```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

          ###   #######  #####    ###       #
  ####    #   #  #     #  #     #  #  #     ##
 #       #     #  #     #  #        #     # # #
  ####  #     #  #     #  ######   #     #   #
      # #      #  #     #  #     #  #     #   #
 #    #  #    #  #     #  #     #  #  #  #   #
  ####    ###    #     #  #####    ###    #####


User: Bogdanov K.E.

Request id: pri-193    Printer: pri

Wed Dec 12 20:37:16 EST 1990
```

```
uses Wind,Crt;
{   text editor
    Text format
          name
    }

        const  base_length=60;
               empty_line='                                                  ';
        type s__s=string[base_length];

         continue1=^continue;
         continue=record
                    cont:continue1;
                    line:s__s;
                  end;
         tx1=^tx;




         ln1=^ln;
         ln=record
              red,green:boolean;
              blockbeg,blockend:integer;
              pred,next:ln1;
              cont:continue1;
            end;

         tx=record
               fil:sttr;
               len,maxx,maxy,showlinebegin,cursorx,cursory,startx,starty:integer
;
               insert,modified:boolean;
               bb,be,currline,screentop:ln1;
            end;

        var tex:tx;
            a:char;

  procedure save_text(fil:sttr;tet:tx1);forward;

  procedure init_text(text:tx1;name:sttr);
  var d:ln1;
  begin
               text^.startx:=1;
               new(d);
               text^.maxx:=32767;
               text^.maxy:=32767;
               text^.screentop:=d;
               text^.currline:=d;
               text^.len:=1;
               d^.red:=false;
               d^.green:=false;
               d^.blockend:=0;
               d^.blockbeg:=0;
               d^.next:=nil;
               d^.pred:=nil;
               d^.cont:=nil;
               if name='' then text^.fil:='NO_NAME.NAM'else
                text^.fil:=name;
               text^.starty:=1;text^.startx:=1;
               text^.showlinebegin:=length(text^.fil)+21;
               text^.modified:=false;
               text^.insert:=true;
               text^.cursorx:=1;
               text^.cursory:=1;
   end;

  function te(text:tx;open:boolean):char;
     var number:integer;

     procedure showscreen;forward;

     function getcurrent:continue1;
     var d:continue1;g:integer;
     begin
           number:=text.cursorx+text.startx-1;
           d:=text.currline^.cont;
           if d=nil then
             begin
                 getcurrent:=nil;
                 exit;
             end;
```

```
      while (d^.cont<>nil) and (number>base_length) do
       begin
         d:=d^.cont;number:=number-base_length;




       end;
    getcurrent:=d;
 end;

procedure cursor_right;
begin
   if text.cursorx=text.maxx then exit;
   text.cursorx:=text.cursorx+1;
   if text.cursorx>currwindow^.lenx-2 then
     begin
        text.cursorx:=text.cursorx-1;
        text.startx:=text.startx+1;
        showscreen;
     end;
end;

procedure cursor_left;
begin
   text.cursorx:=text.cursorx-1;
   if text.cursorx=0 then
     begin
        if text.startx>1 then
        begin
           text.startx:=text.startx-1;
           showscreen;
        end;
        text.cursorx:=1;
     end;

end;

procedure displline(l:ln1);
 var e,p:integer;nc:boolean;ll:continue1;
 begin
   backk(blue);
   ll:=l^.cont;
   setcolor(yellow);
   nc:=false;e:=0;
   if l^.green then begin setcolor(white);backk(cyan); nc:=true end else
   if l^.red then begin backk(red);setcolor(white);nc:=true;end;
       if ll<>nil then
       while (ll^.cont<>nil) and ((text.startx-e)> base_length) do
        begin
             e:=e+base_length;
             ll:=ll^.cont;
        end;

    for p:=1 to currwindow^.lenx do
     begin
        if (not nc) then
          begin
             if (p=l^.blockbeg ) then backk(white) else
             if (p=l^.blockend ) then backk(blue);
          end;
     if ll=nil then wr(' ') else
     if ((text.startx+p-1-e)> base_length) then
        begin
         if (ll^.cont<>nil) then
          begin
             e:=e+base_length;
             ll:=ll^.cont;
             wr(ll^.line[1]);
          end
           else wr(' ');
        end else
             wr(ll^.line [text.startx+p-1-e]);




    end;
  end; {   displline   }

  procedure disp_pos;
  var g3,g4:string[5];
  begin
      setcolor(cyan);
      str(text.cursorx+text.startx-1:3,g3);
      str(text.cursory+text.starty-1:3,g4);
      gxy(text.showlinebegin+7,1);
      wrstr(g3);wr(' ');wrstr(g4);
  end;


  procedure show_firstline;
  var g,g1:string[10];
  begin
      gxy(1,1);
      backk(blue);
      setcolor(white);
      if text.modified then g:='*' else g:=' ';
      if text.insert   then g1:='Insert' else g1:='      ';
      wrstr(concat('   ',g,'   ',g1,'File :',text.fil));
      setcolor(cyan);gxy(text.showlinebegin,1);
      wrstr('Line :');
      wrln;
  end;

  procedure show_currline;
  begin
          gxy(text.startx,text.cursory+1);
          displline(text.currline);
          if text.modified then
            disp_pos
            else begin
               text.modified:=true;
```

```
                show_firstline;
                 end;
            cursor_right;
    end;

    procedure delete(n:integer;r:char;d:continuel);
    var ss,qq:char;i:integer;
    begin
       qq:=r;
       for i:=base_length downto n do
        begin
           ss:=d^.line[i];d^.line[i]:=qq;qq:=ss;
        end;
    end;


    procedure delete_pressed;
    var d,k:continuel;
    begin
       d:=getcurrent;
       if number>base_length then exit;
       if d=nil then exit;
       k:=d;
       if d^.cont <>nil then
         begin
            delete(number,d^.cont^.line[1],d);
            d:=d^.cont;
            number:=1;
         end;
            while d^.cont <> nil do begin




              delete(1,d^.cont^.line[1],d);k:=d;
              d:=d^.cont;
                                   end;
          delete(number,' ',d);

       k:=text.currline^.cont;
       if k<>d then
       while k^.cont<>d do k:=k^.cont;
       if (d^.line=empty_line) (and (d<>text.currline^.cont)) then
        begin
           k^.cont:=nil;dispose(d);
        end;
       show_currline;cursor_left;
    end;

    procedure bks_pressed;
    var t:continuel;
    begin
       if text.cursorx>1 then begin
       cursor_left;
       delete_pressed;
       end;{
       else
        if text.currline^.pred<>nil then
         begin
            t:=text.currline^.pred;
            if text.currline^.cont=nil then
              begin
                 text.currline^.pred:=text.currline^.next;
                 text.currline:=text.currline^.pred;
                 if text.screentop=text.currline then text.screentop:=text.cur
-line^.pred;
                 text.currline:=text.currrline^.pred;
                 showscreen;
              end else
              begin
                 if t.cont=nil


          )
    end;


    procedure showscreen1(p:integer);
     var j,i:integer;h:ln1;g,g1:string[80];
     begin
          backk(blue);
          show_firstline;h:=text.screentop;
          i:=2;setcolor(yellow);backk(blue);
          gxy(1,2);
          while (i<=(currwindow^.leny-2)) do
           begin
             if (h=nil) then for j:=1 to currwindow^.lenx do wr(' ')
              else
                 begin
                      if (i-1)>=p then displline(h);
                      h:=h^.next;
                 end;
              if i<currwindow^.leny then wrln;
              i:=i+1;
           end;
     end;

    procedure showscreen;
    begin




        showscreen1(1);
    end;

    procedure curup;
    begin
       if (text.cursory=1) then
        begin
           if (text.screentop^.pred<>nil) then
```

```
          begin
             text.screentop:=text.screentop^.pred;
             text.starty:=text.starty-1;
             text.currline:=text.currline^.pred;
             showscreen;
          end;
      end else
        begin
           text.cursory:=text.cursory-1;
           text.currline:=text.currline^.pred;
        end;

end;


procedure curdown;
begin
    if (text.currline^.next<>nil) then
     begin
     if (text.cursory=currwindow^.leny-3) then
       begin
          text.screentop:=text.screentop^.next;
          text.currline:=text.currline^.next;
          text.starty:=text.starty+1;
          showscreen;
       end
     else
      begin
         text.cursory:=text.cursory+1;
         text.currline:=text.currline^.next;
      end;
end;
end;

procedure  enter;
var g:ln1;d,t:continue1;i:integer;
begin
       if text.len=text.maxy then exit;
       text.len:=text.len+1;
       new(g);
       g^.red:=false;
       g^.green:=false;
       g^.blockbeg:=text.currline^.blockbeg;
       g^.blockend:=text.currline^.blockend;
       d:=getcurrent;
       if number> base_length then
        begin
           g^.next:=text.currline^.next;
           g^.pred:=text.currline;
           if text.currline^.next<>nil then
           text.currline^.next^.pred:=g;
           text.currline^.next:=g;
           showscreen;
           exit;
        end;
       g^.next:=text.currline;
       g^.pred:=text.currline^.pred;
       if g^.pred<> nil then g^.pred^.next:=g else



       text.screentop:=g;
       text.cursorx:=1;
       text.startx:=1;
       text.currline^.pred:=g;
       text.currline:=g;
       if text.cursorx=1 then g^.cont:=nil else
        begin
           new(t);
           t^:=d^;
           g^.cont:=t;
           d^.cont:=nil;
           i:=number;
           while number<=base_length do
            begin
               d^.line[number]:=' ';number:=number+1;
            end;
           text.currline:=g;
           while i>0 do begin
            i:=i-1;delete_pressed;
            end;
        showscreen;
       end;
end;

function insert_char(dd:char;element:integer;dt:continue1):char;
 var a,p:char;
 begin
    a:=dd;
    while element<=base_length do
     begin
        p:=dt^.line[element];
        dt^.line[element]:=a;
        a:=p;
        element:=element+1;
     end;
    insert_char:=p;
  end;

 procedure showtext    )


function string_length(li:ln1):integer;forward;

procedure insertstr;
var y:integer;d,t:continue1;p:char;
begin
  if text.insert and (string_length(text.currline)=text.maxx) then exit;
  if text.currline^.cont=nil then
   begin
      new(d);
      number:=text.cursorx+text.startx-1;
      text.currline^.cont:=d;
```

```
        d^.line:=empty_line;
        if number<=base_length then
         begin
             d^.line[number]:=a;
             d^.cont:=nil;
             exit;
         end;
        while number>base_length do
         begin
            new(t);d^.cont:=t;
            t^.line:=empty_line;d:=t;
            number:=number-base_length;
         end;
         d^.cont:=nil;




         d^.line[number]:=a;
         exit;
      end;
    d:=getcurrent;
    if number>base_length then
     begin
         while number>base_length do
           begin
              new(t);
              t^.line:=empty_line;
              d^.cont:=t;d:=t;number:=number-base_length;
           end;
           d^.cont:=nil;
           d^.line[number]:=a;
    end else
      begin
         if text.insert then
          begin
             while d^.cont<>nil do
               begin
                    a:=insert_char(a,number,d);
                    number:=1;
                    d:=d^.cont;
                end;
                a:=insert_char(a,number,d);
                if a<>' ' then
                 begin
                    new(t);
                    d^.cont:=t;
                    t^.cont:=nil;
                    t^.line:=empty_line;
                    t^.line[1]:=a;
                  end;
           end
           else
              begin
                 d^.line[number]:=a;
              end;


 end;
 end;

 text.currline^.cont^.cont^.cont^.cont^.cont^     )

function string_length(li:ln1):integer;
var d:continue1;l,i:integer;
begin
    if li^.cont=nil then
     begin
        string_length:=1;
        exit;
     end;
    d:=text.currline^.cont;
    l:=0;
    while d^.cont<>nil do
     begin l:=l+base_length;d:=d^.cont;end;
    i:=base_length;
    while (i>0) and (d^.line[i]=' ') do
     i:=i-1;

    string_length:=l+i+1;
end;









procedure set_to_end;
var i,j:integer;
begin
   j:=string_length(text.currline);

   if j<=(text.startx-1) then
    begin
       text.startx:=j;
       text.cursorx:=1;
       showscreen;
    end
   else
     begin j:=j-text.startx+1;
      if j>(currwindow^.lenx-2) then
       begin
          text.startx:=text.startx+j-(currwindow^.lenx-2);
          text.cursorx:=currwindow^.lenx-2;
          showscreen;
       end
       else text.cursorx:=j;
   end;

end;
```

```
procedure page_up;
var i:integer;
begin
    i:=currwindow^.leny-4;
    while (text.screentop^.pred<>nil) and (i>0) and ((text.cursorx+text.s
tartx)<text.len)  do
     begin
        i:=i-1;text.screentop:=text.screentop^.pred;
        text.curline:=text.curline^.pred;
        text.starty:=text.starty-1;
     end;
       showscreen;
end;

procedure page_down;
var i:integer;
begin
    i:=currwindow^.leny-4;
    while (text.screentop^.next<>nil) and (i>0) and ((text.cursory+text.s
tarty)<text.len) do
     begin
        i:=i-1;text.screentop:=text.screentop^.next;
        text.curline:=text.curline^.next;
        text.starty:=text.starty+1;
     end;
     if text.screentop^.next=nil then text.curline:=text.screentop;
     showscreen;
end;

procedure delline;
var t:ln1;
begin
    if (t^.pred=nil) and (t^.next=nil) then exit;
    t:=text.curline;
    if t^.pred<> nil then t^.pred^.next:=t^.next;
    if t^.next<>nil then t^.next^.pred:=t^.pred;
    if t^.next= nil then
     begin
         text.curline:=t^.pred;
         curup;
     end else
          text.curline:=t^.next;




    text.len:=text.len-1;
    dispose(t);
    showscreen;
end;

procedure make_block(bb,be:integer);
var t:ln1;
begin
   t:=text.curline;

   text.curline^.blockbeg:=1;
   text.curline^.blockend:=string_length(text.curline);
   show_currline;
end;



procedure ctrlk;
begin
   gxy(1,1);
   wrstr('^K          ');
   show_cursor;
   gxy(3,1);
   a:=readkey;
   delay(50);
   hide_cursor;
   show_firstline;
{   if a='b' then text.stbl:=text.cursory+text.starty;  }
   make_block(1,1);
end;

begin
   if open then showscreen;


   repeat
   disp_pos;
   gxy(text.cursorx,text.cursory+1);
   show_cursor;
   a:=readkey;
   hide_cursor;
   if a=#0 then
      begin
           a:=readkey;
           case a of
             #75:{  left  }
                   cursor_left;
             #77:{  right }
                   cursor_right;
             #71:   begin
                      text.cursorx:=1;
                      if text.startx<>1 then
                        begin
                          text.startx:=1;
                          showscreen;
                        end;
                      end;
             #82:  begin
                    text.insert:=not text.insert;
                    show_firstline;
                    end;
             #79:  set_to_end;
             #83:  {  del   }
                   delete_pressed;
             #72:  curup;
             #80:  curdown;
```

```
                    #60:  save_text(text.fil,@text);
                    #73:  page_up;
                    #81:  page_down;
                       else if ord(a)>32  then begin
                            {  exit  }
                             te:=a;
                             exit;                end;

                     end;
              end
              else
                if a=#25 then delline else
                if a=#11 then ctrlk else
               if a=#8 then
                {  bks }
                bks_pressed
                else
               if a=#13 then
                    enter
              else
              begin
               insertstr;
               show_currline;
              end;

          until false;
       end;

  procedure save_text;
  var line:ln1;a,od,oa:char;d:continue1;f:file of char;j,i:integer;
  begin
     j:=currwindow^.level;od:=#13;oa:=#10;
     line:=tet^.screentop;
     while line^.pred<>nil do line:=line^.pred;
     assign(f,fil);
     {$i-} rewrite(f); {$i+}
     error:=ioresult;
     if error<>0 then exit;
         new_window(17,3,30,10,14,lightblue,2,5,'STATUS');
         wrstr(' Savivg... ');
         open_window;
     while line<>nil do
      begin
         d:=line^.cont;
         while d<> nil do
          begin
             for i:=1 to length(d^.line) do write(f,d^.line[i]);
             d:=d^.cont;
          end;
         if line^.next<>nil then begin write(f,od);write(f,oa);end;
         line:=line^.next;
       end;
     close(f);
     delete_window;
     go_window(j);
end;



  function hmenu(x,y,w,l:integer;s:sttr):integer;
  var p,i,j,d,ll:integer;
  procedure ds(k,g,v:integer);
  var i:integer;
  begin
     gxy(j*ll+1,1);backk(k);setcolor(g);




     for i:=(j*ll+1) to (j+1)*ll do
      if s[i]=' '  then exit else
      if ord(s[i])<96 then
       begin
          setcolor(v);
          wr(s[i]);setcolor(g);
       end else wr(s[i]);
  end;

  begin
     ll:=length(s) div l+1;p:=currwindow^.level;
     new_window(length(s)+3,3,x,y,w,2,3,4,'Menu :');
     open_window;
     hide_cursor;
     for j:=0 to l-1 do ds(white,green,red);j:=0;
     repeat
         ds(yellow,blue,yellow);
         a:=readkey;
         if a=#0 then
           begin
              a:=readkey;
              ds(white,green,red);
              if a=#75 then
                if j=0 then j:=l-1 else j:=j-1
              else if a=#77 then
                if j=(l-1) then j:=0 else j:=j+1;
              a:=' ';
           end else
           begin
              d:=1;
              while (d<=length(s)) and (s[d]<>upcase(a)) do d:=d+1;
              if d<=length(s) then begin
              a:=#13;j:=d div ll;end;
           end;
     until a=#13;
     delete_window;go_window(p);show_cursor;
     hmenu:=j+1;
  end;
```

```
procedure error_message(mes:sttr);forward;
procedure load_text(fil:sttr;tet:txt;max:integer);
    label 1;
    var
        f:file of char;
        ll,line:lnl;
        dd,d:continuel;
        text:tx;
        gg:integer;
        a:char;
        v:sttr;

    procedure nnew;
    begin
       new(dd);
       dd^.line:=empty_line;
       dd^.cont:=nil;
       dd^.line[1]:=a;
       gg:=2;
    end;

    begin
       error:=0;
       assign(f,fil);
        {$i-} reset(f); {$i+}
       error:=ioresult;if error<>0 then
          begin error_message(concat('Can''t load file ',fil));exit;end;



       new_window(25,3,30,10,14,lightblue,9,5,'STATUS');
       wrstr(' Loading ...');
       open_window;
       init_text(tet,fil);
       setcolor(Green);
       tet^.maxy:=max;
       d:=tet^.currline^.cont;line:=tet^.currline;


       while not eof(f) do
         begin
            read(f,a);
            if a=#13 then
              begin
                 if tet^.len=max then
                  begin
                     new_window(32,3,37,8,15,1,5,7,'Message:');
                     wrstr(' File contains too many lines');
                     open_window;
                     a:=readkey;delete_window;go_window(14);
                     goto 1;
                  end;
                 new(ll);
                 line^.next:=ll;
                 tet^.len:=tet^.len+1;
                 if tet^.len mod 10 = 0 then begin
                 str(tet^.len:5,v);gxy(17,1);wrstr(v);end;
                 ll^.pred:=line;
                 ll^.cont:=nil;gg:=1;
                 ll^.red:=false;
                 ll^.green:=false;
                 ll^.blockbeg:=0;
                 ll^.blockend:=0;
                 read(f,a);
                 line:=ll;
              end
             else begin
                    if (line^.cont=nil) then
                      begin
                         nnew;
                         d:=dd;
                         line^.cont:=d;
                      end else
                      if gg>base_length then
                      begin
                         nnew;
                         d^.cont:=dd;
                         d:=dd;
                      end
                      else
                        begin
                           if (ord(a)>=32) and (ord(a)<127) then begin
                           d^.line[gg]:=a;
                           gg:=gg+1;end;
                        end;
                 end;
          end;
:
           line^.next:=nil;
           close(f);
           delete_window;
       end;

  procedure error_message(mes:sttr);
  var a:char;
      i:integer;




  begin
      setcolor(blue);backk(yellow);

      i:=currwindow^.level;
      new_window(length(mes)+6,3,(76-length(mes)) div 2,10,maximum_windows-1,bl
ue,3,4,'ERROR :');
      wrstr(concat(mes,' '));
      setcolor(lightred);
      backk(lightgreen);wrstr('ESC');
```

```
      open_window;
      repeat a:=readkey until a=#27;
      delete_window;
      go_window(i);
    end;


egin {  main  }
   init_windows;
   set_window(1);
   new_window(23,10,29,9,2,4,3,lightgreen,'');
   gxy(8,4);
   setcolor(green);backk(blue);wrstr('N');backk(yellow);wrstr('   A  M');
   wrln;wrln;
   setcolor(black);
   wrstr('   (c) by        KIRR');wrln;
   wrstr('            NIKITA     ');
   setcolor(yellow);backk(green);
   gxy(10,8);
   wrstr('Ok');
   open_window;
   hide_cursor;

      repeat a:=readkey; until a=#13;
      delete_window;
      init_text(@tex,'NONAME.NAM');
      {
      load_text('exp.txt',@tex,500);
      if error<>0 then
       begin
          error_message('Can''t open file');
          halt;
       end;
      }
      write(hmenu(1,1,5,4,'File             Edit             Debug            Run
  '));
      new_window(76,22,2,1,8,9,lightred,yellow,'Edit 1');
      load_text('noname.nam',@tex,100);
      go_window(8);
      open_window;
      a:=te(tex,true);
      set_window(0);
  end.
  →
```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX