

Stručný obsah

Stručný obsah	6
Poděkování	8
O autorovi	9
Obsah	11
Úvod	19
1. Chyby: Kde vznikají a jak je řešit	25
2. Začínáme s laděním	47
3. Ladění při psaní programu	88
4. Podpora ladění v operačním systému a funkce ladicích programů prostředí Win32	132
5. Pokročilé techniky použití ladicího programu ve Visual Studio .NET	175
6. Pokročilé ladění aplikací .NET s Visual Studio .NET	193
7. Pokročilé techniky pro práci s nativním kódem ve Visual Studio .NET	222
8. Pokročilé techniky nativního kódu ve WinDBG	297
9. Rozšíření rozhraní Visual Studio .NET	342
10. Sledování řízených výjimek	378
11. Trasování toku programového kódu	398
12. Nalezení zdrojového souboru a řádku pomocí kolizní adresy	406
13. Co dělat v případě zhroucení aplikace	420

14. Ladění služeb Windows a DLL do nich zaváděných	444
15. Deadlock při běhu několika vláken	456
16. Automatizované testování	497
17. Ladicí knihovna běhového prostředí jazyka C a správa paměti	524
18. FastTrace: Nástroj pro efektivní trasování pro serverové aplikace	564
19. Ladění pracovní množiny programu	569
Příloha A: Jak číst logové soubory nástroje Dr. Watson	592
Příloha B: Zdroje pro vývojáře na platformách .NET a Windows	602
Příloha C: Anglicko-český slovníček	610
Příloha D: Česko-anglický slovníček	621
Příloha E: Význam identifikátorů	631
Rejstřík	636

Obsah

Stručný obsah	6
Poděkování	8
O autorovi	9
Obsah	11
Úvod	19
Kdo by si měl tuto knihu přečíst?	20
Jak číst tuto knihu a co je nového ve druhém vydání.....	21
Systémové požadavky.....	22
Co najdete v ukázkových souborech?	22
Důležité! Windows 98/Me a kódování ANSI.....	23
Důležité! Engine symbolů DBGHELP.DLL	23
Zpětná vazba	24
Podpora od Microsoft Press	24
1. Chyby: Kde vznikají a jak je řešit	25
1.1 Chyby a ladění.....	25
1.2 Co jsou to chyby?.....	26
Nekonzistentní uživatelská rozhraní	26
Nesplněná očekávání	27
Špatná výkonnost	27
Kolize programů a poškození dat	28
1.3 Procesní chyby a řešení.....	28
Krátké nebo nespílitelné termíny	29
Přístup „Nejprve programuj, pak přemýšlej“	29
Špatně pochopené požadavky	32
Ignorantský přístup autora nebo nesprávné školení.....	33
Podceňování důležitosti kvality.....	34
Plánování ladění	36
1.4 Předpoklady pro ladění	37
Soubor dovedností.....	37
Učení se dovednostem	39
1.5 Proces ladění.....	40
První krok: Reprodukuj chybu.....	41
Druhý krok: Popiš chybu.....	42
Třetí krok: Vždy předpokládej, že jde o tvou chybu	42
Čtvrtý krok: Rozděl a panuj.....	43
Pátý krok: Uvažuj kreativně	43
Šestý krok: Použij efektivní nástroje	44
Sedmý krok: Pusť se do tvrdého ladění.....	44

Osmý krok: Ověř opravu chyby	45
Devátý krok: Uč se a sdílejte zkušenosti s ostatními	46
Závěrečné tajemství procesu ladění	46
1.6 Shrnutí	46
2. Začínáme s laděním	47
2.1 Sledujte změny, dokud projekt nevyhodíte	47
Systémy řízení verzí	48
Systémy sledování programových chyb	51
Zvolte si systém, který vám nejlépe vyhovuje	52
2.2 Začněte do harmonogramu čas pro vytvoření ladicího systému	53
Vytvořte veškerý sestavený kód s ladicími symboly	54
V řízeném kódu pracujte s varováním jako s chybou	59
V nativním kódu většinou pracujte s varováním jako s chybou	61
V nativním kódu byste měli vědět, kde jsou zaváděny knihovny DLL	64
Výchozí adresy a řízené moduly	68
Vytvořte zjednodušený diagnostický systém pro finální sestavený kód	75
2.3 Častá sestavení a zahoření programu jsou nutná	76
Častá sestavení programu	76
Zahořování	77
2.4 Vytvoření instalačního programu	78
2.5 Zkušební technici musí testovat kód sestavený pro ladění	79
2.6 Nainstalujte symboly operačního systému a nastavte úložiště symbolů	79
Zdrojový kód a symbolové servery	87
2.7 Shrnutí	87
3. Ladění při psaní programu	88
3.1 Aserce	89
Jak a co doplnit asercemi	89
Aserce ve formulářových a konzolových aplikacích Windows a v prostředí .NET	97
Aserce v aplikacích ASP.NET a webové služby XML	104
Aserce v aplikacích nativního C++	113
Různé typy asercí prostředí Visual C++	115
3.2 Trasování	124
Trasování formulářových a konzolových aplikací Windows v prostředí .NET	124
Trasování v aplikacích ASP.NET a webové služby XML	126
Trasování v aplikacích nativního C++	128
3.3 Komentáře	128
3.4 Důvěřujte si, ale ověřujte svou práci (testování jednotek)	129
3.5 Shrnutí	131
4. Podpora ladění v operačním systému a funkce ladicích programů prostředí Win32	132
4.1 Typy ladicích programů Windows	133
Ladicí programy uživatelského režimu	133
Ladicí programy privilegovaného režimu	135
4.2 Podpora ladění v operačním systému Windows	137

Ladění Just-In-Time (JIT)	137
Automatický start v ladícím programu	140
4.3 MinDBG: jednoduchý ladící program Win32	141
4.4 WDBG: skutečný ladící program	157
Čtení a zápis do paměti	159
Body přerušení a krokování programu	162
Tabulky symbolů, engine symbolů a procházení zásobníkem	166
Funkce Step Into, Step Over a Step Out	171
4.5 Jak napsat vlastní ladící program	172
4.6 Jaký vývoj WDBG lze očekávat?	173
4.7 Shrnutí	174

5. Pokročilé techniky použití ladícího programu ve Visual Studio .NET 175

5.1 Pokročilé body přerušení a jejich použití	176
Tipy pro body přerušení	176
Rychlé zastavení na libovolné funkci	178
Modifikátor místních bodů přerušení	184
5.2 Počet přístupů	184
5.3 Výrazy s podmínkou	186
Více bodů přerušení na jednom řádku	187
5.4 Okno Watch	188
Volání metod v okně Watch	189
Příkaz Set Next Command	190
5.5 Shrnutí	192

6. Pokročilé ladění aplikací .NET s Visual Studio .NET 193

6.1 Pokročilé body přerušení a programy .NET	193
Výrazy s podmínkou	193
6.2 Okno Watch	197
Automatické rozvinutí vlastních typů	198
6.3 Tipy a triky	201
Atributy DebuggerStepThroughAttribute a DebuggerHiddenAttribute	201
Smišený způsob ladění	202
Vzdálené ladění	204
6.4 ILDASM a Microsoft Intermediate Language	205
Začátky práce s ILDASM	207
Základy CLR	212
MSIL, lokální proměnné a parametry	212
Důležité instrukce	214
Další nástroje pro zpětný překlad	219
6.5 Shrnutí	220

7. Pokročilé techniky pro práci s nativním kódem ve Visual Studio .NET 222

7.1 Pokročilé body přerušení pro nativní aplikace	222
Syntaxe pokročilých bodů přerušení	222
Body přerušení v systémových nebo v exportovaných funkcích	223
Výrazy s podmínkou	226

Datové body přerušení	228
Zlepšené datové body přerušení	231
7.2 Okno Watch	232
Formátování dat a vyhodnocování výrazů	232
Měření času pomocí okna Watch	235
Nedokumentované pseudoregistry	235
Automatické rozvinutí vašich datových typů	235
Přidávání vlastních hodnot HRESULT	237
Přidání speciálně upraveného zobrazení do okna Watch	237
7.3 Vzdálené ladění	242
7.4 Tipy a triky	245
Ladění vloženého kódu	245
Okno Memory a automatické vyhodnocování paměti	246
Monitorování výjimek	246
Další tipy pro práci se symboly	249
Odpojení od procesů Windows 2000	249
Práce se soubory s výpisem paměti	250
7.5 Assembler x86	252
Základy fungování procesoru	252
Registry	253
Formát instrukcí a adresování paměti	257
Několik slov o vkládaném kódu assembleru – Visual C++ .NET Inline Assembler	258
Instrukce, které je třeba znát	259
Častá sekvence: vstup do funkce a její opuštění	262
Volání procedur a návrat	263
Konvence volání	264
Příklad konvencí volání	267
Přístup k proměnným: globální proměnné, parametry a lokální proměnné	269
Další instrukce, které je třeba znát	274
Operace s daty	274
Operace s řetězci	279
7.6 Časté konstrukty assembleru	282
Přístup do registru FS	282
Odkazy na struktury a třídy	284
Úplná ukázka	285
Okno Disassembly	287
Příkaz Set Next Statement	290
Ruční procházení zásobníkem	291
Tipy a triky	294
Nadbytečný kód	295
7.7 Shrnutí	296

8. Pokročilé techniky nativního kódu ve WinDBG	297
8.1 Než začnete	298
8.2 Základy	300
8.3 Různé situace při ladění	303
Získání nápovědy	304
Načtení správných symbolů	304
Procesy a vlákna	308

8.4 Obecné ladění v okně Command	313
Hledání a vyhodnocení proměnných	313
Spuštění, krokování a trasování	315
Body přerušení	319
Výjimky a události	322
Správa WinDBG	324
8.5 Rozšíření	325
Nahrávání a správa rozšíření	326
Důležité příkazy rozšíření	327
8.6 Práce se soubory s výpisem z paměti	331
Vytvoření souboru s výpisem z paměti	332
Otevření souboru s výpisem	333
Ladění výpisu z paměti	334
8.7 Son of Strike (SOS)	334
Používání SOS	335
8.8 Shrnutí	341

9. Rozšíření rozhraní Visual Studio .NET **342**

9.1 Rozšíření pomocí makra	343
Parametry makra	344
Problémy s projekty	345
Prvky kódu	346
9.2 Makro Commentator	348
9.3 Úvod k doplňkům	353
Oprava kódu doplňku automaticky vygenerovaného pomocníkem	354
Práce s tlačítky panelu nástrojů	357
Vytváření oken s nástroji	358
Karty s vlastnostmi pro řízený kód v dialogovém okně Options	360
9.4 Doplňek SuperSaver	364
9.5 Doplňek SettingsMaster	369
Implementace doplňku SettingsMaster	376
Další případná vylepšení doplňku SettingsMaster	377
9.6 Shrnutí	377

10. Sledování řízených výjimek **378**

10.1 Úvod do Profiling API	378
Spuštění profilovacího programu	385
Knihovna ProfilerLib	387
10.2 Utilita ExceptionMon	389
Ladění uvnitř procesů pomocí utility ExceptionMon	390
10.3 Využití výjimek v .NET	395
10.4 Shrnutí	397

11. Trasování toku programového kódu **398**

11.1 Sledování funkcí v rámci Profiling API	398
Hlášení vstupu do funkce a výstupu z ní	398
Implementace sledovacích funkcí	399

Vkládání inline kódu.....	400
FunctionIDMapper	401
11.2 Použití programu FlowTrace	401
11.3 Implementace nástroje FlowTrace	403
11.4 Další vývoj nástroje FlowTrace	404
11.5 Shrnutí	405
12. Nalezení zdrojového souboru a řádku pomocí kolizní adresy	406
<hr/>	
12.1 Vytváření a čtení souborů typu MAP	408
12.2 Obsah souboru typu MAP.....	409
Nalezení zdrojového souboru, názvu funkce a čísla řádku	412
PDB2MAP – dodatečné mapování souborů	413
Použití nástroje CrashFinder	415
Důležitá hlediska implementace	417
12.3 Co dál s nástrojem CrashFinder?	418
12.4 Shrnutí	419
13. Co dělat v případě zhroutil aplikace	420
<hr/>	
13.1 Strukturovaná obsluha výjimek versus obsluha výjimek C++	420
Strukturovaná obsluha výjimek.....	421
Obsluha výjimek jazyka C++	423
Vyhněte se obsluze výjimek C++	424
Za žádnou cenu nepoužívejte catch (...)!.....	425
Nepoužívejte _set_se_translator	426
Funkce API SetUnhandledExceptionFilter.....	427
Použití API CrashHandler.....	429
Převod struktur EXCEPTION_POINTERS	431
Minivýpisy	433
13.2 Shrnutí	442
14. Ladění služeb Windows a DLL do nich zaváděných	444
<hr/>	
14.1 Základy služeb	444
Služby a API	445
Služby a bezpečnost.....	446
14.2 Ladění služeb	446
Ladění kódu jádra	447
Odladte kód pro spouštění	451
14.3 Shrnutí	454
15. Deadlock při běhu několika vláken	456
<hr/>	
15.1 Tipy a triky programování s více vlákny.....	456
Vyhněte se programování s více vlákny.....	456
Omezte počet použitých vláken	457
Omezte vlákna pouze na malé oddělené části kódu	457
Synchronizujte na co nejnižší úrovni	457
„Rotujte“ své kritické sekce.....	461
Nepoužívejte CreateThread/ExitThread	462
Implicitní správa paměti vás může zabít.....	463

Pracujte s výpisy.....	464
Opakovaně revidujte svůj kód.....	464
Testujte na počítačích s více procesory.....	465
15.2 Požadavky kladené na nástroj DeadlockDetection.....	467
15.3 Problematika návrhu s DeadlockDetection.....	468
15.4 Použití programu DeadlockDetection.....	469
15.5 Implementace programu DeadlockDetection.....	472
Sledování importovaných funkcí.....	472
Problematika implementace.....	480
15.6 Jaký vývoj programu DeadlockDetection lze očekávat?.....	495
15.7 Shrnutí.....	496
16. Automatizované testování.....	497
<hr/>	
16.1 Prokletí testování jednotek: uživatelská rozhraní.....	497
Požadavky utility Tester.....	498
16.2 Použití utility Tester.....	499
Skripty Testera.....	499
Nahrávání skriptů.....	504
16.3 Implementace utility Tester.....	507
Oznámení TESTER.DLL a implementace zpětného přehrávání.....	507
16.4 Implementace TESTREC.EXE.....	510
Zpracování úhozů.....	511
Zpracování vstupu myši.....	519
16.5 Co dále s utilitou Tester?.....	522
16.6 Shrnutí.....	523
17. Ladicí knihovna běhového prostředí jazyka C a správa paměti.....	524
<hr/>	
17.1 Charakteristika Debug C Run-Time Library.....	524
17.2 Použití Debug C Run-Time Library.....	526
Chyba v DCRT.....	528
Užitečné funkce DCRT.....	532
17.3 Výběr správné knihovny běhového prostředí C pro vaši aplikaci.....	533
17.4 Použití MemDumperValidator.....	535
Použití MemDumperValidator v jazyku C++.....	537
Použití MemDumperValidator v jazyku C.....	538
Hloubková ověřování.....	538
17.5 Implementace MemDumperValidator.....	542
Spouštění a ukončování programů v C++.....	544
Kam zmizela všechna oznámení o ztracené paměti?.....	545
17.6 Použití rozšíření MemStress.....	546
Zajímavé problémy související se zátěží aplikací v provozu.....	547
17.7 Hromady operačního systému.....	548
17.8 Triky pro odhalování problémů s pamětí.....	551
Hledání zápisů s neinicializovanou pamětí.....	551
Hledání přetečení paměti.....	552
17.9 Nejdůležitější přepínače.....	558
Přepínače kontrol chyb za běhu programu.....	558
Bezpečnostní přepínač bufferu.....	562
17.10 Shrnutí.....	563

18. FastTrace: Nástroj pro efektivní trasování pro serverové aplikace	564
18.1 Základní problém a jeho řešení	564
18.2 Použití programu FastTrace	565
Spojení protokolovacích souborů trasování.....	567
18.3 Implementace programu FastTrace	568
18.4 Shrnutí	568
19. Ladění pracovní množiny programu	569
19.1 Ladění pracovní množiny	569
19.2 Použití programu SWS	574
Nastavení příkladu programu SWS	574
Spuštění aplikace pomocí programu SWS	576
Vytvoření a použití souboru s pořadím	577
19.3 Implementace programu SWS	579
Funkce _penter	579
Souborový formát .SWS a výčet symbolů	583
Běžové prostředí a ladění	588
19.4 Jaký vývoj SWS lze očekávat?	591
19.5 Shrnutí	591
Příloha A: Jak číst logové soubory nástroje Dr. Watson	592
Logovací soubory nástroje Dr. Watson	594
Příloha B: Zdroje pro vývojáře na platformách .NET a Windows	602
Knihy	602
Vývoj programových aplikací	602
Ladění a testování aplikací	604
Technologie .NET	604
Jazyk C/C++	605
Technologie Windows	605
Procesory Intel a Hardware	606
Programové nástroje	607
Webové stránky	608
Příloha C: Anglicko-český slovníček	610
Příloha D: Česko-anglický slovníček	621
Příloha E: Význam identifikátorů	631
Rejstřík	636