COMP2001/COMP2401

Introduction to System Programming

Summer 2012, Assignment 2

Deadline : May 30, 2012, at 12:00 (noon), submitted via WebCT

Objectives:

You will write a program in C under Linux to create very basic text-art works, which is also called digital micrography. These kinds of art works use different character patterns to represent simple shapes. The goal is to design four patterns (also called drawings or shapes) with different characters, to transform them into another pattern with bit operators (**ONCE**), and to display them on the screen by selecting an interactive menu. By completing this assignment, you will be able to

- manipulate advanced data types: 1D and 2D array and structures;
- write a simple user-interactive program;
- create and call functions by passing parameters;
- use bit operators;
- manipulate pointers in a very basic way.

You **MUST** submit your source code and any other files required to compile and run your program through WebCT. All submitted files should be packaged as **one single .tar or .zip file**. TAs should be able to compile and run your program under our SCS Linux machines. [Total 15 marks]

Description:

Digital micrography is an interesting artistic style, which uses different characters to create different patterns and thus to represent different shapes. You are going to create a simple interactive system to display this style by placing character-based shapes at specified locations. Your system should allow users to transform these patterns into another patterns **once** by adapting bit operators. The requested system and the output patterns are demonstrated in the **assign2log.txt** file. Your system should accomplish similar jobs as demonstrated. The basic skeleton of this program is provided by assign2.c, which gives you some basic impression about your program. You can either build your own program on this skeleton code or create your own source code. The program does the following steps:

(1) initialize the shapes by using initShape function; six shapes must be initialized to produce the sample output: two ALINEs at positions (10,10) and (30,20), a GRECTANGLE at position (45,25), a CCROSS at position (20,15), and two DSQUAREs at positions (15,15) and (40,7).
(2) create an interactive menu and allow users to select three different options:

- select 0: print out the space map displaying all those shapes, as found in the sample output;
- select 1: transform the current patterns into another patterns;
- select 2: exit system.

Transformation is a **one-time** action. After selecting '1' **once**, this option will be not usable any more.

Requirements:

1. You must define two structures called **Position** and **Shape**. The Shape structure uses the Position structure as a field to specify the starting location of the shape. Each shape must have a position where it will be displayed on the space map and a **two-dimensional character image** to represent its pattern. For efficiency in initializing and drawing the shapes, each shape needs to keep track of **the size of its image**, both in the horizontal and vertical directions.
2. There are **four types** of shapes constructed with capital letters ('A', 'C, 'D', and 'G') in this program: ALINE, a vertical line of three 'A' characters; CCROSS, a cross with two perpendicular lines intersected with each other at the middle point (there are 7 'C' characters on each line); DSQUARE, a square of 2 by 2 'D' characters; and GRECTANGLE, a rectangle of 8 by 2 'G' characters.
3. You must use enumerated data type called **PatternType** to name the different shape types.
4. You have to implement a few functions called: **printMap**, **initShape**, **printMenu**, **transformMap**. You could add other functions if it is needed to complete these functions.

- printMap: display all of shapes on the screen
- initShape: initialize a shape and load information for a shape; there has to be a way to differentiate the shape types in this function. This function initializes all the values of the given shape (including its image), based on the given shape type, and positioned at the given x and y coordinates.
- printMenu: display system menu
- transformMap: transform the characters used in shapes into another characters by using right-shifting bit operator. After the right-shift, every bit value for characters has moved to the right by one bit position. **Note**: your output for the boundary lines might look different as the sample, which is fine.

5.To simplify the problem, the output map is **a fixed-size two-dimentional region** with width = 60 in x direction and height = 30 in y direction. Initially, there are boundary lines around the region map as shown in sample output. If the shapes are outside this region, it will not be shown on the screen. It only shows shapes with valid starting positions.

Marking scheme:

1. **Submission**
   - You must follow all the instructions **exactly**, or you will lose marks
2. **Deductions**
   - 2 marks if the assignment is marked **Late** in WebCT (submitted between 12:00 noon and 12:30 PM)
   - 15 marks if the assignment is marked **Missed** in WebCT (submitted after 12:30 PM)
   - 3 marks if the code does not compile, if any submitted files are missing or corrupt or in the wrong format, or if the program consistently crashes
   - 0.5 marks for missing comments or other bad style (non-standard indentation, improper funct/var names, etc)
3. **Bonus Marks**
   - Up to 3 extra marks are available for fun and creative additional features.