

# Spis treści

<b>Część I: Lektura obowiązkowa</b> .....	1
<b>1 Obsługa błędów</b> .....	3
Definiowanie własnych kodów błędów .....	7
Aplikacja przykładowa <code>ErrorShow</code> .....	8
<b>2 Praca ze znakami i łańcuchami znaków</b> .....	11
Kodowanie znaków.....	12
Znakowe i łańcuchowe typy danych ANSI i Unicode.....	14
Funkcje Unicode i ANSI w Windows .....	16
Funkcje Unicode i ANSI w bibliotece uruchomieniowej C.....	18
Bezpieczne funkcje łańcuchowe w bibliotece uruchomieniowej C.....	19
Wprowadzenie do nowych, bezpiecznych funkcji łańcuchowych .....	20
Jak uzyskać więcej kontroli przy wykonywaniu operacji na łańcuchach? .....	23
Funkcje łańcuchowe Windows .....	25
Dlaczego należy korzystać z Unicode? .....	27
Jak zalecamy pracować ze znakami i łańcuchami? .....	28
Tłumaczenie łańcuchów pomiędzy Unicode a ANSI.....	29
Eksportowanie funkcji ANSI i Unicode z bibliotek DLL .....	31
Określanie, czy dany tekst jest w standardzie ANSI, czy Unicode.....	33
<b>3 Obiekty jądra</b> .....	35
Czym jest obiekt jądra? .....	35
Zliczanie użyć .....	37
Bezpieczeństwo .....	37
Tabela uchwytów obiektów jądra dla danego procesu .....	39
Tworzenie obiektu jądra .....	40
Zamykanie obiektu jądra .....	41
Współdzielenie obiektów jądra pomiędzy granicami procesów.....	45
Korzystanie z dziedziczenia uchwytów do obiektów .....	46
Nadawanie nazw obiektom .....	50
Duplikowanie uchwytów do obiektów .....	62
 <b>Część II: .....</b>	 <b>Realizacja zadań 67</b>
<b>4 Procesy</b> .....	69
Pisanie pierwszej aplikacji dla Windows .....	70
Uchwyt do instancji procesu.....	75
Funkcja <code>CreateProcess</code> .....	91
<i>pszApplicationName</i> i <i>pszCommandLine</i> .....	92
Kończenie procesu.....	107
Koniec działania funkcji początkowej głównego wątku .....	107
Funkcja <code>ExitProcess</code> .....	108
Funkcja <code>TerminateProcess</code> .....	109
Gdy przestaną istnieć wszystkie wątki procesu .....	110
Gdy proces zakończy swoje działanie.....	110
Procesy potomne.....	111

Uruchamianie osobnych procesów potomnych .....	113
Gdy administrator działa z uprawnieniami zwykłego użytkownika .....	113
Automatyczne podnoszenie uprawnień procesu .....	117
Ręczne podnoszenie uprawnień procesu .....	119
Jaki jest bieżący kontekst uprawnień? .....	120
Wyliczanie procesów działających w systemie .....	122
<b>5 Zadania</b> .....	131
Nakładanie ograniczeń na procesy należące do zadania .....	135
Umieszczanie procesu wewnątrz zadania .....	143
Kończenie wszystkich procesów należących do zadania .....	144
Pozyskiwanie statystyk dotyczących zadania .....	144
Powiadomienia o zadaniach .....	147
Aplikacja przykładowa Job Lab .....	150
<b>6 Podstawy wątków</b> .....	153
Kiedy tworzyć wątek? .....	154
Kiedy nie tworzyć wątku .....	156
Pisanie funkcji początkowej nowego wątku .....	157
Funkcja <i>CreateThread</i> .....	158
<i>psa</i> .....	159
<i>cbStackSize</i> .....	159
<i>pfnStartAddr</i> i <i>pvParam</i> .....	160
<i>dwCreateFlags</i> .....	161
<i>pdwThreadID</i> .....	161
Kończenie wątku .....	162
Funkcja wątkowa zwraca wartość .....	162
Funkcja <i>ExitThread</i> .....	162
Funkcja <i>TerminateThread</i> .....	163
Gdy proces kończy działanie .....	164
Gdy wątek kończy działanie .....	164
Wątki od środka .....	165
Uwarunkowania biblioteki uruchomieniowej C/C++ .....	167
Ojej! przez pomyłkę wywołałem <i>CreateThread</i> zamiast <i>_beginthreadex</i> .....	176
Funkcje biblioteki uruchomieniowej C/C++, których nie powinno się nigdy wywoływać .....	177
Zyskiwanie poczucia własnej tożsamości .....	178
Konwertowanie pseudo-uchwyty na prawdziwy uchwyt .....	178
<b>7 Planowanie wątków, priorytety i koligacje</b> .....	181
Wstrzymywanie i wznawianie wątku .....	183
Wstrzymywanie i wznawianie procesu .....	184
Usypianie .....	186
Przełączanie na inny wątek .....	186
Przełączanie na inny wątek na procesorze z hiperwątkowością .....	187
Czasy wykonywania wątku .....	187
CONTEXT we właściwym kontekście .....	191
Priorytety wątków .....	195

Abstrakcyjne spojrzenie na priorytety .....	196
Programowanie priorytetów .....	200
Dynamiczne zwiększanie poziomów priorytetów wątków .....	203
Dostrajanie planowania wątków dla procesu pierwszoplanowego .....	204
Planowanie priorytetów dla żądań wejścia/wyjścia .....	205
Aplikacja przykładowa Scheduling Lab .....	206
Koligacje .....	212
<b>8 Synchronizacja wątków w trybie użytkownika .....</b>	<b>217</b>
Dostęp atomowy: rodzina funkcji blokujących .....	218
Wiersze pamięci podręcznej .....	224
Zaawansowana synchronizacja wątków .....	225
Technika, której należy unikać .....	226
Sekcje krytyczne .....	228
Sekcje krytyczne: uwagi drobnym drukiem .....	230
Sekcje krytyczne i blokady pętlowe .....	233
Sekcje krytyczne i obsługa błędów .....	233
Ograniczone blokady odczytu-zapisu .....	235
Zmienne warunkowe .....	238
Przykładowa aplikacja Queue .....	239
Użyteczne wskazówki i techniki .....	249
<b>9 Synchronizacja wątków przy pomocy obiektów jądra .....</b>	<b>253</b>
Funkcje oczekiwania .....	255
Efekty uboczne udanego oczekiwania .....	258
Obiekty jądra dla zdarzeń .....	260
Aplikacja przykładowa Handshake .....	264
Obiekty jądra dla zegarów .....	268
Kolejkowanie wywołań APC przez zegary .....	272
Brakujące szczegóły dotyczące zegarów .....	273
Obiekty jądra dla semaforów .....	274
Obiekty jądra dla muteksów .....	277
Sprawy związane z porzucaniem .....	279
Muteksy a sekcje krytyczne .....	280
Aplikacja przykładowa Queue .....	280
Wygodne zestawienie obiektów do synchronizacji wątków .....	287
Inne funkcje synchronizujące wątki .....	287
Asynchroniczne, sprzętowe wejście/wyjście .....	288
<i>WaitForInputIdle</i> .....	288
<i>MsgWaitForMultipleObjects(Ex)</i> .....	290
<i>WaitForDebugEvent</i> .....	290
<i>SignalObjectAndWait</i> .....	291
Wykrywanie zakleszczeń przy użyciu Wait Chain Traversal API .....	292
<b>10 Synchroniczne i asynchroniczne operacje wejścia/wyjścia .....</b>	<b>301</b>
Otwieranie i zamykanie urządzeń .....	302
Szczegółowe spojrzenie na <i>CreateFile</i> .....	305
Praca z urządzeniami plikowymi .....	312

Pobieranie rozmiaru pliku.....	313
Ustawianie wskaźnika pliku.....	314
Ustawianie końca pliku.....	316
Wykonywanie synchronicznego, sprzętowego wejścia/wyjścia.....	316
Przerzucanie danych do urządzenia.....	317
Anulowanie synchronicznego wejścia/wyjścia.....	317
Podstawy asynchronicznego, sprzętowego wejścia/wyjścia.....	319
Struktura OVERLAPPED.....	320
Zastrzeżenia do asynchronicznego, sprzętowego wejścia/wyjścia.....	322
Anulowanie zakolejkowanych żądań sprzętowego wejścia/wyjścia.....	324
Otrzymywanie powiadomień o zakończonym żądaniu wejścia/wyjścia.....	325
Sygnalizowanie obiektu jądra dla urządzenia.....	326
Sygnalizowanie obiektu jądra dla zdarzenia.....	327
Powiadamialne wejście/wyjście.....	330
Porty zakończeń wejścia/wyjścia.....	335
<b>11 Pula wątków Windows.....</b>	<b>355</b>
Scenariusz 1: Asynchroniczne wywoływanie funkcji.....	356
Jawne sterowanie zadaniem.....	357
Aplikacja przykładowa Batch.....	358
Scenariusz 2: Wywoływanie funkcji w określonych odstępach czasu.....	362
Aplikacja przykładowa Timed Message Box.....	364
Scenariusz 3: Wywoływanie funkcji, gdy zasygnalizowany zostanie pojedynczy obiekt jądra.....	367
Scenariusz 4: Wywoływanie funkcji, gdy zakończy się żądanie asynchronicznego wejścia/wyjścia.....	369
Działania wykonywane po zakończeniu funkcji zwrotnej.....	371
Niestandardowe pule wątków.....	372
Eleganckie niszczenie puli wątków: grupy porządkowe.....	375
<b>12 Włókna.....</b>	<b>377</b>
Praca z włóknami.....	377
Aplikacja przykładowa Counter.....	381
<b>Część III:..... Zarządzanie pamięcią 385</b>	
<b>13 Architektura pamięci Windows.....</b>	<b>387</b>
Wirtualna przestrzeń adresowa procesu.....	387
Jak jest podzielona wirtualna przestrzeń adresowa?.....	388
Partycja przypisywania pustego wskaźnika.....	389
Partycja trybu użytkownika.....	389
Partycja trybu jądra.....	392
Regiony w przestrzeni adresowej.....	392
Przydzielanie fizycznej pamięci dla regionu.....	393
Pamięć fizyczna a plik wymiany.....	394
Pamięć fizyczna nieutrzymywana w pliku wymiany.....	396
Atrybuty ochrony.....	398

Dostęp z kopiowaniem przy zapisie .....	399
Flagi atrybutów ochrony dla specjalnego dostępu .....	400
Zebranie wszystkiego razem .....	400
Regiony od środka .....	406
Znaczenie wyrównywania danych .....	410
<b>14 Badanie pamięci wirtualnej .....</b>	<b>415</b>
Informacje systemowe .....	415
Aplikacja przykładowa System Information .....	418
Stan pamięci wirtualnej .....	424
Zarządzanie pamięcią na maszynach NUMA .....	425
Aplikacja przykładowa Virtual Memory Status .....	426
Określanie stanu przestrzeni adresowej .....	428
Funkcja <i>VMQuery</i> .....	430
Aplikacja przykładowa Virtual Memory Map .....	435
<b>15 Korzystanie z pamięci wirtualnej we własnych aplikacjach .....</b>	<b>439</b>
Rezerwowanie regionu w przestrzeni adresowej .....	439
Przydzielanie pamięci w zarezerwowanym regionie .....	442
Jednoczesne rezerwowanie regionu i przydzielanie pamięci .....	442
Kiedy przydzielać pamięć fizyczną? .....	444
Oddzielanie pamięci fizycznej i zwalnianie regionu .....	446
Kiedy oddzielać pamięć fizyczną? .....	447
Aplikacja przykładowa Virtual Memory Allocation .....	448
Zmienianie atrybutów ochrony .....	454
Odnawianie zawartości pamięci fizycznej .....	455
Aplikacja przykładowa MemReset .....	456
Rozszerzenia okien adresowych .....	459
Aplikacja przykładowa AWE .....	462
<b>16 Stos wątku .....</b>	<b>469</b>
Funkcja sprawdzania stosu z biblioteki uruchomieniowej C/C++ .....	473
Aplikacja przykładowa Summation .....	475
<b>17 Pliki mapowane w pamięci .....</b>	<b>481</b>
Mapowane w pamięci pliki wykonywalne i biblioteki DLL .....	482
Dane statyczne nie są współdzielone przez wiele instancji pliku wykonywalnego lub biblioteki DLL .....	483
Mapowane w pamięci pliki danych .....	493
Metoda 1: Jeden plik, jeden bufor .....	493
Metoda 2: Dwa pliki, jeden bufor .....	494
Metoda 3: Jeden plik, dwa bufory .....	494
Metoda 4: Jeden plik, zero buforów .....	495
Korzystanie z plików mapowanych w pamięci .....	495
Krok 1: Tworzenie lub otwieranie obiektu jądra dla pliku .....	495
Krok 2: Tworzenie obiektu jądra dla mapowania pliku .....	497
Krok 3: Mapowanie danych pliku do przestrzeni adresowej procesu .....	500
Krok 4: Odłączanie mapowania danych pliku od przestrzeni adresowej procesu .....	503

Kroki 5 i 6: Zamykanie obiektu mapowania pliku i obiektu pliku .....	505
Aplikacja przykładowa File Reverse .....	506
Przetwarzanie dużego pliku przy użyciu plików mapowanych w pamięci .....	512
Pliki mapowane w pamięci a spójność .....	513
Określanie adresu bazowego pliku mapowanego w pamięci .....	514
Szczegóły implementacyjne plików mapowanych w pamięci .....	515
Korzystanie z plików mapowanych w pamięci do współdzielenia danych pomiędzy procesami .....	517
Pliki mapowane w pamięci wspierane przez plik wymiany .....	518
Aplikacja przykładowa Memory-Mapped File Sharing .....	519
Rzadko przydzielane pliki mapowane w pamięci .....	522
Aplikacja przykładowa MMF Sparse .....	524
<b>18 Sterty</b> .....	<b>535</b>
Domyślna sterła procesu .....	535
Powody tworzenia dodatkowych stert .....	536
Ochrona składników .....	537
Efektywniejsze zarządzanie pamięcią .....	537
Dostęp lokalny .....	538
Unikanie kosztów synchronizacji wątków .....	539
Szybkie zwalnianie pamięci .....	539
Jak utworzyć dodatkową stertę? .....	539
Alokowanie bloku pamięci ze sterty .....	541
Zmienianie rozmiaru bloku .....	543
Otrzymywanie rozmiaru bloku .....	544
Zwalnianie bloku .....	544
Niszczenie sterty .....	544
Używanie stert w języku C++ .....	545
Różne funkcje sterty .....	548

#### **Część IV:..... Dynamicznie dołączane biblioteki 551**

<b>19 Podstawy DLL</b> .....	<b>553</b>
Biblioteki DLL a przestrzeń adresowa procesu .....	554
Ogólny obraz sytuacji .....	556
Budowanie modułu DLL .....	559
Budowanie modułu wykonywalnego .....	564
Uruchamianie modułu wykonywalnego .....	567
<b>20 Zaawansowane techniki DLL</b> .....	<b>571</b>
Jawne ładowanie modułu DLL i dołączanie symboli .....	571
Jawne ładowanie modułu DLL .....	573
Jawne wyładowywanie modułu DLL .....	576
Jawne dołączanie do wyeksportowanego symbolu .....	579
Funkcja wejściowa biblioteki .....	580
Powiadomienie DLL_PROCESS_ATTACH .....	581
Powiadomienie DLL_PROCESS_DETACH .....	582

Powiadomienie DLL_THREAD_ATTACH .....	585
Powiadomienie DLL_THREAD_DETACH .....	586
Szeregowane wywołania <i>DllMain</i> .....	587
<i>DllMain</i> a biblioteka uruchomieniowa C/C++ .....	589
Ładowanie biblioteki DLL z opóźnieniem .....	590
Aplikacja przykładowa <i>DelayLoadApp</i> .....	596
Przekierowania funkcji .....	602
Znane biblioteki DLL .....	602
Przekierowywanie bibliotek DLL .....	604
Zmienianie adresów bazowych modułów .....	605
Wiązanie modułów .....	611
<b>21 Pamięć lokalna dla wątku .....</b>	<b>615</b>
Dynamiczna pamięć TLS .....	616
Korzystanie z dynamicznej pamięci TLS .....	618
Statyczna pamięć TLS .....	620
<b>22 Wszczepianie bibliotek DLL i podczepianie API.....</b>	<b>623</b>
Wszczepianie biblioteki DLL: przykład.....	624
Wszczepianie biblioteki DLL przy użyciu rejestru .....	626
Wszczepianie biblioteki DLL przy użyciu haków Windows .....	627
Narzędzie Desktop Item Position Saver (DIPS) .....	629
Wszczepianie biblioteki DLL przy użyciu zdalnych wątków.....	638
Aplikacja przykładowa <i>Inject Library</i> .....	642
Biblioteka DLL <i>Image Walk</i> .....	647
Wszczepianie biblioteki DLL przy użyciu konia trojańskiego.....	649
Wszczepianie biblioteki DLL przez debugger .....	650
Wszczepianie kodu przy użyciu <i>CreateProcess</i> .....	650
Podczepianie API: przykład.....	651
Podczepianie API poprzez nadpisywanie kodu .....	652
Podczepianie API poprzez manipulowanie sekcją importu modułu .....	653
Aplikacja przykładowa <i>Last MessageBox Info</i> .....	656
<b>Część V:Strukturalna obsługa wyjątków.....</b>	<b>671</b>
<b>23 Procedury obsługi sytuacji krańcowych .....</b>	<b>673</b>
Zrozumienie procedur obsługi sytuacji krańcowych poprzez przykład .....	674
<i>Funcenstein1</i> .....	675
<i>Funcenstein2</i> .....	675
<i>Funcenstein3</i> .....	677
<i>Funcfurter1</i> .....	678
<b>Czas na Quiz: <i>FuncuDoodleDoo</i></b> .....	679
<i>Funcenstein4</i> .....	680
<i>Funcarama1</i> .....	681
<i>Funcarama2</i> .....	682
<i>Funcarama3</i> .....	682
<i>Funcarama4: ostatnia granica</i> .....	683

Uwagi na temat bloku <i>finally</i> .....	685
<i>Funcfurter2</i> .....	686
Aplikacja przykładowa SEH Termination .....	687
<b>24 Procedury obsługi wyjątków i wyjątki programowe .....</b>	<b>693</b>
Zrozumienie filtrów wyjątków i procedur obsługi wyjątków poprzez przykład .....	694
<i>Funcmeister1</i> .....	694
<i>Funcmeister2</i> .....	694
<b>EXCEPTION_EXECUTE_HANDLER</b> .....	697
Kilka użytecznych przykładów .....	698
Globalne odkręcanie .....	701
Zatrzymywanie globalnego odkręcania .....	704
<b>EXCEPTION_CONTINUE_EXECUTION</b> .....	705
Ostrożne korzystanie z <b>EXCEPTION_CONTINUE_EXECUTION</b> .....	706
<b>EXCEPTION_CONTINUE_SEARCH</b> .....	707
<i>GetExceptionCode</i> .....	709
Wyjątki związane z pamięcią .....	709
Wyjątki związane z wyjątkami .....	710
Wyjątki związane z debugowaniem .....	710
Wyjątki związane z liczbami całkowitymi .....	710
Wyjątki związane z liczbami zmiennoprzecinkowymi .....	710
<i>GetExceptionInformation</i> .....	713
Wyjątki programowe .....	717
<b>25 Nieobsłużone wyjątki, ukierunkowana obsługa wyjątków i wyjątki C++</b>	<b>721</b>
Wewnątrz funkcji <i>UnhandledExceptionFilter</i> .....	724
Działanie numer 1: pozwalanie na zapisanie zasobu i kontynuowanie wykonania .....	724
Działanie numer 2: powiadamianie debugera o nieobsłużonym wyjątku .....	724
Działanie numer 3: powiadamianie globalnie ustawionej funkcji filtra .....	724
Działanie numer 4: powiadamianie debugera o nieobsłużonym wyjątku (znowu) .....	725
Działanie numer 5: ciche zakończenie procesu .....	725
<i>UnhandledExceptionFilter</i> i interakcja z usługą WER .....	726
Debugowanie dokładnie na czas .....	730
Aplikacja przykładowa Spreadsheet .....	733
Ukierunkowany wyjątek i procedury obsługi kontynuacji .....	742
Wyjątki C++ a wyjątki strukturalne .....	743
Wyjątki i debugger .....	745
<b>26 Raportowanie błędów i przywracanie aplikacji do normalnego stanu....</b>	<b>749</b>
Konsola raportowania błędów systemu Windows .....	749
Programowe raportowanie błędów systemu Windows .....	752
Wyłączanie generowania i przesyłania raportów .....	754
Dostosowywanie wszystkich raportów dotyczących problemów wewnątrz procesu .....	755
Tworzenie i dostosowywanie raportu o problemie .....	756
Tworzenie niestandardowego raportu dotyczącego problemu: <i>WerReportCreate</i> .....	759
Ustawianie parametrów raportu: <i>WerReportSetParameter</i> .....	760
Dodawanie pliku mini-zrzutu do raportu: <i>WerReportAddDump</i> .....	761
Dodawanie arbitralnych plików do raportu: <i>WerReportAddFile</i> .....	762

<b>Modyfikowanie łańcuchów tekstowych w oknie dialogowym:</b>	
<i>WerReportSetUIOption</i> .....	763
<b>Przekazywanie raportu dotyczącego problemu: <i>WerReportSubmit</i></b> .....	764
<b>Zamykanie raportu dotyczącego problemu: <i>WerReportCloseHandle</i></b> .....	766
<b>Aplikacja przykładowa Customized WER</b> .....	766
<b>Automatyczne wznawianie aplikacji i odzyskiwanie danych</b> .....	772
Automatyczne wznawianie aplikacji .....	772
Wsparcie dla odzyskiwania danych przez aplikację .....	774

**Część VI:.....Dodatki 777**

<b>A Środowisko budowania aplikacji</b> .....	779
<b>Plik nagłówkowy <i>CmnHdr.h</i></b> .....	779
<b>Opcja wersji systemu Microsoft Windows</b> .....	780
<b>Opcje Unicode</b> .....	780
<b>Definicje Windows i poziom ostrzeżeń 4</b> .....	781
<b>Pomocnicze makro <i>pragma message</i></b> .....	781
<b>Makro <i>chINRANGE</i></b> .....	782
<b>Makro <i>chBEGINTHREADEX</i></b> .....	782
<b>Ulepszenie <i>DebugBreak</i> dla platform <i>x86</i></b> .....	783
<b>Tworzenie kodów dla wyjątków programowych</b> .....	784
<b>Makro <i>chMB</i></b> .....	784
<b>Makra <i>chASSERT</i> i <i>chVERIFY</i></b> .....	784
<b>Makro <i>chHANDLE_DLGMSG</i></b> .....	784
<b>Makro <i>chSETDLGICONS</i></b> .....	784
<b>Zmuszanie programu łączącego do szukania funkcji początkowej (<i>w</i>)<i>WinMain</i></b> ....	784
<b>Wsparcie dla kompozycji interfejsu użytkownika przy użyciu dyrektywy <i>pragma</i></b>	785
<b>B Makra rozbijające komunikaty, makra pól potomnych i makra API</b> .....	791
Makra rozbijające komunikaty .....	792
Makra pól potomnych .....	794
Makra API .....	795