

Obsah

1 Úvodem	17
1.1 O struktuře knihy	18
1.2 Použité typografické konvence	19
1.3 Pravidla pro programové příklady	20
2 Síť a jejich práce	21
2.1 IP adresa	21
2.1.1 IPv6 tvar adresy	22
2.2 Paket	22
2.3 Rámec (frame)	23
2.4 Nejpoužívanější protokoly IP, TCP a UDP	23
2.4.1 IP (internet protokol)	24
2.4.2 TCP (transmission internet protokol)	24
2.4.3 UDP (user datagram protokol)	25
2.5 Jak vypadá klasické odesílání z aplikace až po vrstvu sítě TCP/IP?	25
2.6 Stručný popis putování paketu po síti Internet	26
2.6.1 Co je to TCP okno (TCP window)?	26
2.6.2 TCP TIME_WAIT stav	26
2.6.3 NAGLE algoritmus	26
2.7 Co je to port?	27
2.8 DNS (domain name system)	27
2.9 Router	27
2.10 MTU (maximum transmission unit)	28
2.11 Formáty dat LITTLE a BIG endian	28
2.12 Co je to broadcasting?	28
2.13 Co je multicasting?	28
2.14 OOB data (out-off-band data)	29
2.15 QOS (quality of services)	29
2.16 MAC adresa (media access control)	29
2.17 RFC (request for comments)	29

3	Využívané pomocné síťové nástroje z prostředí Microsoft Windows	30
3.1	winipcfg – zjištění IP adresy počítače	30
3.2	netstat – statistika spojení	30
4	Windows Socket	35
4.1	Podpora WinSock pro Microsoft Windows	35
4.2	Objasnění pojmů v přímé souvislosti na knihovnu WinSock	35
4.3	Co je to vlastně socket?	35
4.3.1	Spojově-orientované sockety (stream sockety)	36
4.3.2	Nespojově-orientované sockety (datagramové sockety)	36
4.4	Blokující, neblokující a asynchronní sockety	36
4.4.1	Blokující sockety	36
4.4.2	Neblokující sockety	36
4.4.3	Asynchronní sockety	36
4.5	Priorita skupin socketů (group sockets priority)	37
4.6	Příznak keepalive („nechat přežít“)	37
4.7	Broadcast zpráva (broadcasting)	37
4.8	Zajišťovatelé služeb (Service Providers)	38
4.9	Blokovací funkce	38
4.10	Objekt události (objekt event) [pouze winsock2]	38
4.11	Nevyřízená spojení (backlog)	38
4.12	Maximální počet otevřených socketů	39
4.13	Volba strategie vývoje aplikace v závislosti na množství spojení	39
4.14	Model „centrálního vlákna“	40
4.15	Server versus klient – obecná doporučení	40
4.16	WinSock2 – nové vlastnosti	40
5	Zdrojový kód WinSock	42
5.1	DLL knihovny	42
5.2	Hlavičkové soubory	42
5.2.1	Berkeley	42
5.2.2	Microsoft Windows	42
5.3	Používané LIB soubory	42
5.4	Kompatibilita zdrojového kódu jednotlivých verzí a odlišných platforem	43
5.5	Nově zavedené typy	43
5.6	Rozdíly typové velikostí datových typů mezi verzemi 16 a 32 bitů	43

6 Praktické příklady a zkušenosti s knihovnou WinSock	44
6.1 Ověření čísla aktivované verze při inicializaci WinSock	44
6.2 Získání IP adresy lokální stanice	45
6.3 Převod adresy z vnitřního u_long formátu na tečkovou notaci bez použití inet_ntoa()	47
6.4 Zjištění vlastností socketu	48
6.4.1 Zjištění podpory broadcastingu	48
6.4.2 Naplnění struktury WSAPROTOKOL_INFO [WinSock2]	49
6.5 Daytime klient	50
6.6 Tvorba Daytime serveru (s využitím local loopback adresy) a jeho komunikace s Daytime klientem	53
6.7 Jednoduchý HTTP klient - web prohlížeč (web browser)	57
6.8 Příklad získání informací ze vzdáleného připojeného počítače	62
6.9 Příklad asynchronního zjištění informací o protokolu	64
6.10 Jak správně provést uzavření TCP socketu	66
6.11 SMTP protokol – odeslání e-mailové zprávy	66
6.12 POP3 klient – vybrání emailové schránky	74
6.13 Zjištění aktivních portů vzdáleného počítače	81
6.14 Odeslání broadcast zprávy	86
6.15 Zjištění dostupných transportních protokolů [WinSock2]	91
6.16 Zjištění dostupných jmenných prostorů [WinSock2]	95
6.17 Implementace kompletní WinSock úlohy – program CHAT	97
7 Přehled konstant, struktur a funkcí WinSock	110
7.1 Konstanty knihovny WINSOCK	110
7.1.1 FD_SETSIZE	110
7.1.2 INADDR_BROADCAST	110
7.1.3 INADDR_NONE	110
7.1.4 IPPORT_xxx -identifikátory nejvýznamnějších služeb portů	110
7.1.5 IPPROTO_xxx	111
7.1.6 MAXGETHOSTSTRUCT	111
7.1.7 SOMAXCONN	111
7.1.8 WSADESCRIPTION_LEN	111
7.1.9 WSASYS_STATUS_LEN	112
7.2 Struktury WinSock	112
7.2.1 fd_set	112
7.2.2 hostent	112
7.2.3 linger	113

7.2.4	protoent	113
7.2.5	SERVENT	114
7.2.6	SOCKADDR_IN	115
7.2.7	timeval	116
7.2.8	WSABUF [WinSock2]	116
7.2.9	WSAData	117
7.2.10	WSAEVENT [pouze WinSock2]	118
7.3	Makra	118
7.3.1	FD_CLR(s, *set)	118
7.3.2	FD_ISSET(s, *set)	118
7.3.3	FD_SET(s, *set)	118
7.3.4	FD_ZERO(*set)	119
7.3.5	timerclear	119
7.3.6	timercmp	119
7.3.7	timerisset	119
7.3.8	WSAGETASYNCBUFLLEN	119
7.3.9	WSAGETSELECTERROR	119
7.3.10	WSAGETSELECTEVENT	120

8 Popisy nejvyžívanějších funkcí WinSock..... 121

8.1	accept()	121
8.2	bind()	122
8.3	connect()	123
8.4	gethostbyaddr()	124
8.5	gethostbyname()	124
8.6	getpeername()	125
9.7	getprotobyname()	126
8.8	getprotobynumber()	128
8.9	getservbyname()	128
8.10	getservbyport()	129
8.11	getsockname()	130
8.12	getsockopt()	131
8.13	htonl()	134
8.14	htons()	135
8.15	inet_addr()	135
8.16	inet_ntoa()	136
8.17	ioctlsocket()	137
8.18	listen()	138
8.19	ntohl()	139

8.20 ntohs()	139
8.21 recv()	139
8.22 recvfrom()	140
8.23 select()	140
8.24 setsockopt()	143
8.25 shutdown()	146
8.26 socket()	147
8.27 sendto()	148
8.28 WSAAAsyncGetHostByAddr()	149
8.29 WSAAAsyncGetHostByName()	150
8.30 WSAAAsyncGetProtoByName()	151
8.31 WSAAAsyncGetProtoByNumber()	152
8.32 WSAAAsyncGetServByName()	153
8.33 WSAAAsyncGetServByPort()	154
8.34 WSAAAsyncSelect()	155
8.35 WSACancelBlockingCall() [zrušeno od WinSock 2.2]	157
8.36 WSACleanup()	157
8.37 WSACreateEvent() [pouze WinSock2]	158
8.38 WSAEventSelect() [pouze WinSock2]	158
8.39 WSAGetLastError	159
8.40 WSAIsoBlocking() [pouze WinSock 1.1]	160
8.41 WSAREcvEx()	161
8.42 WSAResetEvent() [pouze WinSock2]	161
8.43 WSASetBlockingHook() [zrušeno od WinSock 2.2]	162
8.44 WSASetEvent() [pouze WinSock2]	163
8.45 WSASetLastError()	163
8.46 WSAStartup()	163
8.47 WSAUnhookBlockingHook() [zrušeno od WinSock 2.2]	164
9 RAW sockety	165
9.1. Hlavičky (headers) protokolů	165
9.1.1 Hlavička IP	165
9.1.2 Hlavička ICMP	169
9.1.3 Hlavička TCP	170
9.1.4 Komunikace TCP spojení	174
9.1.5 Hlavička UDP	175
9.1.6 Ethernet	176
9.1.7 Tvar typického paketu ethernetového TCP/IP	176
9.1.8 Co se děje s ethernetovým paketem na straně routeru?	177

9.1.9	Jak probíhá příjem typického ethernetového paketu?	177
9.1.10	Postup programového vytvoření paketu	177
9.2	Doporučené zdroje dalších informací	180
10	Jmenný prostor (name space) [WinSock2]	181
10.1	Poskytovatel jmenných prostorů (name space provider)	181
10.2	GUID (globally unique identifier)	181
10.2.1	uuidgen.exe	181
10.2.2	Převodní rutina z ukazatele GUID struktury na ukazatel na konstantní řetězec	182
11	Rozšíření WinSock2 pro protokoly TCP/IP - ws2tcpip.h	183
11.1	Základní pravidla pro používání socketů při formátu IPv6	183
12	Chybová hlášení Windows Sockets	184
12.1	Obecné formáty chyb	184
12.2	Seznam chybových konstant	184
12	Chybová hlášení Windows Sockets	184
12.1	Obecné formáty chyb	184
12.2	Seznam chybových konstant	184
13	Overlapped I/O rozhraní [WinSock2]	188
13.1	Výhody overlapped přístupu	188
13.2	Obecný popis overlapped I/O z programátorského hlediska	188
13.3	Užitečné overlapped funkce a makra	190
13.3.1	Canceled()	190
13.3.2	HasOverlappedIoCompleted	191
13.4	WinSock a overlapped interface	194
13.5	Podrobnější popis uvedených rozdílů	195
13.5.1	Struktura WSAOVERLAPPED	195
13.5.2	Prototyp LPWSAOVERLAPPED_COMPLETION_ROUTINE	195
14	Podpora Windows Socket v MFC a VCL	196

15	Winsock ve VISUAL C++ 6.0 MFC	197
15.1	Umístění zdrojů	197
15.2	Globální funkce	197
15.2.1	AfxSocketInit()	197
15.2.2	AfxSocketTerm()	198
15.3	Třídy	198
15.3.1	CAsyncSocket	198
15.3.2	CSocket	200
15.3.3	CSocketFile	200
15.3.4	CSocketWnd	200
15.4	Praktický příklad využití MFC Socket	200
15.4.1	MFC Socket inicializace	200
15.4.2	DAYTIME klient	200
16	WinSock v DELPHI 5 VCL	203
16.1	Umístění zdrojů	203
16.2	Nové důležité typy	203
16.2.1	ESocketError (exception)	203
16.2.2	TClientType	203
16.2.3	TErrorEvent	203
16.2.4	TServerType	204
16.2.5	TSocketEvent	204
16.3	Nové zprávy CM_SOCKETMESSAGE, CM_DEFERFREE a CM_LOOKUPCOMPLETE	205
16.4	Globální procedury a funkce	205
16.4.1	SetErrorProc()	205
16.4.2	Očekávané globální procedury a funkce	206
16.5	Třídy	206
16.6	Praktické ukázky využití DELPHI 5 VCL	207
16.6.1	Stažení HTTP source přes neblokovaný TClientSocket	207
16.6.2	Komunikace TServerSocket a TClientSocket	210
17	ICMP protokol (ICMP.DLL)	212
17.1	Hlavičkový soubor pro ICMP.DLL	212
17.1.1	ip_option_information	212
17.1.2	icmp_echo_reply	213
17.2	Funkce ICMP.DLL	214

17.2.1	IcmpCloseHandle()	214
17.2.2	IcmpSendEcho()	214
17.2.3	IcmpCreateFile()	216
17.3	Praktické příklady ICMP.DLL	216
17.3.1	Odeslání ICMP ping echo požadavku přes ICMP.DLL	216
17.3.2	trace – program pro sledování postupu paketu v síti	221
18	NETBIOS – (network basic input output system)	227
18.1	Identifikace stanice v NETBIOS	228
18.2	Soubor LMHOSTS	228
18.3	NETBIOS datagramy (datagram services)	229
18.4	NETBIOS session (session services)	230
18.5	NCB (network control block)	230
18.6	NETBEUI	230
19	NETBIOS z bližšího pohledu	231
19.1	Umístění Netbios	231
19.2	Důležité konstanty	231
19.2.1	NCBNAMSZ	231
19.2.2	MAX_LANA	231
19.3	Důležité struktury	231
19.3.1	ADAPTER_STATUS	231
19.3.2	LANA_ENUM	232
19.3.3	NCB	232
19.3.4	SESSION_HEADER, SESSION_BUFFER	237
19.4	Nastavení asynchronního tvaru požadavku – ASYNCH	238
19.5	Funkce Netbios()	238
20	Praktické ukázky	239
20.1	Zjištění MAC adres adaptérů LANA	239
20.2	Reset příslušného LAN adaptéru s alokací nových zdrojů (NCBRESET)	243
20.3	Přidání jména (NCBADDNAME)	245
21	Programování v doménách lokálních sítí NT	246
21.1	Identifikace NT síťových funkcí	246

21.2	WNet versus Net	246
21.3	Umístění hlavičkových a LIB souborů	247
21.4	Hierarchie NT sítě	247
21.5	Nové konstanty	247
21.5.1	NET_API_STATUS	248
21.6	Nové struktury	248
21.6.1	CONNECTDLGSTRUCT struktura	248
21.6.2	NETRESOURCE struktura	249
21.7	Praktické programátorské příklady WNet (enumerace dostupných prostředků sítě, sdílení, vzdálené jednotky)	251
21.7.1	Enumerace dostupných zařízení v síti NT	251
21.7.2	Zobrazení dialogu pro připojení síťové jednotky	257
21.7.3	Programové připojení síťové jednotky bez dialogu	258
21.7.4	Odpojení připojené síťové jednotky bez dialogu	260
21.7.5	Zjištění vzdáleného jména lokální jednotky	261
21.7.6	Zjištění jména poskytovatele specifikované sítě	263
21.7.7	Další způsob připojení síťové jednotky dialogem s možností procházení sítě	264
21.7.8	Zjištění čísla a popisu speciální chyby ERROR_EXTENDED_ERROR	267
21.7.9	Zjištění UNC pojmenování souboru z lokálně připojené síťové jednotky	268
21.7.10	Zjištění jména pod jakým jsme přihlášení ke vzdálenému disku + zjištění jména lokální stanice	271
21.8	Praktické programátorské příklady Net funkcí	272
21.8.1	Enumerace uživatelů NT serveru	273
21.8.2	Přidání nového NT uživatele	277
21.8.3	Zjištění informací o konkrétním uživateli s jeho následným smazáním	278
21.8.4	Enumerace serverů sítě - resp. připojených stanic	279
21.8.5	Zjednodušená enumerace pomocí NetQueryDisplayInformation()	281
22	WinInet (windows internet functions)	284
22.1	Umístění WinInet	284
22.2	Chybové stavy	284
22.3	Způsoby přístupu do internetové sítě	284
22.4	Požadavky HTTP (HTTP requests)	285
22.5	Asynchronní zpracování	285

22.6	Stručně o WinInet v MFC VISUAL C++	285
22.7	Protokol FTP	286
23	WinInet nejdůležitější typy a struktury	288
23.1	HINTERNET	288
23.2	URL_COMPONENTS struct	288
24	Praktické příklady	289
24.1	Obecné příklady	289
24.1.1	Test na připojení k Internetu	289
24.1.2	Zjištění způsobu aktivního připojení internetové linky	290
24.1.3	Inicializace WinInet knihovny – získání handle internetového spojení	292
24.1.4	Získání informací o specifické URL	293
24.1.5	Zjištění a změna nastavených parametrů pro síť Internet	296
24.1.6	“Kanonizace” URL – odstranění nebezpečných znaků z URL ...	299
24.1.7	Implementace callback funkce	301
24.2	Praktické ukázky HTTP	309
24.2.1	Připojení k HTTP stránce a vypsání jejího zdrojového kódu	309
24.2.2	Stažení gif obrázku z HTTP serveru přes “GET” požadavek	311
24.3	Praktické ukázky FTP	321
24.3.1	Připojení k FTP serveru – jednoduchý FTP klient	321
24.3.2	Zjištění adresářů a souborů v aktuálním adresáři připojeného FTP	323
24.3.3	Stažení vzdáleného souboru z FTP serveru na lokální počítač ...	326
24.3.4	Vložení souboru na FTP server	327
25	Programování prohlížeče Microsoft Internet Explorer	329
25.1	Potřebné soubory	329
25.2	Praktická ukázka	329
25.2.1	Dynamické vytvoření prohlížeče a dynamický kontakt s prvky HTML stránky z programu	329
26	RAS API (remote access services – služby vzdáleného přístupu)	338
26.1	Fyzické umístění knihovny a zdrojových souborů	338
26.2	Phonebook soubory	338

26.3	RAS struktury	339
29.3.1	struct RASDEVINFO	339
26.3.2	RASMONITORDLG	340
26.4	Praktické ukázky	341
26.4.1	Zjištění připojených nainstalovaných RAS zařízení	341
26.4.2	Funkce zjišťující počet nainstalovaných modemů	343
26.4.3	Založení nového phonebooku	345
26.5	Zobrazení monitoru	346
	Rejstřík	348