

Stručný obsah

1. Operační systémy	17
2. Architektura rodiny operačních systémů Windows NT	45
3. Vývoj ovladačů jádra	65
4. Synchronizace	113
5. Výjimky, přerušování a systémová volání	147
6. Správce objektů (Object Manager)	201
7. Procesy a vlákna	259
8. Správce vstupně/výstupních operací	305
9. Správa paměti	327
10. Registr	393
11. Souborové systémy	431
Použité zdroje	461
Rejstřík	463

Obsah

Úvod	13
Zdrojové kódy projektů	13
Co v knize najdete	14
Zpětná vazba od čtenářů	15
Zdrojové kódy ke knize	16
Errata	16

KAPITOLA 1

Operační systémy **17**

Základní pojmy	17
Procesor, úroveň oprávnění a systémová volání	17
Virtuální paměť	18
Procesy, vlákna, joby	20
Knihovny DLL a rozhraní Windows API	21
Služby a ovladače	23
Historie Windows	23
Windows 1	23
Windows 2	24
Windows 3 a OS/2	24
Windows NT	24
Windows 95, Windows 98 a Windows Me	25
Windows 2000	25
Windows XP	26
Windows Vista	26
Windows 7	27
Serverové verze	29
Základní datové struktury užívané v operačních systémech	29
Pole	30
Spojové seznamy	32
Zásobník	36
Fronta	37
Hašovací tabulky	38
Stromy	40

KAPITOLA 2

Architektura rodiny operačních systémů Windows NT

45

Mikrojádru a monolitický operační systém	45
Windows NT a jeho součásti	47
Vrstva abstrakce hardwaru (Hardware Abstraction Layer – HAL)	47
Tvrdé jádro	48
Ovladače	48
Exekutiva	48
Subsystémy	52
Systémové procesy	55

OKAPITOLA 3

Vývoj ovladačů jádra

65

Co je to ovladač	65
Prostředí pro programování	68
Jak přeložit ovladač	68
Načtení ovladače do jádra	72
Čistý a oficiální způsob	72
Méně známý způsob (nativní funkce NtLoadDriver)	75
Méně známý způsob (nativní funkce NtSetSystemInformation)	79
Jednoduchý příklad: Klasické „Hello World!“	81
Několik poznámek k ladění ovladačů	83
DbgPrint	84
DbgPrintEx	84
ASSERT	86
KdPrint a KdPrintEx	87
WinDbg	87
Modrá obrazovka smrti	91
Okolnosti vzniku	92
Průběh	92
Nastavení výpisu příčin selhání	94
Zjišťování příčin modrých obrazovek	96
Závěrečný příklad	97
Způsob uchovávání událostí	98
Použité rozhraní pro zachytávání událostí	100
Inicializace a úklid	102
Komunikace s aplikací	106

KAPITOLA 4

Synchronizace 113**Modelový příklad 113****Kritická sekce 115****Vybraná řešení problému kritické sekce 116**

Zakázání přerušení 116

Atomické operace 117

Instrukce test and set (TSL) a zámky 117

Známé synchronizační problémy 118

Synchronizační primitiva implementovaná na systémech Windows NT 126

Spinlock 126

Spinlock s frontou (Queued Spinlock) 127

Zásobníkový spinlock s frontou (In Stack Queued Spinlock) 128

Složitější atomické operace (Interlocked operations) 129

Objekt dispatcher 130

Událost (event) 131

Semafor (semaphore) 135

Mutex (Mutant) 136

Rychlé a strážené mutexy (fast mutexes, guarded mutexes) 138

Zámky reader-writer určené pro ovladače (Executive resources) 139

Pushlock 141

Kritická sekce (critical section) 142

Událost na klíč (Keyed Event) 143

Podoba konečného řešení problému s kritickými sekcemi 144

Zámek reader-writer pro uživatelský režim (Slim Reader Writer Lock – SRW Lock) 145

Další synchronizační primitiva? 146

KAPITOLA 5

Výjimky, přerušení a systémová volání 147**Komunikace s hardware 147**

Dotazování (polling) 147

Obsluha přerušení (interrupt handling) 148

Obsluha přerušení na architekturách x86 a x64 148

Výjimky 150

Hardwarová přerušení ve Windows a hardwarové priority 155

Hardwarové priority (IRQL) 157

Předdefinované hodnoty IRQL 158

Odložené volání procedur (Deferred Procedure Call – DPC) 160

Využití objektů DPC 162

Pracovní vlákna (worker threads)	169
Asynchronní volání procedur (Asynchronous Procedure Call – APC)	170
Systémová volání	171
Průběh systémového volání	173
SSDTInfo: Získání informací o tabulkách systémových volání	186
Interrupt Counter: Monitorování přerušení	189
Syscallmon: Monitorování systémových volání	195

KAPITOLA 6

Správce objektů (Object Manager) 201

Požadavky	201
Objekty exekutivy	202
Struktura objektu exekutivy	205
Hlavička (object header)	205
Hlavička OBJECT_HEADER_NAME_INFO	209
Hlavička OBJECT_HEADER_CREATOR_INFO	210
Hlavička OBJECT_HEADER_HANDLE_INFO	210
Hlavičky OBJECT_HEADER_QUOTA_INFO a OBJECT_HEADER_PROCESS_INFO	211
Struktura OBJECT_CREATE_INFORMATION	212
Příklady	213
Tělo objektu	215
Objekty reprezentující typ (struktura OBJECT_TYPE)	218
Struktura objektu ObjectType	219
Struktura OBJECT_TYPE_INITIALIZER	225
Handle a jejich tabulky	235
Vlastnosti handle	235
Skutečný význam hodnoty handle	237
Zranitelnosti v bezpečnostním software	239
Detaily struktury HANDLE_TABLE_ENTRY	239
Popis některých funkcí pro práci s handle	240
Jména objektů, adresáře a symbolické odkazy	246
Adresáře	248
Symbolické odkazy	250
Důležité oblasti jmenného prostoru	251
Relace (sessions)	252
Kradení jmen (object name squatting)	253
Počítání referencí	253
Příklad: ObjView	256
Příklad: ObInIt	257

KAPITOLA 7

Procesy a vlákna 259

Obecně o procesech a vláknech	259
Definice	259
Stavy procesů a vláken	262
Plánování	265
Možnosti implementace	270
Procesy a vlákna ve Windows	272
Reprezentace	272
Lokální úložiště vláken (Thread Local Storage)	281
Vznik nových procesů a vláken	282
Plánovač procesů a vláken	284
Chráněné procesy	297
Objekty Job	302
Fibery – vlákna implementovaná čistě v uživatelském režimu	304

KAPITOLA 8

Správce vstupně/výstupních operací 305

Základní přehled	305
Standardní způsob komunikace mezi aplikací a ovladačem	310
Způsoby přenosu dat zprávy	313
Bufferovaná metoda (METHOD_BUFFERED)	314
Metoda přímého vstupu (METHOD_IN_DIRECT)	317
Metoda přímého výstupu (METHOD_OUT_DIRECT)	317
Metoda nulové režie (METHOD_NEITHER)	319
Obsluha požadavků	320
Funkční ovladače	320
Filtry	321
Rychlý vstup a výstup (Fast I/O)	324

KAPITOLA 9

Správa paměti 327

Historický úvod	327
Virtuální paměť	329
Segmentace	332
Stránkování	335
Výběr oběti	338

Příklad implementace virtuální paměti: Intel x86	344
Datové segmenty	346
Kódové segmenty	346
Selektor	347
Stránkování	348
Přidělování bloků paměti proměnlivé velikosti	353
Správa paměti ve Windows NT	355
Struktura virtuálního adresového prostoru jádra	356
Práce s virtuální pamětí	358
Address Windowing Extension (AWE)	367
Paměťově mapované soubory	369
Interní reprezentace struktury virtuálního adresového prostoru	378
MDL (Memory Descriptor List)	380
Práce na haldě	384

KAPITOLA 10

Registr **393**

Pohled shora	394
Operace nad registrem	396
Podpora starších 32bitových aplikací na 64bitových platformách	406
Virtualizace (Registry Virtualization)	406
Přesměrování (Registry Redirector) a reflexe (Registry Reflection)	407
Interní struktura	411
Soubory registru (hive)	411
Buňka (cell)	416
Monitorování a filtrování operací nad registrem	424
Kontrola registru na bázi modifikace tabulky systémových volání	425
Kontrola registru pomocí speciálního rozhraní	426

KAPITOLA 11

Souborové systémy **431**

FAT	432
Adresáře	437
Dlouhá jména	439
NTFS	440
Bezpečnostní model	441
Hard linky	441
Soft linky (symlinky, symbolické odkazy)	442
Alternativní datové proudy	443

Řídké soubory	444
Defragmentace	444
Komprese a šifrování	447
Žurnálování a transakce	448
Žurnál USN (USN Change Journal)	449
Interní struktura	450
Speciální soubory	456

Použité zdroje **461**

Rejstřík **463**

Úvod

Tato kniha v jedenácti kapitolách pojednává o různých aspektech jádra operačních systémů rodiny Windows NT. Neklade si však za cíl toto téma zpracovat do nejmenších podrobností; spíše se čtenáři snaží vstřípnit základní informace a obohatit je o některé zajímavosti, na něž může při průzkumu jádra narazit.

Kniha se snaží postupy a algoritmy používané v jádře Windows uvést do širších souvislostí teorie operačních systémů. Z tohoto důvodu jsou některé kapitoly rozděleny na dvě části – první se věnuje dané problematice (například synchronizaci či správě paměti) obecně a druhá ukazuje, jaké poznatky a algoritmy se vývojáři a návrháři jádra Windows rozhodli použít. Tímto uspořádáním se kniha snaží ukázat, že mnohé postupy si Microsoft nevymyslel na „zelené louce“, ale vychází z teoreticky podložených faktů. Varianty některých zde popsaných algoritmů se nacházejí i v jádrech jiných operačních systémů, například těch založených na Unixu.

Dalším a posledním cílem knihy je naučit čtenáře (pokud bude chtít) pohybovat se v jádře a samostatně zkoumat jeho vnitřní mechanismy a zákonitosti. Z tohoto důvodu kniha zahrnuje i popis v tomto ohledu užitečných nástrojů a snaží se poskytnout neformální základy programování ovladačů. Za tímto účelem spolu s touto publikací vzniká webová stránka <http://www.jadro-windows.cz>, na které naleznete okomentované zdrojové kódy ovladačů jádra, jež prakticky ukazují některé aspekty probírané v jednotlivých kapitolách. Na části těchto zdrojových kódů narazíte i v textu knihy formou výpisů. Dále na zmíněném webu naleznete materiály vhodné pro další rozšiřování znalostí a užitečné nástroje.

Zdrojové kódy projektů

Konkrétně na webové stránce [jadro-windows.cz](http://www.jadro-windows.cz) naleznete zdrojové kódy následujících projektů:

- **Dllhide** – program demonstruje, jak lze manipulaci s interními datovými strukturami procesu skrýt knihovny DLL, které používá.
- **Drv** – tento program na základě argumentů příkazového řádku dokáže načítat a odstraňovat ovladače jádra. Ukazuje, jak tyto operace provádět různými způsoby.
- **Filemptest** – ukazuje, jakým způsobem je možné využít sdílené paměti vytvořené pomocí paměťově mapovaného souboru a přenášet data mezi dvěma procesy.
- **Hello** – velmi jednoduchý ovladač, který pouze vypíše několik oznámení do debuggeru jádra. Ačkoliv neprovádí prakticky žádné operace, jeho zdrojový kód ukazuje, co musí každý ovladač minimálně umět.
- **Intcount** – ovladač a aplikace, které si kladou za cíl zjistit statistiku vykonávání jednotlivých přerušení. Cílem této ukázky je demonstrovat práci s tabulkou vektorů přerušení a synchronizaci více procesorů.
- **Keyedevent** – jádro Windows implementuje zajímavá synchronizační primitiva, která jsou v této knize označována jako události na klíč (keyed events). Ačkoliv je mohou používat i normální aplikace, příslušné rozhraní není dokumentováno. Projekt `keyedevent` práci s tímto rozhraním zjednodušuje a dokumentuje jej.

- **Listdll** – tento program tvoří protipól k projektu `dllhide`. Na základě argumentů příkazového řádku se snaží zjistit seznam knihoven DLL používaných cílovým procesem. Ukazuje různé způsoby, jak tyto informace zjistit.
- **Logptm** – projekt ukazuje, jak pomocí relativně jednoduchého ovladače jádra monitorovat spouštění a ukončování procesů a vláken a načítání knihoven DLL a dalších spustitelných souborů do paměti. Ovladač k tomuto účelu využívá velmi staré a dokumentované rozhraní.
- **Ntqueryobject** – jednoduchý program, který demonstruje použití nativní funkce `NtQueryObject` ke zjištění různých zajímavých informací.
- **Obinit** – tento projekt ukazuje, jak pomocí techniky DKO (Direct Kernel Object Hooking) monitorovat přístupy k různým objektům operačního systému (souborům, klíčům registru, procesům, vláknům a dalším).
- **Objview** – projekt, který ukazuje, jakým způsobem je možné implementovat funkce, kterými disponuje utilita `WinObj` ze serveru www.sysinternals.com. Jedná se o prohlížeč pojmenovaných objektů existujících v jádře operačního systému.
- **Ppoc** – Windows Vista mimo jiné zavádí nový druh procesů – tzv. *chráněné procesy* (protected processes). Cílem projektu `pproc` je umožnit vám libovolný proces označit jako chráněný a opačně. Dále projekt také obsahuje testovací program, který prakticky demonstruje, jaké možnosti a omezení chráněné procesy s sebou přináší.
- **Registrymon** – projekt ukazuje, jak implementovat funkcionalitu podobnou aplikaci `Regmon`, kterou jste mohli dříve nalézt na serveru www.sysinternals.com, tedy monitorování operací nad Registrem Windows.
- **SSDTInfo** – systémová volání patří k jednomu z nejdůležitějších mechanismů ve Windows. Projekt `SSDTInfo` ukazuje, jak zjistit zajímavé informace o interních datových strukturách, které implementace tohoto mechanismu používá.
- **Syscallmon** – projekt ukazuje, jakým způsobem lze monitorovat systémová volání.
- **Vad** – bloky alokované a rezervované paměti procesu reprezentuje jádro Windows pomocí struktur `VAD` (Virtual Address Descriptor). Projekt `vad` demonstruje, jak s těmito strukturami pracovat a získat z nich užitečné informace.

Všechny zdrojové kódy jsou psány v programovacích jazycích C a Object Pascal a vývojových prostředích Delphi XE 2010 a Microsoft Visual Studio. Jednotlivé programy a ovladače, až na výjimky, fungují na 32bitových i 64bitových verzích Windows XP, Windows Server 2003, Windows Vista a Windows 7. Delphi je použito převážně z důvodu snadné tvorby grafického uživatelského rozhraní.

Co v knize najdete

První tři kapitoly knihy tvoří úvod do problematiky operačních systémů rodiny Windows NT. První kapitola nejprve vysvětluje základní pojmy, jako je proces, vlákno, systémové volání či handle. Dále pokračuje stručným popisem jednotlivých verzí Windows; od Windows 1 až po současná Windows 7. Ve své třetí části kapitola popisuje základní datové struktury, mezi které patří pole, spojový seznam, fronta či zásobník a které jádra operačních systémů často využívají k uchování různých informací.

Druhá kapitola již trochu sestupuje z teoretických výšin kapitoly první a obecně pojednává o jednotlivých částech jádra Windows a dalších komponentách, bez kterých by operační systém nefungoval. Dočtete se v ní také o službách – programech (a ovladačích), jejichž posláním je vykonávat důležité činnosti na pozadí.

Třetí kapitola popisuje některé postupy a principy programování ovladačů jádra. Dozvíte se, jaké nástroje potřebujete a jak ovladač načíst do paměti jádra. V závěrečné části je popsána struktura ovladače `logptm.sys`. Kapitola se informace snaží podávat méně formálním způsobem; kterým se autor této knihy učil poznávat zákoutí jádra Windows.

Další kapitoly se již věnují jednotlivým aspektům operačního systému, i když u některých nechybí obecný úvod. Ve čtvrté kapitole se dočtete o způsobech řízení přístupu více aplikací (vláken) ke sdíleným prostředkům. Pátá kapitola popisuje mechanismy obsluhy přerušení, odloženého volání procedur a systémových volání. Následující kapitola pojednává o tom, jak jádro Windows využívá principů objektivě orientovaného programování.

Sedmá kapitola se věnuje procesům, vláknům a jejich plánování. V osmé se dočtete o způsobech předávání dat mezi aplikací a ovladačem a mezi ovladači navzájem. Jedná se o rozšíření poznatků neformálně sdílených ve třetí kapitole. Devátá kapitola se zabývá správou paměti. Popisuje jak různé operace s virtuální pamětí, které jádro Windows podporuje, tak dává lehce nahlédnout i do interních datových struktur, které správce paměti používá.

Desátá a jedenáctá kapitola popisují formáty, které Windows používají pro ukládání dat na externí média, například pevné disky. Desátá kapitola se věnuje Registru, struktuře určené ukládání nastavení aplikací a celého operačního systému. Popisuje tento formát z hlediska programátora a uživatele. Její podstatná část pojednává i o tom, jak je tato „databáze“ fyzicky uložena na pevném disku. Jedenáctá kapitola popisuje interní datové struktury dvou na Windows nejrozšířenějších souborových systémů – FAT a NTFS.

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

redakce PC literatury
Computer Press
Spielberk Office Centre
Holandská 3
639 00 Brno

nebo

sefredaktor.pc@cpress.cz

Zdrojové kódy ke knize

Z adresy <http://knihy.cpress.cz/K1741> si po klepnutí na odkaz Soubory ke stažení můžete přímo stáhnout archiv s ukázkovými kódy.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už v textu nebo v kódu, budeme rádi, pokud nám ji nahlásíte. Ostatní uživatelé tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese <http://knihy.cpress.cz/K1741> po klepnutí na odkaz Soubory ke stažení.