

H1 CS205 C/ C++ Programming - Lab Assignment 4

Name: 邱煜 (Qiu Yu)

SID: 11611127

H2 Part1 - Analysis

This program is to load the codepoint data from a file `Blocks.txt` to a structure, then use that to determine the input stream's name of the block to which most its characters belongs.

H2 Part2 - Code

```
1  //
2  //  main.cpp
3  //  lab4
4  //
5  //  Created by 邱煜 on 2019/4/12.
6  //  Copyright © 2019 邱煜. All rights reserved.
7  //
8
9  #include <cstdlib>
10 #include <iostream>
11 #include <sstream>
12 #include <fstream>
13 #include <map>
14 #include "main.hpp"
15
16 using namespace std;
17
18 int cnt = 0;
19 int maxOccurence = 0;
20 Blocks blocks[300];
21 map<int, int> occurenceMap;
22
23 int main(int argc, char **argv){
24     ifstream file ("Blocks.txt");
25     // invalid case
26     if(!file.is_open()){
27         cout << "Open file blocks.txt failed!" << endl;
28         exit(9);
```

```

29     }
30     // useless information
31     string value;
32     for (int i = 0; i < 34; i++) {
33         getline(file, value);
34     }
35     // store data into blocks
36     int numInDecimal = 0;
37     stringstream ss;
38     while(file.good() && cnt<262){
39         getline(file, value, '.');
40         if(value.length()){
41             ss << value;
42             ss >> hex >> numInDecimal;
43             ss.clear();
44             blocks[cnt].id = cnt;
45             blocks[cnt].startCode = numInDecimal;
46             // cout << cnt << "->" << value << "," <<
numInDecimal ;
47         }
48         getline(file, value, '.');
49         getline(file, value, ';');
50         if(value.length()){
51             ss << value;
52             ss >> hex >> numInDecimal;
53             ss.clear();
54             blocks[cnt].endCode = numInDecimal;
55         }
56         getline(file, value);
57         if(value.length()){
58             trim(value);
59             blocks[cnt].codeName = value;
60         }
61         cnt++;
62     }
63
64     string input;
65     char *str;
66     unsigned char *pt;
67     int lenptr;
68     int codept;
69     int readLength;
70     int codeId;
71
72     while (getline(cin, input)) {
73         str = (char *)input.data();
74         pt = (unsigned char *)str;

```

```

75         readLength = 0;
76         while(readLength < input.length()){
77             lenptr = 0;
78             codept = utf8_to_codepoint(pt, &lenptr);
79             codeId = findId(codept);
80             if(codeId != -1) {
81                 addOccurence(codeId);
82             }
83             pt += lenptr;
84             readLength += lenptr;
85         }
86     }
87
88     cout << blocks[findMaxId()].codeName << endl;
89
90     return 0;
91 }
92
93 int findId(int codepoint){
94     for (int i = 0; i < cnt; i++) {
95         if (blocks[i].startCode <= codepoint &&
96             blocks[i].endCode >= codepoint) {
97             return i;
98         }
99     }
100     return -1;
101 }
102
103 void addOccurence(int key){
104     if(occurenceMap.count(key)!=0){
105         occurenceMap[key]++;
106         maxOccurence = occurenceMap[key]>maxOccurence ?
5         occurenceMap[key] : maxOccurence;
107     }else{
108         occurenceMap[key] = 0;
109     }
110 }
111
112 int findMaxId(){
113     for(map<int, int>::iterator mapItr =
2     occurenceMap.begin(); mapItr != occurenceMap.end();
++mapItr){
114         if (mapItr->second == maxOccurence) {
115             return mapItr->first;
116         }
117     }
118     return -1;

```

```

17 }
18
19 void trim(string &str){
20     if(!str.empty()){
21         str.erase(0, str.find_first_not_of(" "));
22         str.erase(str.find_last_not_of(" ") + 1);
23     }
24 }
25
26 unsigned int utf8_to_codepoint(const unsigned char *u, int
27 *lenptr) {
28     // Returns 0 if something goes wrong
29     // Passes back the length
30     unsigned int cp = 0;
31
32     *lenptr = 0;
33     if (u) {
34         if (*u < 0xc0) {
35             cp = (unsigned int)*u;
36             *lenptr = 1;
37         } else {
38             *lenptr = isutf8(u);
39             if (*lenptr == 0) {
40                 return 0;
41             }
42             switch (*lenptr) {
43                 case 2:
44                     cp = (u[0] - 192) * 64 + u[1] - 128;
45                     break;
46                 case 3:
47                     cp = (u[0] - 224) * 4096
48                         + (u[1] - 128) * 64 + u[2] - 128;
49                     break;
50                 default:
51                     cp = (u[0] - 240) * 262144
52                         + (u[1] - 128) * 4096
53                         + (u[2] - 128) * 64 + u[3] - 128;
54                     break;
55             }
56         }
57     }
58     return cp;
59 }
60
61 int isutf8(const unsigned char *u) {
62     // Validate utf8 character.
63     // Returns the length, 0 if invalid.

```

```

18     int len = 0;
19
15     if (u) {
16         if (*u < 0xc0) {
17             len = 1;
18         } else {
19             if ((*u & 0xe0) == 0xc0) {
20                 // U-00000080 - U-000007FF : 110xxxxx
21                 1 10xxxxxx
22                 len = 2;
23             } else if ((*u & 0xf0) == 0xe0) {
24                 // U-00000800 - U-0000FFFF : 1110xxxx
25                 4 10xxxxxx 10xxxxxx
26                 len = 3;
27             } else if ((*u & 0xf8) == 0xf0) {
28                 // U-00010000 - U-001FFFFF : 11110xxx
29                 7 10xxxxxx 10xxxxxx 10xxxxxx
30                 len = 4;
31             } else {
32                 // malformed UTF-8 character
33                 return 0;
34             }
35             // Check that the UTF-8 character is OK
36             int i;
37             for (i = 1; i < len; i++) {
38                 if ((u[i] & 0xc0) != 0x80) {
39                     return 0;
40                 }
41             }
42         }
43     }
44     return len;
45 }

```

```

1 //
2 // main.hpp
3 // lab4
4 //
5 // Created by 邱煜 on 2019/4/12.
6 // Copyright © 2019 邱煜. All rights reserved.
7 //
8
9 #ifndef main_hpp
10 #define main_hpp
11

```

```

12 struct Blocks {
13     int id;
14     int startCode;
15     int endCode;
16     std::string codeName;
17 };
18
19
20 // return the id of the block of the corresponding
    codepoint
21 // return -1 if not found
22 int findId(int codepoint);
23
24 // trim the spaces in the starting and ending of the str
25 void trim(std::string &str);
26
27 // set the value of the key +1
28 // if this key not exist, add this key add set its value to
    1
29 void addOccurence(int key);
30
31 // find the code block index which has the maximum
    occurence
32 int findMaxId(void);
33
34 // utf8_to_codepoint returns 0 if conversion fails. If it
    succeeds,
35 // the value pointed by lenptr is set to the number of
    bytes of the
36 // UTF-8 character
37 unsigned int utf8_to_codepoint(const unsigned char *u, int
    *lenptr);
38
39 // Returns the length in bytes, 0 if invalid
40 int isutf8(const unsigned char *u);
41 #endif /* main_hpp */

```

H2 Part 3 - Result & Verification

H3 Test case

Test case #1:

1 Input :

```
2 1. ./main < samples/sample.txt
3 2. ./main < samples/sample2.txt
4 3. ./main < samples/sample3.txt
5 4. ./main < samples/sample4.txt
6 5. ./main < samples/sample5.txt
7 6. ./main < samples/sample6.txt
8
9 Output:
10 1. Armenian
11 2. Georgian
12 3. Lao
13 4. Malayalam
14 5. Devanagari
15 6. Georgian
16
17 Verification:
18 1. Armenian
19 2. Georgian
20 3. Lao
21 4. Malayalam
22 5. Devanagari
23 6. Georgian
```

```
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample.txt
Armenian
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample2.txt
Georgian
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample3.txt
Lao
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample4.txt
Malayalam
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample5.txt
Devanagari
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ ./main < samples/sample6.txt
Georgian
qiuy@wind-SYS-4028GR-TR-Invalid-entry-length-16-Fixed-up-to-11:~/c/lab4$ █
```

H2 Part 4 - Difficulties & Solutions

1. The given files are c files. I copied the functions I used to my own files.