

CRISTIAN CEMÎRTAN

Documentație
pentru produsul soft
„Graph Paint”

Chișinău, 2022

CUPRINS

I. PROGRAM ÎN LIMBAJUL DE ASAMBLARE	3
1. 1. Condiția problemei	3
1. 2. Instrucțiunile de utilizare a programului	3
1. 3. Logica programului	3
1. 4. Plasarea unui pixel.....	4
1. 5. Desenarea unui pătrat	5
1. 6. Desenarea unui cerc.....	5
1. 7. Trasarea unei linii	7
1. 8. Redimensionarea razei sau semilaturii	10
II. PROGRAM ÎN LIMBAJUL C	11
2. 1. Condiția problemei	11
2. 2. Instrucțiunile de utilizare și logica programului.....	11
2. 3. Construirea programului.....	11
2. 4. Utilizarea programului.....	12
CONCLUZII.....	13
BIBLIOGRAFIE	14

I. PROGRAM ÎN LIMBAJUL DE ASAMBLARE

1. 1. Condiția problemei

De realizat un produs program grafic ce permite utilizatorului să picteze un tablou utilizând instrumentele: ștampila ce imprimă o circumferință de cerc sau un pătrat opac, și un trasator de linii.

1. 2. Instrucțiunile de utilizare a programului

ESC – ieșire din program;

Click stânga (principal) – desenează un cerc sau trasează o linie;

Click dreapta (secundar) – desenează un pătrat;

A – alege între cerc și trasarea liniei;

Z X – schimbă crescător/descrescător culoarea instrumentului principal;

C V – schimbă crescător/descrescător culoarea instrumentului secundar;

Săgețile sus/jos – micșorează/crește raza cercului/semilatura pătratului;

Space – permite sau nu efectul de suprapunere;

N – resetează zonei de desen.

1. 3. Logica programului

La lansarea programului, se verifică prezența pilotului (driver) ce este responsabil pentru controlul mouse-ului [1]. Dacă pilotul este absent, programul iese din execuție, afișând la ecran mesajul corespunzător. Dacă pilotul este prezent, se setează modul video la rezoluția de 640x480 cu 16 culori [2].

Pentru a pregăti spațiul de lucru pentru desen, se imprimă bara status împreună cu următoarele stări: culoarea principală/secundară curentă, starea de suprapunere, dimensiunea razei și instrumentul principal curent. Apoi, se colorează spațiul de lucru în culoarea neagră, pentru a asigura un tablou inițial curat, utilizând acces direct la *VRAM* (memoria video de lucru) [3].

Programul intră într-o buclă, ce permite utilizatorului să beneficieze de o opțiune menționată în paragraful precedent.

¹ Jurgens D., *INT 33, 0 – Mouse Reset/Get Mouse Installed Flag*, HelpPC Reference Library, https://www.stanislavs.org/helppc/int_33-0.html.

² Jurgens D., *INT 10, 12 – Video Subsystem Configuration (EGA/VGA)*, HelpPC Reference Library, https://www.stanislavs.org/helppc/int_10-12.html.

³ OSDev, *Drawing In a Linear Framebuffer*, https://wiki.osdev.org/Drawing_In_a_Linear_Framebuffer.

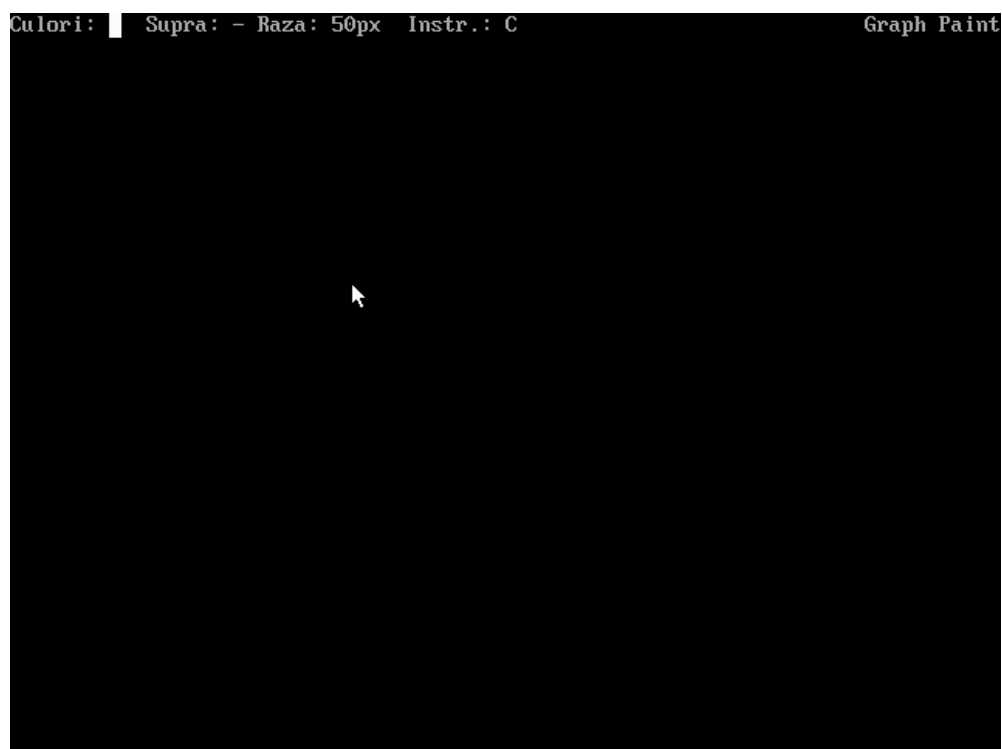


Figura 1. 1. Tabloul de desen inițial

1. 4. Plasarea unui pixel

Acest program conține o procedură ce permite, în mod sigur, unui instrument de desen să-și plaseze punctele colorate (pixeli) pe un tablou de desen. La apelarea procedurii, se verifică poziția pixelului ce va fi modificat, și dacă este în afara zonei tabloului, returnează fără să plaseze un pixel.

Dacă este activat efectul de suprapunere (se schimbă cu tasta **Space**), se compară valoarea culorii din parametru cu cea de pe ecran, și dacă sunt diferite, atunci valoarea culorii a pixelului de pe ecran devine o disjuncție exclusivă (XOR) între ele.

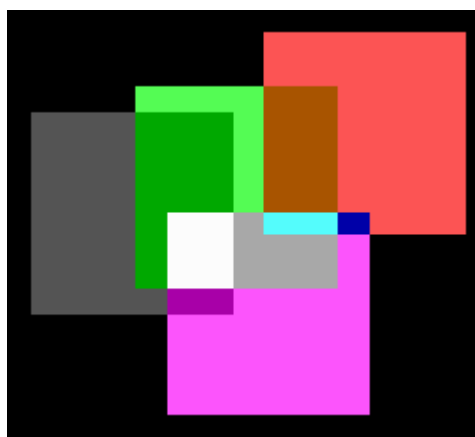


Figura 1. 2. Efectul de suprapunere activat

1. 5. Desenarea unui pătrat

Când utilizatorul face click din partea dreapta a mouse-ului, are loc desenarea unui pătrat opac poziționat centrat cursorului mouse. Algoritmul este simplu, și este reprezentat de o parcurgere liniară a unei bucle bidimensionale, unde la fiecare pas se plasează un pixel de culoarea secundară.

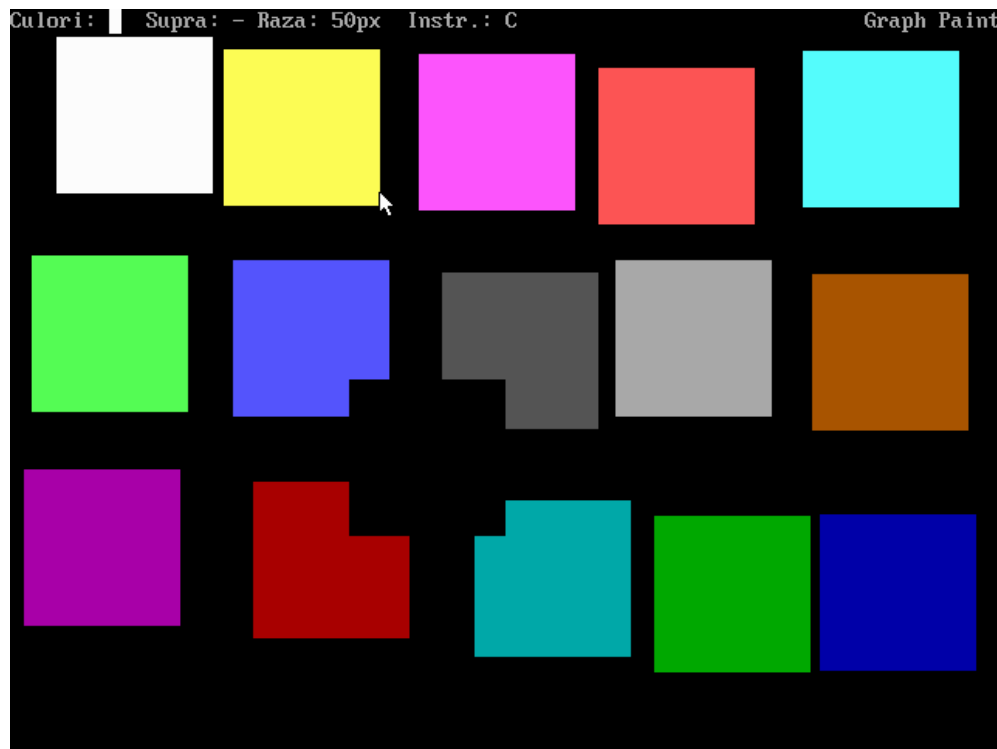


Figura 1. 3. Pătratele desenate sub 16 culori diferite

1. 6. Desenarea unui cerc

Când utilizatorul face click din partea stângă a mouse-ului, și dacă indicatorul instrumentului este C, are loc desenarea unei circumferințe de cerc poziționat centrat cursorului mouse. Algoritmul este cercetat de Bresenham, și este reprezentat de o trasare simetrică a celor 8 octante de 45°.

Avantajul principal este că algoritmul lui Bresenham se bazează pe aritmetica numerelor întregi, și este mai performant în comparație cu prelucrarea numerelor reale [4].

⁴ Pradham S., *Bresenham's circle drawing algorithm*, GeeksforGeeks, <https://www.geeksforgeeks.org/bresenham-circle-drawing-algorithm>.



Figura 1. 4. Trasarea cercului observată pe pași

Varianta originală în C	Varianta tradusă în limbajul de asamblare
<pre> #define symm_pixel()\ draw_pixel(xc + x, yc + y);\ draw_pixel(xc - x, yc + y);\ draw_pixel(xc + x, yc - y);\ draw_pixel(xc - x, yc - y);\ draw_pixel(xc + y, yc + x);\ draw_pixel(xc - y, yc + x);\ draw_pixel(xc + y, yc - x);\ draw_pixel(xc - y, yc - x) // (si, di, ebx) void circle(const int xc, const int yc, const int r) { int x = 0; // eax int y = r; // edx int d = 3 - 2 * r; // ecx for (; ;) { symm_pixel(); if (x > y) break; ++x; if (d > 0) { --y; d += 10 + 4 * (x - y); } else d += 6 + 4 * x; } } </pre>	<pre> PROC cerc push eax ebx ecx edx ; Fie eax = x, edx = y, ecx = d (precizie) movzx ebx, r xor eax, eax mov edx, ebx ; raza ; d = 3 - 2 * r lea ecx, [2 * ebx - 3] neg ecx cond: symm_pixel ; anexa cerc.asm cmp eax, edx jg proc_end inc eax ; determinam daca precizia este negativa test ecx, ecx jns jos ; d += 6 + 4 * x lea ecx, [ecx + 4 * eax + 6] jmp cond jos: dec edx ; d += 10 + 4 * (x - y) mov ebx, eax sub ebx, edx lea ecx, [ecx + 4 * ebx + 10] jmp cond proc_end: pop edx ecx ebx eax ret ENDP cerc </pre>

Figura 1. 5. Algoritmul cercului a lui Bresenham

1. 7. Trasarea unei linii

Când utilizatorul face click din partea stângă a mouse-ului, și dacă indicatorul instrumentului este L, are loc trasarea unei linii având capătul inițial poziționat cursorului mouse. Ca să fie trasată o linie pe ecran, utilizatorul mai face încă o dată click-ul. În mod asemănător cu algoritmul menționat precedent, algoritmul eficient privind trasarea unei linii tot este cercetat de Bresenham [5].

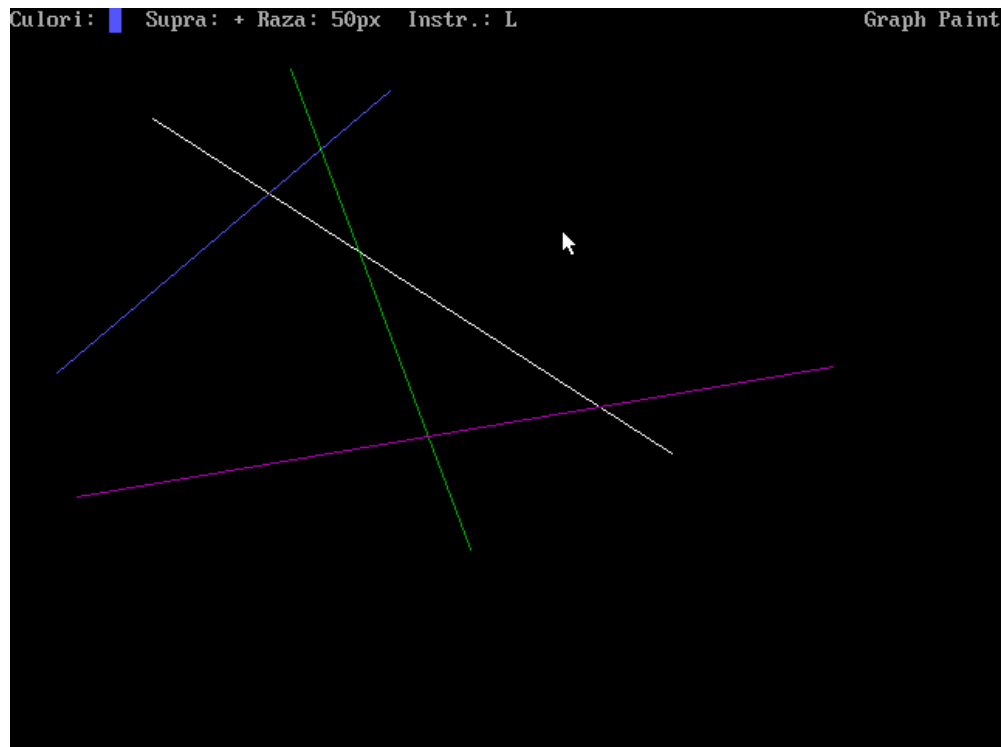


Figura 1. 6. Liniile trasate

⁵ Rosetta Code, Bitmap/Bresenham's line algorithm,
http://rosettacode.org/wiki/Bitmap/Bresenham%27s_line_algorithm.

Varianta originală în C	Varianta tradusă în limbajul de asamblare
<pre> // (cx, dx, si, di) void line(int x0, int y0, const int x1, const int y1) { const int sx = x0 < x1 ? 1 : -1; // [bp - 2] const int sy = y0 < y1 ? 1 : -1; // [bp - 4] int dx = abs(x1 - x0); // ax int dy = abs(y1 - y0); // bx int err = (dx > dy ? dx : -dy) >> 1; // [bp - 6] int temp; // [bp - 8] for (; ;) { if (!draw_pixel(x0, y0)) break; if (x0 == x1 && y0 == y1) break; temp = err; if (temp > -dx) { err -= dy; x0 += sx; } if (temp < dy) { err += dx; y0 += sy; } } } </pre>	<pre> PROC line push eax ebx cx dx si di bp mov bp, sp ; ax - dx, bx - dy ; cx - x0, dx - y0, si - x1, di - y1 mov cx, [bp + 22] ; xc precedent mov dx, [bp + 20] ; yc precedent mov si, xc mov di, yc line\$sx_sy: cmp cx, si setl al cmp dx, di setl bl movzx eax, al movzx ebx, bl lea eax, [eax + eax - 1] ; -1 daca 0, 1 daca 1 lea ebx, [ebx + ebx - 1] push ax bx ; sx, sy in stiva [bp - 2], [bp - 4] line\$dx: mov ax, si sub ax, cx jns line\$dy neg ax ; abs line\$dy: mov bx, di sub bx, dx jns line\$err neg bx ; abs line\$err: cmp ax, bx jle line\$push_neg_dy push ax ; [bp - 6] jmp line\$div2 line\$push_neg_dy: push bx ; [bp - 6] neg WORD PTR [esp] line\$div2: sar WORD PTR [esp], 1 </pre>

	<pre> push ax ; temp, ax nu are rol specific [bp - 8] line\$loop: push ax mov al, color call draw_pixel jc line\$ret cmp cx, si ; x0 == x1 sete al cmp dx, di ; y0 == y1 sete ah test al, ah jnz line\$ret ; temp > -dx mov ax, [bp - 6] mov [bp - 8], ax ; se salveaza err pop ax neg ax cmp [bp - 6], ax jle line\$err_lt_dy sub [bp - 6], bx add cx, [bp - 2] ; temp < dy line\$err_lt_dy: neg ax cmp [bp - 8], bx jge line\$loop add [bp - 6], ax add dx, [bp - 4] jmp line\$loop line\$ret: mov sp, bp pop bp di si dx cx ebx eax ret ENDP line </pre>
--	---

Figura 1. 7. Algoritmul liniei a lui Bresenham

1. 8. Redimensionarea razei sau semilaturii

Utilizând tastele de săgeată de sus sau jos, utilizatorul are la dispoziție să mărească sau să micșoreze raza cercului sau semilatura pătratului. Valoarea razei poate fi între 1 și 239.

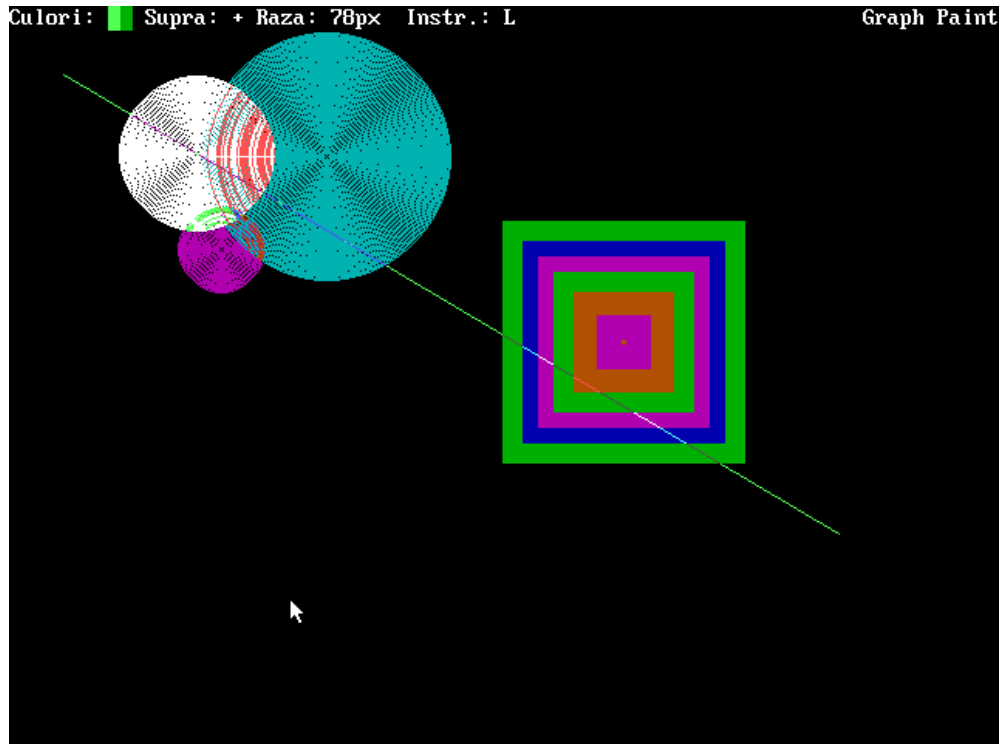


Figura 1. 8. Un tablou care conține figuri geometrice cu mărimi diverse

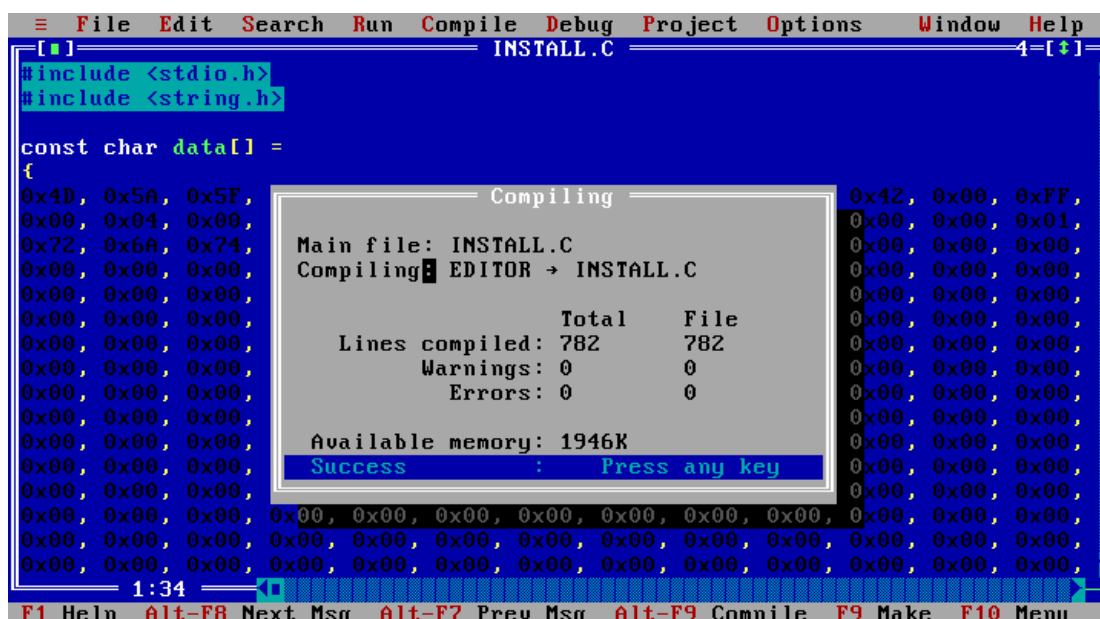


Figura 2. 2. Programul compilat cu succes

2. 4. Utilizarea programului

În cazul nostru, specificăm următoarea calea absolută: C:\grpaint, unde vor fi extrase programul „Graph Paint” și ghidul său de utilizare.

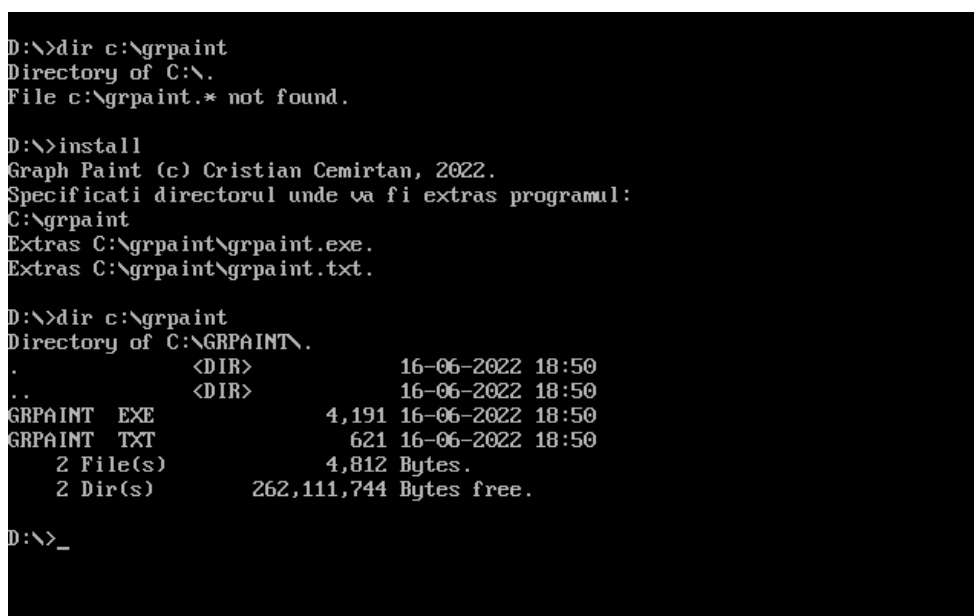


Figura 2. 3. Instalarea produsului program „Graph Paint”

CONCLUZII

Produsul program „Graph Paint” a fost inițial elaborat în anul 2021. Prima versiune suporta numai desenarea circumferințelor de cerc, și spațiul tabloului de desen era mai mic cu 16 pixeli, ca să fie afișată legenda operațiunilor în loc ca programul să fie însoțit de un ghid sub formă de fișier text.

Am reactualizat produsul program cu opțiunile noi menționate în conținutul documentației. Pentru afișarea literelor și a cifrelor pe ecran, am utilizat o bibliotecă scrisă de mine tot în anul precedent. Biblioteca, `stdlibc`, implementează parțial funcțiile/procedurile prevăzute în standardul limbajului de programare C. Această bibliotecă era creată cu scopul de a-mi ușura realizarea lucrărilor de laborator pentru disciplina „Arhitectura calculatorului și limbajul de asamblare”.

Pentru a face produsul program portabil, am decis să realizez un instalator, scris în limbajul C, ce dezarchivează programul însoțit cu ghidul de utilizare, într-un director specificat de utilizator. În realizarea instalatorului, am utilizat un bloc de cod, scris în limbajul de asamblare, ce permite creării unui director în caz dacă nu există [8].

⁸ Jurgens D., *INT 21, 39 – Create Subdirectory (mkdir)*, HelpPC Reference Library, https://www.stanislaivs.org/helppc/int_21-39.html.

BIBLIOGRAFIE

1. Jurgens D., *INT 33, 0 – Mouse Reset/Get Mouse Installed Flag*, HelpPC Reference Library, https://www.stanislavs.org/helppc/int_33-0.html.
2. Jurgens D., *INT 10, 12 – Video Subsystem Configuration (EGA/VGA)*, HelpPC Reference Library, https://www.stanislavs.org/helppc/int_10-12.html.
3. OSDev, *Drawing In a Linear Framebuffer*, https://wiki.osdev.org/Drawing_In_a_Linear_Framebuffer.
4. Pradham S., *Bresenham's circle drawing algorithm*, GeeksforGeeks, <https://www.geeksforgeeks.org/bresenham-circle-drawing-algorithm>.
5. Rosetta Code, *Bitmap/Bresenham's line algorithm*, http://rosettacode.org/wiki/Bitmap/Bresenham%27s_line_algorithm.
6. Ostrowski T., *File to hexadecimal converter*, https://tomeko.net/online_tools/file_to_hex.php.
7. WinWorld, *Borland Turbo C++ 3.x (DOS)*, <https://winworldpc.com/product/turbo-c/3x>.
8. Jurgens D., *INT 21, 39 – Create Subdirectory (mkdir)*, HelpPC Reference Library, https://www.stanislavs.org/helppc/int_21-39.html.