```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <ctype.h>
5 #include <windows.h>
6
7 //#define DUMP
8
9 char toescape(char c)
10 {
11     switch(c) {
12     case '7': return '\7';
13     case '(':  return '(';
14     case ')':  return ')';
15     case 't':  return '\t';
16     case 'n':  return '\n';
17     case 'r':  return '\r';
18     case '\\': return '\\';
19     case '\"': return '\"';
20     }
21     return c;
22 }
23
24 char todigit(char c)
25 {
26     char d = toupper(c);
27     if(d >= '0' && d <= '9') d = d - '0';
28     else d = d - 'A' + 10;
29     return d;
30 }
31
32 int isescape(char c)
33 {
34     if(c == '(' || c == ')' || c == 't' || c == 'n' || c == 'r' || c == '\\' || c == '\"')
    return 1;
35     return 0;
36 }
37
38 int findString(char* str, char* find)
39 {
40     for(unsigned int i=0; i<strlen(str); i++){
41         if(!strncmp(str+i,find,strlen(find))) return i;
42     }
43     return 0;
44 }
45
46 void dump(unsigned char* str, int sn)
47 {
48     for(int i=0; i<sn; i++){ printf("%02x, ",str[i]); }
49     printf(" : %s \n",str);
50 }
51
52 #define BOM ((wchar_t)0xFEFF) // UTF-16 BOM
53
```



```
54 int isEnglish(char* str, int sn)
55 {
56     for(int i=0; i<sn; i++){ if(!isascii(str[i])) return 0; }
57     return 1;
58 }
59
60 void wcs2sjis(char* str, char* sjis, int sn)
61 {
62     if(isEnglish(str,sn)){
63         memset(sjis,0,sizeof(sjis));
64         strncpy(sjis,str,strlen(str)+1);
65         return;
66     }
67     wchar_t string[2048];
68     wchar_t *dest = string;
69     for(int i=0; i<sn; i++){
70         wchar_t wc = (wchar_t)(((unsigned char)str[i] << 8) | (unsigned char)str[i+1]);
71         if(wc != BOM) *dest++ = wc;
72         i++;
73     }
74     *dest = L'\0';
75     memset(sjis,0,sizeof(sjis));
76     int nLen = ::WideCharToMultiByte(CP_ACP, 0, string, -1, NULL, 0, NULL, NULL );
77     ::WideCharToMultiByte(CP_ACP, 0, string, -1, sjis, nLen, NULL, NULL );
78 }
79
80
81 int getPDFString(char* dst, char* src, int si, char* find)
82 {
83     int b_esc = 0,ci = 0;
84     int b_start = 0;
85     memset(dst,0,sizeof(dst));
86     for(unsigned int i=si+strlen(find); i<strlen(src); i++){ if(src[i] != ' ') { si = i;
  break; } }
87     for(unsigned int i=si; i<strlen(src); i++){
88         if(src[i] == '(') { b_start = 1; si = i+1; break; }
89         else if(src[i] == '<') { b_start = 2; si = i+1; break; }
90         else if(src[i] != '(' && src[i] != '<') { si = i; break; }
91
92     }
93     int iskip = (b_start == 2) ? 2 : 1;
94     for(unsigned int i=si; i<strlen(src); i+=iskip){
95         if(!b_esc && src[i] == '\\'){ b_esc = 1; continue; }
96         if(b_esc){
97             b_esc = 0;
98             if(isescape(src[i])){
99                 dst[ci++] = toescape(src[i]);
100 #ifdef DUMP
101                 printf("[\\%c], ",src[i]);
102 #endif
103                 continue; }
104             unsigned int ii, di = 0;
105             char dv[10]; memset(dv,0,sizeof(dv));
106             int b_flag = 0;
```

```
107             for(ii=i; b_flag != 1; ii++){
108                 if(!(src[ii] >= '0' && src[ii] <= '9')) { b_flag = 1; continue; }
109                 if(ii == i+3){ b_flag = 1; continue; }
110                 dv[di++] = src[ii];
111             }
112             char *es;
113             int base8 = strtol(dv,&es,8);
114             dst[ci++] = base8;
115             printf("[\\%o], ",base8);
116             i = ii-2;
117             continue;
118         }
119         if(src[i] == ')' && b_start == 1) break;
120         if(src[i] == '>' && b_start == 2) break;
121         if((src[i] == ' ' || src[i] == '/') && b_start == 0) break;
122         if(b_start == 2) {
123             if(src[i] == '0' && src[i] == '0') { dst[ci++] = 0; printf("0, "); continue;
    }
124             dst[ci++] = (todigit(src[i])<<4) + todigit(src[i+1]);
125 #ifdef DUMP
126             printf("([%d,%02x,%c]: %c%c), ",(unsigned char)dst[ci-1],(unsigned char)dst[
    ci-1],dst[ci-1],src[i],src[i+1]);
127 #endif
128         } else {
129             dst[ci++] = src[i];
130 #ifdef DUMP
131             printf("%c[%02x], ",src[i],(unsigned char)src[i]);
132 #endif
133         }
134     }
135 #ifdef DUMP
136     printf("\n");
137 #endif
138     return ci;
139 }
140
141 int main()
142 {
143     const int BUFSIZE = 2048;
144     FILE* fp = fopen("aaa.pdf","rb");
145     char readbuf[BUFSIZE];
146     if(fp == NULL) return 0;
147     int count = 0;
148     int b_title = 0, b_author = 0, b_count = 0;
149     char title[BUFSIZE],author[BUFSIZE],scount[BUFSIZE];
150     char title_s[BUFSIZE],author_s[BUFSIZE],scount_s[BUFSIZE];
151     memset(readbuf,0,sizeof(readbuf));
152     memset(title,0,sizeof(title));
153     memset(author,0,sizeof(author));
154     memset(scount,0,sizeof(scount));
155     memset(title_s,0,sizeof(title));
156     memset(author_s,0,sizeof(author));
157     memset(scount_s,0,sizeof(scount_s));
158
```



```
159     while(fgets(readbuf,BUFSIZE,fp)){
160         int si = 0;
161         if((si = findString(readbuf,"/Title")) > 0 && b_title == 0){
162             int cn = getPDFString(title,readbuf,si,"/Title");
163             wcs2sjis(title,title_s,cn);
164 #ifdef DUMP
165             printf("title: "); dump((unsigned char*)title,cn);
166 #endif
167             b_title = 1;
168         }
169         if((si = findString(readbuf,"/Author")) > 0 && b_author == 0){
170             int cn = getPDFString(author,readbuf,si,"/Author");
171             wcs2sjis(author,author_s,cn);
172 #ifdef DUMP
173             printf("author: "); dump((unsigned char*)author,cn);
174 #endif
175             b_author = 1;
176         }
177         if((si = findString(readbuf,"/N ")) > 0 && b_count == 0){
178             int cn = getPDFString(scount,readbuf,si,"/N ");
179             if(strlen(scount) < 4){
180 #ifdef DUMP
181                 printf("count: "); dump((unsigned char*)scount,cn);
182 #endif
183                 count = atoi(scount);
184                 b_count = 1;
185             }
186         }
187         if((si = findString(readbuf,"/Count")) > 0 && b_count == 0){
188             int cn = getPDFString(scount,readbuf,si,"/Count");
189 #ifdef DUMP
190             printf("count: "); dump((unsigned char*)scount,cn);
191 #endif
192             count = atoi(scount);
193             b_count = 1;
194         }
195         if(b_title && b_author && b_count) break;
196         memset(readbuf,0,sizeof(readbuf));
197     }
198     fclose(fp);
199
200     if(b_title) printf("title is %s\n",title_s);
201     if(b_author) printf("author is %s\n",author_s);
202     if(b_count) printf("count is %d\n",count);
203
204     return 0;
205 }
```