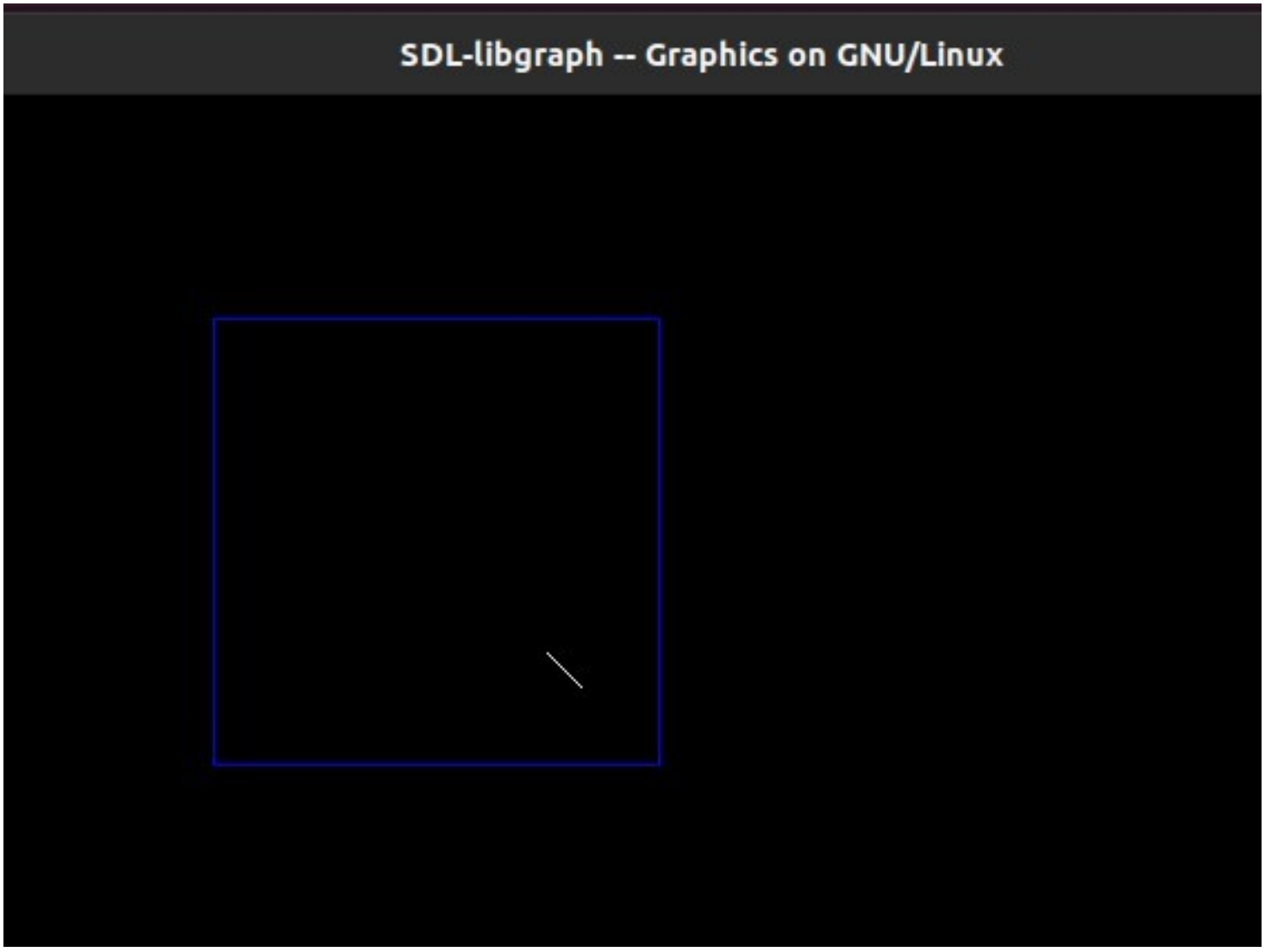
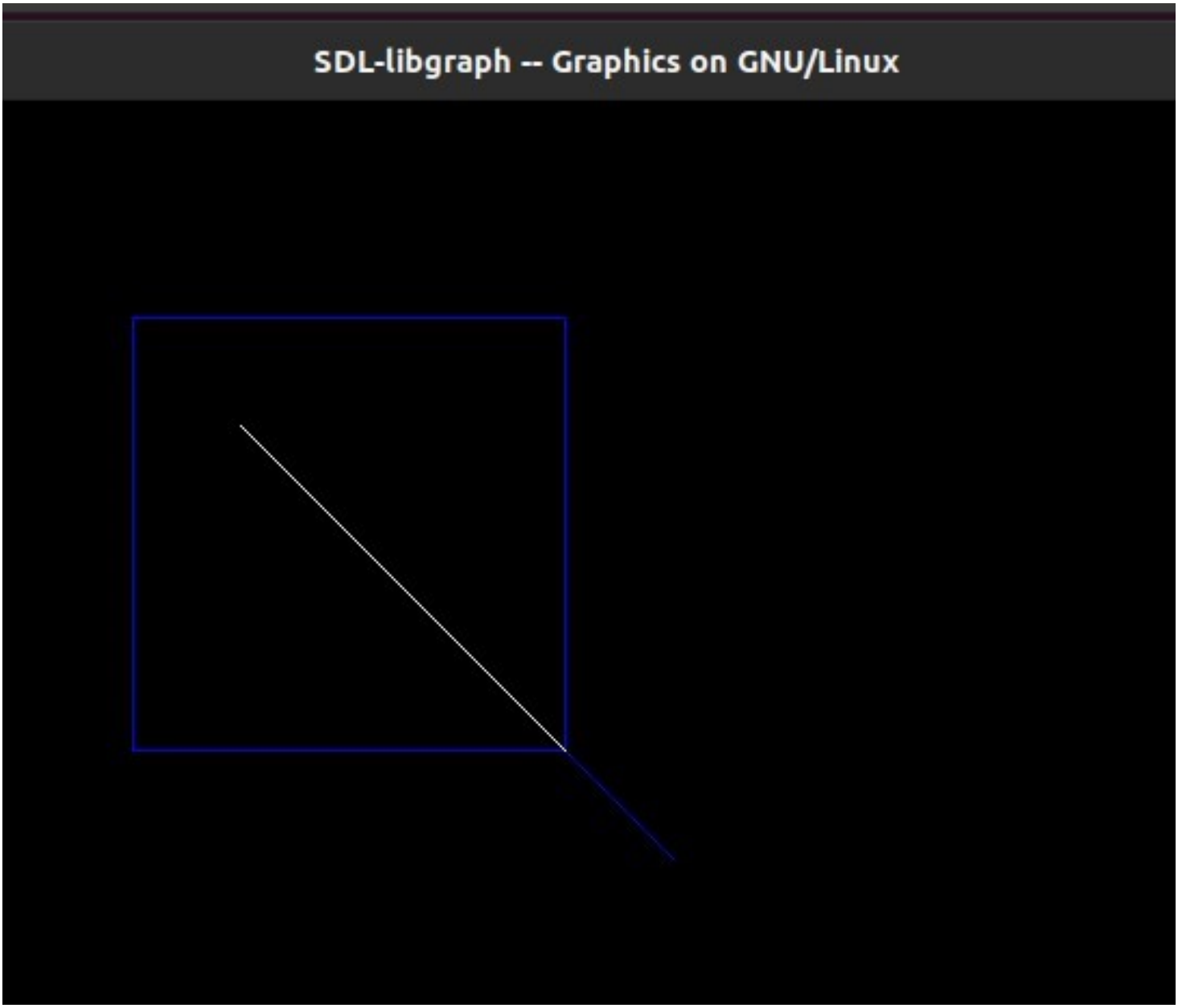
```cpp
#include <iostream>
#include <graphics.h>
using namespace std;
static int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8, xl, yl, xh, yh;
int getcode(int x, int y)
{
    int code = 0;
    // Perform Bitwise OR to get outcode
    if (y > yh)
        code |= TOP;
    if (y < yl)
        code |= BOTTOM;
    if (x < xl)
        code |= LEFT;
    if (x > xh)
        code |= RIGHT;
    return code;
}
int main()
{
    int gdriver = DETECT, gmode;
    initgraph(&gdriver, &gmode, NULL);
    setcolor(BLUE);
    cout << "Enter bottom left and top right co-ordinates of window: ";
    cin >> xl >> yl >> xh >> yh;
    rectangle(xl, yl, xh, yh);
    int x1, y1, x2, y2;
    cout << "Enter the endpoints of the line: ";
    cin >> x1 >> y1 >> x2 >> y2;
    line(x1, y1, x2, y2);
    getch();
    int outcode1 = getcode(x1, y1), outcode2 = getcode(x2, y2);
    int accept = 0; // decides if line is to be drawn
    while (1)
    {
        float m = (float)(y2 - y1) / (x2 - x1);
        // Both points inside. Accept line
        if (outcode1 == 0 && outcode2 == 0)
        {
            accept = 1;
            break;
        }
        // AND of both codes != 0.Line is outside. Reject line
        else if ((outcode1 & outcode2) != 0)
        {
            break;
        }
        else
        {
            int x, y;
            int temp;
            // Decide if point1 is inside, if not, calculate intersection
            if (outcode1 == 0)
                temp = outcode2;
            else
                temp = outcode1;
            // Line clips top edge
            if (temp & TOP)
            {
```

```
60                  x = x1 + (yh - y1) / m;
61                  y = yh;
62              }
63              else if (temp & BOTTOM)
64              { // Line clips bottom edge
65                  x = x1 + (yl - y1) / m;
66                  y = yl;
67              }
68              else if (temp & LEFT)
69              { // Line clips left edge
70                  x = xl;
71                  y = y1 + m * (xl - x1);
72              }
73              else if (temp & RIGHT)
74              { // Line clips right edge
75                  x = xh;
76                  y = y1 + m * (xh - x1);
77              }
78              // Check which point we had selected earlier as temp, and replace its
    coordinates
79              if (temp == outcode1)
80              {
81                  x1 = x;
82                  y1 = y;
83                  outcode1 = getcode(x1, y1);
84              }
85              else
86              {
87                  x2 = x;
88                  y2 = y;
89                  outcode2 = getcode(x2, y2);
90              }
91          }
92      }
93      setcolor(WHITE);
94      cout << "After clipping:";
95      if (accept)
96          line(x1, y1, x2, y2);
97      return 0;
98      closegraph();
99  }
```


SDL-libgraph -- Graphics on GNU/Linux


SDL-libgraph -- Graphics on GNU/Linux