

Obsah

Úvod	10
Typografické a syntaktické konvence	12
Obecně platné poznámky	12
I ANSI C	15
1 Komplexní pohled na souborový vstup a výstup	16
1.1 Rozdíl mezi binárním a textovým souborem	17
1.1.1 Binární soubory	17
1.1.2 Textové soubory	18
1.2 Otevření souboru	21
1.2.1 Různé způsoby otevírání souboru	22
1.3 Uzavření souboru	24
1.4 Ošetření chyb	27
1.5 Čtení a zápis dat	29
1.5.1 Formátované vstupy a výstupy	29
1.5.2 Neformátované vstupy a výstupy	30
1.6 Bufferování	37
1.6.1 Základní informace	37
1.6.2 Možnosti bufferování poskytnuté ANSI C	39
1.6.3 Funkce pro nastavení velikosti bufferu	39
1.6.4 Řešení problému nebufferovaného <code>stdin</code>	42
1.7 Funkce pro zápis bufferu na disk	44
1.7.1 Funkce <code>fflush()</code>	44
1.8 Přímý přístup do souboru	45
1.8.1 Funkce <code>fseek()</code> – posun v souboru	45
1.8.2 Funkce <code>ftell()</code> – zjištění pozice v souboru	46
1.8.3 Typické použití přímého přístupu	47

1.9	Zbývající užitečné funkce ze <code><stdio.h></code>	51
1.9.1	Funkce <code>freopen()</code> – přesměrování proudu	51
1.9.2	Funkce <code>rename()</code> – přejmenování souboru	52
1.9.3	Funkce <code>remove()</code> – zrušení souboru	53
1.9.4	Funkce <code>tmpfile()</code> – pomocný soubor	53
1.9.5	Funkce <code>tmpnam()</code> – jméno pomocného souboru	53
1.10	Ukázka použití funkcí z UNIXové knihovny	54
2	Návaznost překladače ANSI C na okolní prostředí	56
2.1	Standardní hlavičkové soubory	56
2.2	Rezervovaná jména	58
2.3	Standardní hlavičkové soubory bez funkčních prototypů	59
2.3.1	Soubor <code><errno.h></code>	60
2.3.2	Soubor <code><limits.h></code>	60
2.3.3	Soubor <code><float.h></code>	61
2.3.4	Soubor <code><stddef.h></code>	61
2.4	Předdefinovaná makra preprocesoru	62
2.4.1	Makro <code>__STDC__</code>	62
2.4.2	Makro <code>__FILE__</code>	63
2.4.3	Makro <code>__LINE__</code>	63
2.4.4	Makro <code>__DATE__</code>	63
2.4.5	Makro <code>__TIME__</code>	63
2.5	Externí proměnné	63
2.5.1	Proměnná <code>_ctype</code> z <code><ctype.h></code>	64
2.5.2	Proměnná <code>errno</code> z <code><errno.h></code>	64
2.5.3	Proměnná <code>sys_errlist</code> ze <code><stdlib.h></code>	65
2.5.4	Proměnná <code>sys_nerr</code> ze <code><stdlib.h></code>	66
2.5.5	Proměnná <code>environ</code> ze <code><stdlib.h></code>	66
2.5.6	Proměnná <code>_streams</code> ze <code><stdio.h></code>	67
3	Popis funkcí a maker ze standardních knihoven	68
3.1	<code><assert.h></code> – makro používané při ladění	68
3.2	<code><ctype.h></code> – práce se znaky	68
3.3	<code><math.h></code> – matematické funkce v přesnosti double	69
3.4	<code><locale.h></code> – přizpůsobení C národnímu prostředí	71
3.5	<code><setjmp.h></code> – umožnění běžně nedovolených skoků	71
3.6	<code><signal.h></code> – zpracování signálů	74
3.7	<code><stdarg.h></code> – práce s proměnným počtem parametrů	76
3.8	<code><stdio.h></code> – funkce pro vstup a výstup	76
3.9	<code><stdlib.h></code> – obecně užitečné funkce	77
3.9.1	Konverze řetězců na čísla	77

3.9.2	Generátor pseudonáhodných čísel	79
3.9.3	Funkce pracující s dynamickou pamětí	80
3.9.4	Funkce pro spolupráci s operačním systémem	80
3.9.5	Funkce pro hledání a třídění	82
3.10	<string.h> – zpracování řetězců	87
3.10.1	Funkce <code>strspn()</code>	88
3.10.2	Funkce <code>strcspn()</code>	88
3.10.3	Funkce <code>strpbrk()</code>	88
3.10.4	Funkce <code>strtok()</code>	88
3.10.5	Funkce <code>memchr()</code>	89
3.10.6	Funkce <code>memcmp()</code>	89
3.10.7	Funkce <code>memcpy()</code>	90
3.10.8	Funkce <code>memmove()</code>	90
3.10.9	Funkce <code>memset()</code>	90
3.11	<time.h> – práce s datem a časem	92
3.11.1	Funkce <code>clock()</code>	93
3.11.2	Funkce <code>time()</code>	94
3.11.3	Funkce <code>difftime()</code>	94
3.11.4	Funkce <code>localtime()</code>	94
3.11.5	Funkce <code>asctime()</code>	95
3.11.6	Funkce <code>ctime()</code>	95
3.11.7	Funkce <code>mktime()</code>	96
4	Funkce s proměnným počtem parametrů	98
4.1	Úvodní informace	98
4.2	Základní princip	100
4.3	Standardní hlavičkový soubor <code><stdarg.h></code>	101
4.4	Funkce <code>vprintf()</code> , <code>vfprintf()</code> , <code>vsprintf()</code>	102
4.5	Praktické příklady použití FPPP	103
4.5.1	Typické použití <code>vprintf()</code> , <code>vfprintf()</code> , <code>vsprintf()</code>	103
4.5.2	Přesměrování výstupu za běhu programu	105
4.5.3	Seznam parametrů ukončen domluvenou hodnotou	108
4.5.4	Počet parametrů je předán v pevném parametru	109
4.5.5	Odstrašující případ zcela netypického použití	110
5	Ladění v jazyce C	113
5.1	Ladicí výpisy na <code>stderr</code>	115
5.2	Využití podmíněného překladu pomocí příkazů C-preprocesoru	116
5.2.1	Podm. překlad využívající existence symb. konstanty	117
5.2.2	Podm. překlad využívající hodnotu symb. konstanty	118
5.3	Self-testy modulů nebo funkcí	121

5.4	Využití makra ze souboru <code><assert.h></code>	124
5.4.1	Princip <i>asercce</i>	127
5.4.2	Chybné použití <i>asercce</i>	128
5.5	Prostředky pro ladění dynamických procesů	130
II	Borland C	141
6	Využití registrů procesoru z Borland C	142
6.1	Registrové pseudoproměnné	142
6.2	Union REGS	144
6.3	Struktura SREGS	144
6.4	Struktura REGPACK	145
6.5	Registrové proměnné	146
7	Volání přerušení (služeb) MS-DOSu	147
7.1	Makro <code>geninterrupt()</code>	147
7.2	Funkce <code>int86()</code>	148
7.3	Funkce <code>int86x()</code>	149
7.4	Funkce <code>intr()</code>	150
7.5	Funkce <code>intdos()</code>	151
7.6	Funkce <code>intdosx()</code>	152
7.7	Funkce <code>bdos()</code>	152
8	Práce s porty a s CMOS	154
8.1	Porty	154
8.2	Paměť CMOS	154
9	Funkce pro obsluhu přerušení	156
9.1	Základní znalosti	156
9.2	Ukázka obsluhy časového přerušení	157
9.3	Spolupráce obsluhy přerušení s programem	159
10	Near, far, huge a ostatní typy ukazatelů	163
10.1	Obecné informace	163
10.2	Ukazatel typu <i>near</i>	164
10.2.1	Modifikátory segmentu ukazatele typu <i>near</i>	165
10.3	Ukazatel typu <i>far</i>	165
10.4	Ukazatel typu <i>huge</i>	166
10.5	Ukazatel typu <code>_seg</code>	168
10.6	Funkční modifikátory <i>near</i> , <i>far</i> a <i>huge</i>	169

11 Paměti v PC a jejich programové ovládání	170
11.1 Základní přehled problematiky	170
11.1.1 Konvenční paměť	171
11.1.2 Rezervovaná paměť – <i>Upper Memory</i>	171
11.1.3 Paměť UMB – <i>Upper Memory Blocks</i>	171
11.1.4 Stránková paměť – <i>Expanded Memory</i>	172
11.1.5 Rozšířená paměť – <i>Extended Memory</i>	172
11.1.6 Paměť HMA – <i>High Memory Area</i>	172
11.1.7 Paměť EMB – <i>Extended Memory Blocks</i>	173
11.1.8 Stínová paměť – <i>Shadow RAM</i>	173
11.2 Programové ovládání pamětí EMS a EMB	173
11.2.1 Programové ovládání EMS	174
11.2.2 Programové ovládání EMB	188
A Minimální znalosti při programování pod UNIXem	199
A.1 Editor	200
A.2 Překladač a sestavovací program	200
A.3 Kontrola zdrojových textů v C	202
A.4 Práce na velkých projektech	203
A.5 Ladicí prostředky	207
B Malé úvahy o velkých projektech	210
B.1 Fáze vývoje velkých programů	211
B.2 Přesná specifikace programu	211
B.3 Návrh hlavních částí programu	213
B.4 Plán projektu a odhad nákladů	214
B.5 Smlouva s uživatelem	217
B.6 Určení způsobu spolupráce	218
B.7 Volba softwarových nástrojů	220
B.8 Určení způsobu ladění	221
B.9 Testování	222
B.10 Analýza výkonnosti a optimalizace	224
B.11 Zpracování dokumentace	225
B.12 Předání programu	227
B.13 Údržba programu	227
C Desatero příkázání pro programátory v C	229
Literatura	231
Rejstřík	232