

1 簡単な時計のプログラム 完成版

前回までに動作確認をした部品を統合しました。

```
/*
    clock.c
*/
#include <time.h>
#include <stdio.h>
#include <string.h>
#include <windows.h>

char c[10]; // 現在の時刻 (文字列)
char buf[8][64]; // 画面表示用 8x64 文字

void put_time();
void Set0(int j); // 0 を設定
void Set1(int j); // 1 を設定
void Set2(int j); // 2 を設定
void Set3(int j); // 3 を設定
void Set4(int j); // 4 を設定
void Set5(int j); // 5 を設定
void Set6(int j); // 6 を設定
void Set7(int j); // 7 を設定
void Set8(int j); // 8 を設定
void Set9(int j); // 9 を設定
void SetC(int j); // : を設定

int main( )
{
    int i,j,k;

    while(1){
        put_time(); // 現在時間を取得

        for(j=0,k=0; j<8; j++, k+=8){

            switch( c[j] ){ // 時分秒の文字 XX:XX:XX
                case '0': Set0(k); break; // 0 を設定
                case '1': Set1(k); break; // 1 を設定
                case '2': Set2(k); break; // 2 を設定
                case '3': Set3(k); break; // 3 を設定
                case '4': Set4(k); break; // 4 を設定
                case '5': Set5(k); break; // 5 を設定
                case '6': Set6(k); break; // 6 を設定
                case '7': Set7(k); break; // 7 を設定
                case '8': Set8(k); break; // 8 を設定
                case '9': Set9(k); break; // 9 を設定
                case ':': SetC(k); break; // : を設定
            }
        }
    }
}
```

```

    }
}

system("cls"); // 画面消去

for(i=0; i<8; i++){ // 第0行~第7行まで
    printf("\n%.64s",&buf[i][0]); // 横1行64文字表示
}

fflush(stdout); // 画面出力

Sleep(1000); // 1秒待つ
}

return 0;
}

void put_time() // 現在時間を取得
{
    time_t current;
    struct tm *local;

    time(&current);
    local = localtime(&current);

    sprintf(c, "%02d:%02d:%02d\n",local->tm_hour, local->tm_min, local->tm_sec);
}

// 0を設定
void Set0(int j)
{
    memcpy(&buf[0][j], "##### ",8);
    memcpy(&buf[1][j], " # # ",8);
    memcpy(&buf[2][j], " # # ",8);
    memcpy(&buf[3][j], " # # ",8);
    memcpy(&buf[4][j], " # # ",8);
    memcpy(&buf[5][j], " # # ",8);
    memcpy(&buf[6][j], "##### ",8);
}

// 1を設定
void Set1(int j)
{
    memcpy(&buf[0][j], " # ",8);
    memcpy(&buf[1][j], " # ",8);
    memcpy(&buf[2][j], " # ",8);
}

```

```

        memcpy(&buf[3][j], "      # ",8);
        memcpy(&buf[4][j], "      # ",8);
        memcpy(&buf[5][j], "      # ",8);
        memcpy(&buf[6][j], "      # ",8);
    }

```

// 2を設定

```

void Set2(int j)
{
    memcpy(&buf[0][j], "##### ",8);
    memcpy(&buf[1][j], "      # ",8);
    memcpy(&buf[2][j], "      # ",8);
    memcpy(&buf[3][j], "##### ",8);
    memcpy(&buf[4][j], "      # ",8);
    memcpy(&buf[5][j], "      # ",8);
    memcpy(&buf[6][j], "##### ",8);
}

```

// 3を設定

```

void Set3(int j)
{
    memcpy(&buf[0][j], "##### ",8);
    memcpy(&buf[1][j], "      # ",8);
    memcpy(&buf[2][j], "      # ",8);
    memcpy(&buf[3][j], "##### ",8);
    memcpy(&buf[4][j], "      # ",8);
    memcpy(&buf[5][j], "      # ",8);
    memcpy(&buf[6][j], "##### ",8);
}

```

// 4を設定

```

void Set4(int j)
{
    memcpy(&buf[0][j], " # # ",8);
    memcpy(&buf[1][j], " # # ",8);
    memcpy(&buf[2][j], " # # ",8);
    memcpy(&buf[3][j], "##### ",8);
    memcpy(&buf[4][j], "      # ",8);
    memcpy(&buf[5][j], "      # ",8);
    memcpy(&buf[6][j], "      # ",8);
}

```

// 5を設定

```

void Set5(int j)
{
    memcpy(&buf[0][j], "##### ",8);
    memcpy(&buf[1][j], "      # ",8);
    memcpy(&buf[2][j], "      # ",8);
    memcpy(&buf[3][j], "##### ",8);
}

```

```

        memcpy(&buf[4][j], "      # ",8);
        memcpy(&buf[5][j], "      # ",8);
        memcpy(&buf[6][j], " ##### ",8);
    }

```

// 6を設定

```

void Set6(int j)
{
    memcpy(&buf[0][j], " ##### ",8);
    memcpy(&buf[1][j], " #      ",8);
    memcpy(&buf[2][j], " #      ",8);
    memcpy(&buf[3][j], " ##### ",8);
    memcpy(&buf[4][j], " # #    ",8);
    memcpy(&buf[5][j], " # #    ",8);
    memcpy(&buf[6][j], " ##### ",8);
}

```

// 7を設定

```

void Set7(int j)
{
    memcpy(&buf[0][j], " ##### ",8);
    memcpy(&buf[1][j], "      # ",8);
    memcpy(&buf[2][j], "      # ",8);
    memcpy(&buf[3][j], "      # ",8);
    memcpy(&buf[4][j], "      # ",8);
    memcpy(&buf[5][j], "      # ",8);
    memcpy(&buf[6][j], "      # ",8);
}

```

// 8を設定

```

void Set8(int j)
{
    memcpy(&buf[0][j], " ##### ",8);
    memcpy(&buf[1][j], " # #    ",8);
    memcpy(&buf[2][j], " # #    ",8);
    memcpy(&buf[3][j], " ##### ",8);
    memcpy(&buf[4][j], " # #    ",8);
    memcpy(&buf[5][j], " # #    ",8);
    memcpy(&buf[6][j], " ##### ",8);
}

```

// 9を設定

```

void Set9(int j)
{
    memcpy(&buf[0][j], " ##### ",8);
    memcpy(&buf[1][j], " # #    ",8);
}

```

```

        memcpy(&buf[2][j], " # # ", 8);
        memcpy(&buf[3][j], " ##### ", 8);
        memcpy(&buf[4][j], "      # ", 8);
        memcpy(&buf[5][j], "      # ", 8);
        memcpy(&buf[6][j], " ##### ", 8);
    }

    // :を設定
    void SetC(int j)
    {
        memcpy(&buf[0][j], "      ", 8);
        memcpy(&buf[1][j], "      ", 8);
        memcpy(&buf[2][j], " #    ", 8);
        memcpy(&buf[3][j], "      ", 8);
        memcpy(&buf[4][j], " #    ", 8);
        memcpy(&buf[5][j], "      ", 8);
        memcpy(&buf[6][j], "      ", 8);
    }

```

2 練習問題 53 「C 言語」242 頁

練習問題 53

例題 53 で書き出したシーケンシャル・ファイルのデータをリードするプログラムを作りなさい。

```

/*
   rensyu53.c
*/

#include <stdio.h>
#include <process.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    FILE *fp;
    int age;
    char name[20];

    if((fp = fopen(argv[1], "r")) == NULL){
        printf("can't open %s\n", argv[1]);
        exit(1);
    }

    while(fscanf(fp, "%s %d", name, &age) != EOF){
        printf("%20s%3d\n", name, age);
    }
}

```

```
}  
  
fclose(fp);  
  
return 0;  
}
```

3 応用問題

下記は、2013年度の阪神タイガースの野手の背番号、名前、打率一覧です。このファイルを読んで、画面に出力するプログラムを作成してください。また、平均打率、最高打率、最低打率等を表示してください。

```
0 柴田 講平 0.229  
0 大和 0.273  
1 鳥谷 敬 0.282  
3 関本 賢太郎 0.279  
4 上本 博紀 0.254  
5 浅井 良 0.22  
7 西岡 剛 0.29  
8 福留 孝介 0.198  
9 マートン 0.314  
25 新井 貴浩 0.267  
31 林 威助 0  
32 新井 良太 0.238  
33 西田 直斗 0  
35 坂 克彦 0.229  
37 日高 剛 0.289  
38 黒瀬 春樹 0.231  
39 小宮山 慎二 0.167  
42 コンラッド 0.175  
45 清水 誉 0.233  
49 今成 亮太 0.265  
50 藤井 彰人 0.258  
51 伊藤 隼太 0.145  
58 荒木 郁也 0  
61 田上 健一 0.286  
62 高山 久 0.25  
65 緒方 凌介 0  
67 野原 将志 0  
68 俊介 0.292  
69 森田 一成 0.211  
99 狩野 恵輔 0.2
```