušení od zařízení a povolila přerušení.
2. Zavolá se ovladač videa pro zavádění (`\Windows\System32\Bootvid.dll`), který zase zobrazí spouštěcí obrazovku Windows. (Na systémech Windows XP a Windows Server 2003 nabídne tento ovladač stejnou grafiku, jakou již dříve umístil na obrazovku zavaděč `Ntldr`.)
3. Zavolá se inicializace správce napájení.
4. Inicializuje se systémový čas (voláním `HalQueryRealTimeClock`) a pak se uloží jako čas spuštění systému.
5. Na víceprocesorovém systému se inicializují zbývající procesory a začnou svá vykonávání.
6. Lišta postupu se nastaví na 5 procent.
7. Správce objektů vytvoří kořenový adresář oboru názvů (`\`), adresář `\ObjectTypes` a adresář mapování (připojení) názvů zařízení systému DOS (`\??` na Windows 2000 a `\Global??` na Windows XP a Windows Server 2003). Pak vytvoří symbolický odkaz `\DosDevices` ukazující na adresář mapování názvů zařízení pro systém DOS.
8. Zavolá se výkonná část, aby vytvořila své typy objektů včetně semaforu, mutexu, události a časovače.
9. Jádro inicializuje datové struktury plánovače (odesílatele) a tabulku odesílání systémových služeb.
10. Monitor bezpečnostních referencí vytvoří v oboru názvů správce objektů adresář `\Security` a inicializuje auditovací datové struktury, je-li auditování zapnuté.
11. Lišta postupu se nastaví na 10 procent.
12. Zavolá se správce paměti, aby vytvořil objekt sekce a svá systémová pracovní vlákna (která si vysvětlíme v kapitole 7).
13. Do systémového prostoru se připojí tabulky podpory národních jazyků (National Language Support – NLS).
14. Do systémového adresového prostoru se připojí (mapuje) `Ntdll.dll`.
15. Správce mezipaměti (cache) inicializuje datové struktury mezipaměti souborového systému a vytvoří svá pracovní vlákna.
16. Správce konfigurace vytvoří objekt klíče `\Registry` v oboru názvů správce objektů a zkopíruje počáteční data registru předaná zavaděčem `Ntldr` do podregistrů `HARDWARE` a `SYSTEM`.

17. Inicializují se globální datové struktury ovladače systému souborů.
18. Správce Plug and Play zavolá prvek Plug and Play systému BIOS.
19. Lišta postupu se nastaví na 20 procent.
20. Subsystém LPC (lokálního volání procedur) inicializuje objekt typu portu LPC.
21. Pokud byl systém spuštěn s protokolováním (`/BOOTLOG`), inicializuje se protokolovací soubor spouštění.
22. Lišta postupu se nastaví na 25 procent.
23. Nyní dojde k inicializaci správce I/O (vstupů a výstupů). Tato fáze spouštění systému je komplexní a představuje 50 % „postupu", jak jej hlásí informační lišta. Správce I/O považuje každé úspěšné zavedení ovladače za další 2 procenta postupu spouštění. (Pokud se zavádí více než 25 ovladačů, tak se lišta postupu zastaví na 75 %.)

    Správce I/O nejprve inicializuje různé interní struktury a vytvoří typy objektů ovladače a zařízení. Pak zavolá správce Plug and Play, správce napájení a vrstvu HAL, kde začnou různé fáze dynamického zjišťování zařízení a inicializace. (Protože tento proces je složitý a specifický pro konkrétní systém I/O, podrobnosti si necháme až na kapitolu 9.) Následně je inicializován subsystém Windows Management Instrumentation (WMI), který zajišťuje ovladačům zařízení podporu WMI. (Další informace najdete v oddílu „Windows Management Instrumentation" v kapitole 4.) Pak se zavolají všechny startovací ovladače, aby si zajistily své specifické inicializace, a zavedou se a inicializují ovladače zařízení pro spouštění systému. (Podrobnosti o zpracování informací o řízení zavádění ovladačů jsou také popsány v kapitole 9.) Nakonec se v oboru názvů správce objektů vytvoří názvy zařízení systému MS-DOS jako symbolické odkazy.
24. Lišta postupu se nastaví na 75 procent.
25. Spouští-li se počítač v bezpečném režimu, zaznamená se tento fakt do registru.
26. Není-li v registru specificky zakázáno, aktivuje se stránkování kódu v režimu jádra (v `Ntoskrnl` a ovladačích).
27. Lišta postupu se nastaví na 80 procent.
28. Zavolá se správce napájení, aby inicializoval různé struktury správy napájení.
29. Lišta postupu se nastaví na 85 procent.
30. Zavolá se monitor bezpečnostních referencí, aby mohl vytvořit vlákno příkazového serveru (Command Server Thread) komunikující s LSASS. (Více se o tom, jak je v systému Windows vynucováno zabezpečení, dozvíte v oddílu „Komponenty systému zabezpečení" v kapitole 8.)
31. Lišta postupu se nastaví na 90 procent.
32. Posledním krokem je vytvoření procesu subsystému správce relací (Session Manager Subsystem – `SMSS`), jak byl popsán v kapitole 2. `SMSS` zodpovídá za vytvoření prostředí pro uživatelský režim poskytující systému Windows viditelné rozhraní a kroky jeho inicializace si přiblížíme v dalším oddílu.
33. Lišta postupu se nastaví (konečně) na 100 %.

Jako poslední krok předtím, než je inicializace výkonné části a jádra považována za dokončenou, čeká vlákno fáze 1 inicializace na manipulátor procesu správce relací

s dobou vypršení 5 sekund. Dojde-li k ukončení procesu správce relací před uplynutím 5 sekund, systém se zhroutí s řídicím kódem chyby SESSION5_INITIALIZATION_FAILED.

Uplyne-li pětisekundové čekání, předpokládá se úspěšné spuštění správce relací a funkce fáze 1 inicializace zavolá funkci vlákna stránky nula (vysvětlenou v kapitole 7) správce paměti. Toto systémové vlákno se tak stane vláknem stránky nula po zbytek běhu systému.

SMSS, CSRSS a Winlogon

SMSS je jako každý jiný proces v uživatelském režimu, až na dvě výjimky: Zaprvé, Windows považují SMSS za důvěryhodnou součást operačního systému. Zadruhé, SMSS je *nativní* aplikace. Jelikož se jedná o důvěryhodnou součást operačního systému, může SMSS vykonávat i akce, k jakým má oprávnění jen několik málo dalších procesů, jako například vytvářet tokeny zabezpečení. Jelikož je to nativní aplikace, SMSS nepoužívá rozhraní API systému Windows – využívá jen základní rozhraní API výkonné části společně označované za nativní API Windows. SMSS nevyužívá rozhraní API Windows, protože v okamžiku spuštění SMSS ještě není vykonáván subsystém Windows. Ve skutečnosti je jednou z prvních úloh SMSS právě spustit subsystém Windows.

SMSS pak zavolá subsystém výkonné části správce konfigurace, aby mohl dokončit inicializaci registru, který bude obsahovat všechny své klíče. Správce konfigurace je naprogramován tak, že ví, kde na disku jsou základní podregistry uloženy (s výjimkou podregistrů odpovídajících uživatelským profilům). Cesty k zaváděným podregistrům zaznamenává do klíče HKLM\SYSTEM\CurrentControlSet\Control\hivelist.

Hlavní vlákno SMSS vykoná následující kroky inicializace:

1. Vytvoří objekt portu volání LPC (\SmApiPort) a dvě vlákna čekající na klientské požadavky (jako je zavedení nového subsystému nebo vytvoření relace).
2. Definuje symbolické odkazy pro názvy zařízení v systému MS-DOS (např. COM1 a LPT1).
3. Jsou-li instalovány terminálové služby, vytvoří v oboru názvů správce objektů podadresář \Sessions (pro několik relací).
4. Spustí jakýkoli program definovaný v HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\BootExecute. Tato hodnota obvykle obsahuje jeden příkaz ke spuštění Autochk (verze Chkdsk pro okamžik spouštění).
5. Vykoná operace zpožděného přejmenování a odstranění souborů, jak je nařizují klíče HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\PendingFileRenameOperations a HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\PendingFileRenameOperations2.
6. Otevře známé knihovny DLL a vytvoří pro ně objekty sekcí v adresáři \Knowndlls oboru názvů správce objektů. Seznam knihoven DLL považovaných za známé se nachází v klíči HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs a cesta k adresáři, v němž se tyto knihovny DLL nacházejí, je uložena v hodnotě Dlldirectory uvedeného klíče. Informace

o tom, jak se sekce známých knihoven DLL používají během zavádění knihoven DLL, najdete v kapitole 6.

7. Vytvoří dodatečné stránkovací soubory. Konfigurace stránkovacího souboru je uložena pod klíčem `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PagingFiles`.
8. Inicializuje registr. Správce konfigurace sestaví registr zavedením podregistrů `HKLM\SAM`, `HKLM\SECURITY` a `HKLM\SOFTWARE`. I když jsou soubory podregistrů na disku uvedeny v `HKLM\SYSTEM\CurrentControlSet\Control\hivelist`, správce konfigurace je vytvořen tak, aby je hledal v `\Windows\System32\Config`.
9. Vytvoří systémové proměnné prostředí, jak jsou definovány v `HKLM\System\CurrentControlSet\Session Manager\Environment`.
10. Zavede tu část subsystému Windows (`Win32k.sys`), která pracuje v režimu jádra. `SMSS` stanoví umístění `Win32k.sys` a dalších zaváděných komponent zjištěním jejich cest z `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager`. Inicializační kód ve `Win32k.sys` používá k přepnutí obrazovky na rozlišení definované výchozím profilem ovladače videa, takže právě v tomto okamžiku přejde displej z režimu VGA využívaného ovladačem videa pro spouštění do výchozího rozlišení navoleného na systému.
11. Spustí procesy subsystému včetně `Csrss`. (Jak bylo uvedeno v kapitole 2, na Windows 2000 se subsystémy POSIX a OS/2 spouštějí až na explicitní vyžádání.)
12. Spustí přihlašovací proces (`Winlogon`). Kroky spuštění `Winlogon` si popíšeme dále.
13. Vytvoří porty LPC pro zprávy událostí ladění (`DbgSsApiPort` a `DbgUiApiPort`) a vlákna naslouchající na těchto portech.

Pozdržené operace přejmenování souborů

Skutečnost, že jsou spustitelné obrazy a knihovny DLL při svém použití mapovány do paměti, umožňuje aktualizovat základní systémové soubory, jakmile se dokončí spouštění systému Windows. Rozhraní API `MoveFileEx` systému Windows umožňuje specifikovat přesunutí souboru až na okamžik následujícího spouštění. Servisní balíčky a opravy, které musí aktualizovat používané, do paměti mapované soubory, instalují nahrazující soubory na dočasná místa systému a používají API `MoveFileEx` k náhradě jinak používaných souborů. Při použití s příslušnou volbou `MoveFileEx` prostě jen zaznamená příkazy do hodnot `PendingFileRenameOperations` a `PendingFileRenameOperations2` pod `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager`. Tyto hodnoty registru jsou typu `MULTI_SZ` a každá operace je specifikována dvojicí názvů souborů: První název souboru je zdrojovým místem a druhý pak cílovým umístěním. Operace odstranění používají jako cílovou cestu prázdný řetězec. K zobrazení zaregistrovaných zpožděných příkazů přejmenování a odstranění můžete využít utilitu `Pendmoves` ze sídla `www.sysinternals.com`.

Po provedení těchto inicializačních kroků čeká hlavní vlákno v `SMSS` donekonečna na manipulátory procesů `CSRSS` a `Winlogon`. Dojde-li k neočekávanému ukončení jednoho z těchto procesů, `SMSS` způsobí zhroucení systému, protože Windows bez nich nemohou fungovat. (Pokud ve Windows XP a novějších systémech z nějakého důvodu selže `CSRSS`, způsobí zhroucení systému jádro a nikoli `SMSS`.)

`Winlogon` následně zajistí své spouštěcí kroky, jako je vytvoření počátečních objektů stanice oken a pracovní plochy. Je-li v `HKLM\Software\Microsoft\Windows NT\Current`

Version\WinLogon\GinaDLL zadána nějaká knihovna DLL, Winlogon použije tuto knihovnu jako GINA; jinak využije výchozí knihovnu GINA společnosti Microsoft, tedy Msgina (\Windows\System32\Msgina.dll), jež zobrazuje standardní přihlašovací dialogové okno Windows. Winlogon pak vytvoří proces správce řízení služeb (Service Control Manager – SCM –\Windows\System32\Services.exe), jenž zavede všechny služby a zařízení označená k automatickému spuštění, a proces subsystému lokálního ověřování zabezpečení (Local Security Authentication Subsystem – LSASS – \Windows\System32\LSASS.exe). (Další detaily o spouštěcí sekvenci součástí Winlogon a LSASS najdete v oddílu „Inicializace Winlogon" v kapitole 8.)

Jakmile správce SCM inicializuje automaticky spouštěné služby a ovladače a dojde k úspěšnému přihlášení uživatele ke konzole, považuje SCM spuštění za úspěšné. Poslední známá funkční konfigurace registru (jak ji indikuje HKLM\SYSTEM\Select\LastKnownGood) se aktualizuje, aby odpovídala \CurrentControlSet.

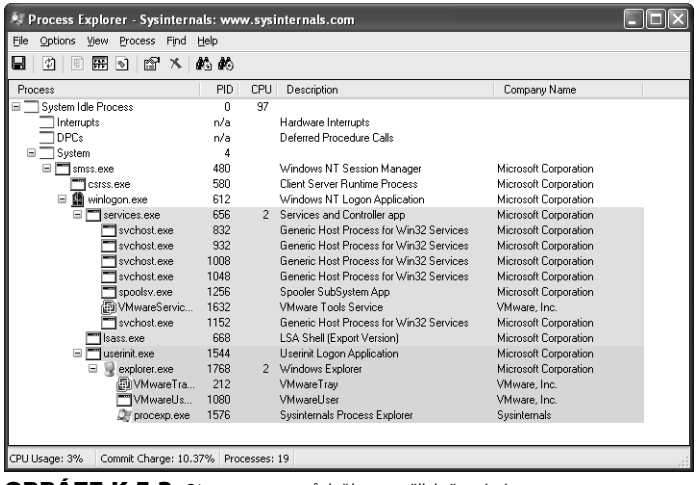
Poznámka Jelikož na neinteraktivních serverech nemusí nikdy dojít k interaktivnímu přihlášení, nemusí se aktualizovat jejich konfigurace LastKnownGood tak, aby odrážela řídicí množinu použitou při úspěšném spuštění.
Definici úspěšného spuštění můžete překrýt nastavením HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ReportBootOk na hodnotu 0, napsáním vlastního programu ověření spuštění, který volá API NotifyBootConfigStatus systému Windows, jakmile je spuštění úspěšné, a zadáním cesty k tomuto ověřovacímu programu do HKLM\System\CurrentControlSet\Control\BootVerificationProgram.

Po spuštění SCM čeká Winlogon na upozornění na interaktivní přihlášení od GINA. Když toto přihlášení obdrží a ověří je (což je proces, s nímž se blíže seznámíme v oddílu „Kroky přihlašování uživatelů" v kapitole 8), zavede Winlogon podregistr z profilu přihlašovaného uživatele a připojí jej k HKCU. Pak nastaví uživatelské proměnné prostředí uložené v HKCU\Environment a upozorní balíčky Winlogon zaregistrované v HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify, že došlo k přihlášení.

Winlogon následně řekne GINA, že má spustit systémové prostředí. V reakci na tento požadavek spustí Msgina vykonatelný soubor nebo soubory specifikované v HKLM\Software\Microsoft\WindowsNT\CurrentVersion\WinLogon\Userinit (přičemž více vykonatelných souborů se vzájemně odděluje čárkami). Tato hodnota ve výchozím stavu ukazuje na \Windows\System32\Userinit.exe. Userinit.exe zajistí toto:

1. Zpracuje uživatelské skripty zadané v HKCU\Software\Policies\Microsoft\Windows\System\Scripts a skripty přihlašování k počítači v HKLM\Software\Policies\Microsoft\Windows\System\Scripts. (Jelikož se skripty počítače spouštějí až po uživatelských skriptech, mohou překrýt uživatelská nastavení.)
2. Specifikují-li zásady skupiny nějakou kvótu uživatelského profilu, spustí \Windows\System32\Proquota.exe, čímž bude vynucována kvóta pro aktuálního uživatele.
3. Spustí prostředí (pokud je jich více, pak jsou oddělená čárkami) specifikované v HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell. Pokud tato hodnota neexistuje, spustí Userinit.exe prostředí (nebo více prostředí) zadané v HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell, kterým je standardně Explorer.exe.

Winlogon pak upozorní registrované poskytovatele sítě, že došlo k přihlášení uživatele. Poskytovatel sítí Microsoft neboli Multiple Provider Router (\Windows\System32\Mpr.dll) obnoví trvalé přiřazení písmen jednotek a tiskáren daného uživatele, jak je uloženo v HKCU\Network resp. HKCU\Printers. Obrázek 5.3 zachycuje strom procesů, jak jej zaznamenává Process Explorer během přihlašování před ukončením Userinit.



OBRÁZEK 5.3: Strom procesů během přihlašování

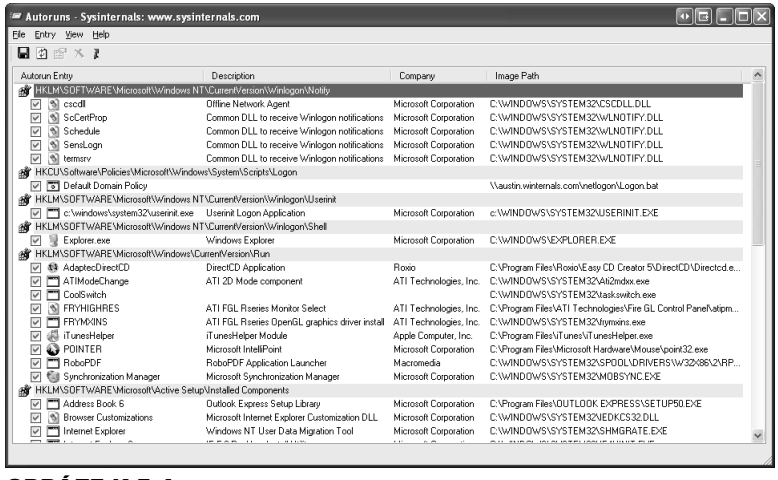
Automaticky spouštěné obrazy

Kromě hodnot registru Userinit a Shell v klíči součásti Winlogon je ještě mnoho dalších míst a adresářů registru, která výchozí komponenty systému kontrolují a zpracovávají při zjišťování automaticky zaváděných procesů během procesu spouštění a přihlašování. Utilita Msconfig (obsažená ve Windows XP a Windows Server 2003 v adresáři \Windows\System32\Msconfig.exe) ukazuje nakonfigurované obrazy v několika místech. Nástroj Autoruns sídla Sysinternals, který si můžete stáhnout z www.sysinternals.com a který je uveden na obrázku 5.4, zkoumá více míst než Msconfig a zobrazuje více údajů o obrazech nastavených na automatický běh. Autoruns standardně zobrazuje pouze ta místa, kde je nastaveno automatické vykonání alespoň jednoho obrazu, po zaškrtnutí položky Include Empty Locations v nabídce View však nástroj Autoruns zobrazuje všechna prozkoumávaná umístění. Nabídka View obsahuje rovněž volby zobrazování informací o dalších typech automaticky zaváděných obrazů, jako jsou služby Windows a doplňky Exploreru (Průzkumníku).

EXPERIMENT: Nástroj Autoruns

Mnoho uživatelů vůbec neví, kolik programů se vykonává jako součást jejich přihlášení. Výrobci originálního vybavení (OEM) často konfigurují své systémy pomocí doplňkových utilit, jež se vykonávají na pozadí s využitím hodnot registru nebo určitých adresářů systému souborů, do nichž se ukládají automaticky spouštěné položky, které nejsou normálně viditelné. Všechny programy nakonfigurované na automatické spouštění na počítači si zobrazíte nástrojem Autoruns ze sídla www.sysinternals.com. Porovnejte si seznam zobrazený utilitou Autoruns s tím, co ukazuje Msconfig (tento nástroj je k dispozici na systémech Windows XP

a Windows Server 2003), a vysledujte rozdíly. Následně se přesvědčte, že rozumíte účelu každého z těchto programů.



OBRÁZE K 5.4: Nástroj Autoruns ze sídla www.sysinternals.com

5.2 Řešení potíží se startem a spouštěním

V tomto oddílu najdete přístupy k řešení potíží, jež se mohou vyskytnout během procesu spouštění Windows a jsou důsledkem poškození obsahu na pevném disku, poškození souborů, chybějících souborů a chyb v ovladačích nezávislých výrobců. Nejprve si popíšeme tři režimy obnovení Windows při potížích se spuštěním: poslední známou funkční konfiguraci, nouzový režim a konzolu pro zotavení. Pak si ukážeme obvyklé problémy při spouštění a přístupy k jejich řešení. Tyto přístupy využívají poslední známou funkční konfiguraci, nouzový režim, konzolu pro zotavení a další nástroje dodávané s Windows.

Poslední známá dobrá konfigurace

Poslední známá funkční konfigurace (Last Known Good – LKG) je užitečný mechanismus pro přestavení systému, který se během spouštění hroutí, zpět do spustitelného stavu. Protože se konfigurační nastavení systému ukládají do klíče `HKLM\System\CurrentControlSet\Control` a konfigurace ovladačů a služeb do klíče `HKLM\System\CurrentControlSet\Services`, změny těchto částí registru mohou učinit systém nespustitelným. Když kupříkladu nainstalujete nějaký ovladač zařízení obsahující chybu, jež způsobuje hroucení systému při startování, můžete během příštího spouštění stisknout klávesu F8 a ze zobrazené nabídky zvolit poslední známou funkční konfiguraci. Systém označí řídicí množinu, kterou dosud používal ke spuštění, za selhavší tím způsobem, že nastaví hodnotu `Failed` klíče `HKLM\System\Select`. Pak změní `HKLM\System\Select\Current` na hodnotu uloženou v `HKLM\System\Select\LastKnownGood`. Rovněž aktualizuje symbolický odkaz `HKLM\System\CurrentControlSet`, aby ukazoval na řídicí množinu `LastKnownGood`. Jelikož se v podklíči `Services` řídicí množiny `LastKnownGood` klíč nového ovladače nevyskytuje, systém se úspěšně spustí.

Stav nouze

Zřejmě nejčastější příčinou nemožnosti spustit systém Windows je to, že nějaký ovladač zařízení zapříčiňuje hroucení systému během sekvence startování/spouštění. Protože se konfigurace hardwaru a softwaru mohou časem měnit, mohou se kdykoli projevit latentní chyby v ovladačích. Systém Windows nabízí správci možnost vyřešit takové potíže pomocí spouštění ve stavu nouze (safe mode). Stav nouze je princip, který si systém Windows vypůjčuje od Consumer Windows – jedná se o spouštěcí konfiguraci sestávající z minimální množiny ovladačů zařízení a služeb. Když využívá pouze ty ovladače a služby, které jsou ke spuštění skutečně nezbytné, vyhýbá se systém Windows zavádění ovladačů třetích výrobců a jiných nedůležitých ovladačů, které mohou způsobit zhroucení.

Při startování Windows stiskněte klávesu F8, čímž vstoupíte do speciální spouštěcí nabídky obsahující i volby spuštění ve stavu nouze. Většinou máte možnost vybrat si ze tří variant: stav nouze, stav nouze s prací v síti a stav nouze se systémem MS-DOS, tedy s příkazovým řádkem. Standardní stav nouze zahrnuje minimální počet ovladačů zařízení a služeb nezbytných k úspěšnému spuštění. Stav nouze s prací v síti přidává síťové ovladače a služby k těm, které zahrnuje standardní nouzový režim. Konečně stav nouze se systémem MS-DOS odpovídá standardnímu nouzovému režimu, jenom Windows v okamžiku, kdy systém aktivuje režim GUI (grafický), spouštějí jako prostředí aplikaci příkazového řádku (`Cmd.exe`), a nikoli Windows Explorer.

Systém Windows zahrnuje ještě čtvrtý nouzový režim – režim obnovení adresářové služby. Ten se liší od předchozích režimů. Režim obnovení adresářové služby (Directory Services Restore) lze použít ke spuštění systému v režimu, kdy je adresářová služba Active Directory řadiče domény offline (vypnutá) a její databáze neotevřená. To vám umožní vykonávat opravné operace na dané databázi nebo ji obnovit ze záložního média. Při spouštění systému v režimu obnovení adresářové služby se zavádějí všechny ovladače a zařízení s výjimkou služby Active Directory. V případě, že se nemůžete přihlásit na systém kvůli poškození databáze Active Directory, vám tento režim umožní narušení odstranit.

Zavádění ovladačů v bezpečném režimu

Jak systém Windows ví, které ovladače zařízení a služby jsou součástí standardního a síťového bezpečného režimu? Odpověď se nachází v klíči registru `HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot`. Ten obsahuje podklíče `Minimal` a `Network`. Každý podklíč obsahuje další podklíče specifikující názvy ovladačů zařízení nebo služeb či skupin ovladačů. Kupříkladu podklíč `vga.sys` identifikuje ovladač zařízení displeje VGA, který je součástí startovací konfigurace. Ovladač zařízení displeje VGA nabízí základní služby všech grafických adaptérů kompatibilních se standardem PC. Systém využívá tento ovladač jako grafický ovladač při nouzovém režimu místo jiného ovladače, který může využívat pokročilé hardwarové prvky, ale zároveň zabraňovat systému ve spuštění. Každý podklíč pod SafeBoot má určitou výchozí hodnotu popisující, co daný podklíč identifikuje; výchozí hodnotou podklíče `vga.sys` je „Driver“.

Podklíč `Boot` file system má jako výchozí hodnotu zadáno „Driver Group“. Když vývojáři navrhují instalační skript ovladače zařízení, mohou také specifikovat, že daný ovladač patří do určité skupiny ovladačů. Skupiny ovladačů definované systé-

mem jsou uvedeny v hodnotě `List` klíče `HKLM\SYSTEM\CurrentControlSet\Control\ServiceGroupOrder`. Vývojář specifikuje ovladač jako člena určité skupiny proto, aby systému Windows řekl, v jakém okamžiku procesu spouštění se má tento ovladač zavést. Hlavním smyslem klíče `ServiceGroupOrder` je definovat pořadí, v jakém se zavádějí skupiny ovladačů; některé typy ovladačů se musejí zavádět před jinými typy ovladačů nebo po nich. Hodnota `Group` pod klíčem registru konfigurace ovladače přiřazuje daný ovladač do nějaké skupiny.

Klíče konfigurace ovladačů a služeb se nacházejí pod `HKLM\SYSTEM\CurrentControlSet\Services`. Podíváte-li se do tohoto klíče, najdete tu klíč `VgaSave` sloužící ovladači zařízení displeje VGA, který je v registru uveden jako člen skupiny Video Save. Všechny ovladače systému souborů, které systém Windows vyžaduje pro přístup k systémové jednotce Windows, se nacházejí ve skupině Boot file system. Má-li systémová jednotka formát NTFS, pak je součástí této skupiny ovladač NTFS. (Hodnotou `Group` pod klíčem `Ntfs` je Boot file system.) Jinak je součástí této skupiny ovladač systému souborů Fastfat (jenž podporuje jednotky FAT12, FAT16 a FAT32 ve Windows). Ostatní ovladače systému souborů jsou součástí skupiny File system, kterou konfigurace standardního a síťového nouzového režimu také zahrnují.

Když spouštíte systém v konfiguraci nouzového režimu, předá startovací zavaděč (`Ntldr`) přidružený přepínač jádru (`Ntoskrnl.exe`) jako parametr příkazového řádku společně s ostatními přepínači, které jste přiřadili spouštěné instalaci v souboru `Boot.ini`. Spouštíte-li nějaký nouzový režim, předává `Ntldr` přepínač `/SAFEBOOT:`. `Ntldr` přidává k `/SAFEBOOT:` jeden nebo více doplňkových řetězců podle toho, jaký typ tohoto režimu zvolíte. V případě standardního nouzového režimu doplňuje `Ntldr` označení MINIMAL a v případě nouzového režimu s prací v síti používá NETWORK. Zavaděč `Ntldr` přidává MINIMAL(ALTERNATESHELL) u Stavu nouze se systémem MS-DOS a DSREPAIR u režimu obnovení adresářové služby.

Jádro Windows skenuje spouštěcí parametry a hledá přepínače bezpečného režimu již velmi brzy v procesu spouštění. Interní proměnnou `InitSafeBootMode` pak nastavuje na hodnotu odrážející přepínače, jež nalezne. Jádro zapisuje hodnotu `InitSafeBootMode` do hodnoty registru `HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Option\OptionValue`, takže komponenty uživatelského režimu, jako např. správce SCM, dokážou stanovit, v jakém režimu spouštění se systém nachází. Navíc platí, že pokud se systém spouští ve Stavu nouze se systémem MS-DOS, jádro nastaví `HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Option\UseAlternateShell` na hodnotu 1. Jádro zaznamená parametry, které mu předal zavaděč `Ntldr`, do hodnoty `HKLM\SYSTEM\CurrentControlSet\Control\SystemStartOptions`.

Když subsystém jádra správce I/O zavádí ovladače zařízení specifikované klíčem `HKLM\SYSTEM\CurrentControlSet\Services`, vykonává správce I/O funkci `IopLoadDriver`. Jakmile správce Plug and Play detekuje nějaké nové zařízení a chce dynamicky zavést jeho ovladač, vykoná funkci `IopCallDriverAddDevice`. Obě tyto funkce volají funkci `IopSafeBootDriverLoad`, než zavedou příslušný ovladač. `IopSafeBootDriverLoad` kontroluje hodnotu `InitSafeBootMode` a stanovuje, zda se má daný ovladač zavést. Pokud je systém kupříkladu spuštěn ve standardním nouzovém režimu, `IopSafeBootDriverLoad` si pod klíčem `Minimal` vyhledá skupinu daného ovladače, máli nějakou. Najde-li tu `IopSafeBootDriverLoad` uvedenou skupinu, indikuje svému volajícímu, že se daný ovladač může zavést. Jinak `IopSafeBootDriverLoad` hledá název

daného ovladače pod klíčem Minimal. Je-li název ovladače uveden jako podklíč, může se daný ovladač zavést. Nedokáže-li funkce IopSafeBootDriverLoad najít podklíče skupiny ovladače ani názvu ovladače, nemůže se daný ovladač zavést. Pokud je systém spuštěn v nouzovém režimu s prací v síti, vykoná funkce IopSafeBootDriverLoad stejná hledání v podklíči Network. Není-li systém spuštěn v nouzovém režimu, IopSafeBootDriverLoad umožní ovladači zavedení.

Existuje tu určitá výjimka související s ovladači, které nouzový režim ze spouštění vyloučí: Ntldr, a nikoli jádro, zavede všechny ovladače s nulovou hodnotou Start v jejich klíčích registru, což specifikuje zavádění ovladačů během startování systému. Ntldr neprověřuje klíč registru SafeBoot, protože předpokládá, že jakýkoli ovladač s nulovou hodnotou Start je nezbytný k úspěšnému spuštění systému. Jelikož zavaděč Ntldr neprověřuje klíč registru SafeBoot a nestanovuje, které ovladače se mají zavést, Ntldr zavede všechny ovladače pro spouštění (a Ntoskrnl je později spustí).

Uživatelské programy s podporou nouzového režimu

Když se během procesu spouštění inicializuje komponenta uživatelského režimu správce řízení služeb (SCM), kterou implementuje Services.exe, prověří SCM hodnotu HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Option\OptionValue a stanoví, zda se systém rozbíhá v nouzovém režimu. Pokud ano, zrcadlí SCM akce funkce IopSafeBootDriverLoad. Přestože SCM zpracovává všechny služby uvedené pod HKLM\SYSTEM\CurrentControlSet\Services, zavádí pouze ty, jejichž názvy příslušný podklíč nouzového režimu specifikuje. Více informací o procesu inicializace SCM najdete v oddílu „Služby“ v kapitole 4.

Userinit (\Windows\System32\Userinit.exe) je další komponentou běžící v uživatelském režimu, která musí vědět, zda se systém rozbíhá v nouzovém režimu. Userinit, což je komponenta inicializující prostředí uživatele při jeho přihlašování, kontroluje HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Option\UseAlternateShell. Je-li tato hodnota zadána, spustí Userinit program specifikovaný v HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell jako uživatelské prostředí místo Explorer.exe. Windows během instalace zapisují do hodnoty AlternateShell program Cmd.exe, takže výchozím prostředím pro nouzový režim se systémem MS-DOS systému Windows se stává právě příkazový řádek. Prostředím je sice příkazový řádek, přesto do něj ale můžete zadat Explorer.exe a spustit si Průzkumník Windows, stejně jako jakýkoli jiný grafický program.

Jak aplikace zjišťují, zda se systém spouští v nouzovém režimu? Voláním funkce GetSystemMetrics(SM_CLEANBOOT) systému Windows. Dávkové skripty, které musí vykonat určité operace v případě, že se systém spouští v nouzovém režimu, hledají proměnnou prostředí SAFEBOOT_OPTION, protože tu systém definuje pouze při svém spouštění v nouzovém režimu.

Protokolování spouštění v nouzovém režimu

Když systém nasměrujete na spouštění v nouzovém režimu, předá zavaděč Ntldr řetězec specifikovaný volbou /BOOTLOG jádru Windows jako parametr společně s parametrem požadujícím nouzový režim. Při inicializaci jádro kontroluje existenci parametru protokolování spouštění, ať už je parametr nouzového režimu použit nebo ne. Jestliže jádro odhalí řetězec protokolování spouštění, začne zaznamenávat

své akce s každým ovladačem zařízení, jehož zavedení zvažuje. Když kupříkladu funkce IopSafeBootDriverLoad řekne správci I/O, že určitý ovladač nemá zavádět, zavolá tento správce IopBootLog, aby byla tato skutečnost zaznamenána. Podobně platí, že po úspěšném zavedení ovladače, který je součástí konfigurace nouzového režimu, funkcí IopLoadDriver, zavolá tato funkce IopBootLog a nechá zavedení daného ovladače zaznamenat. Na protokoly spouštění se pak můžete podívat a zjistit, které ovladače zařízení jsou součástí konfigurace spouštění.

Protože se jádro chce vyhnout modifikacím disku, dokud neproběhne Chkdsk, k čemuž dochází až později v procesu spouštění, nemůže funkce IopBootLog prostě zprávy zapisovat do nějakého protokolového souboru. Místo toho zaznamenává IopBootLog zprávy do hodnoty registru HKLM\SYSTEM\CurrentControlSet\BootLog. Jako první komponenta běžící v uživatelském režimu, která se během spouštění zavádí, vykoná správce relací (\Windows\System32\SMSS.exe) program Chkdsk kontrolující integritu systémových jednotek a následně dokončí inicializaci registru provedením systémového volání NtInitializeRegistry. Jádro činí tuto akci jako náznak, že lze bezpečně otevřít protokolový soubor na disku, což také učiní a zavolá funkci IopCopyBootLogRegistryToFile. Ta vytvoří soubor Ntbtlog.txt v systémovém adresáři Windows (standardně \Windows) a zkopíruje do něj obsah hodnoty registru BootLog. Funkce IopCopyBootLogRegistryToFile rovněž nastavuje pro IopBootLog příznak, který této funkci oznamuje, že nyní může zapisovat přímo do protokolového souboru a nemusí již zaznamenávat zprávy do registru. Následující výstup zachycuje částečný obsah ukázkového protokolu spouštění:

```
Service Pack 1 3 30 2004 14:05:21.500
Loaded driver \WINDOWS\system32\ntoskrnl.exe
Loaded driver \WINDOWS\system32\hal.dll
Loaded driver \WINDOWS\system32\KDCOM.DLL
Loaded driver \WINDOWS\system32\BOOTVID.dll
Loaded driver ACPI.sys
Loaded driver \WINDOWS\System32\DRIVERS\WMILIB.SYS
Loaded driver pci.sys
Loaded driver isapnp.sys
Loaded driver intelide.sys
Loaded driver \WINDOWS\System32\DRIVERS\PCIIDEX.SYS
Loaded driver MountMgr.sys
Loaded driver ftdisk.sys
Loaded driver dmload.sys
Loaded driver dmio.sys Microsoft (R) Windows 2000 (R) Version 5.0 (Build
2195)
 2 11 2000 10:53:27.500
Loaded driver \WINNT\System32\ntoskrnl.exe
Loaded driver \WINNT\System32\hal.dll
Loaded driver \WINNT\System32\BOOTVID.DLL
Loaded driver ACPI.sys
Loaded driver \WINNT\System32\DRIVERS\WMILIB.SYS
Loaded driver pci.sys
Loaded driver isapnp.sys
Loaded driver compbatt.sys
Loaded driver \WINNT\System32\DRIVERS\BATTC.SYS
Loaded driver intelide.sys
```

```
Loaded driver \WINNT\System32\DRIVERS\PCIIDEX.SYS
Loaded driver pcmcia.sys
Loaded driver ftdisk.sys
Loaded driver Diskperf.sys
Loaded driver dmload.sys
Loaded driver dmio.sys
§
Did not load driver \SystemRoot\System32\Drivers\lbrtfdc.SYS
Did not load driver \SystemRoot\System32\Drivers\Sfloppy.SYS
Did not load driver \SystemRoot\System32\Drivers\i2omgmt.SYSDid not load
driver Media
                    Control Devices
Did not load driver Communications Port
Did not load driver Audio Codecs
§
```


Konzola pro zotavení

Stav nouze je uspokojivá možnost na systémech, které náhle nelze spustit kvůli tomu, že během spouštěcí sekvence dochází ke zhroucení nějakého ovladače zařízení. V určitých případech ale spuštění v nouzovém režimu rozběh systému nezajistí. Když je kupříkladu onen ovladač, jenž zabraňuje systému ve spuštění, členem skupiny Safe, tak selže i spouštění v bezpečném režimu. Další situace, kdy nouzový režim nepomůže se spuštěním systému, nastává, když spuštění systému zabraňuje nějaký ovladač jiného tvůrce, například antivirový program zaváděný při startování. (Startovací ovladače se zavádějí bez ohledu na to, zda je systém v nouzovém režimu.) Další situace, kdy selže spouštění v nouzovém režimu, nastávají, když se poškodí nějaký systémový modul nebo soubor kritického ovladače zařízení, který je součástí konfigurace nouzového režimu, nebo je-li poškozen hlavní spouštěcí záznam (Master Boot Record – MBR) disku. Tyto potíže můžete vyřešit pomocí konzoly pro zotavení (Recovery Console) systému Windows. Konzola pro zotavení vám umožňuje spustit omezené prostředí příkazového řádku z CD Windows nebo spouštěcích disků a instalaci opravit, aniž byste ji museli využívat ke spuštění systému.

Při spouštění systému z CD Windows nebo spouštěcích disket nakonec uvidíte obrazovku nabízející vám buď instalaci Windows, nebo opravu existující instalace. Vyberete-li opravu instalace, systém vás vyzve k vložení CD Windows (není-li zatím v jednotce CD systému) a pak k výběru jedné ze dvou možností opravy: spuštění konzoly pro zotavení nebo inicializaci procesu opravy v nouzi. Stisknete-li na uvítací obrazovce instalace Windows klávesu F10, přeskočíte možnosti nabídky a dostanete se přímo do konzoly pro zotavení.

Při spuštění konzoly pro zotavení se vám nabídne k výběru seznam instalací třídy Windows NT a Windows, který konzola vytvořila skenováním pevných disků počítače. Jakmile vyberete jeden systém, vyzve vás konzola k zadání hesla účtu Administrator, čímž se k instalaci přihlásíte jako správce. Po úspěšném přihlášení vás systém přenese do příkazového řádku podobného prostředí systému MS-DOS. Příkazová množina je flexibilní a dovoluje vám provádět jednoduché operace se soubory (jako je kopírování, přejmenování a odstranění), aktivovat a deaktivovat služby a ovladače a dokonce i opravovat záznamy MBR a spouštěcí záznamy. Konzola pro zotavení vám neumožní přístup k jiným adresářům než kořenovým, k systémovému adresáři instalace, k níž

jste přihlášeni, a k adresářům na výměnných jednotkách, jako jsou CD a diskety, pokud to tedy nastavení lokálních zásad zabezpečení, jak jsou uložena v podregistru SECURITY registru příslušné instalace, nezakazují. Tím je zajištěna určitá úroveň zabezpečení dat, k nimž nemusí mít administrátor obvykle přístup. Toto omezení můžete překonat pomocí editoru zásad místního zabezpečení (Local Security Policy – secpol.msc), v němž lze konfigurovat nastavení konzoly pro zotavení ve složce Security Options místních zásad při normálně spuštěném systému.

Konzola pro zotavení používá k vykonávání operací I/O podpory příkazů jako Cd, Rename a Move nativní rozhraní systémových volání Windows. Příkazy Enable a Disable, jež vám dovolují změnit režimy spouštění ovladačů zařízení a služeb, však fungují jinak. Když kupříkladu řeknete konzole pro zotavení, že chcete zakázat nějaký ovladač zařízení, „sáhne" do klíče Services dané instalace a upraví hodnotu Start klíče specifikovaného ovladače, kterou změní na SERVICE_DISABLED. Při dalším spouštění této instalace se příslušný ovladač zařízení nezavede. (Konzola pro zotavení zavádí rovněž podregistr SYSTEM [\Windows\System32\Config\System] instalace, k níž se přihlašujete. Tento podregistr obsahuje informace uložené v klíči registru HKLM\SYSTEM\CurrentControlSet\Services.)

Při spouštění z CD Windows nebo spouštěcích disket máte v okamžiku, kdy vám systém nabídne možnost nainstalovat nebo obnovit Windows, již spuštěnou kopii jádra Windows včetně všech potřebných podpůrných ovladačů zařízení (kupříkladu ovladače NTFS nebo FAT, ovladače SCSI, ovladač grafiky). Na systémech x86 je spouštění z CD řízeno souborem Txtsetup.sif v adresáři I386 na CD systému Windows; tento soubor obsahuje direktivy identifikující soubory, které se mají zavést, a také jejich umístění na CD. Podobně jako při spouštění Windows z pevného disku je prvním programem uživatelského režimu, který jádro vykoná, správce relací (Session Manager – SMSS.exe) umístěný ve složce I386\System32. Správce relací používaný instalačním programem Windows se liší od správce relací ve standardní instalaci. Ona první komponenta vám nabídne nabídky umožňující nainstalovat nebo opravit Windows a nabídku dotazující se na typ opravy, kterou chcete provést. Pokud systém Windows instalujete, právě správce relací je tou komponentou, jež vás provádí volbou oddílu k instalaci a jež kopíruje soubory na pevný disk.

Když spouštíte konzolu pro zotavení, nahraje se správce relací a spustí dva ovladače zařízení, které konzolu pro zotavení implementují: Spcmdcon.sys a Setupdd.sys. Spcmdcon.sys nabízí interaktivní příkazový řádek a vykonává zpracování vysokoúrovňových příkazů. Setupdd.sys je podpůrný ovladač, který nabízí Spcmdcon.sys sadu funkcí umožňujících spravovat diskové oddíly, zavádět podregistry a zobrazovat a spravovat grafický výstup. Setupdd.sys také komunikuje s diskovými ovladači při správě diskových oddílů a využívá základní podporu grafiky, vestavěnou do jádra Windows, k zobrazování zpráv na monitoru.

Jakmile zvolíte instalaci, k níž se chcete přihlásit, a konzola pro zotavení přijme vaše heslo, musí konzola váš pokus o přihlášení ověřit. To vše v situaci, kdy subsystém zabezpečení Windows dané instalace neběží. Proto musí sama konzola pro zotavení stanovit, zda vaše heslo odpovídá účtu Administrator daného systému. Prvním krokem konzoly pro zotavení v tomto procesu je využití Setupdd.sys k zavedení podregistru správce účtů zabezpečení (Security Accounts Manager – SAM) dané instalace, který ukládá informace o heslech. Podregistr SAM se nachází

v \Windows\System32\Config\Sam. Po zavedení tohoto podregistru z disku vyhledá konzola pro zotavení systémový klíč v registru dané instalace a použije jej k dešifrování paměťové kopie SAM. Šifrování podregistru správce SAM je prvek zavedený ve Windows NT 4 Service Pack 3, který doplňuje ochranu před dosovými programy zjišťování hesel, jež se snaží číst hesla přímo ze souboru podregistru.

Konzola pro zotavení (Spcmdcon.sys) následně vyhledá v SAM heslo účtu Administrator a v posledním kroku ověření použije konzola pro zotavení hešovací algoritmus MD5 – tentýž, jaký používá přihlašovací proces systému Windows – k hešování zadaného hesla a porovnání výsledného kódu s heslem uloženým správcem SAM. Zjistí-li konzola pro zotavení shodu, systém vás považuje za přihlášené. Pokud konzola pro zotavení shodu nepotvrdí, systém vám odmítne přístup ke konzole pro zotavení.

Řešení obvyklých potíží se spouštěním

Tento oddíl shrnuje potíže, jež se mohou projevit během procesu spouštění, popisuje jejich symptomy, příčiny a přístupy k jejich řešení. Abyste dokázali lépe najít problém, s nímž jste se setkali, jsou uspořádány podle místa, kde k nim při startování/spouštění dochází.

Poškození záznamu MBR

- **Příznaky** Systém s poškozeným hlavním spouštěcím záznamem (Master Boot Record – MBR) vykoná samokontrolu po zapnutí (power-on self test – POST) systému BIOS, zobrazí informace o verzi BIOSu a značce OEM, přejde zpět na černou obrazovku a pak se zastaví. V závislosti na typu poškození, jaké záznam MBR utrpěl, můžete spatřit jednu z následujících zpráv: „Invalid Partition Table“ (neplatná tabulka oddílů), „Error Loading Operating System“ (chyba při zavádění operačního systému) nebo „Missing Operating System“ (chybějící operační systém).
- **Příčina** Záznam MBR se může poškodit díky chybám pevného disku, narušení disku v důsledku chyby nějakého ovladače při běhu Windows nebo úmyslným poškozením způsobeným virem.
- **Řešení** Spusťte konzolu pro zotavení (Recovery Console) a vykonejte příkaz fixmbr. Ten nahradí vykonatelný kód v MBR. Bohužel ale neopraví tabulku oddílů. Jedinou možností, jak obnovit poškozenou tabulku diskových oddílů, je použití záložní kopie nebo speciálního nástroje opravy poškození pevného disku.

Poškození spouštěcího sektoru

- **Příznaky** Poškození spouštěcího sektoru může vypadat jako porušení záznamu MBR, protože systém se zablokuje na obrazovce POST systému BIOS. Můžete také vidět zprávy „A disk read error occurred“ (došlo k chybě při čtení z disku), „NTLDR is missing“ (chybí zavaděč NTLRD) nebo „NTLDR is compressed“ (zavaděč NTLDR je zkomprimován).
- **Příčina** Spouštěcí sektor se může poškodit díky chybám pevného disku, narušení disku v důsledku chyby nějakého ovladače při běhu Windows nebo úmyslným poškozením způsobeným virem.
- **Řešení** Spusťte konzolu pro zotavení a vykonejte příkaz fixboot. Ten přepíše spouštěcí sektor jednotky, kterou zadáte. Uvedený příkaz byste měli vykonat na systémové i spouštěcí jednotce, pokud jsou rozdílné.

Špatná konfigurace souboru Boot.ini

- **Příznak** Po testu POST systému BIOS uvidíte zprávu začínající „Windows could not start because of a computer disk hardware configuration problem“ (systém Windows nemohl být spuštěn kvůli problému s konfigurací diskového hardwaru počítače), „Could not read from selected boot disk“ (nebylo možné číst z vybraného spouštěcího disku) nebo „Check boot path and disk hardware“ (prověřte spouštěcí cestu a diskový hardware).
- **Příčina** Soubor `Boot.ini` byl odstraněn, je poškozen nebo se již neodkazuje na spouštěcí svazek, protože doplněním nějakého oddílu došlo ke změně názvu ARC (Advanced RISC Computing) daného svazku.
- **Řešení** Spusťte konzolu pro zotavení a vykonejte „bootcfg /rebuild“. Tento příkaz způsobí, že konzola pro zotavení projde jednotlivými svazky a bude hledat instalaci systému Windows. Jakmile odhalí nějakou instalaci, zeptá se vás, zda ji má přidat do souboru `Boot.ini` jako volbu spouštění a jaký název v nabídce spouštění jí má přiřadit.

Poškození systému souborů

- **Příznaky** Existuje několik způsobů, jakými se může projevit poškození systémových souborů, což zahrnuje spustitelné soubory, knihovny DLL a ovladače. Jednou z možností je zpráva na prázdné obrazovce po testu POST systému BIOS říkající, že „Windows could not start because the following file is missing or corrupt“ (systém Windows nemohl být spuštěn, protože chybí nebo je poškozen následující soubor) s dále uvedeným názvem příslušného souboru a požadavkem na jeho přeinstalování. Další možností je zhroucení s modrou obrazovkou během spouštění říkající „STOP: `0xC0000135` {Unable to Locate Component}“ (nebylo možné najít komponentu).
- **Příčiny** Svazek, na němž se daný systémový soubor nachází, je poškozen nebo došlo k odstranění či poškození jednoho nebo více systémových souborů.
- **Řešení** Spusťte konzolu pro zotavení a vykonejte příkaz `chkdsk`. `Chkdsk` se pokusí opravit poškozený svazek. Pokud nenahlásí žádné problémy, obstarejte si záložní kopii příslušného systémového souboru. Podívejte se do adresáře `\Windows\System32\DllCache`, kam si systém Windows ukládá kopie mnoha systémových souborů pro přístup součásti ochrany souborů systému Windows (Windows File Protection – viz doplněk). Nemůžete-li tu najít kopii příslušného souboru, podívejte se po ní na jiném systému na síti. Nezapomínejte, že záložní soubor musí být ze stejného servisního balíčku nebo opravy, jako ten nahrazovaný.

V některých případech dojde k odstranění nebo poškození více systémových souborů, takže proces opravy pak může představovat několik spouštění a jejich selhání, jak budete soubory jednotlivě doplňovat. Myslíte-li si, že je poškození systémových souborů rozsáhlé, zvažte obnovu systému ze záložního obrazu, jaký generuje například Automated System Recovery – ASR (Automatické obnovení systému). Když spustíte Backup (nachází se ve složce Accesories – System Tools v nabídce tlačítka Start), můžete vygenerovat záložní obraz ASR zahrnující všechny soubory na systémovém a spouštěcím svazku plus disketu s uloženými údaji o sys-

témových discích a svazcích. Chcete-li obnovit systém z ASR, spusťte jej z instalačního média Windows a po výzvě stiskněte klávesu F2.

Nemáte-li zálohu, z níž by bylo možné systém obnovit, máte ještě možnost vykonat opravnou instalaci Windows: Spusťte systém z instalačního média a postupujte podle průvodce, jako byste vykonávali novou instalaci. Průvodce se vás zeptá, zda chcete provést opravu nebo čistou instalaci. Jakmile mu oznámíte, že se jedná o opravu, přeinstaluje instalační program všechny systémové soubory a nechá vaše data aplikací a nastavení registru beze změn.

Ochrana souborů systému Windows

Kromě své role jako interaktivního přihlašovacího rozhraní a správce relací implementuje součást `Winlogon` také ochranu souborů systému Windows (Windows File Protection – WFP). WFP, jež je implementována ve dvou knihovnách DLL, konkrétně `\Windows\System32\Sfc.dll` a `\Windows\System32\Sfc_os.dll`, monitoruje změny klíčových ovladačů, spustitelných souborů a knihoven DLL v několika adresářích zahrnujících většinu adresářů pod `\Windows`. K tomu využívá nativní verzi API `ReadDirectoryChangesW`. Když WFP spatří, že došlo ke změně nějakého systémového souboru uvedeného ve `\Windows\System32\Sfcfiles.dll` (k zobrazení souborů zaznamenaných v `Sfcfiles.dll` můžete použít nástroj `Strings` ze sídla `www.sysinternals.com`), prověří, zda je daný soubor digitálně podepsán společností Microsoft (to je proces, s nímž se blíže seznámíte v oddílu „Instalace ovladačů" v kapitole 9). Je-li daný soubor digitálně podepsán společností Microsoft, WFP tuto změnu povolí a soubor zkopíruje do svého záložního adresáře. Standardně se tento záložní adresář nachází v `\Windows\System32\DllCache`, tento údaj lze ale překrýt definováním hodnoty registru `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\SFCDllCacheDir`. Opravy a servisní balíčky vždy instalují systémové soubory podepsané společností Microsoft.

Není-li důsledkem modifikace soubor, který není podepsán společností Microsoft, WFP nahradí změnu záložní verzí daného souboru z podadresáře DLLCache. Nedokáže-li v tomto adresáři komponenta `Winlogon` nalézt příslušnou záložní verzi, podívá se také do cesty síťové instalace, když byl systém instalován ze sítě, nebo na instalační médium (k jehož vložení vás vyzve), jestliže byl instalován z lokálního média.

Poškození systémového podregistru

- **Příznaky** Chybí-li nebo je-li poškozen podregistr `System` (který je popsán společně se soubory podregistrů v oddílu „Registr" v kapitole 4), zavaděč `NTLDR` zobrazí na prázdné obrazovce po testu POST systému BIOS zprávu „Windows could not start because the following file is missing or corrupt: `\WINDOWS\SYSTEM32\CONFIG\SYSTEM`" (systém Windows nebylo možné spustit, protože chybí nebo je poškozen následující soubor).
- **Příčiny** Podregistr `System`, který obsahuje konfigurační informace nezbytné pro spuštění systému, je poškozen nebo vymazán.
- **Řešení** Spusťte konzolu pro zotavení a vykonejte příkaz `chkdsk` na spouštěcím svazku, čímž opravíte jeho možné chyby. Není-li problém vyřešen, obstarejte si záložní kopii podregistru `System`. Pokud jste provedli zálohování ASR systému nebo použili nástroj Windows Backup k zálohování stavu systému (což je jedna z voleb uživatelského rozhraní této utility), jsou kopie podregistrů z posledního

zálohování uloženy v adresáři \Windows\Repair. Stačí tedy zkopírovat soubor s názvem System do adresáře \Windows\System32\Config.

Pracujete-li na systému Windows XP s aktivní obnovou systému (System Restore – popsáno v kapitole 12), často máte možnost využít novější zálohu podregistrů, včetně podregistru System, z nejnovějšího bodu obnovení. Z konzoly pro zotavení však nemusíte mít možnost přistupovat k adresáři s uloženými body obnovení, jímž je \System Volume Information. Verze konzoly pro zotavení z Windows XP Service Pack 1 umožňují přístup k tomuto adresáři, to ale neplatí pro starší verze, není-li to přímo zadáno v zásadách lokálního zabezpečení daného systému. Toto omezení můžete v případě potřeby překrýt pomocí editoru zásad místního zabezpečení, kterým změníte nastavení konzoly pro zotavení, jak již bylo popsáno. Pro získání přístupu k dalším adresářům můžete rovněž využít nástroje jiných výrobců. Dokážete-li přistupovat k adresářům bodů obnovení, můžete pro proniknutí k jejich podregistrům využít následující postup:

1. Přejděte do adresáře, jehož název začíná "_restore" a který se nachází pod adresářem \System Volume Information spouštěcího svazku.
2. Vyhledejte podadresář RP s nejvyšší číslem jako příponou (kupříkladu „RP173“).
3. Zkopírujte soubor nazvaný _REGISTRY_MACHINE_SYSTEM z podadresáře snapshot do \Windows\System32\Config\System.
4. Restartujte počítač.

Další možností je pokusit se opravit poškození pomocí nástroje Microsoft ChkReg. Ten se pokouší automaticky opravit narušení registru a spouští se z instalačních disket Windows XP. Tento nástroj a instrukce k jeho použití najdete na adrese http://www.microsoft.com/downloads/details.aspx?displaylang=en&familyid=56d3c201-2c68-4de8-9229-ca494362419c.

Pokud jste systém nikdy nezálohovali, nemáte přístup k bodům obnovení a nástroj ChkReg nedokáže poškození opravit (nebo vám příslušné podregistry chybějí), pak můžete jako poslední pokus použít kopii podregistru System uloženou v adresáři \Windows\Repair. Instalační program Windows vytváří kopii podregistru System po dokončení instalace, takže tím ztratíte všechny od té doby zadané změny konfigurace systému a instalace ovladačů zařízení.

Zhroucení nebo zastavení po úvodní obrazovce

- **Příznaky** Do této kategorie spadají problémy, jež se vyskytují po zobrazení úvodní obrazovky systému Windows, zobrazení pracovní plochy nebo po přihlášení. Mohou se projevovat modrou obrazovkou zhroucení nebo zablokováním, kdy je celý systém „zatuhlý“, popřípadě se sice pohybuje kurzor myši, jinak ale systém nereaguje.
- **Příčiny** Tyto potíže jsou téměř vždy důsledkem chyby v nějakém ovladači zařízení, mohou být ale také výsledkem poškození jiného podregistru registru, než je podregistr System.
- **Řešení** O nápravu problému se můžete pokusit v několika krocích. Nejprve zkuste použít poslední známou funkční konfiguraci (LKG). Jak jsme si říkali dříve v této kapitole a v oddílu „Služby“ v kapitole 4, LKG se skládá z řídicí množiny registru

použité k poslednímu úspěšnému spuštění systému. Protože řídicí sada obsahuje základní konfiguraci systému a databázi registrace ovladačů zařízení a služeb, můžete se použitím té verze, jež nezahrnuje změny nebo nově instalované ovladače či služby, vyhnout zdroji problémů. Přístup k poslední známé funkční konfiguraci získáte stiskem klávesy F8 na počátku procesu spouštění. Zobrazí se stejná nabídka, jejímž prostřednictvím můžete přejít do nouzového režimu.

Jak tu již bylo řečeno, při spuštění LKG systém ukládá řídicí množinu, které se vyhýbáte, a označuje ji za selhávající konfiguraci. Tuto selhávající řídicí množinu můžete využít v případech, kdy se LKG podaří systém spustit, ke zjištění příčiny selhávání systému. Stačí si vyexportovat obsah aktuální řídicí množiny úspěšného spuštění a selhávající řídicí množinu do souborů `.reg`. K tomu využijte funkci exportu nástroje `Regedit`, jak ji najdete v nabídce File (nebo pod nabídkou Registry, pracujete-li na Windows 2000):

1. Spusťte `Regedit` a vyberte `HKLM\System\CurrentControlSet`.
2. Zvolte Export z nabídky File a zadejte uložení do souboru nazvaného `dobry.reg`.
3. Otevřete si `HKLM\System\Select`, načtěte hodnotu `Failed` a vyberte podklíč nazvaný `HKLM\System\Control`*XXX*, kde *XXX* představuje hodnotu `Failed`.
4. Exportujte obsah této řídicí množiny do souboru `spatny.reg`.
5. Pomocí aplikace Wordpad (najdete ji ve složce Accessories v nabídce tlačítka Start) nahraďte všechny výskyty „`CurrentControlSet`“ v souboru `dobry.reg` za „ControlSet“.
6. Pomocí WordPadu změňte všechny výskyty „`Control`*XXX*“ (kde *XXX* nahradíte hodnotou `Failed` řídicí množiny) v souboru `spatny.reg` za „`ControlSet`“.
7. Spusťte `Windiff` ze sady podpůrných nástrojů a oba soubory porovnejte.

Rozdílů mezi selhavší řídicí množinou a tou dobrou může být mnoho, takže své zkoumání byste měli zaměřit na změny pod klíčem `Control` a také pod klíči `Parameters` ovladačů a služeb zaregistrovaných v podklíči `Services`. Nevšímejte si změn podklíčů `Enum` klíčů registru ovladačů ve větvi Services řídicí množiny.

Je-li problém, s nímž bojujete, způsoben nějakým ovladačem nebo službou, která byla na systému od posledního úspěšného spuštění, tak ani LKG nezajistí spuštění systému. Podobně platí, že LKG nepomůže ani v situacích, kdy došlo k problematické změně konfigurace mimo řídicí množinu nebo před posledním úspěšným spuštěním. V takových případech je další možností vyzkoušet nouzový režim (popsaný již dříve v této kapitole). Pokud se systém úspěšně spustí v nouzovém režimu a vy víte, že nějaký konkrétní ovladač způsobuje selhání normálního spuštění, můžete jej nyní deaktivovat pomocí správce zařízení (je přístupný z karty Hardware ovládacího panelu System). Vyberte příslušný ovladač a z místní nabídky pak zvolte Disable. Pracujete-li na systému Windows XP nebo Windows Server 2003, daný ovladač jste nedávno aktualizovali a myslíte-li si, že tato aktualizace zavedla na systém chybu, můžete pomocí správce zařízení zadat také změnu ovladače na jeho původní verzi. Chcete-li takto postupovat, tak nejprve na daný ovladač poklepejte, čímž si otevřete dialogové okno jeho vlastností, a pak stiskněte tlačítko Roll Back Driver (Vrátit změny ovladače) na kartě Drivers (Ovladač).

Na systémech Windows XP s aktivní možností obnovení systému je další možností při selhání LKG vrátit zpět stav systému (jak jej definuje obnovení systému) na nějaký předchozí bod v čase. Nouzový režim detekuje existenci bodů obnovení, a jakmile je najde, dotáže se vás, zda se chcete přihlásit k dané instalaci a zajistit manuální diagnostiku a opravu, nebo zda chcete spustit System Restore Wizard (Průvodce obnovením systému). Využití prvku obnovení systému k opětnému zajištění spustitelnosti systému je zajímavá volba v situaci, kdy znáte příčinu problému a chcete opravu provést automaticky nebo když příčinu sice neznáte, ale nechcete ztrácet čas jejím přesným zjišťováním.

Není-li obnovení systému využitelnou volbou nebo chcete-li stanovit příčinu hroucení při normálním spouštění, když se zároveň systém bez potíží rozbíhá v nouzovém režimu, pokuste se získat protokol spouštění neúspěšného rozběhu systému – stiskem F8 si zobrazte speciální nabídku spouštění a zvolte možnost protokolování spouštění. Jak již bylo v této kapitole vysvětleno, správce relací (`\Windows\System32\SMSS.exe`) ukládá protokol spouštění do souboru `\Windows\ntbtlog.txt`, kde je záznam ovladačů zařízení, které systém zaváděl a které se rozhodl nezavádět. Dojde-li tedy ke zhroucení nebo zablokování až po inicializaci správce relací, můžete získat protokol spouštění. Když pak systém spustíte v nouzovém režimu, přidá nové položky k existujícímu spouštěcímu protokolu. Vyjměte ty části protokolovacího souboru, jež odpovídají selhavšímu pokusu spouštění v bezpečném režimu, do samostatných souborů. Odstraňte řádky obsahující text „Did not load driver“ (nebyl zaveden ovladač) a pak soubory porovnejte nějakým nástrojem textového porovnávání, jako je např. `Windiff`. Postupně deaktivujte ovladače zaváděné během normálního spouštění, nikoli však při spouštění v nouzovém režimu, až dokud se systém úspěšně nerozběhne. (Pak zase povolte ty ovladače, které za problém nebyly odpovědné.)

Nemůžete-li získat protokol z normálního spouštění (kupříkladu proto, že se systém hroutí před inicializací správce relací), hroutí-li se systém také během spouštění v nouzovém režimu nebo když porovnání protokolů normálního spouštění a spouštění v nouzovém režimu neodhalí žádné významné rozdíly (když se kupříkladu ovladač způsobující zhroucení systému v normálním spouštění zavádí až po inicializaci správce relací), pak ještě můžete vyzkoušet Driver Verifier ve spojení s analýzou výpisu zhroucení. (Více se o těchto tématech dozvíte v kapitole 14.)

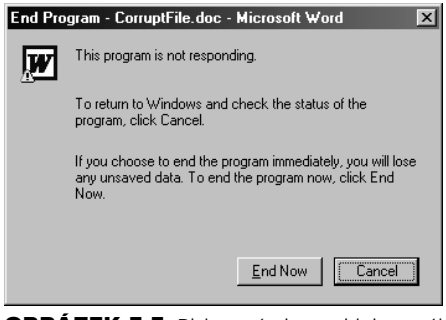
5.3 Vypnutí

Je-li někdo přihlášen a určitý proces iniciuje voláním funkce `ExitWindowsEx` vypnutí systému Windows, odešle se zpráva `CSRSS` instruující subsystém v tom smyslu, že má zajistit vypnutí. `CSRSS` zase převezme identitu volajícího a odešle zprávu systému Windows skrytému oknu vlastněnému komponentou `Winlogon`, v níž říká, že má zajistit vypnutí systému. `Winlogon` následně převezme identitu aktuálně přihlášeného uživatele (který může, ale nemusí mít stejný kontext zabezpečení jako uživatel, jenž vypnutí systému inicioval) a zavolá funkci `ExitWindowsEx` s určitými speciálními interními příznaky. Toto volání znovu způsobí odeslání zprávy `CSRSS` požadující vypnutí systému.

Tentokrát však vidí subsystém `CSRSS` příchozí požadavek od `Winlogon` a v cyklu prochází všemi procesy v dané relaci přihlášení interaktivního uživatele (opět se

nejedná o uživatele, který vypnutí požadoval) v pořadí opačném vzhledem k jejich *úrovni vypnutí* (shutdown level). Voláním funkce `SetProcessShutdownParameters` může proces zadávat úroveň vypnutí indikující systému, kdy se má ukončit vzhledem k ostatním procesům. Platné úrovně vypnutí jsou v rozsahu od 0 do 1023 a výchozí úrovní je 640. Kupříkladu Explorer (Průzkumník) nastavuje svou úroveň vypnutí na hodnotu 2 a správce úloh specifikuje 1. Jednotlivým vláknům s cyklem zpráv Windows každého procesu, který vlastní okno nejvyšší úrovně, odešle `Csrss` zprávu `WM_QUERYENDSESSION`. Pokud vlákno vrátí `TRUE`, může vypínání systému pokračovat. `Csrss` pak takovému vláknu odesílá zprávu `WM_ENDSESSION` požadující jeho ukončení. `Csrss` čeká na ukončení vlákna po dobu zadanou v `HKCU\Control Panel\Desktop\HungAppTimeout` (výchozím údajem je 5000 milisekund).

Pokud dané vlákno neskončí před vypršením intervalu, zobrazí `Csrss` dialogové okno zablokovaného procesu, jak je uvedené na obrázku 5.5. (Toto dialogové okno můžete deaktivovat změnou hodnoty registru `HKCU\Control Panel\Desktop\AutoEndTasks` na 1.) Uvedené dialogové okno říká, že se určitý program dostatečně rychle neukončuje a nabízí uživateli možnost buď daný proces násilně ukončit, nebo přerušit vypínání systému. (Tomuto dialogovému oknu není přiřazena žádná doba vypršení, což znamená, že požadavek na vypnutí se v tomto místě může trvale zastavit.)



OBRÁZEK 5.5: Dialogové okno zablokovaného programu

Skončí-li dané vlákno před vypršením povolené doby, pokračuje `Csrss` v odesílání dvojic zpráv `WM_QUERYENDSESSION` a `WM_ENDSESSION` ostatním vláknům v daném procesu, která vlastní okna. Jakmile skončí všechna vlákna daného procesu, která vlastní okna, ukončí `Csrss` vlastní proces a přejde k dalšímu procesu v interaktivní relaci.

EXPERIMENT: Sledování HungAppTimeout

Využití hodnoty registru `HungAppTimeout` si můžete vyzkoušet, spustíte-li Notepad (Poznámkový blok), zadáte do něj nějaký text a pak se odhlásíte. Jakmile uplyne doba zadaná hodnotou registru `HungAppTimeout`, nabídne vám `Csrss.exe` dialogové okno dotazující se, zda chcete ukončit proces Notepad, který sám neskončil, protože čeká, až mu řeknete, zda má uložit zadaný text do souboru. Stisknete-li v zobrazeném dialogovém okně tlačítko Cancel, přeruší `Csrss.exe` vypínání systému.

Najde-li `Csrss` nějakou konzolovou aplikaci, vyvolá řídicí handler konzoly odesláním události `CTRL_LOGOFF_EVENT`. (Událost `CTRL_SHUTDOWN_EVENT` při vypínání dostávají pouze procesy služeb.) Vrátí-li handler `FALSE`, `Csrss` daný proces násilně ukončí. Pokud handler vrátí `TRUE` nebo nezareaguje po dobu definovanou hodnotou registru

HKCU\Control Panel\Desktop\WaitToKillAppTimeout (výchozí údaj je 20 000 milisekund), Csrss zobrazí okno zablokovaného programu uvedené na obrázku 5.5.

Winlogon dále zavolá funkci ExitWindowsEx, aby subsystém Csrss ukončil všechny procesy COM, jež jsou součástí interaktivní uživatelské relace.

V tomto okamžiku byly ukončeny všechny procesy v interaktivní uživatelské relaci. Opět zavolá ExitWindowsEx, tentokrát však v kontextu systémového procesu, jenž znovu odešle zprávu subsystému CSRSS, který pak hledá všechny procesy patřící systémovému kontextu a odesílá zprávy WM_QUERYENDSESSION/WM_ENDSESSION vláknům GUI (jako předtím). Místo CTRL_LOGOFF_EVENT však konzolovým aplikacím se zaregistrovanými řídicími handlery odesílá CTRL_SHUTDOWN_EVENT. Vzpomeňte si, že SCM je konzolový program, který si registruje řídicí handler. Jakmile obdrží požadavek na vypnutí, odešle zase řídicí zprávu vypnutí všem službám, které si zaregistrovaly upozornění na vypnutí. Podrobnosti o vypínání služeb (například dobu vypršení vypnutí, jakou CSRSS používá pro správce SCM), najdete v oddílu „Služby“ v kapitole 4.

Třebaže CSRSS využívá stejné doby vypršení, jako když ukončuje uživatelské procesy, nezobrazuje žádná dialogová okna ani násilně neukončuje žádné procesy. (Hodnoty registru, odpovídající dobám vypršení platným pro systémový proces, se přebírají z výchozího uživatelského profilu.) Tato vypršení prostě jen dávají systémovým procesům možnost uklidit po sobě a skončit, než dojde k vypnutí systému. Proto při vlastním vypnutí systému ještě mnoho systémových procesů běží; příkladem jsou SMSS, Winlogon, SCM a LSASS.

Jakmile Csrss skončí s upozorňováním systémových procesů na to, že se systém vypíná, dokončí Winlogon proces vypnutí zavoláním funkce NtShutdownSystem subsystému výkonné části. Tato funkce volá funkci NtSetSystemPowerState, která organizuje vypnutí ovladačů a zbytku subsystému výkonné části (správce Plug and Play, správce napájení, běhu systému, správce I/O, správce konfigurace a správce paměti).

Funkce NtSetSystemPowerState kupříkladu volá správce I/O, který má za úkol odeslat vypínací pakety I/O všem ovladačům zařízení, jež požadovaly upozornění na vypnutí. Tato akce dává ovladačům zařízení možnost před vypnutím Windows vykonat všechna zvláštní zpracování vyžadovaná jejich zařízeními. Správce konfigurace zapíše všechna modifikovaná data registru na disk a správce paměti zase zapíše do příslušných souborů všechny změněné stránky obsahující souborová data. Je-li aktivní volba vyčištění stránkovacího souboru při vypínání systému, tak správce paměti stránkovací soubor vyčistí právě nyní. Správce I/O se zavolá podruhé, aby mohl informovat ovladače systému souborů, že dochází k vypnutí systému. Vypnutí systému končí ve správci napájení. Akce prováděná správcem napájení závisí na tom, zda uživatel zadal vypnutí, restartování nebo snížení příkonu.

5.4 Závěr

V této kapitole jsme detailně probrali kroky spouštění a vypínání systému Windows (jak v normálním stavu, tak i při chybách). Zatím jsme se seznamovali s celkovou strukturou Windows a základními mechanismy zajišťujícími fungování systému, jeho běh a nakonec i vypnutí. S těmito základy jsme připraveni na detailnější průzkum jednotlivých komponent výkonné části. Začneme procesy a vlákny.