CS205 C/ C++ Programming Lab Assignment4

> Name: 黄玉安 (Huang Yu'an)
>
> SID: 11610303

Part1 - Analysis

I designed a structure to store each block's start code point and, code point as well as the corresponding name.

```
struct code_point
{
    unsigned int start;
    unsigned int end;
    int chars_num;
    string name;
};
```

I use a 300 size code_point array to store the information reading from blocks.txt. After that, the char form standard input was read and analyzed by function `check(string & line)`.

Finally , I tranversed the struct array to find the max chars_num.

Part2 -Code

Read blocks.txt.

```

void read_blockfile(){
    ifstream ifile("Blocks.txt");

    string line;
    int start, end;
    char name[100];
    while (!ifile.eof())
    {
        getline(ifile, line);
        if(line[0]=='#' || line[0]==' ')
            continue;
        int res = sscanf(line.c_str(), "%x..%x; %[^\n]", &start, &end, name);
        if(res!=3)
            continue;
        else{
            chars[curr].start = start;
            chars[curr].end = end;
            chars[curr].chars_num = 0;
            chars[curr].name = name;
            curr++;
        }
        // cout << name << endl;
    }
    ifile.close();
}
```

Analysis of one string

```
void check(string &line){

    unsigned char *p;
    int            bytes_in_char;
    int            i;
    unsigned int   codepoint;

    char* myc_string = new char[line.size()+1];
    line.copy(myc_string, line.size()+1);
    p = (unsigned char*)myc_string;
    while (*p) {
        codepoint = utf8_to_codepoint(p, &bytes_in_char);
        if (codepoint) {
            _utf8_incr(p);
            find_code_point(codepoint);
        } else {
            printf("%c Invalid UTF-8\n", *p);
            p++;  // Try the next character
        }
    }
    delete[] myc_string;
}
```

```
void find_code_point(unsigned int codepoint){
    for(int i=0;i<curr;i++){
        if(chars[i].start < codepoint && chars[i].end > codepoint){
            chars[i].chars_num ++;
        }
    }
}
```

Finally get the result

```
int found(){
    int max = 0;
    int res = 0;
    for(int i=0;i<curr;i++){
        if(chars[i].chars_num > max){
            max = chars[i].chars_num;
            res = i;
        }
    }
    return res;
}
```

You can find more details in my code.

Part 3 - Result & Verification

I wrote a makefile to test it, this is how I wrote:

```
CC=g++ -std=c++11

SRC=utf8.c check_language.cpp
TARGET=main

debug:$(SRC)
    $(CC) -g -o $(TARGET) $(SRC)


TEST_SRC=test_data
TEST_SOURCE+=$(sort $(wildcard $(TEST_SRC)/*.txt))

test:$(TARGET)
    @$(foreach var, $(TEST_SOURCE),\
        echo $(var);\
        ./$(TARGET) < $(var); \
        echo "";\
    )
```

Type command `make test` can run all the test in the test folder. The result is as following:

Case:

> test_data/sample.txt
> there comes the result, file is:Armenian
>
> test_data/sample2.txt
> there comes the result, file is:Georgian
>
> test_data/sample3.txt
> there comes the result, file is:Lao
>
> test_data/sample4.txt
> there comes the result, file is:Malayalam
>
> test_data/sample5.txt
> there comes the result, file is:Devanagari

> test_data/sample6.txt
> there comes the result, file is:Georgian

The reslut is the same as what we want.

Part 4 - Difficulties & Solutions

It is a little difficult to read file from block.txt, because I need to deal with the special format.