

XDIR



ハードディスク環境を快適に

■ 武石秀男

いまや、“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となりつつあります。

以前発表した“Movingシステム”もこの統合環境化をはかっていますが、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ハード・ディスクがパンク状態となり、80M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、と気軽な気持ちでこのプログラム作り始めたのですが・・・

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の方は、Turbo Cの方言はほとんど使っていないので、他のC言語の処理系でも動作すると思います（ファイル関係が処理系によって違います。X68000にも2、3箇所の手直しで簡単に移植できました）。

しかし、リスト2の方は、画面表示にTurbo Cの方言をふんだんに使っているため、他の処理系では、うまく動かないと思います。また、長年メモリ不足の生活を強いられてきたせいか、桁分割、ビット分割のせこい癖が抜けずリストが読みづらくなっているところもあります。まあ、このビットとポインタ操作ができるからCをやっているところもありますが。



実行方法

書式は

XDIR [〈ドライブ名パス名〉] [〈スイッチ〉]

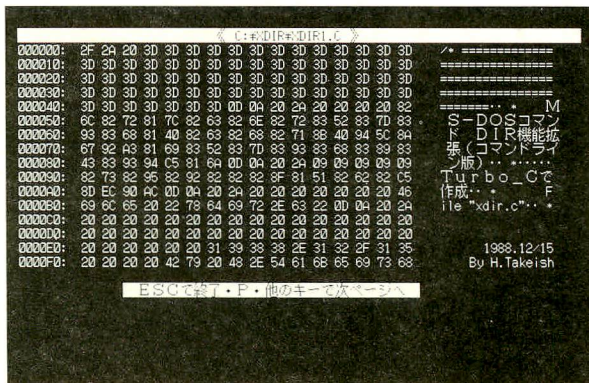
で、ディスクの内容を罫線で階層的に表示します。オプションによって起動時にきめ細かな設定ができます。統合環境版では、起動スイッチの変更も含めて、コマンドにより多彩な指定ができます。

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。以下に、起動時のオプションスイッチの説明をします（*は統合環境版だけです）。

ドライブ名とパス名

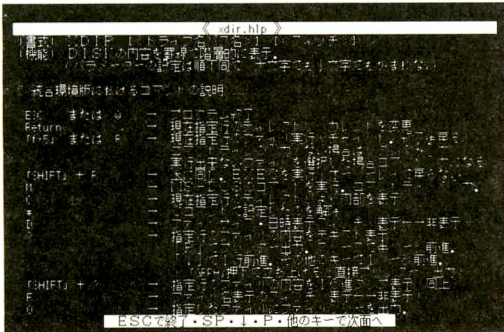
“B:”, “A:USER”のようにドライブ名、またはパス名を指定します。省略した場合はカレント・ドライブのカレント・ディレクトリに設定されます。

16進ダンプ



カーソル移動

ヘルプ画面



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または, CTRL 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 +	ファイル全体の末尾に移動する
HOMECLR	
ドライブ変更	
CTRL+A~Z	指定のドライブに変更 CAPS がロックされている場合はルートより検索 ロックされていない場合はカレントより検索 SHIFT を押下で, CAPS ロックと同じ カナ・キーロックの場合は今までの条件は変わらない

トラブルの発生と対策

ファイル間の移動をスムーズにするためにオートリビート機能を強調しています。

コマンド入力でキーを押し過ぎるとダブル・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。SHIFT キーを押した状態で ↑, ↓, →, ←, ROLLUP, DOWN を操作すると反応が鈍くなります。

統合環境版使用中に, STOP キーを押すとブレーク関数に処理が渡されます。A の入力で終了, I の入力によって継続ができます。ただし, 押した後に, 何かキーを押さないとブレーク関数が呼び出されません。これは, MS-DOS で設定されていないキー (ROLLUP, ROLL DOWN, HELP など) が使えるようにシステム・コールのINT 18H を呼び出しているのが原因のようです。

ディスクの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。その場合, 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=キャンセル)

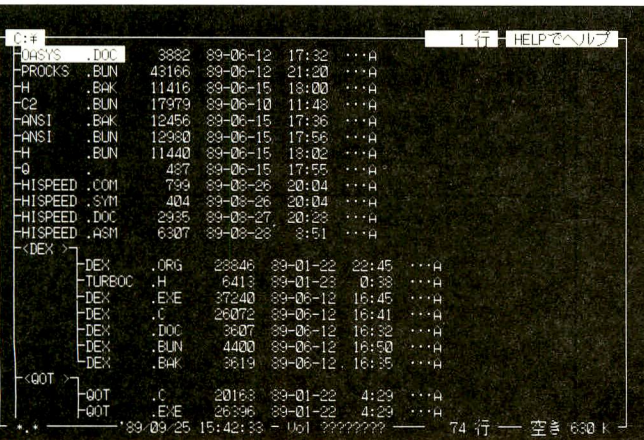
のキー入力で処理してください。また, プリンタ・ドライブを設定しないでプリントしようとした場合も同様に, エラールーチンが呼び出されます。

コマンドの選択によっては, 該当ファイルが存在しない場合があります。

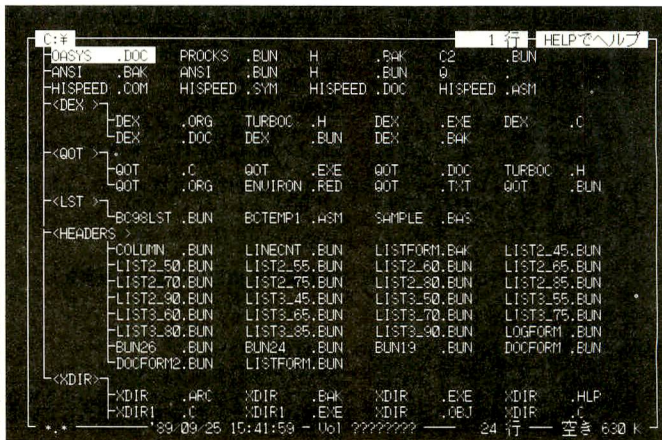
Cコマンドでそうなった場合, ¥ でキャンセル, その他は, ドライブ名の変更をするか, ESC キーで終了してください。

統合環境版

A, Dコマンドを使って日付けとアトリビュートを表示



Wコマンドはファイルが横に表示される



(アルファベット順)

A	アトリビュート属性 (システム・不可視) の表示スイッチ
C	現在指定行のディレクトリより内部を表示
D	日付表示スイッチ (表示←→非表示)
F	ファイル名表示のスイッチ (表示←→非表示) ディレクトリのみ表示したいとき (たとえば、目的の位置にすばやく移動し、そこでファイルを表示する) に活用する
M	MS-DOSのコマンドを実行。コマンドラインの入力
O	指定したファイルにファイル・ツリーを出力 ファイル名を入力 (キャンセルはESCまたはリタンのみ)
P	プリンタに出力 (ESCキーで中止できる)
Q (ESC)	プログラム終了
R (f・5)	現在指定行のファイル実行 (チャイルド。終了後戻る)
R+SHIFT	上に同じ。EXECを実行 (オーバーレイで戻らない)
S	起動時のスイッチ内容を変更 コマンドラインを入力 (例/H4/W60)
T	現在の日時表示のスイッチ (表示←→非表示) (Vと同様目障りなときは消去する)
V	ボリューム名表示スイッチ (表示←→非表示)
V+SHIFT	XDIR のバージョンの最下ライン表示スイッチ
W	ファイル横表示のスイッチ (1個表示←→横に複数表示) M O R S などのコマンドはESCでキャンセルできる
Return	現在指定行のディレクトリヘカレントを変更 コマンドラインの入力。ない場合は、リタンのみ入力
?	指定行のファイルの内容をテキスト表示 P でプリンタ出力。スペースで半ページ前進 ↓ で1行前進。その他のキーで1ページ前進 (GRPH 押下で立ち上げると直接プリンタへ出力)
?+SHIFT	指定行のファイルの内容を16進ダンプ表示 (同上)
¥	Cコマンドで設定したのを解除
DEL	ファイルの削除 (確認有り)
SHIFT+←	罫線段落とし幅を縮小する (階層が深いとき、または横に多くのファイルを表示したいときに利用する)
SHIFT+→	罫線段落とし幅を拡大する
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する (初期値以内)

起動時のオプション

/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
* /FS	システム・不可視 (シークレット) ファイルを表示
* /FA	ファイルのアトリビュート属性を表示 R (リードオンリファイル) S (システムファイル) H (隠しファイル) A (アーカイブファイル)
/Hn	段落としの罫線幅を指定 (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)
/Cn1n2 n3n4n5 n6n7	罫線、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 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「n1」がないので、以下、1個ずつずれる)
*	n1 [現在指定行ファイル名] n2 [罫線] 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]
*	「J」指定でパス (以前の値は保存される) 注) 多用すると色気違いあつかいされるので控えめに
* /R	統合環境版でリダイレクトでプリンタ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している)

物足りないあなたへ

MS-DOSのDIR機能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、軽き指向に傾いているため、始めはほどほどの機能を持つものを作る予定で、でき上がったのがリスト1でした。

ところが、「凝り性には軽きは似合わない」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。

ファイル数は2,000までとしています、ポインタ型の構造体を使い、malloc関数でヒープ領域にファイル属性を格納するようにすれば、メモリの許す限りのファイルが扱えるようになります。

その他、操作・機能の面で、まだまだ、改良の余地があり、現在、プログラムは進化の途中です。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, TYPEにちょび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。

参考文献

- 1) Turbo C ver1.5 ユーザーズガイド
- 2) Turbo C ver1.5 リファレンスガイド
- 3) 実習C言語, ASCII

XDIR.ARCの展開

```

XDIRのディスクを作ります。

ドライブAのルートディレクトリにPKXPAK.COM
ドライブBのルートディレクトリにXDIR.ARC

B:
A:PKXPAK XDIR.ARC

以下のファイルが展開されます。

XDIR      C      34351  89-01-17  14:14
XDIR      EXE    44318  89-01-17  14:15
XDIR      HLP    4981  89-01-17  14:20
XDIR1     C      8497  89-01-13  9:58
XDIR1     EXE    10024 89-01-13  9:58

こここまで展開は終了です。
必ず元のディスクからコピーしたファイルで操作してください。
    
```

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. 1/16 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 <bios98.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() .
  attr_dp() .colehg() .colagn() .underlin(),
  outline() .prnout() .redirec() .structmov(),
  makefna() .dirschr() .vollavel() .swchset(),
  run() .dptext() .dpsbin() .nofna() );
char colset( .fcmp(), *strinput());
int outdisp( .fmake() .fsamen() .keyinp(),
  c_break() .handler() .delfile() );

static int tent .dep .kln .wid .dent .dnum[DIMAX].
  fend .depmax .klnmax .fwp .dps .fdp,
  brkck ;
static double fresiz;
static char atr .flg .pmd,
  col[9] .path[80] .schf[13] .voln[13] .buff[160];

FILE *stream;

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 kmod; /* 罫線モード (ビット) */
} fp[DIMAX];

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

  harderr(handler); /* ハードウェアエラーの関数設定 */
  ctrlbrk(c_break); /* CTRLブレークハンドラの設定 */
  setcbkr(1);
  brkck=0;
  outportb(0xa2,0xc); /* グラフィック画面の停止 */

  /* 初期値設定 */
  dep = 15; /* サブディレクトリの深さ (最大値) */
  flg = 0x35; /* 表示フラッグ (ビット単位で設定) */
  /* 1ビット(0x01) ファイル名を表示する */
  /* 2ビット(0x02) ファイル名を横表示する */
  /* 3ビット(0x04) 罫線を表示 (固定) */
  /* 4ビット(0x08) サイズ日付時刻を表示 */
  /* 5ビット(0x10) 現在の日時表示 */
  /* 6ビット(0x20) ボリューム名表示 */
  /* 7ビット(0x40) アトリビュート属性表示 */
  /* 8ビット(0x80) バージョン番号表示 */
  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) */

  col[0] = 14; /* 現在指定ファイルのカラー表示色 */
  for( j=1; j<9; j++) col[j]=7;

  /* 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 swchset(a);
  }
  if( path[0]==0 ) getcwd(path, 80);
  if( strchr(path, ':') == NULL )
  {
    drv=getdisk(); /* ドライブ名取得 */
    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();
textcurscr(0);
gotoxy(20, 8);
textcolor(0x20*6);
cprintf(" [%s] を検索中です, path);
textcolor(0x20*7);

vollavel(voln, path); /* ボリューム名取得 */
getdfree(path[0]-'@', &dfr); /* ディスク空領域取得 */
fresiz=(dfr.df_sclus*dfr.df_bsec/1024)*dfr.df_aval;

if( path[strlen(path)-1] == ':' )
{
  drv=path[0]-'@'; /* カレントディレクトリ取得 */
  getcurdir(drv, cur);
  strcat(path, "YY");
  strcat(path, cur);
}

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

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

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

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

while( cnt<tcnt && 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<tcnt )
        {
          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]; ) cnt++;
i=cnt-1; /* 以前のファイル指定行に再設定 */

```


XDIR.C

```

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

/* ===== 統合環境での表示開始 ===== */

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

LOOP2:

ibk= -2;
ink= 0;
fwp= 0;

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

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

LOOP3:
brkck=0;
colehg();
gotoxy(1,ypos);
fdisp(i,0);
colagn();
if( i+fwp != ibk ) lbk=outdisp(i,lbk,cur);
ibk=i+fwp;

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

if( ink & 0x1000 ) /* CTRL でドライブ変更 */
{
    buff[0]=(ink & 0xf)+'@';
    if( buff[0]>'A' && buff[0]<='Z' )
    {
        buff[1]=': ';
        buff[2]=0;
        getdfr(buff[0]-'@',&dfr);
        if( dfr.df_sclus == -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,"YY");
            strcpy(path,buff);
            if( (ink & 0x0400)==0 ) /* [加]ロックなし */
            {
                dep= 15;
                flg=0x35;
                strcpy(schf,"*.");
            }
            goto START;
        }
    }
}

/* ----- コマンドの解析と実行 ----- */

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

    case 0x1b: /* 終了 (ESC) */
    case 0x10: /* (Q) */
        textbank(0);
        exit(0);
        break;

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

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

    case 0x66: /* ファイル実行 (f・5) */
    case 0x13: /* (R) */
        if( fp[dnum[i]].attr !=0x10 )
        {
            char p[4],*dp;

            makefna(i,cur);
            printf("Yn%s",cur);
            run(cur,ink & 0x0100 );
            goto LOOP2;
        }
        break;

    case 0x39: /* ファイルの削除 (DEL) */
        if( delfile(i,cur) ) goto LOOP1;
        else goto LOOP2;

    case 0x32: /* テキストファイルの画面表示 (?) */
        if( ink & 0x0800 ) f=3;
        else f=2;
        makefna(i,cur);
        if( ink & 0x0100 ) dpbin(cur,f);
        else dpdtext(cur,f);
        goto LOOP2;

    case 0x3f: /* ヘルプ画面表示 (HELP) */
        dpdtext("xdir.hlp",2);
        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+f1,77,24,3,2);
        gotoxy(1,23-f1);
        for( f=fs ; f<fs+fl ; f++ ) fdisp(f,2);
        if( fend-i<11 ) ypos+=(fend-i-f1-1);
        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: /* 罫線幅縮小 (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;
        }
        fs=dnum[i];
        if( fwp>0 && fsamen(fs,fs+fwp-1) )
            while( fcmp(schf,fp[--fwp+fs].name) );
        break;

    case 0x3c: /* 罫線幅拡大 (SHIFT + [→]) */
        if( ink & 0x0100 && kln < klnmax )

```

```

        kln+=2;
        i=dnum[i];
        goto LOOP1;
    }
    /* 階層表示拡張 (GRPH + [→]) */
    if( ink & 0x0800 && dep<depmax )
    {
        dep++;
        i=dnum[i];
        goto LOOP1;
    }

    fs=dnum[i];
    fl=dnum[i+1]-1;
    if( fl>=tcnt ) fl=tcnt-1;
    if( fs+fw<fl && fsamen(fs+fw+1,fl) )
        while( fcmp(schf,fp[fw+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);
        }
        fw=0;
    }
    break;

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

case 0xae: /* 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) */
    flg +=(1-(flg & 1)*2);
    i=dnum[i];
    goto LOOP1;

case 0x11: /* ファイル横表示スイッチ (W) */
    flg +=(2-(flg & 2)*2);
    if( flg & 2 ) flg=(flg & 0xf7);
    i=dnum[i];
    goto LOOP1;

case 0x1f: /* 日時表示スイッチ (D) */
    flg +=(8-(flg & 8)*2);
    if( flg & 8 ) flg=(flg & 0xfd);
    goto LOOP1;

case 0x1d: /* アトリビュート属性表示スイッチ (A) */
    flg +=(0x40-(flg & 0x40)*2);
    goto LOOP1;

case 0x14: /* 現在の日時表示スイッチ (T) */
    underlin();
    flg +=(0x10-(flg & 0x10)*2);
    ibk=-2;
    goto LOOP2;

case 0x2c: /* ボリューム名表示スイッチ (V) */
    underlin();
    if( ink & 0x0100 )
        flg +=(0x80-(flg & 0x80)*2);
    else flg +=(0x20-(flg & 0x20)*2);
    ibk=-2;
    goto LOOP2;

case 0x0d: /* カレントディレクトリ復元 (Y) */
    dst=0;
    dps=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].deep+1;
        i=0;
        goto LOOP1;
    }
    break;

case 0x18: /* 指定ファイルに出力 (O) */
    gotoxy(1,23);
    textcolor(0x20*6);
    a=strinput("YnYn 出力ファイル名 ");

```

```

    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(0x20*6);
    a=strinput("YnYn スイッチライン ");
    while( *a )
    {
        if( *a == '/' ) swchset(a);
        a++;
    }
    goto LOOP1;

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

if( ink & 0x0100 ) /* SHIFT 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[13];

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

    n=fp[ptr].deep;
    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( fcmp(schf,p) == 0 )
                {
                    if( fct > dnum[i] )
                        dispstr(pmd,7,"");
                    if( (dp=strchr(p,'.') != 0 ) *dp==0;
                    sprintf(buff,"%-8s.%-4s",p,dp+1);
                    if( fwp == 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*kln-15);
            if( flg & 0x40 && n*kln+15+(flg & 8)*3+7 < wid )
                attr_dp(fp[ptr].attr);
        }
    }
}

else /* --- サブディレクトリの表示 --- */
{
    int j;
    char ksn[40];

```

```

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

/* ----- 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 3: fprintf(stream, "%s", s);
        case 2: if( c & 8 ) textreverse(REVERSE);
                textcolor( 0x20*(c & 7) );
                fprintf("%s", s);
                if( c & 8 ) textreverse(NOREVERSE);
                break;

        case 4: fprintf("%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; /* 各ファイル毎ビット単位で設定 */
    /* 1ビット ~ 16ビット */
    buff[0]=0; /* 0階層の状態 15階層の状態 */
    for( j=dps; j<n; j++)
    {
        if( flg & 4 && (kfg & (1<<j)) ) strcat(buff, "|");
        else strcat(buff, "-");
        for( k=0; k<kln-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>8 ) /* ファイルサイズ表示 */
    {
        sprintf(buff, "%8lu", fp[i].size);
        dispstr(pmd.col[4], buff);
    }
    if( l>8+10 ) /* 年月日表示 */
    {
        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>8+10+7 ) /* 時分秒表示 */
    {
        unsigned int t;
        t=fp[i].time;
        sprintf(buff, "%2u:%02u", t>>11, (t & 0x7ff)>>5);
        dispstr(pmd.col[6], buff);
    }
}

/* ----- アトリビュート属性の表示 ----- */
void attr_dp( char atr )
{
    if( atr & 1 ) dispstr(pmd.col[6], "R"); /* リードオンリ */
    else dispstr(pmd.col[6], "-");
    if( atr & 2 ) dispstr(pmd.col[6], "H"); /* 不可視 */
    else dispstr(pmd.col[6], "-");
    if( atr & 4 ) dispstr(pmd.col[6], "S"); /* システム */
    else dispstr(pmd.col[6], "-");
    if( atr & 0x20 ) dispstr(pmd.col[6], "A"); /* アーカイブ */
    else dispstr(pmd.col[6], "-");
}

/* ----- 現在指定行のファイル表示色変更 ----- */
void colehg()
{
    col[7]=col[0];
    col[8]=col[0];
}

/* ----- 現在指定行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 8~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 i, int lbk, char *gur )
{
    int fln, j;

    dirsch(dnum[i].cur);
    window(1, 1, 80, 25);
    gotoxy(3, 1);
    sprintf(buff, "%s", cur);
    fln=strlen(buff);
    dispstr(2, 5+8, buff);
    textcolor(0x20*4);
    while( (lbk==2)>fln-2 ) fprintf("-");
    gotoxy(3, 25);
    textcolor(0x20*5);
    if( flg & l ) /* ファイル名表示 */
    {
        sprintf(buff, "%s", schf);
        dispstr(2, 5, buff);
    }
    else fprintf(" dir "); /* ディレクトリ名のみ表示 */
    textcolor(0x20*4);
    while( wherex()<15 ) fprintf("-");
    textcolor(0x20*7);

    gotoxy(55, 1); /* 現在指定行 */
    sprintf(buff, "%5d行", i+1);
    dispstr(2, 5+8, buff);
    gotoxy(55, 25);

    if( fend == 0 ) /* ファイル数が0の場合の処理 */
    {
        textcolor(0x20*4);
        fprintf("-");
        gotoxy(26, 12);
        textcolor(0x20*5);
        fprintf("ファイルはありません !!!");
        gotoxy(26, 15);
        if( dps>0 ) fprintf("「V」でカレントへ");
        else fprintf("ドライブ名バスの変更");
    }
    else
    {
        sprintf(buff, "%5d行", fend); /* 全行数 */
        dispstr(2, 5, buff);
    }
    gotoxy(67, 25); /* ドライブ空き領域の表示 */
    if( fresiz<1000 )
        sprintf(buff, "空き%4.0f K", fresiz); /* キロバイト単位 */
    else
    {
        fresiz=fresiz/1000; /* メガバイト単位 */
        if( fresiz<10 )
            sprintf(buff, "空き%4.2f M", fresiz);
        else
            sprintf(buff, "空き%4.1f M", fresiz);
    }
    dispstr(2, 5+0, buff);

    textcolor(0x20*4);
    if( flg & 0x10 && (flg & 0x80)==0 ) /* 現在日時の表示 */
    {
        struct date dt;
        struct time tm;
        int y;

        getdate(&dt);
        y=(dt.da_year % 100);
        gotoxy(17, 25);
        fprintf("%02d/%02d/%02d", y, dt.da_mon, dt.da_day);
        gettime(&tm);
        gotoxy(27, 25);
        fprintf("%02d:%02d:%02d", tm.ti_hour, tm.ti_min, tm.ti_sec);
    }
}

```

```

if( flg & 0x20 && (flg & 0x80)==0 ) /* ボリューム名表示 */
{
    gotoxy(37,25);
    cprintf(" Vol %s ",voln);
}
if( flg & 0x80 ) /* バージョン番号表示 */
{
    gotoxy(17,25);
    cprintf(" XDIR Ver1.0 Copyright(c) 1989.1 H.T ");
}
window(3,2,78,24);
return fln;
}

/* ----- 外枠の下線修復 ----- */

void underlin()
{
    window(1,1,80,25);
    textcolor(0x20*4);
    gotoxy(17,25);
    cprintf("-----");
    textcolor(0x20*7);
    window(3,2,78,24);
}

/* ----- 外枠の作成・表示 ----- */

void outline()
{
    int i;

    window(1,1,80,25);
    clrscr();
    textcolor(4*0x20);
    for( i=2 ; i<25 ; i++ )
    {
        gotoxy(1,i);
        cprintf(" | ");
        gotoxy(79,i);
        cprintf(" | ");
    }
    gotoxy(1,25);
    cprintf(" ^ ");
    for( i=0 ; i<38 ; i++ ) cprintf(" --- ");
    cprintf(" ^ ");
    gotoxy(1,1);
    inline();
    cprintf(" r ");
    for( i=0 ; i<38 ; i++ ) cprintf(" --- ");
    cprintf(" ^ ");
    textmode(VL8025);
    gotoxy(65,1);
    dispstr(2,5+8," HELPでヘルプ ");

    window(3,2,78,24);
}

/* ----- プリンターまたは指定ファイルへ一覧出力 ----- */

void prnout()
{
    int fs;

    pmd =3;
    if( stream == stdprn )
    {
        window(1,1,80,25);
        gotoxy(3,25);
        textcolor(0x20*5);
        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() == 0x1b ) break;
            }
        }
    }
    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() == 0x1b ) break;
    }
}

/* ===== デレクトリとファイルの検索格納 ===== */

int fmake( char n , char *path , int cbk , int kfg )
{
    struct ffbk ffbk; /* ファイル情報格納バッハ領域 */
    unsigned char file[80],*p;
    int done, i, fcnt, scnt, ptr;

    sent=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 && tcnt < DIMAX )
    {
        p=ffbk.ff_name;
        if( *p != '.' )
        {
            int j,atr,fpt;

            atr=ffbk.ff_attrib;
            if( atr == 0x10 || ( flg & 1 && fcmp(schf,p)==0 ) )
            {
                if( atr != 0x10 ) fpt=fcnt++;
                else fpt=scnt;
                for( j=tcnt-1 ; j>=fpt ; j-- )
                    structmov(j+1,j);
                fp[ptr].deep = n;
                fp[ptr].kmod = kfg | (1<<n);
                fp[ptr].atr = ffbk.ff_attrib;
                fp[ptr].time = ffbk.ff_time;
                fp[ptr].date = ffbk.ff_date;
                fp[ptr].size = ffbk.ff_size;
                strcpy(fp[ptr].name,p);
                scnt++;
                tcnt++;
            }
        }
        done=findnext(&ffbk);
    }
    if( cbk < sent ) fp[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=fp[ptr].name;
            strcpy(dir,path);
            strcat(dir,p);
            strcat(dir,"YY");
            ptr=fmake(n+1,dir,ptr+1,fp[ptr].kmod);
        }
    }
    return ptr;
}

/* ----- 構造体メンバーのコピー ----- */

void structmov( int d , int s )
{
    fp[d].atr = fp[s].atr;
    fp[d].time = fp[s].time;
    fp[d].date = fp[s].date;
    fp[d].size = fp[s].size;
    fp[d].deep = fp[s].deep;
    fp[d].kmod = fp[s].kmod;
    strcpy(fp[d].name,fp[s].name);
}

/* ===== 指定領域の検索ファイル数カウント ===== */

int fsamen( int i , int iend )
{
    int cnt;

    cnt=0;
    while( i <= iend )
        if( fcmp(schf,fp[i++].name) == 0 ) cnt++;
}

```


XDIR.C

```

    return cnt;
}

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

    if( strcmp(p1,"*.*") == 0 ) return 0;
    if( *p1 != '*' )
    {
        for( i=0; i<8 ; i++ )
        {
            d1=*p1;
            d2=*p2;
            if( d1 == '*' ) 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++ != '*' )
    {
        p2++;
        for( i=0; i<3 ; i++ )
        {
            d1=*p1++;
            d2=*p2++;
            if( d1 == '*' ) break;
            if( d1 != '?' && d1 != d2 ) return 1;
            if( d1 == 0 ) return 0;
        }
    }
    return 0;
}

/* ===== 現在ファイルの完全ファイル名作成 ===== */
void makefna( int i , char str[] )
{
    if( str[strlen(str)-1] != '\0' )
        strcat(str,"\\");
    strcat(str,fp[dnum[i]+fwp].name);
}

/* ===== 現在ファイルまでのパス名作成 ===== */
void dirschr( int i , char *str )
{
    int j,dep;
    char dir[16][13];

    dep=-1;
    for( j=0 ; j<=i ; j++ )
    {
        if( fp[j].attr == 0x10 )
        {
            dep=fp[j].deep;
            strcpy(dir[dep],fp[j].name);
        }
    }
    strcpy(str,path);
    for( j=0 ; j<=dep ; j++ )
    {
        strcat(str,dir[j]);
        if( j<dep ) strcat(str,"\\");
    }
}

/* ===== ボリュームラベル作成 ===== */
void vollavel( char *vna , char *pash )
{
    struct ffbk ffbk; /* ファイル情報格納バッハ領域 */
    unsigned char file[10];
    int done;

    strcpy(file,pash);
    file[3]=0;
    strcat(file,"*.*");
    if( findfirst(file,&ffbk,FA_LABEL) )
        strcpy(vna,"????????");
    else strcpy(vna,ffbk.ff_name);
}

/* ===== スイッチライン入力 ===== */
void swchset( char *a )
{
    if( strcmp(a,"/D".2) == 0 )
    {

```

```

        dep=atoi(a+2);
        if( dep>15 ) dep = 15;
    }
    if( strcmp(a,"/W".2) == 0 )
    {
        wid=atoi(a+2);
        if( wid<40 ) wid = 40;
        if( wid>136 ) wid = 136;
    }
    if( strcmp(a,"/R".2) == 0 ) pmd=4;
    if( strcmp(a,"/F".2) == 0 )
    {
        flg=(flg | 1);
        if( *(a+2) == 'W' ) flg=(flg | 0x02); /* 横複数 */
        if( *(a+2) == 'D' ) flg=(flg | 0x08); /* 日時 */
        if( *(a+2) == 'A' ) flg=(flg | 0x40); /* 日時 */
        if( *(a+2) == 'S' ) atr=(atr | 7); /* 不可視 */
    }
    if( strcmp(a,"/H".2) == 0 )
    {
        flg=(flg | 4);
        if( *(a+2) ) kln=atoi(a+2);
        if( kln & 1 ) kln--;
        if( kln<2 ) kln = 2;
        if( kln>40 ) kln = 40;
    }
    if( strcmp(a,"/S:".3) == 0 )
    {
        flg=(flg | 1);
        if( strlen(a+3) < 13 )
            strcpy(schf,a+3);
    }
    if( strcmp(a,"/C".2) == 0 )
    {
        int j;
        j=0;
        a+=2;
        while( *a && j<7 )
        {
            if( *a == '.' )
            {
                j++;
                a++;
            }
            else col[j++]=colset(*a++);
        }
    }
    if( kln>wid-13 ) kln = 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,*a;
    int x,y;

    textcursor(1);
    textreverse(REVERSE);
    cprintf("%s",p);
    textreverse(NOREVERSE);
    cprintf(" ");
    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);
            clrreol();
        }
        else
            if( c==' ' && c!='z' || c == 0x1b )
                *a++ = c;
        *a=0;
        gotoxy(x,y);
        cprintf("%s",bfin);
    }
    a=strupr(bfin);
    return a;
}

/* ===== キー入力 (INT 18H) ===== */
int keyinp()
{
    KEY_INFO ink;
    int key1,key2,asc,t;

    ink.cmd=3; /* キーバッハの初期化 */
    bios98key(&ink);
    for( t=0 ; t<1000 ; t++ ) /* キータッチ・ダミー */
        key2=kbhit(); /* ブレーク取得ダミー関数 */
    if( brkck == 1 ) return 0x7f; /* ブレークからの復帰 */
    ink.cmd=0; /* キー情報の取得 */
    key1=bios98key(&ink);
    ink.cmd=2;
}

```

```

key2=bios98key(&ink);
asc=ink.inter_code;
if( brkck==1 ) return 0x7f; /* ブレークからの復帰 */
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;
}

/* ===== コントロールブレークハンドラ ===== */
int c_break(void)
{
char c;

if( brkck == 1 ) return 1; /* ダブルストップ解除 */
beep(1000);
brkck=1;
setcbkr(0);
textreverse(REVERSE);
textcolor(0x20*4);
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 (I) ");
textreverse(NOREVERSE);
textcolor(0x20*7);
c=0;
while( c !='A' && c !='I' && c !='f' && c !='c' )
c=toupper(getch());
setcbkr(1);
if( c =='I' || c =='c' )
{
textcursor(0);
return 1;
}

textbank(0);
printf("YnS T O P キーにより異常終了しますYn");
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( brkck == 2 ) return 0;
beep(1000);
c=0;
brkck=2;
textreverse(REVERSE);
textcolor(0x20*4);
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 (I) ");
}
else
{
if( stream == stdprn )
cprintf(" プリンターが使用できません !!! ");
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 (I) ");
}
textreverse(NOREVERSE);
textcolor(0x20*7);
while( c !='A' && c !='R' && c !='I' && c !='f' && c !='s' && c !='c' )
c=toupper(getch());

if( c =='R' || c =='s' ) 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, fbuf[80], ex[4], c, r, sta, *argv[10];
int argc;

gotoxy(1, 23);
textcolor(0x20*6);
sprintf(fbuf, "YnYn %s ", cur);

strcpy(ex, " ");
if( (p=strchr(cur, ' ')) !=0 )
strcpy(ex, p+1);
if( strcmp(ex, "EXE")==0 || strcmp(ex, "COM")==0 )
{
a=strinput(fbuf, buff);
if( *a==0x1b ) return;
textbank(0);

```

```

argc=0;
argv[argc++]=cur;
p=fbuf;
while( *a != 0 )
{
while( *a ==' ' ) a++;
argv[argc]=p;
while( *a && *a !=' ' ) *p++=*a++;
*p++=0;
printf(" %s", argv[argc]);
argc++;
}
printf("Yn");
argv[argc]=0;
if( flg ) /* S H I F Tキー押下チェック */
sta=execv(cur, argv); /* オバーレイ */
else
sta=spawnv(P_WAIT, cur, argv); /* 子プロセス */
if( sta == -1 )
{
beep(1000);
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=strinput(fbuf, buff);
if( *a==0x1b ) return;
strcat(cur, " ");
strcat(cur, a);
}
else
{
a=strinput("YnYn コマンド ", buff);
if( *a==0 || *a==0x1b ) return;
strcpy(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("Yxlb[%c%c", 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("Yxlb[m");
getch();
outportb(0xa2, 0x0c); /* グラフィック画面の停止 */
textbank(1);
outline();
}

/* ===== ファイル(ヘルプ)の内容をテキスト表示 ===== */
void dptext( char *fna , int pfg )
{
char i, page, *p, *str, txbf[82];
FILE *fp;
int ink;

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

if( pfg==3 ) stream=stdprn;
page=23;
window(1, 1, 80, 25);
clrscr();
for( i=0 ; i<80 ; i++ ) buff[i]=' ';
dispstr(2, 5+8, buff);
gotoxy(36-strlen(fna)/2, 1);
sprintf(buff, " %s ) Yn", fna);
dispstr(pfg, 5+8, buff);
if( pfg == 3 ) fprintf(stream, "YrYx0aYrYx0a");
window(1, 2, 80, 25);
textcolor(0x20*7);
ink=0;
i=page;
while( ink !=0x1b )
{
if( i == page && pfg == 2 ) clrscr();
while( i-- || pfg==3 )
{
p=fgets(txbf, 80, fp);
if( p !=NULL )
{
sprintf(buff, "%s", txbf);
dispstr(pfg, 7, buff);
if( pfg==3 )
fprintf(stream, "YrYx0a");
if( pfg==3 && kbhit() )
if( getch() == 0x1b ) break;

```



```

    }
    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・S P ・↓・P ・他のキーで次面へ ";
dispstr(2,col[0],str);
textcolor(0x20*7);
pfg=2;
ink=toupper(getch());
if( ink=='P' || ink=='t' ) /* 「P」でプリンター出力 */
{
    pfg=3;
    fclose(fp);
    goto AGAIN;
}
gotoxy(1,wherey());
delline();
if( p == NULL ) break;
if( ink == 10 ) i=i-1; /* 「↓」で1行前進 */
else
    if( ink==' ' ) i=i+1; /* スペースで半ページ前進 */
    else i=page; /* 他のキーで1ページ */
}
fclose(fp);
outline();
}

/* ===== ファイルの内容を16進メモリーダンプ ===== */

void dpbin( char *fna , int pfg )
{
    char i,j,k,ink,*p,*str,cbk,kjfg;
    unsigned char c[18];
    FILE *fp;

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

    if( pfg==3 ) stream=stdprn;
    window(1,1,80,25);
    clrscr();
    for( i=0 ; i<80 ; i++ ) buff[i]=' ';
    dispstr(2,5+8,buff);
    gotoxy(36-strlen(fna)/2,1);
    sprintf(buff,"《 %s 》Yn",fna);
    dispstr(pfg,5+8,buff);
    if( pfg==3 ) fprintf(stream,"YrYx0aYrYx0a");

    window(1,2,80,25);
    textcolor(0x20*7);
    ink=0;
    c[17]=0;
    kjfg=0;
    cbk=(getc(fp) & 0x00ff);
    while( ink !=0x1b )
    {
        for( i=0 ; i<16 || pfg==3 ; i++ )
        {
            c[0]=cbk;
            sprintf(buff,"%061X: ",ftell(fp)-1);
            dispstr(pfg,7,buff);
            for( j=1 ; j<17 ; j++ )
            {
                c[j]=(getc(fp) & 0x00ff);
                if( ferror(fp) || feof(fp) )
                {
                    for( k=j ; k<17 ; k++ ) c[k]=0;
                    break;
                }
            }
            for( j=0 ; j<16 ; j++ )
            {
                sprintf(buff,"%02X",c[j]);
                dispstr(pfg,7,buff);
            }
            if( kjfg == 1 ) c[0]=' ';
            for( j=kjfg ; j<16 ; j++ )
            {
                if( iskanji(c[j]) && iskanji2(c[j+1]) )
                    j++;
                else
                    if( c[j]!='.' && isgrkana(c[j])==0 )
                        c[j]='.';
            }
        }

        cbk=c[16];
        kjfg=0;
        if( j==17 ) kjfg=1; /* 漢字1バイト目で終わり */
        else c[16]=0;
        sprintf(buff,"%sYn",c);
        dispstr(pfg,7,buff);
        if( pfg==3 )
            fprintf(stream,"YrYx0a");
        if( pfg==3 && (i % 16)==15 )

```

```

            fprintf(stream,"YrYx0a");
            if( ferror(fp) || feof(fp) ) break;
            if( pfg==3 && kbhit() )
                if( getch() == 0x1b ) 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・P ・他のキーで次ページへ ";
        dispstr(2,col[0],str);
        textcolor(0x20*7);
        pfg=2;
        ink=toupper(getch());
        if( ink=='P' || ink=='t' ) /* 「P」でプリンター出力 */
        {
            pfg=3;
            fclose(fp);
            goto AGAIN;
        }
        if( ferror(fp) || feof(fp) ) break;
        delline();
        gotoxy(1,wherey());
    }
    fclose(fp);
    outline();
}

/* ----- ノーファイルメッセージ ----- */

void nofna( char *fna )
{
    beep(1000);
    clrscr();
    gotoxy(15,12);
    dispstr(2,col[0]," ファイルが見つかりません !!! ");
    gotoxy(15,14);
    sprintf(buff,"%s ",fna);
    dispstr(2,col[0],buff);
    getch();
}

/* ===== ファイルの削除 (DELETE) ===== */

int delfile( int i , char *cur )
{
    char c;

    makefna(i,cur);
    gotoxy(1,23);
    sprintf(buff,"YnYn %s を削除してもいいですか (Y/N) ",cur);
    dispstr(2,col[0],buff);
    c=0;
    while( c !=0x1b && c !='Y' && c !='N' && c !='y' && c !='n' )
        c=toupper(getch());
    if( c =='Y' || c=='y' )
    {
        if( remove(cur) != -1 )
        {
            int j;
            struct dfree df;

            for( j=dnum[i]+fwp; j<tcnt-1; j++ )
                structmov(j,j+1);
            tcnt--;
            getdfree(path[0]+'@',&df);
            fresize(df.df_sclus*df.df_bsec/1024)*df.df_aval;
            return 1;
        }
        beep(1000);
        gotoxy(1,23);
        sprintf(buff,"%s は削除できません !!! ",cur);
        dispstr(2,col[0],buff);
        getch();
    }
    return 0;
}

/* ===== */

```

XDIR1.C

```

/* =====
* MS-DOSコマンド D I R機能拡張 (コマンドライン版)
* Turbo_Cで作成
* File "xdir.c"
* 1988.12/15 By H.Takeishi
* ===== */

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

#define DIMAX 200 /* ディレクトリ内ファイル属性格納の最大数 */

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

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

    /* 初期値設定 */
    dep = 15; /* サブディレクトリの深さ */
    flg = 4; /* 表示フラッグ (罫線あり) */
    /* 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 罫線を表示する */
    /* 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; /* ディレクトリの罫線深さ表示幅 */
    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 (/Dn/Wn/F/Hn/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) == 'W') 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+=2;
                while (*a && j<7)
                    col[j++]=colset(*a++);
            }
        }
    }

    if (kln & 1) kln--;
    if (kln<2) kln = 2;
    if (kln>40) kln = 40;
    if (dep>15) dep = 15;
    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++) kmd[cnt]=1;

    if (strchr(path, ':') == NULL)
    {
        drv=getdisk();
        cur[0]='A'+drv; cur[1]=': '; cur[2]=0;
        strcat(cur, path);
        strcpy(path, cur);
    }
    if (path[strlen(path)-1] == '.')
    {
        drv=path[0]-'@';

```

```

        getcurdir(drv, cur);
        strcat(path, "Y");
        strcat(path, cur);
    }
    strcpy(kmd+20, "");

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

    fdisp(path, sch, atr, 0, dep, flg, kln, kmd, 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, kmd, wid, col)
unsigned char *path, *sch,
    atr, n, dep, flg, kln,
    *kmd, wid, *col;
{
    struct ffbk ffbk; /* ファイル情報格納領域 */
    unsigned char file[100], *p, kfg, fdir[DIMAX][13];
    long siz[DIMAX];
    int dat[DIMAX], tim[DIMAX],
        done, i, cnt, fct, xpo;

    cnt=0;
    fct=0;
    xpo=n*kln;
    strcpy(file, path);
    strcat(file, *sch);

    /* --- 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<DIMAX)
        {
            if (ffbkl.ff_attrib == 0x10)
            {
                siz[cnt]=ffbkl.ff_fsize;
                dat[cnt]=ffbkl.ff_fdate;
                tim[cnt]=ffbkl.ff_fctime;
                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_fctime;
                    strcpy(fdir[fct++], p);
                    cnt++;
                }
            }
        }
        done=findnext(&ffbkl);
    }

    /* --- 罫線の頭処理 ----- */
    if (*col != 0) tcolor(*col+1);
    if (cnt != 0 && flg & 4)
        printf("%sYn", kmd+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, kmd, col);
                if ((fln>wid || cnt>fct) && i<cnt-1)
                    kfg=1;
            }
            else
                kfg=3;

```



```

        keisen(kfg, n, kmd);
    }
    else
    {
        if( xpo>wid-18 )
        {
            printf("Yn");
            spc(flg, n, kln, kmd, 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, kmd);
        }
        if( *col !=0 ) tcolor(*(col+3));
        printf("%-8s.%-4s  ", p, dp+1);
        xpo+=16;
    }
    else
    {
        spc(flg, n, kln, kmd, col);
        if( i<cnt-1 ) kfg=1; else kfg=3;
        keisen(kfg, n, kmd);
        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("Yn");
    }
}
if( flg & 2 && fct>0 ) printf("Yn");
/* --- サブディレクトリの表示 および 再帰呼び出し ----- */
for( i=fct : i<cnt : i++ )
{
    char dir[100], j;

    p=fdir[i];
    spc(flg, n, kln, kmd, col);
    if( i<cnt-1 ) kfg=1; else kfg=3;
    keisen(kfg, n, kmd);
    if( strlen(p) & 1 ) strcat(p, " ");
    strcpy(kmd+20, "");
    if( flg & 4 )
        for( j=strlen(p); j<kln-4; j+=2 )
            strcat(kmd+20, "--");
    if( strlen(p)<kln-2 ) strcat(kmd+20, "\r\n");
    if( *col !=0 ) tcolor(*(col+2));
    printf("<%=>", p);
    if( n<dep )
    {
        strcpy(dir, path);
        strcat(dir, p);
        strcat(dir, "Y");
        fdisp(dir, sch, atr, n+1, dep, flg, kln, kmd, wid, col);
    }
    else
        printf("Yn");
}
}
/* ===== 罫線を含むタブ (スペース) の表示 ===== */
spc(flg, n, kln, kmd, col)
unsigned char flg, n, kln, *kmd, *col;
{
    char j, k;
    if( *col !=0 ) tcolor(*(col+1));
    for( j=0; j<n; j++ )
    {
        if( flg & 4 && *(kmd+j)==1 )
            printf(" ");
        else
            printf(" ");
        for( k=0; k<kln-2; k++ ) printf(" ");
    }
}
/* ===== 罫線の終端処理 ===== */
keisen(kfg, n, kmd)
unsigned char kfg, n, *kmd;
{
    switch( kfg )
    {
        case 1: *(kmd+n)=1;
                printf(" |");
                break;
        case 2: *(kmd+n)=1;
                printf(" |");
                break;
        case 3: *(kmd+n)=0;
                printf(" |");
    }
}
/* ===== ワイルドカードを含むファイル名の比較 ===== */
char fcmp( char *p1, char *p2 )

```

```

char d1, d2, i;

if( strcmp(p1, ".*") == 0 ) return 0;
if( *p1 != '*' )
{
    for( i=0; i<8; i++ )
    {
        d1=*p1;
        d2=*p2;
        if( d1 == '*' ) 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++ != '*' )
{
    p2++;
    for( i=0; i<3; i++ )
    {
        d1=*p1++;
        d2=*p2++;
        if( d1 == '*' ) break;
        if( d1 != '?' && d1 != d2 ) return 1;
        if( d1 == '0' ) return 0;
    }
}
return 0;
}
/* ===== ファイルサイズ日付時間の表示 ===== */
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("%8u", s);
    }
    if( l>15+8+10 )
    {
        unsigned char y;
        y=80+(d>>9);
        if( *col ) tcolor( *(col+5) );
        printf(" %2u-%02u-%02u", (y % 100), (d & 0x1ff)>>5, d & 0x1f);
    }
    if( l>15+8+10+7 )
    {
        if( *col ) tcolor( *(col+6) );
        printf(" %2u-%02u-%02u", t>>11, (t & 0x7ff)>>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<='9' ) return a-'0';
    if( a>='A' && a<='F' ) return (a-'A')+10;
    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("Yx1b[3%cm", c);
    }
    if( 7 < a )
    {
        if( 15 < a ) a = 15;
        a -= 8;
        b = (a & 1) * 4 + (a >> 1);
        c = '0' + b;
        printf("Yx1b[4%cm", c);
    }
}
/* ===== */

```