

第一部分

示范程序和字符方式 下的应用程序接口

在第一部分将提供 21 个示范程序的源代码。这些程序可以在 Borland 公司的 C++ 或 IBM 公司的 C/SET 2 环境下编译。如果你选择了 Borland 公司的编译程序,你将会得到运行在 DOS 环境下的实地址模式程序。如果你选择了 IBM 公司的编译程序,你将会得到运行在 OS/2 环境下的 32 位平面内存模式的全屏幕程序。

在示范程序中尽量减少了条件性的编译语句。只有在极少情况下,调用采用了虚函数(函数中不包含任何指令)以便保持在 DOS 和 OS/2 环境下源代码的兼容性。

第一章 开发统一的 OS/2 和 DOS 下的 全屏幕字符方式 API

我记得这种转变源于一次谈话。我的好友(一个程序开发天才)Marc Neuberger 打电话给我并且对我喊道:“现在将你的计算机上装入 IBM 的 OS/2 2.0 版!”。尽管 Marc 和我偶尔对于各种出版的系统有不同的观点,但我还是按他的建议做了。真高兴,我做对了。

在第一天使用 OS/2 系统之后我就对它着了迷。这里简述一下到底发生了什么。当我打开一个 DOS 窗口开始将超过 250 个的文件压缩并从硬盘拷贝到软盘;处理该过程将花费较长的时间。启动该过程后,又移动鼠标指向了 32 位的图形管理器的通讯程序并且开始拨叫 BBS 系统。线路占线,通讯程序进入了后台运行;之后我发觉只有自己还坐在计算机前。

我将鼠标指向了图形管理器的 Reversi 游戏程序,之后双击鼠标键运行它。这些使我对 OS/2 系统非常敬佩。我的磁盘压缩和文件拷贝都在高速地运行着,通讯程序也在反复地呼叫 BBS 系统,同时我还可以运行游戏 Reversi。我不必对这些应用程序作任何的决定。而这些只是在一个 33MHz 主频的 386 微机上得到的。

我的联想被 BBS 的应答所打断,并且游戏 Reversi 还在等待着我。

哇! 这一切太精彩了。

1.1 函数前缀列表

在本书中对所提供的 OS/2 和 DOS 库文件中的每一类函数都采用了标准的函数前缀名。表 1.1 给出了函数前缀列表。

表 1.1 函数前缀列表

前缀	类别
cu-	光标
kb	键盘
ms-	鼠标
mk-	make
scrn-	屏幕
wind-	窗口
print-	打印机

1.2 统一的 OS/2 全屏幕和 DOS 下的字符方式 API

应用程序开发接口的目的在于可以使程序员使用同时满足于两种环境的一个源程序,编写出鼠标和键盘驱动用户接口、打印机、键盘及鼠标事件处理器等等。为 DOS 和 OS/2 程

序员建立一个统一的 API 驱动的工作已经完成。

程序 1.1 提供了头文件 TPROTO.H 的源代码。该文件包含了所有出现在本书中的 API 函数原型。

程序 1.1 TPROTO.H 源程序清单

```

////////////////////////////////////
//
// tproto.h
//
// Function prototype file for
// library functions
////////////////////////////////////
//
// include files here
//
#include "keyboard.h"
#include "ascii.h"
#include "tstruct.h"
////////////////////////////////////
//
// OS2 defines
//
#ifdef OS2_PROG
#ifdef BCOS2_COMP
#define itoa itoa
#define ltoa ltoa
#endif
#endif

////////////////////////////////////
//
// High level keyboard routines
//
int kb_edit(char * response,
            int row,
            int column,
            int dlen,
            int opt,
            UCHAR attr);

int kb_read(void);

```

```

int    kb_status(void);
char   kb_char(void);
UCHAR  kb_scan(void);

//////////

// void  kb_cap_on(void);
// void  kb_cap_off(void);
// void  kb_ins_in(void);
// void  kb_ins_off(void);
// void  kb_num_on(void);
// void  kb_num_off(void);
// int   gtKBflag(void);
// int   gtKBflsh(int);

//////////

//////////
//
// High level cursor routines
//
void cu_get_loc(int *, int *);
int  cu_get_shape(void);
void cu_move(int, int);
void cu_relative_move(int, int);
void cu_remove(void);
void cu_display(void);
void cu_save_size(void);
void cu_rest_size(void);
void cu_set_size(int, int);
void cu_save_loc(void);
void cu_set_shape(int);
void cu_rest_loc(void);

void putCR(void);
void putLF(void);
void putCRLF(void);

// rectangle routines

void vdBox(RECT *, int, unsigned char);
RECT * setRect(RECT *, int, int, int, int);
void addRect(RECT *, RECT *);
void subRect(RECT *, RECT *);
void dupRect(RECT *, RECT *);
void dpyRect(RECT *);
void offRect(RECT *, int, int);
void boxRect(RECT *, int, UCHAR attr);

```

```
void clrRect(RECT *);
void fillRect(RECT *, int);
void saveRect(RECT *);
UINT sizeRect(RECT *);
void restRect(RECT *);
void scUp(RECT *, int, int);
void scDn(RECT *, int, int);

////////////////////////////////////
//
// High level print routines
//

int print_open(int num);
int print_close(int num);
int print_newline(int num);
int print_cr(int num);
int print_string(int num, char *);
int print_char(int, char);
int print_scrn(int);
int print_scrnFF(int);
int print_status(int);
void print_set_column(int, int);

////////////////////////////////////
//
// High level screen routines
//

void scrn_write(int row, int col, int len, char *str, UCHAR attr);
void scrn_init(void);
int scrn_read_char(unsigned short, unsigned short);
void scrn_save(void);
void scrn_restore(void);
void scrn_clear(void);
void scrn_attr(int, int, int, unsigned char);
void scrn_change_attr(unsigned char);
void scrn_char(int row, int col, char ch, UCHAR attr);
void scrn_repeat_char(int row, int col, int len, char ch, UCHAR attr);
void scrn_chr(int, int, int);

void ascup(int, int, int, int, int, int, int);
void putChr(char);
void putCRLF(void);
void putLF(void);
void putCR(void);
```

```

void putStr(char *);
int rdChar(void);
void vdHoriz(int, int, int, int);
int vdprompt(char *, int, int, int, int);
void vdStr(int, int, int, char *, char);
void vdVert(int, int, int, int);
void wrChar(char, int);

////////////////////
//
// High level mouse routines
//

int ms_init(void);
void ms_on(void);
void ms_off(void);
int ms_status(int *x, int *y);
void ms_map_display(int row, int col, int key_val);

////////////////////
//
// High level window routines
//

WIND *wind_init(WIND *W_PTR,
                int ulr,
                int ulc,
                int lrr,
                int lrc,
                UCHAR attr,
                int border,
                char *title);
int wind_kb_edit(WIND *W,
                 char *response,
                 int row,
                 int column,
                 int dlen,
                 int opt,
                 UCHAR attr);
void wind_display(WIND *);
void wind_remove(WIND *);
void wind_attr(WIND *, int, int, int, unsigned char);
void wind_write(WIND *, int, int, int, char *, UCHAR);
void wind_char(WIND *, int row, int col, char ch, UCHAR attr);
void wind_repeat_char(WIND *, int row, int col, int len, char ch, UCHAR attr);
void wind_destroy(WIND *);

```

```

int    wind_read_char(WIND *, int, int);
void   wind_clear(void);
void   wind_cursor_move(WIND *, int, int);

////////////////////////////////////
//
// Low level window routines
//

void   setattr(WIND *, unsigned char);
void   setBord(WIND *, int);
void   setTitle(WIND *, char *);
WIND * setWind(WIND *, int, int, int, int);
void   strtWind(WIND *);
UINT   sizeImg(WIND *);
void   wrBox(WIND *);
void   wrImg(WIND *);
void   wrWind(WIND *);
void   wvdHoriz(WIND *, int, int, int, int);
int     wvdPrmpt(WIND *, char *, int, int, int, int);
void   wvdVert(WIND *, int, int, int, int);
void   wvdScdn(WIND *, int);
void   wvdScup(WIND *, int);
void   wvdStr(WIND *, int, int, int, char *, char);
void   rdImg(WIND *);
void   rdWind(WIND *);

////////////////////////////////////
//
// Make Attributes and
// Tokens
//
////////////////////////////////////

UCHAR mk_attr(unsigned char, unsigned char, unsigned char, unsigned char);
UCHAR mk_attr_intense(unsigned char);
UCHAR mk_attr_intense_blink(unsigned char);
UCHAR mk_attr_inverse(unsigned char);
UCHAR mk_attr_blink(unsigned char);

void   mk_char_attr(int token, char *ch, UCHAR *attr);
int     mk_token(unsigned char, unsigned char);

```

程序 1.2 到 1.4 给出 OS/2 全屏幕和 DOS 程序员调用的 API 的其他头文件。附程序 1.2 列出了 KEYBOARD.H 文件清单。程序 1.3 给出了 ASCII.H 文件清单，它包含了 ASCII 码和其它定义。程序 1.4 给出了 TSTRUCT.H 文件清单，它包含了被应用在 OS/2 和 DOS

下的 API 的结构。

程序 1.2 KEYBOARD.H 源程序清单

```

////////////////////////////////////
//
// keyboard.h
//
// 16-bit Key Codes
//
////////////////////////////////////

#define INSERT      0x5200
#define DELETE     0x5300
#define SPACE      0x3920
#define ESC        0x011b
#define ESCAPE     0x011b
#define PGDN       0x5100
#define PGUP       0x4900
#define PERIOD     0x342e
#define TAB        0x0f09
#define RT_SQUARE  0x1b5d
#define LT_SQUARE  0x1a5b
#define RT_BRACKET  0x1b7d
#define LT_BRACKET 0x1a7b
#define CNTL_HOME  0x7700
#define CNTL_END   0x7500
#define CNTL_ENTER 0x1c0a
#define CNTL_BS    0x0e7f
#define HOME       0x4700
#define END        0x4f00
#define s_BS       0x0008
#define BS         0x0e08
#define BACKSPACE  0x0e08
#define s_CR       0x000d
#define CR         0x1c0d
#define ENTER      0x1c0d
#define UP_ARROW   0x4800
#define RIGHT_ARROW 0x4d00
#define LEFT_ARROW 0x4b00
#define DOWN_ARROW 0x5000
#define UP_ARROW_K 0x48e0
#define RIGHT_ARROW_K 0x4de0

```

```
#define LEFT_ARROW_K    0x4be0
#define DOWN_ARROW_K    0x50e0
#define F1               0x3b00
#define F2               0x3c00
#define F3               0x3d00
#define F4               0x3e00
#define F5               0x3f00
#define F6               0x4000
#define F7               0x4100
#define F8               0x4200
#define F9               0x4300
#define F10              0x4400
#define SHIFT_TAB        0x0f00
#define SHIFT_HOME       0x4737
#define SHIFT_END        0x4f31
#define SHIFT_INSERT     0x5230
#define SHIFT_DELETE     0x532e
#define SHFT_INSERT      0x5230
#define SHFT_F1          0x5400
#define SHFT_F2          0x5500
#define SHFT_F3          0x5600
#define SHFT_F4          0x5700
#define SHFT_F5          0x5800
#define SHFT_F6          0x5900
#define SHFT_F7          0x5a00
#define SHFT_F8          0x5b00
#define SHFT_F9          0x5c00
#define SHFT_F10         0x5d00
#define SH_R_ARROW       0x4d36
#define SH_L_ARROW       0x4b34
#define SH_U_ARROW       0x4838
#define SH_D_ARROW       0x5032
#define CNTL_F1          0x5e00
#define CNTL_F2          0x5f00
#define CNTL_F3          0x6000
#define CNTL_F4          0x6100
#define CNTL_F5          0x6200
#define CNTL_F6          0x6300
#define CNTL_F7          0x6400
#define CNTL_F8          0x6500
#define CNTL_F9          0x6600
#define CNTL_F10         0x6700
#define CNTL_LEFTA       0x7300
```

```
#define CNTL_RIGHTA 0x7400
#define ALT_F1 0x6800
#define ALT_F2 0x6900
#define ALT_F3 0x6a00
#define ALT_F4 0x6b00
#define ALT_F5 0x6c00
#define ALT_F6 0x6d00
#define ALT_F7 0x6e00
#define ALT_F8 0x6f00
#define ALT_F9 0x7000
#define ALT_F10 0x7100
#define ALT_A 0x1e00
#define ALT_B 0x3000
#define ALT_C 0x2e00
#define ALT_D 0x2000
#define ALT_E 0x1200
#define ALT_F 0x2100
#define ALT_G 0x2200
#define ALT_H 0x2300
#define ALT_I 0x1700
#define ALT_J 0x2400
#define ALT_K 0x2500
#define ALT_L 0x2600
#define ALT_M 0x3200
#define ALT_N 0x3100
#define ALT_O 0x1800
#define ALT_P 0x1900
#define ALT_Q 0x1000
#define ALT_R 0x1300
#define ALT_S 0x1f00
#define ALT_T 0x1400
#define ALT_U 0x1600
#define ALT_V 0x2f00
#define ALT_W 0x1100
#define ALT_X 0x2d00
#define ALT_Y 0x1500
#define ALT_Z 0x2c00
#define CNTL_A 0x1e01
#define CNTL_B 0x3002
#define CNTL_C 0x2e03
#define CNTL_D 0x2004
#define CNTL_E 0x1205
#define CNTL_F 0x2106
```

```
#define CNTL_G      0x2207
#define CNTL_H      0x2308
#define CNTL_I      0x1709
#define CNTL_J      0x240a
#define CNTL_K      0x250b
#define CNTL_L      0x260c
#define CNTL_M      0x320d
#define CNTL_N      0x310e
#define CNTL_O      0x180f
#define CNTL_P      0x1910
#define CNTL_Q      0x1011
#define CNTL_R      0x1312
#define CNTL_S      0x1f13
#define CNTL_T      0x1414
#define CNTL_U      0x1615
#define CNTL_V      0x2f16
#define CNTL_W      0x1117
#define CNTL_X      0x2d18
#define CNTL_Y      0x1519
#define CNTL_Z      0x2c1a
#define K_0          0x0b30
#define K_1          0x0231
#define K_2          0x0332
#define K_3          0x0433
#define K_4          0x0534
#define K_5          0x0635
#define K_6          0x0736
#define K_7          0x0837
#define K_8          0x0938
#define K_9          0x0a39
#define ALT_0        0x8100
#define ALT_1        0x7800
#define ALT_2        0x7900
#define ALT_3        0x7a00
#define ALT_4        0x7b00
#define ALT_5        0x7c00
#define ALT_6        0x7d00
#define ALT_7        0x7e00
#define ALT_8        0x7f00
#define ALT_9        0x8000
#define K_SPACE      0x3920
#define K_EXCLAM     0x0221
#define K_QUOTE       0x2822
```

```
#define K_POUND 0x0423
#define K_DOLLAR 0x0524
#define K_PERCENT 0x0625
#define K_AND 0x0826
#define K_APOST 0x2827
#define K_LPAREN 0x0A28
#define K_RPAREN 0x0B29
#define K_STAR 0x092A
#define K_PLUS 0x0D2B
#define K_COMMA 0x332C
#define K_MINUS 0x0C2D
#define K_PERIOD 0x342E
#define K_FSLASH 0x352F
#define K_COLON 0x273A
#define K_SCOLON 0x273B
#define K_LESS 0x333C
#define K_EQUAL 0x0D3D
#define K_GREAT 0x343E
#define K_QUEST 0x353F
#define K_AMPER 0x0340
#define K_A 0x1E61 - 0x20
#define K_B 0x3062 - 0x20
#define K_C 0x2E63 - 0x20
#define K_D 0x2064 - 0x20
#define K_E 0x1265 - 0x20
#define K_F 0x2166 - 0x20
#define K_G 0x2267 - 0x20
#define K_H 0x2368 - 0x20
#define K_I 0x1769 - 0x20
#define K_J 0x246A - 0x20
#define K_K 0x256B - 0x20
#define K_L 0x266C - 0x20
#define K_M 0x326D - 0x20
#define K_N 0x316E - 0x20
#define K_O 0x186F - 0x20
#define K_P 0x1970 - 0x20
#define K_Q 0x1071 - 0x20
#define K_R 0x1372 - 0x20
#define K_S 0x1F73 - 0x20
#define K_T 0x1474 - 0x20
#define K_U 0x1675 - 0x20
#define K_V 0x2F76 - 0x20
#define K_W 0x1177 - 0x20
```

```
#define K_X          0x2D78 - 0x20
#define K_Y          0x1579 - 0x20
#define K_Z          0x2C7A - 0x20
#define K_LBRACK     0x1A5B
#define K_BSLASH     0x2B5C
#define K_RBRACK     0x1B5D
#define K_KARAT      0x075E
#define K_UNDER      0x0C5C
#define K_a          0x1E61
#define K_b          0x3062
#define K_c          0x2E63
#define K_d          0x2064
#define K_e          0x1265
#define K_f          0x2166
#define K_g          0x2267
#define K_h          0x2368
#define K_i          0x1769
#define K_j          0x246A
#define K_k          0x256B
#define K_l          0x266C
#define K_m          0x326D
#define K_n          0x316E
#define K_o          0x186F
#define K_p          0x1970
#define K_q          0x1071
#define K_r          0x1372
#define K_s          0x1F73
#define K_t          0x1474
#define K_u          0x1675
#define K_v          0x2F76
#define K_w          0x1177
#define K_x          0x2D78
#define K_y          0x1579
#define K_z          0x2C7A
```

程序 1.3 ASCII.H 源程序清单

```

////////////////////////////////////
//
// ascii.h
//
// ASCII def file
//
#define aNUL      0      // null \0 delimiter
#define aSOH      1      // ^ A - start of heading
#define aSTX      2      // ^ B - start of text
#define aETX      3      // ^ C - end of text
#define aEOT      4      // ^ D - end of transmission
#define aENQ      5      // ^ E - inquiry
#define aACK      6      // ^ F - affirm acknowledgement
#define aBEL      7      // ^ G - audible bell
#define aBS      8      // ^ H - backspace
#define aTAB      9      // ^ I - horizontal tab
#define aLF      10     // ^ J - line feed
#define aVT      11     // ^ K - vertical tab
#define aFF      12     // ^ L - form feed
#define aCR      13     // ^ M - carriage return
#define aSO      14     // ^ N - shift out
#define aSI      15     // ^ O - shift in
#define aDCE      16     // ^ P - data link escape
#define aDC1      17     // ^ Q - device control 1
#define aDC2      18     // ^ R - device control 2
#define aDC3      19     // ^ S - device control 3
#define aDC4      20     // ^ T - device control 4
#define aNAK      21     // ^ U - neg acknowledgement
#define aSYN      22     // ^ V - synchronous idle
#define aETB      23     // ^ W - end of transmission
#define aCAN      24     // ^ X - cancel
#define aEM      25     // ^ Y - end of medium
#define aSUB      26     // ^ Z - substitute
#define aESC      27     // escape
#define aFS      28     // file separator
#define aGS      29     // group separator
#define aRS      30     // record separator
#define aUS      31     // unlinked separator
#define aSPC      32     // space

```

```
#define aCODE      94      /* character indicating printer command follows
#define aHCR       aEOT    /* Hard carriage return
#define aCENTER    'C'     /* code to center line
#define aDOUBLE    'D'     /* double strike toggle
#define aEXPAND    'E'     /* emphasize toggle
#define aSUPERS    'S'     /* superscript toggle
#define aITALIC    'I'     /* italics toggle
#define aBOLD      'B'     /* bold toggle
#define aTRUE      1       /* true
#define aFALSE     0       /* false
#define ONE_COL    1       /*      1 column format
#define TWO_COL    2       /*      2 column format
#define ONE_TOP    3       /*
#define TWO_TOP    4       /*
#define ONE_BOT    5       /*
#define TWO_BOT    6       /*
#define TWO_LR     7       /*
#define TWO_R      8       /*
#define TWO_UR     9       /*
#define TWO_TB     10      /*
#define VONE_COL   11      /* word per chart format
#define XONE_COL   81      /*      1 column format
#define XTWO_COL   82      /*      2 column format
#define XTHREE_COL 83      /*      3 column format
#define XONE_TOP   84      /*
#define XTWO_TOP   85      /*
#define XTHREE_TOP 86      /*
#define XONE_BOT   87      /*
#define XTWO_BOT   88      /*
#define XTHREE_BOT 89      /*
#define XTHREE_LR  90      /*
#define XTHREE_R   91      /*
#define XTHREE_P1  92      /*
#define XTHREE_P2  93      /*
#define XTHREE_TB  94      /*
#define XTHREE_UR  95      /*
#define XTHREE_2T  96      /*
#define XTHREE_2B  97      /*
```

程序 1.4 TSTRUCT.H 源程序清单

```

////////////////////////////////////
//
// tstruct.h
//

#ifndef IMAGE
#define IMAGE unsigned int
#endif

#ifndef ULONG
#define ULONG unsigned long
#endif

#ifndef USHORT
#define USHORT unsigned short
#endif

#ifndef UINT
#define UINT unsigned int
#endif

#ifndef UINTP
#define UINTP UINT *
#endif

#ifndef UCHAR
#define UCHAR unsigned char
#endif

#ifndef UCHARP
#define UCHARP UCHAR *
#endif

/*
 * structures
 */

////////////////////////////////////
//
// Interface Structure List
//

////////////////////////////////////

#define LOTUS_ITEM_MAX 20
#define MENUBAR_ITEM_MAX 10

```

```

////////////////////////////////////
//
// Structure for Lotus Style Window Interface
//

typedef struct {
    int number;                // number of LOTUS objects
    char * name[LOTUS_ITEM_MAX]; // pointer to item name
    char * explain[LOTUS_ITEM_MAX]; // pointer to item explanation
    int lot_map[LOTUS_ITEM_MAX][2]; // map for lotus item highlights
    int lotus_item;            // highlight and
    int old_lotus;              // item selection data
    int lotus_open;             // status of lotus window
    unsigned int imgbuf[160];    // top two rows screen image
} LOTUS_CLASS;

////////////////////////////////////
//
// Structure for Lotus Style Window Interface
//

typedef struct {
    int number;                // number of MENUBAR objects
    char * name[MENUBAR_ITEM_MAX]; // pointer to item name
    int mb_map[MENUBAR_ITEM_MAX][2]; // map for menubar item highlights
    int key_hst[MENUBAR_ITEM_MAX];
    int menubar_item;          // highlight and
    int old_menubar;            // item selection data
    int menubar_open;           // status of lotus window
    int sj_attr;                // item attribute
    int sinv_attr;              // inverse attribute
    int sk_attr;                // highlight key attribute
    int first_time;             // first time
    unsigned int imgbuf[160];    // top two rows screen image
} MENUBAR_CLASS;

typedef struct {
    int ul_row;                // * upper left row * /
    int ul_col;                // * upper left column * /
    int lr_row;                // * lower right row * /
    int lr_col;                // * lower right column * /
    unsigned int img_size;      // * window img size * /
    unsigned int * img_ptr;      // * pointer scrn image * /
    unsigned int * wind_ptr;     // * pointer scrn image * /
    int box_type;               // * border selection * /
    int attr;                   // * window attribute * /

```