第八次作业大程 Report

1. 源代码

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    FILE* fphzk = NULL;
    FILE* fp = fopen("output.txt", "w");
    int i, j, k, offset, operation;
    int flag;
    unsigned char buffer[32];
    unsigned char word[1000][3];
    unsigned char key[8] = {
        0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01
    };
    fphzk = fopen("hzk16", "rb");
    if (fphzk == NULL) {
        fprintf(stderr, "error hzk16\n");
        return 1;
    }
    //处理汉字
    printf("请输入汉字:\n");
    for (int m = 0;; m++) {
        fgets(word[m], 3, stdin);
        word[m][3] = 0;
        if (word[m][0] == '\n')
            break;
    }
    for (int m = 0;; m++)
    {
        if (word[m][0] == '\n')
            break;
        offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word[m][1] - 0xa0 - 1)) * 32;
        fseek(fphzk, offset, SEEK_SET);
        fread(buffer, 1, 32, fphzk);
        for (k = 0; k < 16; k++) {
            for (j = 0; j < 2; j++) {
                for (i = 0; i < 8; i++) {
```

```
                        flag = buffer[k * 2 + j] & key[i];
                        printf("%s", flag ? "●" : "○");
                    }
                }
                printf("\n");
            }
            printf("\n");
        }

        //选择模式
        while (1)
        {
            printf("请选择模式：\n0.退出\n1.文件输出\n2.放大\n3.斜体\n4.倒立\n5.变色\n6.动态显示\n");
            scanf("%d", &operation);
            system("cls");
            system("color 0F");
            switch (operation)
            {
            case 0:
                return 0;
            case 1:
                for (int m = 0;; m++)
                {
                    if (word[m][0] == '\n')
                        break;
                    offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word[m][1] - 0xa0 - 1)) * 32;
                    fseek(fphzk, offset, SEEK_SET);
                    fread(buffer, 1, 32, fphzk);
                    for (k = 0; k < 16; k++)
                    {
                        for (j = 0; j < 2; j++) {
                            for (i = 0; i < 8; i++) {
                                flag = buffer[k * 2 + j] & key[i];
                                fprintf(fp, "%s", flag ? "● " : "○ ");
                            }
                        }
                        fprintf(fp, "\n");
                    }
                    printf("\n");
                }
                break;
            case 2:
```

```
79.                 for (int m = 0;; m++)
80.                 {
81.                     if (word[m][0] == '\n')
82.                         break;
83.                     offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
       [m][1] - 0xa0 - 1)) * 32;
84.                     fseek(fphzk, offset, SEEK_SET);
85.                     fread(buffer, 1, 32, fphzk);
86.                     for (k = 0; k < 16; k++)
87.                     {
88.                         for (j = 0; j < 2; j++) {
89.                             for (i = 0; i < 8; i++) {
90.                                 flag = buffer[k * 2 + j] & key[i];
91.                                 printf("%s", flag ? "●●" : "○○");
92.                             }
93.                         }
94.                         printf("\n");
95.                         for (j = 0; j < 2; j++) {
96.                             for (i = 0; i < 8; i++) {
97.                                 flag = buffer[k * 2 + j] & key[i];
98.                                 printf("%s", flag ? "●●" : "○○");
99.                             }
100.                        }
101.                        printf("\n");
102.                    }
103.                    printf("\n\n");
104.                }
105.                break;
106.            case 3:
107.                for (int m = 0;; m++)
108.                {
109.                    if (word[m][0] == '\n')
110.                        break;
111.                    offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
       [m][1] - 0xa0 - 1)) * 32;
112.                    fseek(fphzk, offset, SEEK_SET);
113.                    fread(buffer, 1, 32, fphzk);
114.                    for (k = 0; k < 16; k++) {
115.                        for (int n = 0; n < 16-k; n++)
116.                        {
117.                            printf(" ");
118.                        }
119.                        for (j = 0; j < 2; j++) {
120.                            for (i = 0; i < 8; i++) {
```

```
                                flag = buffer[k * 2 + j] & key[i];
                                printf("%s", flag ? "●" : "○");
                            }
                        }
                        printf("\n");
                    }
                    printf("\n");
                }
                break;
            case 4:
                for (int m = 0;; m++)
                {
                    if (word[m][0] == '\n')
                        break;
                    offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
    [m][1] - 0xa0 - 1)) * 32;
                    fseek(fphzk, offset, SEEK_SET);
                    fread(buffer, 1, 32, fphzk);
                    for (k = 15; k >=0; k--) {
                        for (j = 1; j >=0; j--) {
                            for (i = 7; i >= 0; i--) {
                                flag = buffer[k * 2 + j] & key[i];
                                printf("%s", flag ? "●" : "○");
                            }
                        }
                        printf("\n");
                    }
                    printf("\n");
                }
                break;
            case 5:
                system("color 0C");
                for (int m = 0;; m++)
                {
                    if (word[m][0] == '\n')
                        break;
                    offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (word
    [m][1] - 0xa0 - 1)) * 32;
                    fseek(fphzk, offset, SEEK_SET);
                    fread(buffer, 1, 32, fphzk);
                    for (k = 0; k < 16; k++) {
                        for (j = 0; j < 2; j++) {
                            for (i = 0; i < 8; i++) {
                                flag = buffer[k * 2 + j] & key[i];
```

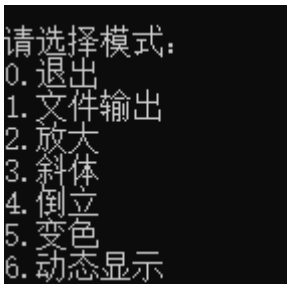
```
                                printf("%s", flag ? "●" : "○");
                            }
                        }
                        printf("\n");
                    }
                    printf("\n");
                }
            case 6:
                for (int n = 1; n <= 20; n++)
                {
                    for (int m = 0;; m++)
                    {
                        if (word[m][0] == '\n')
                            break;
                        offset = (94 * (unsigned int)(word[m][0] - 0xa0 - 1) + (
    word[m][1] - 0xa0 - 1)) * 32;
                        fseek(fphzk, offset, SEEK_SET);
                        fread(buffer, 1, 32, fphzk);
                        for (k = 0; k < 16; k++) {
                            for (j = 0; j < 2; j++) {
                                for (i = 0; i < 8; i++) {
                                    flag = buffer[k * 2 + j] & key[i];
                                    printf("%s", flag ? "●" : "○");
                                }
                            }
                            printf("\n");
                        }
                        printf("\n");
                    }
                    _sleep(100);
                    system("cls");
                    _sleep(100);
                }

            default:
                break;
            }
        }
        fclose(fphzk);
        fphzk = NULL;
        fclose(fp);
        system("pause");
        return 0;
    }
```

2. 结果展示

1) 基本功能：输入汉字（小于 1000 个）

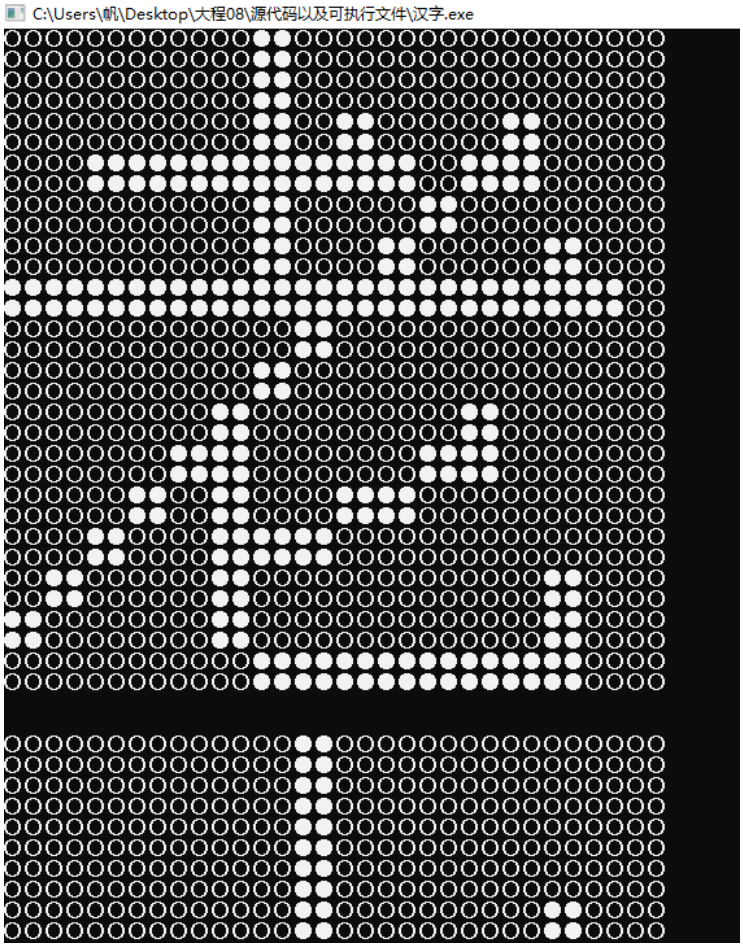
所有功能如下



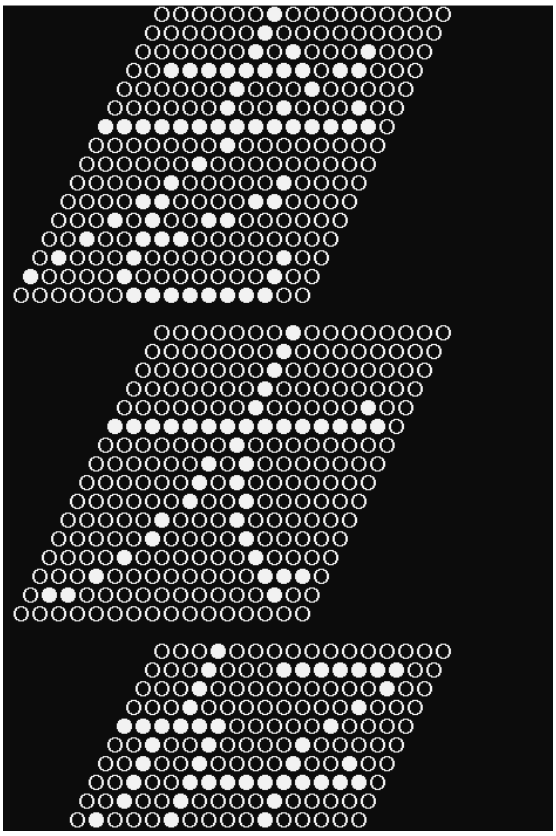
2) **功能扩展：文件输出**

如图选择 1 后可以输出到 output.txt:

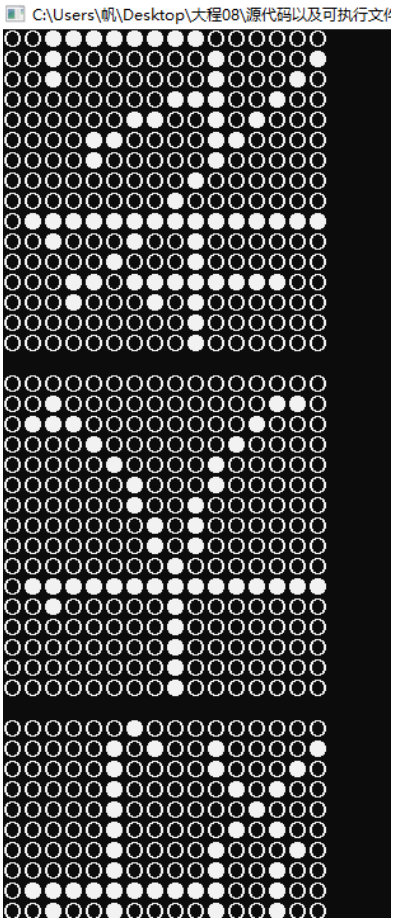
3) **功能扩展：放大**



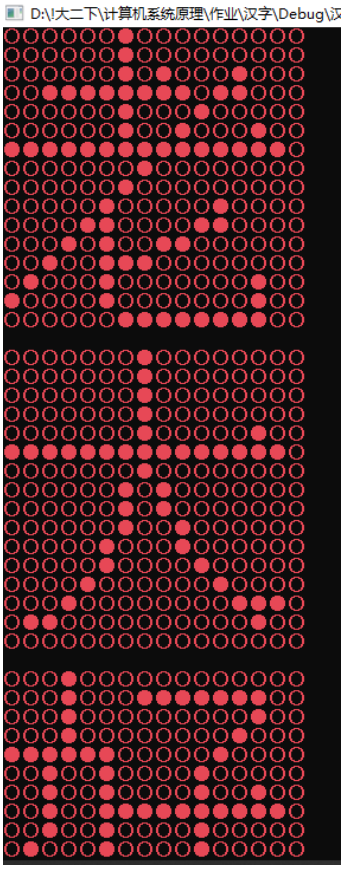
4) 功能扩展：斜体



5) 功能扩展：倒立



6) **功能扩展：变色**



7) **功能扩展：动态显示**

这个功能需要通过在程序中具体体验才可以体会到，欢迎打开汉字显示.exe 进行实际操作。

3. 总结

此次进行汉字显示的大程收获颇多，了解、理解了汉字库以及汉字显示的实际过程。并通过多种方式进行汉字显示，收获了乐趣以及经验，对未来计算机系统原理课程充满了兴趣以及期待。