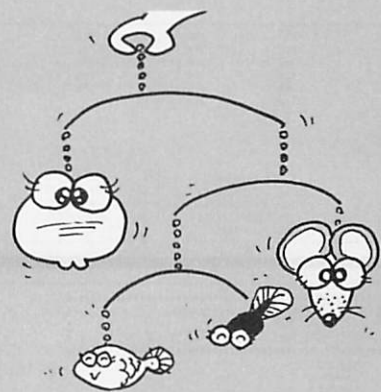


XDIR

ディスクの ディレクトリを 階層表示する



■武石秀男

今や「統合環境」ばやりで、統合化されていないソフトが貧弱に思えるぐらいのブームです。

1/0'88年10月号で発表した「Moving」もこの統合環境化をしていますが、データ・ファイル操作の快適さを追求しているうちに愛用の20Mバイト・ハードディスクがバンク状態になり、80Mバイト・ハードディスクを購入するはめになりました。

手元に届くまでにハードディスク環境の再整備を、と気軽にこのプログラムを作り始めたのですが…

プログラムについて

長い間、「Lattice C」を使ってきたのですが、軽いソフトにもコンパイル時間が結構かかるので、今回は「Turbo C」で作りました。

リスト1は「Turbo C」の方言をほとんど使っていないので、他のC言語の処理系でも作動すると思います。ファイル関係が処理系によって違いましたが、X68Kには、2, 3箇所の手直しで簡単に移植できました。リスト2は画面表示部に方言をふんだんに使っているので、他の処理系ではうまく動かないと思います。

使用方法

書式は

XDIR [<ドライブ, パス名>] [<スイッチ>]

でディスクのディレクトリ構造を罫線で階層的に表示し、起動時のオプションによってきめ細かな設定ができます。

リスト2の統合環境版では、起動スイッチの変更も含めて、コマンドで多彩な指定ができます。

パラメータの設定は順不同で、大文字でも小文字でもかまいません。

表1に起動時のオプション・スイッチの説明をします。

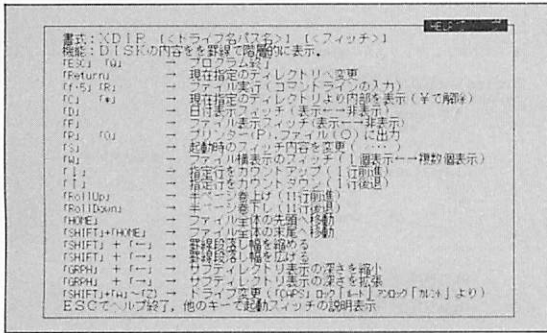
表1 起動時のオプション

スイッチ	内容
/Dn	サブディレクトリ表示の深さを指定 n = 0~20 デフォルトの値は10
/F	ファイル名を表示 (1行に1個ずつ)
/FD	ファイルのサイズ・日付・時刻を表示
/FW	ファイル名を「 DIR /W 」のように横に複数個ずつ表示
/Hn	サブディレクトリを表示する場合の段落とりの幅を指定 n = 2~40 デフォルトの値は8 (罫線が全角なので偶数値)
/Wn	全体の表示幅を設定 n = 40~136 デフォルトは80
/S:F	Fはワイルド・カードを含むファイル名 指定したファイル名だけを表示 (WHEREのように検索表示)
/Cn1n2 n3n4n5n6	罫線、サブディレクトリ名、ファイル名を色分けて表示 0~7, 8を加えた値 (8,9,A~F) を指定した場合は反転表示 プリンタに出力した場合はこのスイッチを指定してはいけない n1: 罫線の色 n2: サブディレクトリの色 n3: ファイル名の色 (/FDでファイルサイズ・日付・時刻表示を指定した場合のみ) n4: ファイルサイズ表示の色 n5: 日付表示の色 n6: 時刻表示の色

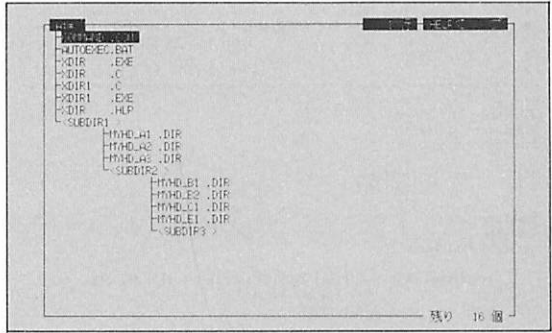
なお、統合環境版ではリダイレクトを行なう場合、「/R」を指定してください (統合環境版では画面表示の高速化のため、直接画面に出力している)。



統合環境版では、こんなに多くの機能が使える



階層ディレクトリもこれなら分かりやすい



トラブルの対策

①ファイル間の移動をスムーズにするために、オート・リピート機能を強調してあります。このため、コマンド入力でキーを押すすぎるとダブル・コマンドになる場合があるので軽いタッチで操作してください。

②統合環境版を使っているときは、むやみに **STOP** キーを押さないでください。テキスト画面の2ページ目を使っているので、画面が戻らない場合があります。

CTRL+**1**によるブレイク・ポイントは用意してありますが、うまく動かない場合があります。

この対策としてリスト2のtextbank(1)をtextbank(0)に変更すれば、常にテキスト画面の1ページ目が使われるのでこのようなトラブルは起こりませんが、統合環境らしさが薄れる気がします。

③ディスクが装着されていないドライブを指定した場合、ハードウェア・エラールーチンが呼び出されます。この場合は通常のディスク・エラーと同じようにA(終了)、R(再実行)、I(無視)のキー入力での処理してください。

④コマンドの選択によっては、該当ファイルがない場合があります。Cコマンドでこうなったときは **Y** でキャンセル、**ESC** で終了、またはドライブ名を変更してください。

おわりに

MS-DOSのDIRコマンドに物足りなさを感じていたので、まず、この機能を拡張することにしました。

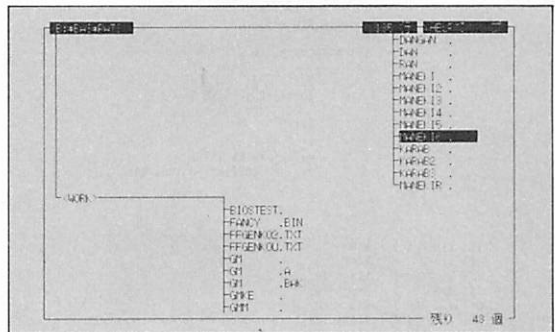
最近、「プログラムの軽さ」に傾いているため、初めはほとんどの機能を持つものを作る予定で出来上がったのがリスト1でした。

ところが、「凝り性には軽さは似合わない」のでしょう。結局、重量感のある統合環境タイプのリスト2になってしまいました。

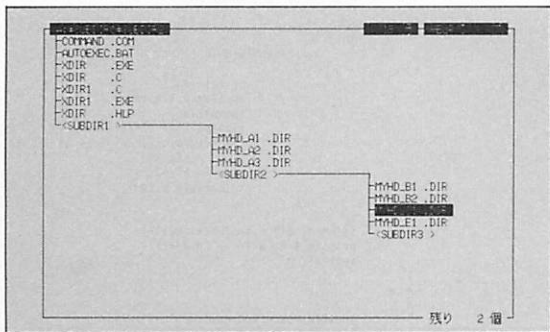
心は「軽さ」を、されどプログラムは「重く」。これが現実のようです。

MS-DOSのDIR、TREEコマンドにちょっぴり不満を持っている方に愛用していただければ幸いです。

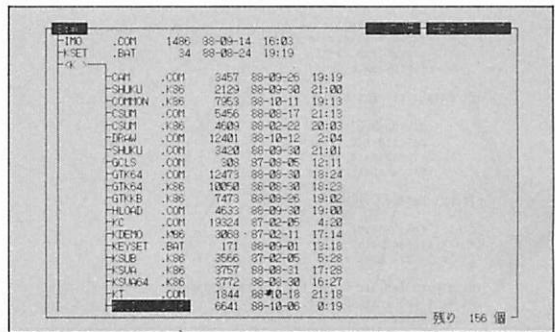
逆スクロールもできる！



罫線の長さを変更



日付けを表示



```

/* =====
 * MS-DOSコマンド DIR機能拡張 (コマンドライン版)
 *                               Turbo_Cで作成
 *                               1988.12/15 By H. Takeishi
 * ===== */

#include <dir.h>
#include <string.h>
#include <stdlib.h>

char fcmp(), colset();
void tcolor(), disp_sdt();

main(argc, argv)
int argc;
unsigned char *argv[];
{
    unsigned char path[80], knd[100], cur[80], col[7], *sch, *a,
    int dep, flg, atr, kln, wid, cnt, drv, c;

    /* 初期値設定 */
    dep = 10; /* サブディレクトリの深さ */
    flg = 4; /* 表示フラグ (昇降あり) */
    /* 1ビット目 ファイルを表示する */
    /* 2ビット目 ファイルを横に表示する */
    /* 3ビット目 昇降を表示する */
    /* 4ビット目 サイズ日付時刻を表示 */
    atr = 0x10; /* アトリビュート */
    kln = 8; /* ディレクトリの昇降深さ表示幅 */
    wid = 80; /* 全体の表示幅 */
    col[0] = 0; /* カラー表示指定 */
    path[0] = 0; /* 指定ドライブ名、パス名 */
    sch = ". *"; /* サーチファイル名 (初期値は全て) */

    /* ドライブ名、パス名
    /* スイッチ ( D n W n F H n S : ) の設定 */

    for( cnt=1; cnt<argc; cnt++ )
    {
        a=strupr(argv[cnt]);
        if( *a != '/' )
            strcpy(path, a);
        else
        {
            if( strncmp(a, "/D", 2) == 0 ) dep=atoi(a+2);
            if( strncmp(a, "/W", 2) == 0 ) wid=atoi(a+2);
            if( strncmp(a, "/F", 2) == 0 )
            {
                flg=(flg | 1);
                if( *(a+2) == 'M' ) flg=(flg | 2);
                if( *(a+2) == 'D' ) flg=(flg | 8);
            }
            if( strncmp(a, "/H", 2) == 0 )
            {
                flg=(flg | 4);
                if( *(a+2) != 0 ) kln=atoi(a+2);
            }
            if( strncmp(a, "/S", 3) == 0 )
            {
                flg=(flg | 1);
                sch=a+3;
            }
            if( strncmp(a, "/C", 2) == 0 )
            {
                int j;
                for( j=0; j<7; j++ )
                    col[j]=7;
                j=1;
                a++;
                while( *a && j<7 )
                    col[j++]=colset(*a++);
            }
        }
    }

    if( kln & 1 ) kln--;
    if( kln<2 ) kln = 2;
    if( kln>40 ) kln = 40;
    if( dep>20 ) dep = 20;
    if( wid<40 ) wid = 40;
    if( wid>136 ) wid = 136;
    if( kln>wid ) kln = wid;
    if( path[0]==0 ) getcwd(path, 80);
    for( cnt=0; cnt<10; cnt++ ) knd[cnt]=1;

    if( strchr(path, ':') == NULL )
    {
        drv=getdisk();
        cur[0]='A'+drv; cur[1]=': '; cur[2]=0;
        streat(cur, path);
        strcpy(path, cur);
    }
    if( path[strlen(path)-1] == ':' )
    {
        drv=path[0]-'0';
        getcurdir(drv, cur);
        streat(path, "YY");
        streat(path, cur);
    }
    strcpy( knd+20, "" );

    printf("Yn%k", path);
    c=path[strlen(path)-1];
    if( c != 'Y' && c != ':' ) streat(path, "YY");

    fdisp(path, sch, atr, 0, dep, flg, kln, knd, wid, col);
    if( *col != 0 ) tcolor(7);
    printf("Yn");
}

```

```

/* === ディレクトリ表示 再帰的ルーチン ===== */

fdisp(path, sch, atr, n, dep, flg, kln, knd, wid, col)
unsigned char *path, *sch,
atr, n, dep, flg,
*knd, wid, *col;
{
    struct ffbk ffbk; /* ファイル情報格納領域 */
    unsigned char file[100], *p, fdir[150][13], kfg;
    long siz[150];
    int dat[150], tim[150],
    done, i, cnt, fct, xpo;

    cnt=0;
    fct=0;
    xpo=n*kln;
    strcpy(file, path);
    strcpy(fdir, ". *");

    /* --- ファイル名とディレクトリ名 (ファイル名の後) の格納 --- */

    done=findfirst(file, &ffbkl, atr);
    while( !done )
    {
        p=ffbkl.ff_name;
        if( *p != '.' && cnt<200 )
        {
            if( ffbkl.ff_attrib == 0x10 )
            {
                siz[cnt]=ffbkl.ff_fsize;
                dat[cnt]=ffbkl.ff_fdate;
                tim[cnt]=ffbkl.ff_ftime;
                strcpy(fdir[cnt++], p);
            }
            else
            {
                if( flg & 1 && fcmp(sch, p) == 0 )
                {
                    int j;
                    for( j=cnt; j>fct; j-- )
                        strcpy(fdir[j+1], fdir[j]);
                    siz[j+1]=siz[j];
                    dat[j+1]=dat[j];
                    tim[j+1]=tim[j];
                }
                siz[fct]=ffbkl.ff_fsize;
                dat[fct]=ffbkl.ff_fdate;
                tim[fct]=ffbkl.ff_ftime;
                strcpy(fdir[fct++], p);
                cnt++;
            }
        }
        done=findnext(&ffbkl);
    }

    /* --- 昇降の処理 --- */

    if( *col != 0 ) tcolor(*col+1);
    if( cnt != 0 && flg & 4 )
        printf("XsYn", knd+20);
    else
        printf("Yn");

    /* --- ファイル名の表示 --- */

    for( i=0; i<fct; i++ )
    {
        char *dp;

        p=fdir[i];
        dp=strchr(p, ':');
        if( dp != 0 ) *dp=0;
        if( flg & 2 )
        {
            int fln;
            fln=xpo*(fct-i)*16+2;
            if( i==0 )
            {
                spc(flg, n, kln, knd, col);
                if( (fln>wid || cnt>fct) && i<cnt-1 )
                    kfg=1;
                else kfg=3;
                keisen(kfg, n, knd);
            }
            else
            {
                if( xpo>wid-18 )
                {
                    printf("Yn");
                    spc(flg, n, kln, knd, col);
                    xpo=n*kln;
                    fln=xpo*(fct-i)*16+2;
                    if( (fln>wid || cnt>fct) && i<cnt-1 )
                        kfg=2;
                    else kfg=3;
                    keisen(kfg, n, knd);
                }
            }
            if( *col != 0 ) tcolor(*col+3);
            printf("Xs-Ys-Xs ", p, dp+1);
            xpo+=16;
        }
        else
        {
            spc(flg, n, kln, knd, col);
        }
    }
}

```

リスト 1 XDIR1.C

```

        if( i<cnt-1 ) kfg+1; else kfg+3;
        keisen(kfg,n,knd);
        if( *col !=0 ) tcolor(*(col+3));
        printf("%-8s.%-4s".p.dp+1);
        if( flg & 8 )
            disp_sdt(siz[i],dat[i].tim[i].col,wid-n*kln);
        printf("\n");
    }
}
if( flg & 2 && fet>0 ) printf("\n");
/* --- サブディレクトリの表示 および 再帰的呼び出し ----- */
for( i=fct : i<cnt : i++ )
{
    char dir[100];
    p=fdir[i];
    spc(flgn,n,kln,knd,col);
    if( i<cnt-1 ) kfg+1; else kfg+3;
    keisen(kfg,n,knd);
    if( strlen(p) & 1 ) strcat(p," ");
    strepy(knd*20,"");
    if( flg & 4 )
        for( j=strlen(p) : j<kln-4 : j+2 )
            strcat(knd*20," ");
    if( strlen(p)<kln-2 ) strcat(knd*20,"-");
    if( *col !=0 ) tcolor(*(col+2));
    printf("%s".p);
    if( n<dep )
    {
        strepy(dir,path);
        strcat(dir,p);
        strcat(dir,"\\");
        fdisp(dir,sch.atr,n+1,dep,flg,kln,knd,wid,col);
    }
    else
        printf("\n");
}
/* ----- 空白を含むタブ (スペース) の表示 ----- */
spc(flgn,n,kln,knd,col)
unsigned char flgn,n,kln,*knd,*col;
{
    char j,k;
    if( *col !=0 ) tcolor(*(col+1));
    for( j=0 : j<n : j++ )
    {
        if( flg & 4 && *(knd+j)==1 )
            printf(" ");
        else
            printf(" ");
        for( k=0 : k<kln-2 : k++ ) printf(" ");
    }
}
/* ----- 空白の終端処理 ----- */
keisen(kfg,n,knd)
unsigned char kfg,n,*knd;
{
    switch( kfg )
    {
        case 1: *(knd+n)+1;
                printf(" ");
                break;
        case 2: *(knd+n)+1;
                printf(" ");
                break;
        case 3: *(knd+n)+0;
                printf(" ");
    }
}
/* ----- ワイルドカードを含むファイル名の比較 ----- */
char fcmp(pl,p2)
char *pl,*p2;
{
    char dl,d2,i;
    if( (i=strcmp(pl,"*.*")) == 0 ) return 0;
    if( *pl != '*' )
    {
        for( i=0; i<8 : i++ )
        {
            dl=*pl;
            d2=*p2;
            if( dl == '*' ) break;
            if( dl != '?' && dl != d2 ) return 1;
        }
    }
}

```

```

        if( dl == '*' ) break;
        if( dl == 0 ) return 0;
        .pl++;
        p2++;
    }
}
while( *pl++ != '*' );
while( *p2++ != '*' );
if( *pl != '*' )
{
    for( i=0; i<3 : i++ )
    {
        dl=*pl++;
        d2=*p2++;
        if( dl == '*' ) break;
        if( dl != '?' && dl != d2 ) return 1;
        if( dl == 0 ) return 0;
    }
}
return 0;
}
/* ----- ファイルサイズ目付時間の表示 ----- */
void disp_sdt(s,d,t,col,l)
long s;
unsigned int d,t;
char *col;
int l;
{
    if( l>15*8 )
    {
        if( *col ) tcolor (*(col+4));
        printf(" %6u".s);
    }
    if( l>15*8*10 )
    {
        if( *col ) tcolor (*(col+5));
        printf(" %2u%02u%02u.%80+(d>>9).(d & 0x1ff)>>5.d & 0x1f);
    }
    if( l>15*8*10*7 )
    {
        if( *col ) tcolor (*(col+6));
        printf(" %2u%02u.%t>>11.(t & 0x3ff)>>5);
    }
}
/* ----- カラーコード取得 (リバースは 8~F) ----- */
char colset(a)
char a;
{
    if( a>='0' && a<='7' ) return a-'0';
    if( a>='A' && a<='F' ) return a-'0'-7;
    return 7;
}
/* ----- テキストカラー ----- */
void tcolor(a)
int a;
{
    int b;
    char c;
    if( a < 7 )
    {
        if( a < 0 ) a = 0;
        b = (a & 1) * 4 + (a >> 1);
        c = '0' + b;
        printf("\x1b[3%c".c);
    }
    if( 7 < a )
    {
        if( 15 < a ) a = 15;
        a -= 8;
        b = (a & 1) * 4 + (a >> 1);
        c = '0' + b;
        printf("\x1b[4%c".c);
    }
}
/* -----

```



リスト 2 XDIR.C

```

/* -----
 * MS-DOSコマンド DIR機能拡張 (統合環境版)
 * Turbo_Cで作成
 * File "xdir.c"
 * 1989. 1/11 By H. Takeishi
 * ----- */
#include <dos.h>
#include <stdio.h>
#include <dir.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <process.h>
#include <bios8.h>
#include <jctype.h>

```

```

#define DIMAX 2000 /* ファイル属性格納構造体配列の最大数 */
void fdisp(), .dispstr(), .keisen(), .sdt_dp(),
colehg(), .colagn(), .outline(), .prnout(),
redirc(), .strucmov(), .dirsch(), .dptext(),
dpbin(), .run(), .nofna(),
char fcmp(), .colset(), .strinput();
int fmkse(), .fsasen(), .keyinp(),
handler(), .c_break();
static int tent, .dep, .kln, .wid, .dnt, .dnun[DIMAX],
fend, .depmax, .kimax, .fwp, .dps, .fdp, .brkck;
static char atr, .flg, .pag,
col[9], .path[80], .schf[15], .buff[160];
FILE *stream;

```

めスクロールできるんです。私はMSXみたいに段々スクロールだと思っていたのですが感動しました。そのうえ、パワーアップすると「スピードアップ」とかしゃべるんですよ。ファミコンもやるもんだと思いました。ドラスビは難しいですね。すぐにダメージをうけてしまいます。でも両面がきれいなので許します。音が今ひとつですが、私は音感はそのほどよくないので満足です。スベハリはキャラクタが小さい



```

struct fgroup      /* ファイル情報格納領域 */
{
    char          attr:      /* アトリビュート */
    unsigned      time:      /* ファイルの時刻 */
    unsigned      date:      /* ファイルの日付 */
    long          size:      /* ファイルのサイズ */
    char          name[13]:  /* ファイル名 */
    char          deep:      /* ファイルの深さ */
    int           kmodes:    /* 表示モード (ビット) */
} fp[DIMAX];

void main( int argc , char *argv[] )
{
    struct dfree    dfr;
    unsigned char  cur[80] , *a;
    int            cnt, drv, c , n , l , j , ypos, lbk, ink, lbk, dst;

    harderr(handler); /* ハードウェアエラーの間数設定 */
    ctrlbrk(c_break); /* CTRLブレークハンドラの設定 */
    setcbk(1);
    brkck=0;

    dep = 15; /* 初期値設定 */
    flg = 5; /* サブディレクトリの深さ (最大値) */
    /* 表示フラグ */
    /* 1ビット目 ファイル名を表示する */
    /* 2ビット目 ファイル名を横表示する */
    /* 3ビット目 属性を表示 (固定) */
    /* 4ビット目 サイズ日付時刻を表示 */
    atr = 0x10; /* アトリビュート */
    kln = 8; /* ディレクトリの罫線深さ表示幅 */
    wid = 76; /* 全体の表示幅 */
    pmd = 2;
    path[0]= 0; /* 指定ドライブ名+パス名 */
    strcpy(schf, ".*");
    /* サーチファイル名 (初期値は全て) */
    /* サーチファイル名 (初期値は全て) */
    col[0] = 14; /* 現在指定ファイルのカラー表示色 */
    for( j=1; j<8; j++) col[j]=7;

    /* ドライブ名+パス名 */
    /* スイッチ (Dn/Wn/F/Hn/S;) の設定 */
    for( cnt=1; cnt<argc; cnt++)
    {
        a=strupr(argv[cnt]);
        if( *a != '/' ) strcpy(path, a);
        else          svchost(a);
    }
    if( path[0]==0 ) getcwd(path, 80);
    if( strchr(path, ':') == NULL )
    {
        drv=getdisk();
        cur[0]='A'+drv; cur[1]=': '; cur[2]=0;
        strcat(cur, path);
        strcpy(path, cur);
    }
}

START:
textbank(1); /* 表示テキスト画面を2ページ目に設定 */
outline(0);
textcursor(0);
gotoxy(20, 8);
textcolor(0x2098);
cprintf(" [%s] を検索中です", path);
textcolor(0x2097);

getdfree(path[0]-'0', &dfr);
if( path[strlen(path)-1] == ':' )
{
    drv=path[0]-'0';
    getcurdir(drv, cur);
    strcat(path, "VY");
    strcat(path, cur);
}

c=path[strlen(path)-1];
if( c != 'VY' && c != ':' ) strcat(path, "VY");

depmax=0; /* ディレクトリの最大の深さ */
tcnt = 0; /* 総ファイル数 */
fnake(0, path, 0, 0); /* ディレクトリのファイル検索へ */

dep=depmax;
l = 0;
lbk=40;
dst= 0;
dps= 0;

LOOP1:
if( dep > dps ) klnax=(wid-13)/(dep-dps);
else            klnax=40;
if( klnax>40 ) klnax=40;
if( klnax < 2 ) klnax--;
if( klnax < 2 ) klnax=2;
fend=0;
cnt =dst;
if( dps>0 ) cnt++;

while( cnt<tent && cnt<DIMAX )
{
    char  fat, *fna, fck;

    fdp=fp[cnt].deep;
    fat=fp[cnt].attr;
    fna=fp[cnt].name;
    if( fdp<dps ) break;
    fck=( fat==0x10 || fcmp(schf, fna)==0 );
    if( fdp<dep && fck && kln<=(fdp-dps)*wid-13 )
    {
        /* ファイルとディレクトリを一個づつ */
        if( (flg & 3)==1 )
            dnum[fend++]=cnt;
    }
}

```

```

/* ディレクトリのみ */
if( (flg & 1) == 0 && fat == 0x10 )
    dnum[fend++]=cnt;

/* ファイルを横に複数、ディレクトリ */
if( (flg & 3) == 3 )
{
    dnum[fend++]=cnt;
    if( fat != 0x10 )
    {
        j=(wid-(fdp-dps)*kln-2)/16;
        while( --j>0 && ++cnt<tent )
        {
            fdp=fp[cnt].deep;
            fat=fp[cnt].attr;
            fna=fp[cnt].name;
            if( fdp<dps ) break;
            if( fat== 0x10 )
            {
                dnum[fend++]=cnt;
                break;
            }
            else if( fcmp(schf, fna) )
                j++;
        }
    }
}

if( fdp<dps || cnt>tent ) break;
cnt++;

if( fend>0 ) dnum[fend]=cnt;

for( cnt=0; cnt<fend; i++)dnum[cnt];
i=cnt-1; /* 以前のファイル指定行に再設定 */

if( pmd == 4 ) /* 指定したリダイレクトデバイスに出力 */
{
    redircf();
    textbank(0);
    exit(0);
}

/* ----- 統合環境での表示開始 ----- */

textcursor(0);
ypos= 1;
colagn(1);

LOOP2:
lbk=-2;
ink= 0;
fep= 0;

if( l>fend ) l=fend-1;
if( l<0 ) l=0;
if( fend<23 || l-ypos+1<0 ) ypos+=1;
else
    if( (j=l-ypos+24-fend) > 0 ) ypos++;

textcursor(0);
gotoxy(1, 1);
for( j=1-ypos+1; j<1-ypos+24; j++)
    fdisp(j, j<1-ypos+23);

LOOP3:
brkck=0;
colehg();
gotoxy(1, ypos);
fdisp(1, 0);
/*
gotoxy(1, ypos); /*
colagn(1);
if( l+ypos != lbk ) lbk=outdisp(1, lbk, cur);
lbk=l+ypos;

ink=keyinp(); /* キーボードからの入力 */
gotoxy(1, ypos);
fdisp(1, 0);

if( ink & 0x1000 ) /* 「CTRL」でドライブ変更 */
{
    buff[0]=(ink & 0x1f)+'0';
    if( buff[0]>'A' && buff[0]<='Z' )
    {
        buff[1]=': ';
        buff[2]=0;
        getdfree(buff[0]-'0', &dfr);
        if( dfr.df_sclua == -1 ) goto LOOP1;
        else /* ドライブエラーなし */
        {
            /* 「CAPS」ロックまたは「SHIFT」 */
            if( ink & 0x0300 ) strcat(buff, "VY");
            strcpy(path, buff);
            dep=15;
            flg=5;
            strcpy(schf, ".*");
            goto START;
        }
    }
}

/* ----- コマンドの解析と実行 ----- */

switch( ink & 0xff )
{
    int  fa, f, fl;

    case 0x1b: /* 終了 (ESC) */
    case 0x10: /* (Q) */
}

```

ですね、でも遊んでいるときはあんまり気になりません。それさえなければ90点は充分につけられる出来です。久々にソフトを買ったわりに、はずれなくて良かったです。(LEI)

リスト 2 XDIR.C

```

textbank(0);
exit(0);
break;

case 0x1c: /* CHDIR (Return) */
  chdir(cur);
  break;

case 0x2f: /* MS-DOS Command */
  strepy(cur, "");
  run(cur, 0);
  goto LOOP2;

case 0x66: /* ファイル実行 (F-5) */
case 0x13: /* (R) */
  if( fp[i].attr != 0x10 )
  {
    char p[4], *dp;
    if( cur[strlen(cur)-1] != '\Y' )
      strcat(cur, "Y");
    fs=dnum[i]+fwp;
    strcat(cur, fp[fs].name);
    printf("\n%a", cur);
    run(cur, ink & 0x0100 );
    goto LOOP2;
  }
  break;

case 0x32: /* ? テキストファイルの画面表示 */
  if( cur[strlen(cur)-1] != '\Y' )
    strcat(cur, "Y");
  fs=dnum[i]+fwp;
  strcat(cur, fp[fs].name);
  if( ink & 0x0100 ) dpbin(cur);
  else dptext(cur);
  goto LOOP2;

case 0x3f: /* ヘルプ画面表示 (HELP) */
  dptext("adir.hlp");
  goto LOOP1;

case 0x36: /* 半ページ巻上げ (Roll Up) */
  fs=i+(24-ypos);
  fl=fend-fs;
  if( fl<=0 )
  {
    fl=fend-i-1;
    if( fl>11 ) fl=11;
    i+=fl;
    ypos+=fl;
    break;
  }
  if( fl>11 ) fl=11;
  movetext(3, 2, fl, 77, 24, 3, 2);
  gotoxy(1, 23-fl);
  for( f=fs ; f<fs+fl ; f++) fdisp(f, 2);
  if( fend-i<11 ) ypos+=(fend-i-fl-1);
  i+=fl;
  gotoxy(1, ypos);
  break;

case 0x37: /* 半ページ巻下ろし (Roll Down) */
  if( i-ypos+1<=0 )
  {
    fl=i;
    if( fl>11 ) fl=11;
    i-=fl;
    ypos-=fl;
    break;
  }
  fs=i-ypos-10;
  fl=fs+11;
  if( fs<=0 )
  {
    movetext(3, 2, 77, 13, 3, 13);
    i-=11;
  }
  else
  {
    movetext(3, 2, 77, 13-fs, 3, 13+fs);
    i=0;
    ypos=1;
    fs=0;
  }
  gotoxy(1, 1);
  for( f=fs ; f<fl ; f++) fdisp(f, 1);
  gotoxy(1, ypos);
  break;

case 0x3a: /* 1行アップ (↑) */
  if( i>0 )
  {
    i--;
    if( --ypos<1 )
    {
      ypos=1;
      gotoxy(1, 1);
      insline();
    }
    fwp=0;
  }
  break;

case 0x3b: /* 縦縮小 (SHIFT + [-]) */
  if( ink & 0x0100 && kln>2 )
  {
    kln-=2;
    i=dnum[i];
    goto LOOP1;
  }
  /* 階層表示縮小 (GRPH + [-]) */
  if( ink & 0x0800 && dep>dps )
  {
    dep--;
    i=dnum[i];
    goto LOOP1;
  }

case 0x3c: /* 縦幅拡大 (SHIFT + [+]) */
  if( ink & 0x0100 && kln < klnmax )
  {
    kln+=2;
    i=dnum[i];
    goto LOOP1;
  }
  /* 階層表示拡張 (GRPH + [+]) */
  if( ink & 0x0800 && dep<dopmax )
  {
    dep++;
    i=dnum[i];
    goto LOOP1;
  }

fs=dnum[i];
fl=dnum[i+1]-1;
if( fl>tcnt ) fl=tcnt-1;
if( fs+fwp<fl && fsamen(fs+fwp+1, fl) )
  while( fcap(schf, fp[++fwp+fs].name) );
break;

case 0x3d: /* 1行ダウン (↓) */
  if( i<fend-1 )
  {
    i++;
    if( ++ypos>23 )
    {
      gotoxy(1, 1);
      delline();
      ypos=23;
      gotoxy(1, ypos);
    }
    fwp=0;
  }
  break;

case 0x3e: /* 先頭へ (HOMECLR) */
  i=0;
  ypos=1;
  goto LOOP2;

case 0x3a: /* 末尾へ (SHIFT + HOMECLR) */
  i=fend-1;
  if( i>22 ) ypos=23;
  else ypos=i+1;
  goto LOOP2;

case 0x20: /* ファイル表示スイッチ (F) */
  i=dnum[i];
  goto LOOP1;

case 0x11: /* ファイル横表示スイッチ (W) */
  flg +=(2-(flg & 2)*2);
  i=dnum[i];
  goto LOOP1;

case 0x1f: /* 日付表示スイッチ (D) */
  flg +=(8-(flg & 8)*2);
  goto LOOP2;

case 0x0d: /* カレントディレクトリ復元 (Ψ) */
  dps=0;
  dst=0;
  if( i==0 ) i=dnum[i]-1;
  else i=dnum[i];
  goto LOOP1;

case 0x2b: /* カレントディレクトリ変更 (C) */
  if( fp[dnum[i]].attr == 0x10 )
  {
    dst=dnum[i];
    dnum[0]=dst;
    dps=fp[dst].decp+1;
    i=0;
    goto LOOP1;
  }
  break;

case 0x18: /* 指定ファイルに出力 (O) */
  gotoxy(1, 23);
  textcolor(0x206);
  a=stripinput("\nfn 出力ファイル名", buff);
  if( *a )
  {
    if( (stream=fopen(a, "w")) != NULL )
    {
      prnout();
      close(stream);
    }
  }
  brkck=0;
  goto LOOP2;

case 0x19: /* プリンター出力 (P) */
  stream=stdprn;
  prnout();
  brkck=0;
  goto LOOP2;

case 0x1e: /* スイッチラインの変更 (S) */
  gotoxy(1, 23);
  textcolor(0x206);
  a=stripinput("\nfn スイッチライン", buff);

```

```

while( *a )
{
    if( *a == '/' ) seekset(a);
    a++;
}
goto LOOP1;

case 0x7f: /* 書き直し */
brkck=0;
lbnk=1;
goto LOOP2;
}

if( lbnk & 0x1100 ) /* SHIFT または CTRL によるダミータイム */
for( j=0; j<20000; j++) j++;
brkck=0;
goto LOOP3;
}

/* ----- ディレクトリまたはファイル表示 ----- */
void fdisp( int i, int lf )
{
    int ptr;
    char n, p[19];

    ptr=dnum[i];
    if( lf==2 )
    {
        sprintf(buff, "Yn");
        dispstr(pmd, 0, buff);
    }
    if( i<0 || i>=fend || ptr>=tcat )
    {
        if( pmd==2 || pmd==3 ) clrroot();
        if( lf==1 )
        {
            sprintf(buff, "Yn");
            dispstr(pmd, 0, buff);
        }
        return;
    }

    n=fp[ptr].deop;
    strcpy(p, fp[ptr].name);
    if( fp[ptr].attr != 0x10 ) /* --- ファイル名の表示 --- */
    {
        char *dp;

        if( flg & 2 )
        {
            int fct;
            char c;

            keisen(fp[dnum[i+1]-1].kmod, n);
            for( fct=ptr; fct<dnum[i+1]; fct++ )
            {
                strcpy(p, fp[fct].name);
                if( fcp[scfh, p] == 0 )
                {
                    if( fct > dnum[i] )
                        dispstr(pmd, 7, " ");
                    if( dp=strchr(p, ".") != 0 ) *dp=0;
                    sprintf(buff, "%-8s.%-4s", p, dp+1);
                    if( fcp == fct-ptr ) c=col[8];
                    else c=col[3];
                    dispstr(pmd, c, buff);
                }
            }
        }
        else
        {
            if( dp=strchr(p, ".") != 0 ) *dp=0;
            keisen(fp[ptr].kmod, n);
            sprintf(buff, "%-8s.%-4s", p, dp+1);
            dispstr(pmd, col[8], buff);
            if( flg & 8 )
                sdt_dp(ptr, wid=n*kin);
        }
    }
    else /* --- サブディレクトリの表示 ----- */
    {
        int j;
        char kn[40];

        keisen(fp[ptr].kmod, n);
        if( strlen(p) & 1 ) strcat(p, " ");
        strcpy(kn, " ");
        j=dnum[i+1];
        if( strlen(p)<kin-2 && fp[j].deop == n+1 && i+1<fend )
        {
            for( j=strlen(p); j<kin-4; j+=2 )
                strcat(kn, " ");
            strcat(kn, " ");
        }
        sprintf(buff, "<ks>", p);
        dispstr(pmd, col[7], buff);
        sprintf(buff, "%s", kn);
        dispstr(pmd, col[1], buff);
    }
    if( pmd==2 || pmd==3 ) clrroot();
    if( lf==1 )
    {
        sprintf(buff, "Yn");
        dispstr(pmd, 0, buff);
    }
}

/* ----- 文字列を指定したデバイスに出力 ----- */

```

```

void dispstr( char pmd, char c, char *s )
{
    switch( pmd )
    {
        case 1: printf("%s", s);
                break;
        case 2: if( c & 8 ) textreverse(REVERSE);
                textcolor( 0x200(c & 7) );
                cprintf("%s", s);
                if( c & 8 ) textreverse(NOREVERSE);
                break;
        case 3: textcolor( 0x200(c & 7) );
                cprintf("%s", s);
                fprintf(stream, "%s", s);
                break;
        case 4: cprintf("%s", s);
                printf("%s", s);
                break;
    }
}

/* ----- 罫線を含むタブ (スペース) 表示と終端処理 ----- */
void keisen( int kfg, unsigned char n )
{
    char j, k;

    buff[0]=0;
    for( j=dps; j<n; j++)
    {
        if( flg & 4 && (kfg & (1<<j)) ) strcat(buff, " | ");
        else strcat(buff, " ");
        for( k=0; k<kin-2; k++) strcat(buff, " ");
    }
    if( kfg & (1<<n) ) strcat(buff, " | ");
    else strcat(buff, " ");
    dispstr(pmd, col[1], buff);
}

/* ----- ファイルサイズ日付時間の表示 ----- */
void sdt_dp( int i, int l )
{
    if( l>15*8 )
    {
        sprintf(buff, "%8u", fp[i].size);
        dispstr(pmd, col[4], buff);
    }
    if( l>15*8+10 )
    {
        unsigned char y, d, t;

        y=(fp[i].date >> 9 )+80;
        d=(fp[i].date & 0x1ff)>>5;
        t=(fp[i].date & 0x1f);
        sprintf(buff, "%2u-%02u-%02u", (y % 100), d, t);
        dispstr(pmd, col[5], buff);
    }
    if( l>15*8+10+7 )
    {
        unsigned int t;
        t=fp[i].time;
        sprintf(buff, "%2u:%02u", t>>11, (t & 0x7ff)>>5);
        dispstr(pmd, col[6], buff);
    }
}

/* ----- 現在位置のファイル表示色変更 ----- */
void colebg()
{
    col[7]=col[0];
    col[8]=col[0];
}

/* ----- 現在位置のファイル表示色復元 ----- */
void colagn()
{
    col[7]=col[2];
    col[8]=col[3];
}

/* ----- カラーコード取得 (リバースは B~F) ----- */
char colset( char a )
{
    if( a=='0' && a<'9' ) return (a-'0');
    if( a=='A' && a<'F' ) return (a-'A'+10);
    return 7;
}

/* ----- 外枠に情報の表示 ----- */
int outdisp( int l, int lbn, char *cur )
{
    int fln;

    dirch(dnum[l], cur);
    window(l, l, 80, 25);
    gotoxy(8, l);
    sprintf(buff, "%s", cur);
    fln=strlen(buff);
    if( fln & 1 ) strcat(buff, " ");
    dispstr(2, 5+8, buff);
    textcolor(0x2004);
    while( lbn-->fln-2 ) cprintf("-");
    gotoxy(8, 25);
    textcolor(0x2005);
    if( flg & 1 )

```

リスト 2 XDIR.C

```

    sprintf(buff, "%s ", schf);
    if( strlen(schf) & 1 ) strcat(buff, " ");
    dispstr(2, 5, buff);
} else cprintf(" dir ");

textcolor(0x204);
cprintf("-----");
textcolor(0x207);

gotoxy(55, 1);
sprintf(buff, "%sd 行 ", i+1);
dispstr(2, 5+8, buff);
gotoxy(63, 25);

if( fend == 0 ) /* ファイル数が0の場合の処理 */
{
    textcolor(0x204);
    cprintf("-----");
    gotoxy(26, 12);
    textcolor(0x205);
    cprintf("ファイルはありません !!! Yn");
    gotoxy(26, 15);
    if( dps>0 ) cprintf("「Y」でカレントへ");
    else cprintf("ドライブ名の変更");
} else cprintf(" %sd 行 ", fend);

window(3, 2, 78, 24);

return fln;
}

/* ----- 外枠の作成・表示 ----- */
void outline()
{
    int i;

    window(1, 1, 80, 25);
    clrscr();
    textcolor(440x20);
    for( i=2; i<25; i++ )
    {
        gotoxy(i, 1);
        cprintf(" | ");
        gotoxy(79, i);
        cprintf(" | ");
    }
    gotoxy(1, 25);
    cprintf("-----");
    for( i=0; i<38; i++ ) cprintf(" - ");
    cprintf(" | ");
    gotoxy(41, 1);
    inline();
    cprintf(" | ");
    for( i=0; i<38; i++ ) cprintf(" - ");
    cprintf(" | ");
    textmode(VL8025);
    gotoxy(65, 1);
    textreverse(REVERSE);
    textcolor(0x205);
    cprintf(" HELPでヘルプ ");
    textreverse(NOREVERSE);
    window(3, 2, 78, 24);
}

/* ----- プリンターまたは指定ファイルへ出力 ----- */
void prnout()
{
    int fs;

    pmd =3;
    if( stream == stdprn )
    {
        window(1, 1, 80, 25);
        gotoxy(3, 25);
        textcolor(0x205);
        cprintf(" E S Cで中止 ");
        window(3, 2, 78, 24);
    }
    clrscr();
    sprintf(buff, "Yn%Yn", path);
    dispstr(pmd, 7, buff);
    if( brkck==0 ) /* デバイスに異常なし */
    {
        if( stream == stdprn ) fprintf(stream, "YrYx0a");
        for( fs=0; fs<fend; fs++ )
        {
            fdisp(fs, 1);
            if( stream == stdprn )
            {
                fprintf(stream, "YrYx0a");
                if( kbhit() )
                    if( getch() == 27 ) fs=fend;
            }
        }
    } else brkck=0;

    pmd=2;
}

/* ----- リダイレクトモードでのデバイス出力 ----- */
void redirec()
{
    int fs;

    sprintf(buff, "Yn%Yn", path);
    dispstr(4, 0, buff);
    for( fs=0; fs<fend; fs++ )

```

```

        fdisp(fs, 1);
        if( kbhit() )
            if( getch() == 27 ) fs=fend;
    }
}

/* ----- ディレクトリとファイルの検索格納 ----- */
/* ----- ディレクトリとファイルの検索格納 ----- */
int fnaek( char n , char *path , int cbk , int kfg )
{
    struct ffbk ffbk; /* ファイル情報格納バッファ領域 */
    unsigned char file[80], *p;
    int done, l, fcnt, scnt, ptr;

    scnt=cbk;
    fcnt=cbk;
    strcpy(file, path);

    gotoxy(20, 12);
    cprintf("%-s", file);
    clrscr();

    strcat(file, ".");

    /* --- ファイル名とディレクトリ名 (ファイル名の後) の格納 --- */
    done=findfirst(file, &ffbk, atr);
    while( !done && tent < DIMAX )
    {
        p=ffbk.ff_name;
        if( *p != '.' )
        {
            int j, atr, fpt;

            atr=ffbk.ff_attrib;
            if( atr == 0x10 || ( flg & 1 && fcup(schf, p) == 0 ) )
            {
                if( atr != 0x10 ) fpt=fcnt++;
                else fpt=scnt;
                for( j=tent-1; j>=fpt; j-- )
                    structmov(j+1, j);
                fpt[fcnt].deep = n;
                fpt[fcnt].kmod = kfg | (1<<n);
                fpt[fcnt].attr = ffbk.ff_attrib;
                fpt[fcnt].time = ffbk.ff_ftime;
                fpt[fcnt].date = ffbk.ff_fdate;
                fpt[fcnt].size = ffbk.ff_fsize;
                strcpy(fpt[fcnt].name, p);
                scnt++;
                tent++;
            }
        }
        done=findnext(&ffbk);
    }
    if( cbk < scnt ) fpt[scnt-1].kmod = kfg;
    if( depmax <= n && cbk < scnt ) depmax=n;

    ptr=fcnt;
    for( i=fcnt; i<scnt; i++ )
    {
        if( n<dep )
        {
            char dir[100];

            p=fpt[ptr].name;
            strcpy(dir, path);
            strcat(dir, p);
            strcat(dir, "Y");
            ptr=fnaek(n+1, dir, ptr+1, fpt[ptr].kmod);
        }
    }
    return ptr;
}

/* ----- 構造体メンバーのコピー ----- */
void structmov( int d , int s )
{
    fpt[d].attr = fpt[s].attr;
    fpt[d].time = fpt[s].time;
    fpt[d].date = fpt[s].date;
    fpt[d].size = fpt[s].size;
    fpt[d].deep = fpt[s].deep;
    fpt[d].kmod = fpt[s].kmod;
    strcpy(fpt[d].name, fpt[s].name);
}

/* ----- 指定領域の検索ファイル数カウント ----- */
int fsaen( int i , int iend )
{
    int cnt;

    cnt=0;
    while( i <= iend )
        if( fcup(schf, fpt[i+1].name) == 0 ) cnt++;
    return cnt;
}

/* ----- ワイルドカードを含むファイル名の比較 ----- */
char fcup( char *p1 , char *p2 )
{
    char d1, d2, i;

    if( strcmp(p1, ".") == 0 ) return 0;
    if( *p1 != '*' )
    {
        for( i=0; i<8; i++ )

```

汗流ってロケテ観戦しました。有難監督は専務だと思いません？素人目にも判定が覆らないことは分かりますよ！；同告を見て思ったこと
 ~あッ！とうとうソフトの広告がなくなったあアッッッ！！・エンサイクローペディア★ファンタニカ風によろしく！！

みんなっ！「ONE MORE KISS」買おうっ！そう僕が「REBECCA」(絵夢絶斗)


```

d1=*p1;
d2=*p2;
if( d1 == 'e' ) break;
if( d1 != '?' && d1 != d2 ) return 1;
if( d1 == '.' ) break;
if( d1 == 0 ) return 0;
p1++;
p2++;
}
}
p1=strchr(p1, '.');
p2=strchr(p2, '.');
if( p1 == NULL || p2 == NULL )
{
if( p1 == p2 ) return 0;
else return 1;
}
if( *p1++ != 'e' )
{
p2++;
for( i=0; i<3; i++ )
{
d1=*p1++;
d2=*p2++;
if( d1 == 'e' ) break;
if( d1 != '?' && d1 != d2 ) return 1;
if( d1 == 0 ) return 0;
}
}
return 0;
}
/* ----- 現在ファイルまでのパス名作成 ----- */
void dirschr( int i, char *s )
{
int j, dep;
char dir[16][16];
dep=-1;
for( j=0; j<=i; j++ )
{
if( fp[j].attr == 0x10 )
{
dep=fp[j].deop;
strcpy(dir[dep], fp[j].name);
}
}
strcpy(s, path);
for( j=0; j<=dep; j++ )
{
strcat(s, dir[j]);
if( j<dep ) strcat(s, "\\");
}
}
/* ----- スイッチライン入力 ----- */
suchset( char *s )
{
if( strncmp(s, "D", 2) == 0 )
{
dep=atoi(s+2);
if( dep>15 ) dep = 15;
}
if( strncmp(s, "W", 2) == 0 )
{
wid=atoi(s+2);
if( wid<40 ) wid = 40;
if( wid>136 ) wid = 136;
}
if( strncmp(s, "R", 2) == 0 ) pd=4;
if( strncmp(s, "F", 2) == 0 )
{
flg=(flg | 1);
if( s(a+2) == 'W' ) flg=(flg | 2);
if( s(a+2) == 'D' ) flg=(flg | 8);
}
if( strncmp(s, "H", 2) == 0 )
{
flg=(flg | 4);
if( s(a+2) ) kin=atoi(s+2);
if( kin & 1 ) kin--;
if( kin<2 ) kin = 2;
if( kin>40 ) kin = 40;
}
if( strncmp(s, "S", 3) == 0 )
{
flg=(flg | 1);
if( strlen(s+3) < 18 )
strcpy(schf, s+3);
}
if( strncmp(s, "C", 2) == 0 )
{
int j;
j=0;
a+=2;
while( s && j<7 )
col[j++] = colset(s+a++);
}
if( kin>wid-13 ) kin = wid-13;
}
/* ----- 文字列の取得 (指定ページにエコー表示) ----- */
char *strinput( char *p, char *bfin )
{
char c, *s;
int x, y;
textcursor(1);
textreverse(REVERSE);
cprintf("%s", p);
textreverse(NOREVERSE);
cprintf("%s", bfin);
}

```

```

x=wherex();
y=wherey();
a=bfin;
c=0;
while( c != 0x0d && c != 0x1b )
{
c=getch();
if( c == 8 && a != bfin ) /* BSキー */
{
a--;
gotoxy(x, y);
clrcool();
}
else if( (c>' ' && c<'z' || c == 0x1b) )
{
*o=0;
gotoxy(x, y);
cprintf("%s", bfin);
}
a=strupr(bfin);
return a;
}
/* ----- キー入力 (INT18H) ----- */
int keyinp()
{
KEY_INFO ink;
int key1, key2, asc, t;
ink.cmd=3; /* キーバハの初期化 */
bios98key(&ink);
for( t=0; t<1000; t++ ) /* キータッチ・ダミー */
key2=hbhit(); /* ブレック取得ダミー・開始 */
if( brck == 1 ) return 0x7f; /* ブレックからの復帰 */
ink.cmd=0; /* キー情報の取得 */
key1=bios98key(&ink);
ink.cmd=2;
key2=bios98key(&ink);
asc=ink.inter_code;
if( brck==1 ) return 0x7f; /* ブレックからの復帰 */
if( key1 == 0x1b ) return key1;
if( key2 & 0x10 ) return (key2<<8)+asc; /* ドライブ変更 */
else return (key2<<8)+ink.scan_code;
}
/* ----- コントロールブレックハンドラ ----- */
int c_break(void)
{
char c;
if( brck == 1 ) return 1; /* ダブルストップ解除 */
beep(1000);
brck=1;
setcbkr(0);
textreverse(REVERSE);
textcolor(0x2004);
gotoxy(20, 12);
cprintf(" S T O P キーが押されました !!! ");
gotoxy(20, 14);
cprintf(" 終了 (A) 無視・キャンセル (1) ");
textreverse(NOREVERSE);
textcolor(0x2007);
c=0;
while( c != 'A' && c != '1' && c != 'z' && c != 'c' )
c=toupper(getch());
setcbkr(1);
if( c == '1' || c == 'z' )
{
textcursor(0);
return 1;
}
textbank(0);
printf("\nS T O P キーにより異常終了します\n");
exit(0);
}
/* ----- ハードウェアエラーハンドラ ----- */
int handler( int errval, int ax )
{
int c;
char cr;
if( brck == 2 ) return 0;
beep(1000);
c=0;
brck=2;
textreverse(REVERSE);
textcolor(0x2004);
gotoxy(20, 12);
if( ax>0 )
{
cr='A'+(ax & 0xff);
cprintf(" ドライブ %c が設定されていません !!! ", cr);
gotoxy(20, 14);
cprintf(" 終了 (A) 再実行 (R) 無視 (1) ");
}
else
{
if( stream == stdprn )
cprintf(" プリンター %c が使用できません !!! ");
else
cprintf(" 指定したデバイスが異常です !!! ");
gotoxy(20, 14);
cprintf(" 終了 (A) 無視・キャンセル (1) ");
}
textreverse(NOREVERSE);
textcolor(0x2007);
while( c != 'A' && c != 'R' && c != '1' && c != 'z' && c != 'c' )
c=toupper(getch());
}

```

▶初投稿がいきなり12月号のBIG I/Oプラザに採用されたのには本当に驚きました。12月号にはほかに「初投稿だ」って言ってた人がいたけど、正直言って僕はページの下の方で小さく載ることを予想していたので、その人とはなにか違った扱いをされているのではないかと気になっていました。あのコーナーは目次にもあるくらいだから、もしかしら読者の5割くらいの方は、僕の名前を覚えてしまったのではないでしようか。(高橋孝次)

```

if( c == 'R' || c == 'X' ) return 1;
if( c == 'I' || c == 'C' ) return 0;

textbank(0);
printf("Ynエラー( %d )発生により異常終了しますYn",errval);
return 2;
}

/* ===== コマンドラインの入力とプログラムの実行 ===== */
void run( char *cur , int flg )
{
    char *a,*p,fnbf[80],ex[4],c,r,sta,*argv[10];
    int argc;
    gotoxy(1,23);
    textcolor(0x2047);
    sprintf(fnbf,"YnYn %s ",cur);

    strepy(ex,"");
    if( (p=stchr(cur," ")) != 0 )
        strepy(ex,p+1);
    if( strcmp(ex,"EXE")==0 || strcmp(ex,"COM")==0 )
    {
        a=stringput(fnbf,buffer);
        if( *a==0x1b ) return;
        textbank(0);
        argc=0;
        argv[argc++]=cur;
        p=fnbf;
        while( *a != 0 )
        {
            while( *a == " " ) a++;
            argv[argc]=p;
            while( *a && *a != " " ) *p++=*a++;
            *p++=0;
            printf(" %s",argv[argc]);
            argc++;
        }
        *argv[argc]=0;
        if( flg ) /* S H I F Tキーのチェック */
            sta=exexcv(cur,argv); /* オバーレイ */
        else
            sta=spawnv(P_WAIT,cur,argv); /* プロセス */
        if( sta == -1 )
        {
            textbank(1);
            clrscr();
            gotoxy(20,12);
            dispstr(2,col[0]," エラーが発生しました ");
            getch();
            return;
        }
    }
    else
    {
        if( strcmp(ex,"BAT")==0 )
        {
            a=stringput(fnbf,buffer);
            if( *a==0x1b ) return;
            strcat(cur," ");
            strcat(cur,a);
        }
        else
        {
            a=stringput("YnYn コマンド ",buffer);
            if( *a==0 || *a==0x1b ) return;
            strepy(cur,a);
        }
        textbank(0);
        system(cur);
    }
    c=(col[0] & 7); /* ファイル指定行と同じ色で表示 */
    if( col[0]>7 ) r='4';
    else r='3';
    printf("Yxib[%c%c]m",r,(c & 1)*4+(c>1)+0');
    printf("Yn X D I Rに戻ります。何かキーを押してください");
    printf("Yxib[m");
    getch();
    textbank(1);
}

/* ===== ファイル (ヘルプ) の内容をテキスト表示 ===== */
void dptext( char *fna )
{
    char i,ink,page,*p,*str,txbf[160];
    FILE *fp;

    if( (fp=fopen(fna,"r")) == NULL )
    {
        nofna(fna);
        return;
    }

    page=24;
    window(1,1,80,25);
    textcolor(0x2047);
    ink=0;
    i=page;
    while( ink != 0x1b )
    {
        if( i == page ) clrscr();
        while( i-- )
        {
            p=fgets(txbf,80,fp);
            if( p != NULL ) cprintf(" %s",txbf);
            else break;
        }
        if( i>0 ) cprintf("Yn");
        cprintf(" ");
        if( p == NULL )
            str=" [EOF] です。何かキーを押してください";
        else

```

```

        str=" E S Cで終了。↓で1行。他のキーで次へ ";
        dispstr(2,col[0],str);
        textcolor(0x2047);
        ink=getch();
        gotoxy(1,wherey());
        delline();
        if( p == NULL ) break;
        if( ink==10 ) i++;
        else i=page;
    }
    fclose(fp);
    outline();
}

/* ===== ファイルの内容を16進メモリーダンプ ===== */
void dpbin( char *fna )
{
    char i,j,k,ink,*p,*str,cbk,kflg,kfbk;
    unsigned char c[16];
    FILE *fp;

    if( (fp=fopen(fna,"rb")) == NULL )
    {
        nofna(fna);
        return;
    }

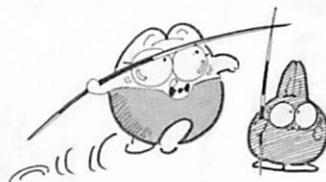
    window(1,1,80,25);
    textcolor(0x2047);
    clrscr();
    ink=0;
    c[17]=0;
    kflg=0;
    cbk=getc(fp) & 0x00ff;
    while( ink != 0x1b )
    {
        for( i=0; i<16; i++ )
        {
            c[i]=cbk;
            cprintf(" %05X : ",ftell(fp)-1);
            for( j=1; j<17; j++ )
            {
                c[j]=(getc(fp) & 0x00ff);
                if( ferror(fp) || feof(fp) )
                {
                    for( k=j; k<17; k++ ) c[k]='.';
                    break;
                }
            }
            for( j=0; j<16; j++ ) cprintf(" %02X",c[j]);
            if( kflg == 1 ) c[0]=' ';
            for( j=kflg; j<16; j++ )
            {
                if( iskanji(c[j]) && iskanji2(c[j+1]) )
                    j++;
                else if( !isalmkana(c[j]) == 0 )
                    c[j]='.';
            }

            cbk=c[16];
            kflg=0;
            if( j==17 ) kflg=1;
            else c[16]=0;
            cprintf(" %sYn",c);
            if( ferror(fp) || feof(fp) ) break;
        }
        cprintf("Yn");
        cprintf(" ");
        if( ferror(fp) || feof(fp) )
            str=" [EOF] です。何かキーを押してください ";
        else
            str=" E S Cで終了。他のキーで次のページ ";
        dispstr(2,col[0],str);
        textcolor(0x2047);
        ink=getch();
        if( ferror(fp) || feof(fp) ) break;
        delline();
        gotoxy(1,wherey());
    }
    fclose(fp);
    outline();
}

/* ===== */
void nofna( char *fna )
{
    clrscr();
    gotoxy(15,12);
    dispstr(2,col[0]," ファイルが見つかりません !!! ");
    gotoxy(15,14);
    sprintf(buffer," %s ",fna);
    dispstr(2,col[0],buffer);
    getch();
}

/* ===== */

```



使用方法
 (書式) XDIR [<ドライブ名パス名>] [<スイッチ>]
 (機能) D I S Kの内容を登録で階層的に表示。
 (パラメーターの設定は順不同で、大文字でも小文字でもかまわない)

(統合環境版におけるコマンドの説明)

- ESC または Q → プログラム終了
- Return → 現在指定行のディレクトリへカレントを変更
- [F-5] または R → 現在指定行のファイル実行 (チャイルド、終了後戻る)
- [SHIFT] + R → 上に同じ。EXECを実行 (オーバーレイ)
- M → MS-DOSのコマンドを実行。コマンドラインの入力。
- C → 現在指定行のディレクトリより内部を表示
- N → Cコマンドで設定したものを解除
- D → ファイルサイズ・日時表示スイッチ (表示←→非表示)
- ? → 指定行のファイルの内容をテキスト表示
- [SHIFT] + ? → 指定行のファイルの内容を16進ダンプ表示
- F → ファイル名表示のスイッチ (表示←→非表示)
- O → 指定したファイルに出力。ファイル名を入力。キャンセルはリタンのみ入力。
- P → プリンターに出力 (ESCキーで中止)
- S → 起動時のスイッチ内容を変更。コマンドラインを入力
- W → ファイル横表示のスイッチ (1個表示←→横に複数表示)

※ 「R M O S」などのコマンドは「ESC」でキャンセルできる

- [SHIFT] + [←] → 登録段落とし幅を縮小
- [SHIFT] + [→] → 登録段落とし幅を拡大
- [GRPH] + [←] → サブディレクトリ表示の深さを縮小する
- [GRPH] + [→] → サブディレクトリ表示の深さを拡張 (初期設定値以内)

(移動)

- ↓ ↑ → 指定行を1行前進、または後退
- → 1行に複数のファイルを表示した場合の横移動
- [RollUp] → 半ページの巻上げをする (11行前進)
- [RollDown] → 半ページの巻下ろしをする (11行後退)

※ 「SHIFT」を押した状態でゆっくり移動できる

- [HOMECLR] → ファイル全体の先頭に移動する
- [SHIFT]+[HOMECLR] → ファイル全体の末尾に移動する

(ドライブ変更)

- [CTRL]+[A]~[Z] → 指定ドライブの変更
 「CAPS」がロック (ルートより検索)
 アンロック (カレントより検索)

「SHIFT」を押下で、「CAPS」ロックと同じ

(起動時のオプションスイッチの説明)

<ドライブ名パス名>

「B:」: 「A:USER」のようにドライブ名、またはパス名を指定。省略した場合、カレントドライブのカレントディレクトリに設定される。

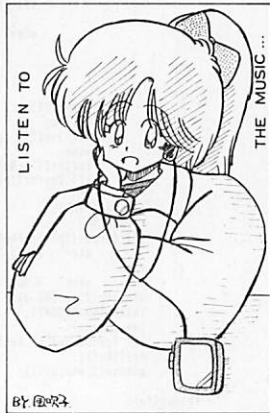
<スイッチ>

- /Dn サブディレクトリ表示の深さを指定する (n = 0~15)
 デフォルトの値は15 (最大の深さに設定)
- /F ファイル名を表示 (1行に1個ずつ)
- /FD ファイルのサイズ・日付・時刻を表示
- /FW ファイルを横に複数個ずつ表示
- /In 段落としの登録幅を指定 (n = 2~40) デフォルトの値は8
 (登録が全角なので偶数値で指定、奇数値を設定した場合は1減される)
- /Wn 全体の表示幅を設定 (n = 40~136) デフォルトは80
 (画面幅を狭めるたり、幅広プリンターに出力する場合に変更する)
- /S: <ファイル名 (ワイルドカードを含む)>
 指定したファイル名だけを表示 (検索表示)
- /Cn1n2n3n4n5n6n7
 登録、ディレクトリ名、ファイル名を色分けして表示
 色コード0~7を指定。8を加えた値 (89ABCDEF) を指定した場合は反転表示
 n1からの色指定は何個でもよい。
 (コマンドライン入力版では「nl」がないので、以下、1個ずつずれる)
 n1 [現在指定ファイル名]
 n2 [登録]
 n3 [サブディレクトリ名]
 n4 [ファイル名]
 (/FD でサイズ・日付時間表示を指定した場合のみ有効)
 n5 [ファイルサイズ]
 n6 [日付]
 n7 [時刻]
- /R 統合環境版でリダイレクトでプリンターやファイルに出力する場合に設定
 (統合環境版では画面表示の高速化のため直接画面に出力)

▼岐阜県 北信



▼富山県 風吹子



▼愛知県 風夢



▼神奈川県 りすとらん



▼埼玉県 あ



▼広島県 久保田武文



▼愛知県 中立国の玉三郎



▼千葉県 Z3

