



北京希望电脑公司 Turbo C++ 技术丛书

TURBOC



张玉亭
韩兰 编译
木林

技术参考大全

海洋出版社

前 言

由美国Borland公司推出的Turbo C目前已发展成为最通用且最受欢迎的C编译程序之一。Turbo C特别以其编译速度之快,所产生的代码有效性之强,与ANSI C兼容性之高而著称。

随着编程技术的提高和不断改进,对编程质量的要求也随之增强。为此,一种新型的编程方法——面向对象的编程(OOP)应运而生。Borland便开始开发Turbo C++,将C++ OOP的基本思想和设计方法溶入功能极强的Turbo C中。Turbo C++继承并发挥了原有Turbo C集成环境的优良特性,并增强了对面向对象(OOP)编程的支持,是目前国际上最受欢迎的面向对象程序软件包。

我们向读者介绍的这本有关Turbo C/C++的参考大全,对这两种语言的各种功能进行详细讲解,并列了众多的实例,以供各个层次的程序员之用。

由于时间仓促,加之我们水平有限,书中难免有不足或疏漏之处,敬请广大读者批评指正。

在本书的出版过程中,得到了海洋出版社的编辑同志,及北京希望电脑公司的热情帮助和大力支持,在此谨致谢意。

编译者

1991.5

目 录

第一部分 C 语言

第一章: C语言回顾

1.1 C语言的起源	(3)
1.2 中级语言	(3)
1.3 结构化语言	(4)
1.4 程序员语言	(5)
1.5 编译程序和解释程序	(6)
1.6 C程序格式	(6)
1.7 术语	(8)

第二章 变量、常数、运算符和表达式

2.1 标识符名	(9)
2.2 数据类型	(10)
2.3 变量说明	(12)
2.4 存储类说明符	(15)
2.5 赋值语句	(19)
2.6 常数	(21)
2.7 运算符	(22)
2.8 表达式	(32)

第三章 程序控制语句

3.1 C语言中的真与假	(35)
3.2 C语言语句	(35)
3.3 条件语句	(35)
3.4 if	(35)
3.5 switch	(40)
3.6 循环	(43)
3.7 for	(43)
3.8 while	(47)
3.9 do/ while	(48)
3.10 break	(49)
3.11 exit ()	(50)
3.12 continue	(51)
3.13 标号与goto	(52)

第四章 函数

4.1 返回语句	(54)
4.2 函数的作用域规则	(56)
4.3 函数变元	(56)
4.4 main () 的变元	(61)
4.5 返回非整型值的函数	(65)
4.6 使用函数原型	(66)
4.7 返回指针	(68)
4.8 传统与现代参数说明	(70)
4.9 递归	(71)
4.10 指向函数的指针	(72)
4.11 补充事宜	(74)

第五章 数组

5.1 一维数组	(75)
5.2 将一维数组传递给函数	(76)
5.3 二维数组	(78)
5.4 多维数组	(82)
5.5 数组和指针	(83)
5.6 已分配数组	(84)
5.7 数组初始化	(87)
5.8 举例	(89)

第六章 指针

6.1 指针是地址	(91)
6.2 指针变量	(92)
6.3 指针运算符	(92)
6.4 指针表达式	(93)
6.5 Turbo C 的动态分配函数	(96)
6.6 指针和数组	(97)
6.7 指向指针的指针	(100)
6.8 初始化指针	(101)
6.9 指向函数的指针	(102)
6.10 指针带来的问题	(104)

第七章 结构、联合和用户定义变量

7.1 结构	(105)
--------------	---------

7.2	结构数组.....	(107)
7.3	将结构传递给函数.....	(112)
7.4	结构指针.....	(114)
7.5	结构中的数组和结构.....	(117)
7.6	位字段.....	(117)
7.7	联合.....	(119)
7.8	枚举.....	(120)
7.9	使用size-of, 确保可移植性	(122)
7.10	类型定义	(123)

第八章 输入、输出和磁盘文件

8.1	流和文件.....	(124)
8.2	ANSI C I/O系统	(125)
8.3	控制台 I/O	(126)
8.4	格式化控制台 I/O	(128)
8.5	ANSI C 文件系统	(133)
8.6	UNIX 类文件例程	(144)
8.7	选择一种方案.....	(149)

第九章 Turbo C预处理器

9.1	Turbo C预处理器	(150)
9.2	# define	(150)
9.3	# error	(152)
9.4	# include	(152)
9.5	条件编译伪指令.....	(153)
9.6	# undef.....	(156)
9.7	# line	(156)
9.8	# pragma	(157)
9.9	预定义宏名.....	(158)

第十章 Turbo C存储模式

10.1	8086系列处理器	(160)
10.2	地址计算	(161)
10.3	near和far指针.....	(161)
10.4	存储模式	(162)
10.5	类型修饰符far, near和huge	(164)
10.6	Turbo C的段说明符	(165)
10.7	存储器显示和更改程序	(165)

第十一章 Turbo C 屏幕和图形函数

- 11.1 PC 视频适配器和操作方式 (168)
- 11.2 文本屏幕函数 (169)
- 11.3 Turbo C 的图形函数 (178)

第二部分 Turbo C 环境

第十二章 Turbo C 综合程序设计环境

- 12.1 执行 Turbo C (192)
- 12.2 主菜单 (192)
- 12.3 编辑和信息窗口 (197)
- 12.4 热键 (197)
- 12.5 TCINST 程序 (199)

第十三章 Turbo C 文本编辑程序

- 13.1 编辑程序命令 (201)
- 13.2 调用编辑程序并输入文本 (201)
- 13.3 删除字符、字和行 (203)
- 13.4 移动、拷贝和删除文本块 (203)
- 13.5 再谈光标移动 (204)
- 13.6 查找 (Find) (204)
- 13.7 保存和装入文件 (205)
- 13.8 理解自动缩排 (205)
- 13.9 将文本块移至 / 移出磁盘文件 (206)
- 13.10 成双匹配 (206)
- 13.11 其它命令 (206)
- 13.12 用文件名调用 Turbo C (207)
- 13.13 命令一览表 (207)

第十四章 编译程序和链接程序选择项

- 14.1 综合开发环境选择项 (209)
- 14.2 编译程序选择项 (209)
- 14.3 链接程序选择项 (214)
- 14.4 环境选择项 (215)
- 14.5 目录选择项 (216)
- 14.6 变元 (216)
- 14.7 保存和装入选择项 (216)

14.8	Turbo C 的命令行版本	(217)
14.9	TLINK, Turbo C 独立链接程序	(218)

第三部分 Turbo C 库

第十五章 链接、库和头文件

15.1	链接程序	(223)
15.2	C 标准库	(224)
15.3	头文件	(224)

第十六章 I/O 函数

16.1	int access(const char *filename, int mode)	(226)
16.2	int chmod(const char *filename, int get_set, int attrib)	(227)
16.3	int chmod(const char *filename, int mode)	(228)
16.4	int chsize(char handle, long size)	(228)
16.5	void clearerr(FILE *stream)	(229)
16.6	int close(int fd) int _close(int fd)	(230)
16.7	int creat(const char *filename, int pmode) int _creat(const char *filename, int attrib) int creatnew(const char *filename, int attrib) int creattemp(const char *filename, int attrib)	(231)
16.8	int dup(int handle) int dup2(int old_handle, int new_handle)	(232)
16.9	int eof(int fd)	(233)
16.10	int fclose(FILE *stream) int fcloseall(void)	(234)
16.11	FILE *fdopen(int handle, char *mode)	(234)
16.12	int feof(FILE *stream)	(235)
16.13	int ferror(FILE *stream)	(235)
16.14	int fflush(FILE *stream)	(236)
16.15	int fgetc(FILE *stream)	(236)
16.16	int fgetchar(void)	(237)
16.17	int *fgetpos(FILE *stream, fpos_t *pos)	(237)
16.18	char *fgets(char *str, int num, FILE *stream)	(238)
16.19	long filelength(int handle)	(239)
16.20	int fileno(FILE *stream)	(239)
16.21	int flushall(void)	(239)

16.22	FILE *fopen (const char *fname, const char *mode)	(240)
16.23	int fprintf (FILE *stream, const char *format, arg-list)	(241)
16.24	int fputc (int ch, FILE *stream)	(242)
16.25	int fputchar (int ch)	(243)
16.26	int fputs (const char *str, FILE *stream)	(243)
16.27	size_t fread (void *buf, size_t size, size_t count, FILE *stream)	(244)
16.28	FILE *freopen (const char *fname, const char *mode, FILE *stream)	(244)
16.29	int fscanf (FILE *stream, const char *format, arg-list)	(245)
16.30	int fseek (FILE *stream, long offset, int origin)	(245)
16.31	int fsetpos (FILE *stream, const fpos_t *pos)	(247)
16.32	int fstat (int handle, struct stat *statbuf)	(247)
16.33	long ftell (FILE *stream)	(248)
16.34	size_t fwrite (const void *buf, size_t size, size_t count, FILE *stream)	(249)
16.35	int getc (FILE *stream)	(249)
16.36	int getch (void) int getche (void)	(250)
16.37	int getchar (void)	(251)
16.38	char *gets (char *str)	(251)
16.39	int getw (FILE *stream)	(252)
16.40	int isatty (int handle)	(253)
16.41	int lock (int handle, long offset, long length)	(253)
16.42	long lseek (int handle, long offset, int origin)	(253)
16.43	int open (const char *filename, int access, unsigned mode) int _open (const char *filename, int access)	(255)
16.44	void perror (const char *str)	(257)
16.45	int printf (const char *format, arg-list)	(257)
16.46	int putc (int ch, FILE *stream)	(259)
16.47	int putchar (int ch)	(260)
16.48	int puts (const char *str)	(260)
16.49	int putw (int i, FILE *stream)	(261)
16.50	int read (int fd, void *buf, unsigned count) int _read (int fd, void *buf, unsigned count)	(261)
16.51	int remove (const char *fname)	(262)

16.52	int rename(const char *oldfname, const char *newfname);	(262)
16.53	void rewind(FILE *stream)	(263)
16.54	int scanf(const char *format, arg-list)	(264)
16.55	void setbuf(FILE *stream, char *buf)	(267)
16.56	int setmode(int handle, int mode)	(267)
16.57	int setvbuf(FILE *stream, char *buf, int mode, size_t size)	(267)
16.58	int sopen(const char *filename, int access, int shflag, int mode)	(268)
16.59	int sprintf(char *buf, const char *format, arg-list)	(269)
16.60	int sscanf(char *buf, const char *format, arg-list)	(270)
16.61	int stat(char *filename, struct stat *statbuf)	(270)
16.62	long tell(int fd)	(271)
16.63	FILE *tmpfile(void)	(271)
16.64	char *tmpnam(char *name)	(272)
16.65	int ungetc(int ch, FILE *stream)	(272)
16.66	int unlink(const char *fname)	(273)
16.67	int unlock(int handle, long offset, long length)	(274)
16.68	int vprintf(const char *format, va_list arg_ptr) int vfprintf(FILE *stream, const char *format, va_list arg_ptr) int vsprintf(char *buf, const char *format, va_list arg_ptr)	(274)
16.69	int vscanf(const char *format, va_list arg_ptr) int vfscanf(FILE *stream, const char *format, va_list arg_ptr) int vsscanf(const char *buf, const char *format, va_list arg_ptr)	(275)
16.70	int write(int handle, void *buf, int count) int _write(int handle, void *buf, int count)	(276)

第十七章 字符串、存储器和字符函数

17.1	int isalnum(int ch)	(277)
17.2	int isalpha(int ch)	(277)
17.3	int isascii(int ch)	(278)
17.4	int iscntrl(int ch)	(278)
17.5	int isdigit(int ch)	(279)
17.6	int isgraph(int ch)	(280)
17.7	int islower(int ch)	(280)
17.8	int isprint(int ch)	(281)

17.9	int ispunct(int ch)	(281)
17.10	int isspace(int ch)	(282)
17.11	int isupper(ch)	(282)
17.12	int isxdigit(int ch)	(283)
17.13	void *memcpy(void *dest,const void *source, int ch, size_t count);	(284)
17.14	void *memchr(const void *buffer,int ch,size_t count)	(284)
17.15	int memcmp(const void *buf1,const void *buf2,size_t count) int memicmp(const void *buf1, const void *buf2,size_t count)	(285)
17.16	void *memcpy(void *dest,const void *source,size_t count)	(286)
17.17	void *memmove(void *dest,const void *source,size_t count)	(286)
17.18	void *memset(void *buf,int ch,size_t count)	(287)
17.19	void movedata(unsigned sourceseg,unsigned sourceoff, unsigned destseg,unsigned destoff,size_t count)	(287)
17.20	void movemem(void *source,void *dest,unsigned count)	(288)
17.21	void setmem(void *buf,unsigned count,char ch)	(288)
17.22	char *strcpy(char *str1,const char *str2)	(288)
17.23	char *strcat(char *str1,const char *str2)	(289)
17.24	char *strchr(const char *str,int ch)	(289)
17.25	int strcmp(const char *str1,const char *str2)	(290)
17.26	int strcoll(char *str1,char *str2)	(290)
17.27	char *strcpy(char str1,const char *str2)	(291)
17.28	size_t strspn(const char *str1,const char *str2)	(291)
17.29	char *strdup(const char *str)	(291)
17.30	char *_strerror(const char *str)	(292)
17.31	char *strerror(int num)	(292)
17.32	int stricmp(const char *str1,const char *str2) int strcmpi (const char *str1,const char *str2)	(293)
17.33	size_t strlen(const char *str)	(293)
17.34	char *strlwr(char *str)	(294)
17.35	char *strncat(char *str1,const char *str2,size_t count)	(294)
17.36	int strncmp(const char *str1,const char *str2,size_t count) int strnicmp(const char *str1,const char *str2,size_t count) int strncmpi(const char *str1,const char *str2, size_t count)	(295)

17.37	char *strncpy(char *dest, const char *source, size_t count)	(296)
17.38	char *strnset(char *str,int ch,size_t count);	(296)
17.39	char *strpbrk(const char *str1,const char str2)	(297)
17.40	char *strchr(const char *str,int ch)	(297)
17.41	char *strrev(char *str)	(298)
17.42	char *strset(char *str,int ch)	(298)
17.43	size_t strspn(const char *str1,const char *str2)	(298)
17.44	char *strstr(const char *str1, const char *str2)	(299)
17.45	char *strtok(char *str1,const char *str2)	(299)
17.46	char *strupr(char *str)	(300)
17.47	size_t strxfrm(char *dest,const char *source,size_t count)	(301)
17.48	int tolower(int ch) int_tolower(int ch)	(301)
17.49	int toupper(int ch) int_toupper(int ch)	(301)

第十八章 数学函数

18.1	double acos(double arg)	(302)
18.2	double asin(double arg)	(303)
18.3	double atan(double arg)	(303)
18.4	double atan2(double y,double x)	(304)
18.5	double cabs(struct complex znum)	(304)
18.6	double ceil(double num)	(305)
18.7	double cos(double arg)	(305)
18.8	double cosh(double arg)	(306)
18.9	double exp(double arg)	(306)
18.10	double fabs(double num)	(306)
18.11	double floor(double num)	(307)
18.12	double fmod(double x,double y)	(307)
18.13	double frexp(double num,int *exp)	(307)
18.14	double hypot(double x,double y)	(308)
18.15	double ldexp(double num,int exp)	(308)
18.16	double log(double num)	(308)
18.17	double log10(double num)	(309)
18.18	int matherr(struct exception *err)	(310)
18.19	double modf(double num,int *i)	(310)

18.20	double poly(double x,int n,double c[])	(311)
18.21	double pow(double base,double exp)	(311)
18.22	double pow10(int n)	(312)
18.23	double sin(double arg)	(312)
18.24	double sinh(double arg)	(313)
18.25	double sqrt(double num)	(313)
18.26	double tan(double arg)	(314)
18.27	double tanh(double arg)	(314)

第十九章 时间、日期和系统相关函数

19.1	int absread(int drive,int numsects,int sectnum,void *buf) int abswrite(int drive,int numsects,int sectnum,void *buf)	(316)
19.2	char *asctime(const struct tm *ptr)	(317)
19.3	int bdos(int fnum,unsigned dx,unsigned al) int bdosptr(int fnum,void *dsdx,unsigned al)	(318)
19.4	int bioscom(int cmd,char byte,int port)	(318)
19.5	int biosdisk(int cmd,int drive,int head,int track,int sector, int nsects,void *buf)	(320)
19.6	int biosequip(void)	(321)
19.7	int bioskey(int cmd)	(322)
19.8	int biosmemory(void)	(323)
19.9	int biosprint(int cmd,int byte,int port)	(324)
19.10	long biostime(int cmd,long newtime)	(324)
19.11	clock_t clock(void)	(325)
19.12	struct country *country(int countrycode,struct country *countryptr)	(325)
19.13	char *ctime(const time_t *time)	(326)
19.14	void ctrlbrk(int (*fptr)(void))	(327)
19.15	void delay(unsigned time)	(328)
19.16	double difftime(time_t time2,time_t time1)	(328)
19.17	void disable(void)	(329)
19.18	int dosexterr(struct DOSERROR *err)	(329)
19.19	long dostounix(struct date *d,struct time *t)	(329)
19.20	void enable(void)	(330)
19.21	unsigned FP_OFF(void far *ptr) unsigned FP_SEG(void far *ptr)	(330)
19.22	void ftime(struct timeb *time)	(330)

19.23	void geninterrupt(int intr)	(331)
19.24	int getcbrk(void)	(331)
19.25	void getdate(struct date *d)	
	void gettime(struct time *t)	(332)
19.26	void getdfree(unsigned char drive, struct dfree *dfptr)	(332)
19.27	char far *getdta(void)	(333)
19.28	void getfat(unsigned char drive, struct fatinfo *fptr) void getfatd(struct fatinfo *fptr)	(333)
19.29	int getftime(int handle, struct ftime *ftptr)	(334)
19.30	unsigned getpsp(void)	(335)
19.31	void interrupt(*getvect(int intr)) ()	(335)
19.32	int getverify(void)	(336)
19.33	struct tm *gmtime(const time_t *time)	(336)
19.34	void harderr(int (*int_handler)()) void hardresume(int code) void hardretn(int code)	(337)
19.35	int inport(int port) unsigned char inportb(int port)	(337)
19.36	int int86(int int_num, union REGS *in_regs, union REGS *out_regs) int int86x(int int_num, union REGS *in_regs, union REGS *out_regs, struct SREGS *segregs)	(338)
19.37	int intdos(union REGS *in_regs, union REGS *out_regs) int intdosx(union REGS *in_regs, union REGS *out_regs, struct SREGS *segregs)	(339)
19.38	void intr(int intr_num, struct REGPACK *reg)	(339)
19.39	void keep(unsigned char status, unsigned size)	(340)
19.40	struct tm *localtime(const time_t *time)	(340)
19.41	time_t mktime(struct tm *p)	(341)
19.42	void far *MK_FP(unsigned seg, unsigned off)	(341)
19.43	void outport(int port, int word) void outportb(int port, unsigned char byte)	(342)
19.44	char *parsfnm(const char *fname, struct fcb *fcbptr, int option)	(342)

19.45	int peek(unsigned seg,unsigned offset)	
	char peekb(unsigned seg,unsigned offset)	
	void poke(unsigned seg, unsigned offset,int word) void	
	pokeb(unsigned seg,unsigned offset,char byte)	(343)
19.46	int randbrd(struct fcb *fcbptr,int count)	
	int randbwr(struct fcb *fcbptr,int count)	(343)
19.47	void segread(struct SREGS *sregs)	(344)
19.48	int setcbrk(int cb)	(344)
19.49	void setdate(struct date *d)	
	void settime(struct time *t)	(345)
19.50	void setdta(char far *dta)	(345)
19.51	int setftime(int handle,struct ftime *t).....	(345)
19.52	void setvect(int intr,void interrupt(*isr)())	(346)
19.53	void setverify(int value)	(346)
19.54	void sleep(unsigned time)	(347)
19.55	int stime(time_t *t)	(347)
19.56	time_t time(time_t *time).....	(348)
19.57	void tzset(void).....	(348)
19.58	void unixtodos(long utime,struct date *d,	
	struct time *t)	(349)

第二十章 动态分配

20.1	int allocmem(unsigned size,unsigned *seg)	(350)
20.2	int brk(void *eds)	(350)
20.3	void *calloc(size_t num,size_t size)	(350)
20.4	unsigned coreleft(void) /* small data models */	
	unsigned long coreleft(void)	
	/* large data models */	(351)
20.5	void far *farcalloc(unsigned long num,	
	unsigned long size	(352)
20.6	unsigned long farcoreleft(void).....	(352)
20.7	void farfree(void far *ptr).....	(352)
20.8	void far *farmalloc(unsigned long size).....	(353)
20.9	void far *farrealloc(void far *ptr,	
	unsigned long newsize).....	(353)
20.10	void free(void *ptr).....	(353)
20.11	int freemem(unsigned seg)	(354)
20.12	int heapcheat(void)	

	int farheapcheck(void)	(354)
20.13	int heapcheckfree(unsigned fill)	
	int farheapcheckfree(unsigned fill)	(355)
20.14	int heapchecknode(void *ptr)	
	int farheapchecknode(void far *ptr).....	(355)
20.15	int heapfillfree(unsigned fill)	
	int farheapfillfree(unsigned fill)	(356)
20.16	int heapwalk (struct heapinfo *hinfo)	
	int farheapwalk (struct farheapinfo *hinfo)	(357)
20.17	void *malloc(size_t size)	(358)
20.18	void *realloc(void *ptr,size_t newsize)	(359)
20.19	void *sbrk(int amount)	(360)
20.20	int setblock(unsigned seg,unsigned size)	(360)

第二十一章 目录函数

21.1	int chdir(const char *path)	(361)
21.2	int findfirst(const char *name,struct fblk *ptr,int attrib)	
	int findnext(struct fblk *ptr);	(361)
21.3	void fmerge(char *path,const char *drive, const char *dir, const char *fname,const char *ext)	
	int fnsplit(const char *path,char *drive,char *dir,char *fname,char *ext)	(362)
21.4	int getcurdir(int drive,char *dir)	(364)
21.5	char *getcwd(char *dir,int len)	(364)
21.6	int getdisk(void)	(365)
21.7	int mkdir(const char *path).....	(365)
21.8	char *mktemp(char *fname)	(366)
21.9	int rmdir(const char *path)	(366)
21.10	char *searchpath(const char *fname).....	(367)
21.11	int setdisk(int drive)	(367)

第二十二章 进程控制函数

22.1	void abort(void)	(368)
22.2	int atexit(void(*func)())	(368)
22.3	int execl(char *fname, char *arg0, ..., char *argN, NULL)	
	int execl(char *fname, char *arg0, ..., char *argN, NULL, char *envp[])	
	int execlp(char *fname, char *arg0, ..., char *argN, NULL)	

int execlpe(char *fname, char *arg0, ..., char *argN, NULL, char *envp[])	
int execv(char *fname, char *arg[])	
int execve(char *fname, char *arg[], char *envp[])	
int execlvp(char *fname, char *arg[])	
int execlvpe(char *fname, char *arg[], char *envp[])	
.....	(369)
22.4 void exit(int status)	
void _exit(int status)	(371)
22.5 int spawnl(int mode, char *fname, char *arg0, ..., char *argN, NULL)	
int spawnle(int mode, char *fname, char *arg0, ..., char *argN, NULL, char *envp[])	
int spawnlp(int mode, char *fname, char *arg0, ..., char *argN, NULL)	
int spawnlpe(int mode, char *fname, char *arg0, ..., char *argN, NULL, char *envp[])	
int spawnv(int mode, char *fname, char *arg[])	
int spawnve(int mode, char *fname, char *arg[], char *envp[])	
int spawnvp(int mode, char *fname, char *arg[])	
int spawnvpe(int mode, char *fname, char *arg[], char *envp[])	(371)

第二十三章 文本屏幕和图形函数

23.1 void far arc(int x, int y, int start, int end, int radius)	(374)
23.2 void far bar(int left, int top, int right, int bottom)	
void far bar3d(int left, int top, int right, int bottom, int depth, int topflag)	(375)
23.3 void far circle(int x, int y, int radius)	(375)
23.4 void far cleardevice(void)	
void far clearviewport(void)	(376)
23.5 void far closegraph(void)	(377)
23.6 void clreol(void)	
void clrscr(void)	(377)
23.7 int cprintf(const char *fmt, ...)	(378)
23.8 int cputs(const char *str)	(379)
23.9 int cscanf(char *fmt, ...)	(380)

23.10	void delline(void).....	(380)
23.11	void far detectgraph(int far *driver,int far *mode).....	(381)
23.12	void far drawpoly(int numpoints,int far *points)	(381)
23.13	void far ellipse(int x,int y,int start,int end,int xradius, int yradius)	(382)
23.14	void far fillellipse(int x, int y, int xr, int yr)	(383)
23.15	void far fillpoly(int numpoints, int far *points)	(383)
23.16	void far floodfill(int x, int y, int border)	(384)
23.17	void far getarccoords(struct arccoordstype far *coords)	(384)
23.18	void far getaspectratio(int far *xasp, int far *yasp)	(385)
23.19	int far getbkcolor(void)	(386)
23.20	int far getcolor(void)	(387)
23.21	struct palettetype *far getdefaultpalette(void)	(387)
23.22	char *far getdrivernam(void)	(387)
23.23	void far getfillpattern(char far *pattern)	(388)
23.24	void far getfillsettings(struct fillsettingstype far *info)	(389)
23.25	int far getgraphmode(void)	(390)
23.26	void far getimage(int left, int top, int right, int bottom, void far *buf)	(391)
23.27	void far getlinesettings(struct linesettingstype far *info)	(392)
23.28	int far getmaxcolor(void)	(393)
23.29	int far getmaxmode(void)	(394)
23.30	int far getmaxx(void)	(394)
	int far getmaxy(void)	(394)
23.31	char *far getmodename(int mode)	(395)
23.32	void far getmoderange(int driver, int far *lowmode, int far *himode)	(396)
23.33	void far getpalette(struct palettetype far *pal).....	(396)
23.34	int far getpalettesize(void)	(398)
23.35	unsigned far getpixel(int x, int y)	(399)
23.36	int gettext(int left, int top, int right, int bottom, void *buf)	(399)
23.37	void far gettextsettings(struct textsettingstype far *info)	(400)