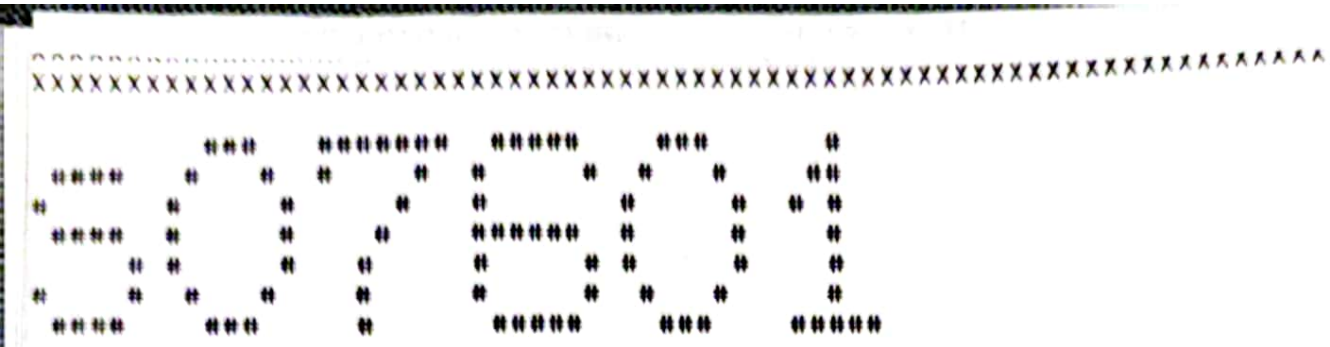

ser: Bogdanov K.E.

equest id: pr1-192   Printer: pr1

ed Dec 12 20:28:46 EST 1990

nit wind;

W I N D O W

U N I T

nterface
    uses crt,dos;
    const maximum_windows=16;
    type sttr=string[80];
          window_type1=^window_type;
          window_type3=record
            while (ds<>nil) and (dr.level>currwindow.level) and (fu.screenwx.wa

```
                        color:byte;
                        back:byte;
                    end;
      window_type2=record
                        data:char;
                        attr:byte;




                        z:byte;
                    end;
        window_type=record
                        cursorx,cursory,startx,starty,lenx,leny,level:inte
r;

                        next:window_type1;
                        screen:array[1..25,1..80] of window_type2;
                        opened:boolean;
                        manager:pointer;
                    end;
        abi=array[1..25,1..80,1..2] of char;
    var error:integer;
        abiabi absolute $b900:0;
        start,currwindow:window_type1;
        attributes:window_type3;
        page,attr:byte;
        cx,cy:integer;
        using:window_type2;


    procedure change_level(n,l:integer);
{   changes  window  level                                                  }
    procedure set_window(n:byte);                                           }
{  set active EGA page                                                      }
    procedure gxy(x,y:integer);                                             }
{     gotoxy(x,y)  in current window                                        }
    procedure wrln;                                                         }
{     writeln                                                               }
    procedure setcolor(c:integer);                                          }
{     textcolor                                                             }
    procedure moveresize(n:integer);                                        }
{     user move and resize of window #n                                     }
    procedure backk(c:integer);                                             }
{     textbackground                                                        }
    procedure wr(sss:char);                                                 }
{     write character                                                       }
    procedure wrstr(str:sttr);                                              }
{      write string                                                         }
    procedure new_window(lx,ly,stx,sty,n,b,frame,nc:integer;name:sttr);     }
{     open new window
      lx,ly     :  it's  length
      stx,sty   :  position on the screen
      n         :  window level
      b         :  color of background ...
      frame     :  frame   color
      nc        :  name color
      name      :  window name                                              }
    procedure rr(n,m:byte);                                                 }
{   set cursor size/shape   n:startline,m:endline                           }
    procedure hide_cursor;                                                  }
{  hides cursor (invisible)                                                 }
    procedure show_cursor;                                                  }
{   makes cursor visible                                                    }
    procedure open_window;                                                  }
{     open window ( display it)                                             }
    procedure go_window(n:integer);                                         }
{     enter window                                                          }
    procedure init_windows;                                                 }
{     initialize windows                                                    }
    procedure close_window;                                                 }
{     hide current window , goto window 1                                   }
    procedure delete_window;                                                }
{     close remove                                                          }
    procedure cls;                                                          }
{     clear screen                                                          }
    function  getstring(len:char;strr:sttr):sttr;




{     get string                                                            }
implementation


    function getstring;
    var s:sttr;
    r:registers;
    begin
       wrstr(strr);
       s[1]:=len;
       error:=0;

             r.ax:=$0A00;
             r.dx:=ofs(s[1]);
             r.ds:=seg(s);
             msdos(r);
      getstring:=copy(s,1,length(s)-1);
    end;

    function biggerwindow(wx,wy:integer):boolean;
      var qq:integer;d:window_type1;
      begin
         d:=start;
         qq:=currwindow^.level+1;
         while (d<>nil) and (d^.level>currwindow^.level) and ((d^.screen[wy,wx]
```

```
data<#2) or not d^.opened) do d:=d^.next;
          biggerwindow:=(d<>nil) and (d^.level>currwindow^.level);
       end;

    procedure ggxy(x,y:byte);
    var r:registers;
    begin
       r.ah:=2;r.bh:=page;r.dl:=x-1;r.dh:=y-1;
       intr($10,r);
    end;

    procedure setcolor;
    begin
{       attributes.color:=c;
       textcolor(c);}
       attr:=(attr and $f0 ) or c;
    end;

    procedure backk;
    begin( red)
{       attributes.back:=c;
       textbackground(c);}
       att:=(attr and $0f ) or ((c and $07)*16);
    end;

    procedure setxy;
     var x1,y1:integer;
     begin
         x1:=currwindow^.startx+currwindow^.cursorx;y1:=currwindow^.starty+curr
indow^.cursory;
         ggxy(x1,y1);
         end;
```

```
   procedure writee(s:char);
   var r:registers;
   begin
      r.ah:=09;
      r.bh:=page;r.al:=ord(s);r.cx:=1;r.bl:=attr;
      intr($10,r);
{      write(s);  }
   end;
```

dEL

```
   procedure wr;
      var i,x1,y1:integer;
      begin
         x1:=currwindow^.startx+currwindow^.cursorx;y1:=currwindow^.starty+curr
indow^.cursory;
         if currwindow^.cursorx<currwindow^.lenx-1 then
          begin
             currwindow^.screen[y1,x1].data:=sss;
             currwindow^.screen[y1,x1].attr:=attr;
             if currwindow^.opened and (not biggerwindow(x1,y1)) then begin
               ab[y1,x1,1]:=sss;ab[y1,x1,2]:=chr(attr); end;                error:
0;
          end else error:=-1;
          currwindow^.cursorx:=currwindow^.cursorx+1;
          if currwindow^.cursorx>currwindow^.lenx-1 then
          begin
             currwindow^.cursorx:=currwindow^.cursorx-1;
             setxy;
          end;
     end;
    procedure wrstr;
      var i:integer;
      begin
         for i:=1 to length (str) do
         wr(str[i]);
      end;

    procedure wrln;
    begin
       currwindow^.cursory:=currwindow^.cursory+1;
       currwindow^.cursorx:=1;
       if currwindow^.cursory>currwindow^.leny-3 then currwindow^.cursory:=curr
indow^.leny-2;
       setxy;
    end;

    procedure seen_value(use_current:boolean;var i,j:integer);
    var d:window_type1;
    begin
       d:=start;error:=0;
       if use_current then
         begin
          while (d[redacted]=currwindow[redacted]) or (not d^.opened) or (d^.screen[
,j].data<#2)
             do
             begin
                if d^.screen[i,j].data=#1 then error:=1;
                d:=d^.next;
             end;
         end
       else  begin
          while (d^.screen[i,j].data=#0) or (d^.screen[i,j].data=#1) do
            begin
              if (d^.screen[i,j].data=#1) then error:=1;
              d:=d^.next;
            end;
            end;

     using.data:=d^.screen[i,j].data;
     using.attr:=d^.screen[i,j].attr;
    end;

    procedure cls;
```

```
var i,j:integer;a:char;
begin
    currwindow^.cursorx:=1;




    currwindow^.cursory:=1;
       for i:=currwindow^.starty to currwindow^.starty+currwindow^.leny-1 d
       begin
       for j:=currwindow^.startx to currwindow^.startx+currwindow^.lenx-1 d
          begin
            currwindow^.screen[i,j].attr:=attr;
            currwindow^.screen[i,j].data:=a;
            ggxy(i,j);
            write(a);
          end;
       end;

       ggxy(currwindow^.startx,currwindow^.starty);)



end;

procedure gxy;
 begin
    currwindow^.cursorx:=x;currwindow^.cursory:=y;
    setxy;
 end;

procedure go_window;
 var d:window_type1;x1,y1:integer;
 begin
     d:=start;error:=0;
     while (d<>nil) and (d^.level<>n) do d:=d^.next;
     if (d=nil)  then error:=1
     else
       begin
          currwindow:=d;
          setxy;
       end;
 end;


procedure init_windows;
var i,j,qq:integer;
begin
   backk(black);
   setcolor(white);
   new(start);
   page:=0;
   with start^ do
    begin
       cursorx:=1;
       cursory:=1;
       startx:=1;
       starty:=1;
       lenx:=79;
       leny:=23;
       level:=1;
       next:=nil;
       opened:=false;
    end;
    clrscr;
    currwindow:=start;
    error:=0;
    for i:=1 to 25 do for j:=1 to 80 do
     begin
       currwindow^.screen[i,j].data:=#176;
       currwindow^.screen[i,j].attr:=attr;
       ab[i,j,2]:=chr(attr);ab[i,j,1]:=#176;




      end;
      currwindow^.opened:=true;
    end;



procedure new_window;
 var d,d1:window_type1;i,j:integer;
 begin
    error:=-1;
    if n>maximum_windows then exit;
    d:=start;d1:=d;
   while (d<>nil) and (d^.level>n) do begin
       d1:=d;d:=d^.next;
       end;
    if not((d=nil) or (d^.level<>n) ) then exit;
     setcolor(frame);
     backk(b);
      if d=d1 then
       begin
          new(d);
          start:=d;
          d^.next:=d1;
       end else
       begin
          new(d);
          d^.next:=d1^.next;
          d1^.next:=d;
       end;
    with d^ do
     begin
```

```
            level:=n;
            lenx:=lx;
            leny:=ly;
            cursorx:=1;
            cursory:=1;
            startx:=stx;
            starty:=sty;
            opened:=false;
          end;

          for i:=1 to 24 do
          for j:=1 to 80 do
            begin
              if (j>=d^.startx) and (j<d^.startx+lx) and (i>=d^.starty) and (i
d^.starty+ly)
               then
                begin
                    d^.screen[i,j].data:=' 'id^.screen[i,j].attr:=attr;end
               else d^.screen[i,j].data:=#0;
               d^.screen[i,j].attr:=attr;
            end;
          for i:=d^.startx to d^.startx+lx-1 do
            begin
              if i>(d^.startx+1) then begin
              d^.screen[d^.starty+1y,i].data:=#1;
              d^.screen[d^.starty+ly,i].attr:=0;
                                    end;
              d^.screen[d^.starty,i].data:=chr(205);
              d^.screen[d^.starty+d^.leny-1,i].data:=chr(205);
              d^.screen[d^.starty,i].attr:=(attr and $f0)+frame;
            end;
          for i:=d^.starty+1 to d^.starty+ly do
           begin
              d^.screen[i,d^.startx+lx].data:=#1;
              d^.screen[i,d^.startx+lx+1].data:=#1;




              d^.screen[i,d^.startx+lx].attr:=112;
              if i<d^.starty+ly then
                begin
                    d^.screen[i,d^.startx].data:=#186;
                    d^.screen[i,d^.startx+d^.lenx-1].data:=#186;
                    d^.screen[i,d^.startx].attr:=(d^.screen[i,d^.startx].attr an
$f0)+frame;
                end;

              d^.screen[d^.starty,d^.startx].data:=chr(201);
              d^.screen[d^.starty+d^.leny-1,d^.startx].data:=chr(200);
              d^.screen[d^.starty+d^.leny-1,d^.startx+d^.lenx-1].data:=chr(188);
              d^.screen[d^.starty,d^.startx+d^.lenx-1].data:=chr(187);
           end;
              currwindow:=d;
              j:=(d^.lenx-length(name)) div 2-1;
              if (j>0) and (length(name)>0) then
                begin
                  attributes.color:=green;
                  with currwindow^.screen[d^.starty,currwindow^.startx+j] do
                  begin
                    attr:=(attr and $f0)+nc;
                    data:=' ';
                  end;
                  setcolor(nc);
                  for i:=1 to length (name) do begin
                  with currwindow^.screen[d^.starty,currwindow^.startx+j+i] do
                  begin
                    attr:=(attr and $f0)+nc;
                    data:=name[i];
                  end;
                     currwindow^.screen[d^.starty,d^.startx+j+i+1].attr:=attr;
                     currwindow^.screen[d^.starty,d^.startx+j+i+1].data:=' ';
                end;

    end;

    procedure delete_info;
     var i,j,qq:integer;dd:window_type1;
     begin
          for i:=currwindow^.starty to currwindow^.starty+currwindow^.leny+1 do
          begin
          for j:=currwindow^.startx to currwindow^.startx+currwindow^.lenx+2 do
             begin
               seen_value(true,i,j);
               attr:=using.attr;
               if error=1 then
                 backk(0);
               ab[i,j,1]:=using.data;ab[i,j,2]:=chr(attr);
             end;
        end;
     end;

    procedure close_window;
     var i,j:integer;
     begin
        if currwindow^.level>1 then
         begin
            error:=0;
            delete_info;
            currwindow^.opened:=false;
        end else error:=1;
     end;

    procedure delete_window;
     var d,d1:window_type1;
```

```
      begin
         close_window;
         if (error=0) then begin
         d:=start;d1:=nil;
         while d^.level<>currwindow^.level do
           begin
              d1:=d;
              d:=d^.next;
           end;
         if d1=nil then start:=d^.next else
         d1^.next:=d^.next;
         dispose(currwindow);
         go_window(1);
         end;
      end;


      procedure open_window;
       var i,j,l:integer;tt:window_type1;
       begin
          hide_cursor;
          currwindow^.opened:=true;
            for i:=currwindow^.starty to currwindow^.starty+currwindow^.leny+1 do
            begin
            for j:=currwindow^.startx to currwindow^.startx+currwindow^.lenx+2 do
             begin
                  if (currwindow^.screen[i,j].data<>#0) then
                   begin
                     ggxy(j,i);
                     seen_value(false,i,j);
                     if error=0 then begin
                     attr:=using.attr;
                     end else
                     begin
                     setcolor(black);
                     backk(black);
                     end;
                     ab[i,j,1]:=using.data;ab[i,j,2]:=chr(attr);
                   end;
              end;
          end;
        show_cursor;
        setxy;
      end;

      procedure moveresize(n:integer);
      var hx,hy,lx,ly:integer;

     procedure seeen(i,j:integer);
     var d:window_type1;
     begin
        d:=start;error:=0;
        while (d^.screen[i,j].data=#0) do d:=d^.next;
     with using do begin
      data:=d^.screen[i,j].data;attr:=d^.screen[i,j].attr;
      if data=#1 then data:=' ';
     end;

     end;
      procedure horizontal_line(l:integer);
      var x:integer;
      begin
         textcolor(brown);
         ggxy(hx+1,l);
         for x:=hx+1 to lx-1 do begin




           seeen(l,x);
           textbackground(using.attr.back);
           writee(chr(205));
                                end;
         end;

      procedure delete_horizline(l:integer);
      var x:integer;d:window_type1;
      begin
         ggxy(hx+1,l);
         for x:=hx+1 to lx-1 do begin
           seeen(l,x);
           textbackground(using.attr.back);
           textcolor(using.attr.color);
           writee(using.data);
             end;
         end;

    begin
       hx:=1;lx:=60;
       horizontal_line(1);
       for ly:=1 to 23 do
        begin
           delete_horizline(ly);horizontal_line(ly+1);
           delay(600);
           end;
   end;

   procedure change_level;
   var i:integer;
   begin
      i:=currwindow^.level;
      go_window(i);
      if error<>0 then exit;
      close_window;
      currwindow^.level:=1;
```

```
    open_window;
    go_window(i);
  end;

  procedure set_window;
  var r:registers;
  begin
     r.ah:=5;
     r.al:=n;intr($10,r);

  end;

  procedure rr(n,m:byte);
  var r:registers;
  begin
     r.ah:=1;
     r.ch:=n;r.cl:=m;
     intr($10,r);
  end;

  procedure hide_cursor;
  begin
     rr($20,0);
     [redacted]
  end;

  procedure show_cursor;
  begin
     rr($5,$8);
  end;




  procedure moveresize;
  begin end;
end.
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX