

2021 年度 プログラミング III 第 8～11 回 レポート 書籍管理システム

学籍番号：32114153

WEN Chenhang

2021 年 12 月 20 日

1 はじめに

このレポートは、第 8～11 回の演習課題で作成した書籍管理システムについて報告する。今回のプログラムを正確に実行するために、プログラムのルートディレクトリで行う。また、分かりやすいため、ファイルの対応は以下になる。

task8.c — LibrarySystem.c

task8 — LibrarySystem

task8.dat — bookInfo.dat

なお、本レポートで表れた書籍は、ダミーデータであり、現実世界に関係ない。

2 システム紹介

本システムは、主に七つの機能より、構築されている。初回実行時に、ルートディレクトリに”bookInfo.txt”ファイルが生成される。二回以降の実行は、”bookInfo.txt”ファイルより、書籍の情報を読み込む。

2.1 ナビゲーションバー

システムが起動されると、ナビゲーションバーが表示される。指示にしたがって、動作番号を入力することで、各機能を実行できる。

書籍管理システム

0. システムログアウト

1. 書籍登録

2. 書籍削除

3. 書籍閲覧

4. 検索

5. 貸出

6. 返還

動作番号 (0-6) を入力してください:

図 1: ナビゲーションバー

2.2 システムログアウト

番号 0 を入力し、システムログアウトの画面に入り、ログアウトの機能が実行される。この操作より、システムからログアウトできる。

```
[-----「システムログアウト」-----  
ログアウトできました！
```

図 2: システムログアウト画面

2.3 書籍登録

書籍登録の機能より、新規書籍をシステムに登録することができる。

書籍登録の画面に入り、指示にしたがって新規書籍の情報を入力する。具体的な実行結果は、図 3 に示す。

登録された書籍は、直ちに”bookInfo.txt” ファイルに保存される。

登録完了後、指示にしたがってメインメニューに戻ることができる。また、そのまま終了することもできる。

```
[-----「書籍登録」-----  
書籍 ID(数字): 0  
書名: C言語の難しさ  
著者: WEN  
登録完了!  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 3: 書籍登録の実行

2.4 書籍の削除

登録された書籍を削除したい場合は、書籍削除機能を使う。

削除したい書籍の ID を指定し、削除操作を行う。具体的な操作は図 4 に示す。削除が行われるたびに、ファイルへの保存も行われる。

```
[-----「書籍削除」-----  
削除の書籍 ID: 0  
削除完了!  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 4: 書籍削除の実行

2.5 書籍閲覧

書籍閲覧機能より、現時点に登録されている書籍を確認できる。

ここで、貸出可能のフィールドが示しているのは、書籍の貸出可能の状態である。1 の場合は、貸出可能の状態であるに対して、0 の場合は既に貸出されており、貸出不可の状態である。

```
[-----「書籍閲覧」-----  
id      図書      著者      貸出可能  
  
0        C言語の難しさ    WEN        1  
1        日本語の難しさ    CHENHANG    1  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 5: 書籍閲覧の実行

2.6 検索

検索機能は、書名に基づいて検索を行う。検索の画面で、書名を入力する。該当の書籍が登録されている場合は、その書籍の情報を印刷する。詳細は、図 6 に示す。

```
[-----「検索」-----  
検索の書名: C言語の難しさ  
id      図書      著者      貸出可能  
  
0        C言語の難しさ    WEN        1  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 6: 書籍検索

また、該当の書籍が登録されていない場合は、「該当の書籍は見つかりませんでした」を返す。

```
[-----「検索」-----  
検索の書名: JAVAの難しさ  
該当の書籍は見つかりませんでした。  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 7: 書籍検索（検索失敗）

2.7 貸出

貸出機能で、書籍を貸出することができる。貸出したい書籍の id を入力することで、該当の書籍の貸出を行われる。具体的な操作は図 8 に示す。

```
[-----「貸出」-----  
貸出書籍のid: 0  
貸出完了!  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 8: 書籍閲覧の貸出

貸出の実行によって、該当の書籍の貸出可能の状態が変わる。図 9 に示すように、貸出可能の状態が 1 より 0 に変わった。つまり、貸出不可になっている。

```
[-----「書籍閲覧」-----  
id      図書      著者      貸出可能  
  
0        C言語の難しさ    WEN        0  
1        日本語の難しさ    CHENHANG    1  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 9: 書籍が貸出不可になる

また、貸出不可の書籍、つまり既に貸出されている書籍を貸出しようとする、図 10 に示すように、「該当の書籍は既に貸出されています」と返されます。

```
[-----「貸出」-----  
貸出書籍のid: 0  
該当の書籍は既に貸出されています。  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 10: 貸出失敗

2.8 返還

貸出中の書籍を返還したい場合は、返還機能を使う。貸出の操作とほぼ同じになっている。流れは、図 11 に示す。

```
[-----「返還」-----  
貸出書籍のid: 0  
返還完了!  
メインメニューに戻りますか? /yes...1 no...0: █
```

図 11: 書籍返還

3 プログラム

3.1 コンパイル

今回の課題は、標準ライブラリを使っている。よって、ルートディレクトリで以下のコマンドにより、コンパイルできる。

```
gcc -o LibrarySystem ./LibrarySystem.c
```

また、プログラムの実行は、ルートディレクトリで以下のコマンドの実行より、行われる。

```
./LibrarySystem
```

3.2 制御構文の利用

今回のプログラムに、制御構文がたくさん含まれている。以下は、メインメソッドにある一例を示す。この制御構文によって、プログラムの一回の実行で何回もメインメニューに戻ることができ、操作を繰り返すことができる。

Listing 1: 制御構文の例

```
1      int FLAGE = 1;
2      while (FLAGE)
3      {
4          makeNav();
5          interaction();
6          pauseContro(&FLAGE);
7      }
```

3.3 関数 (自作) の利用

プログラムを簡潔にできるように、機能をラップして関数として使われる。以下に示しているのは、メインメニューを印刷する関数である。

Listing 2: 関数 (自作) の利用

```
1      // Navigation Bar
2      void makeNav()
3      {
4          puts("-----\n");
5          puts("書籍管理システム\n");
6          puts("\t0. □ システムログアウト\n");
7          puts("\t1. □ 書籍登録\n");
8          puts("\t2. □ 書籍削除\n");
9          puts("\t3. □ 書籍閲覧\n");
10         puts("\t4. □ 検索\n");
11         puts("\t5. □ 貸出\n");
```

```
12         puts("\t6.  返還\n");
13         puts("-----\n");
14         printf("動作番号を入力してください(0-6): ");
15     }
```

3.4 ポインタの利用

3.2 節で既に表示されているよう、while の制御は関数 `pauseContro()` によって行なっている。確実に変数 `FLAG` の値を変えるために、変数 `FLAG` のポインタを渡している。関数 `pauseContro()` の中の操作は、以下に示す。

Listing 3: ポインタの利用

```
1     // Pause
2     void pauseContro(int *FLAG)
3     {
4         int temp = 0;
5         printf("メインメニューに戻りますか? /yes...1 no...0: ");
6         scanf("%d", &temp);
7         clear();
8         *FLAG = temp;
9     }
```

3.5 構造体の利用

構造体は、書籍と Linked List の情報に使われている。以下は、構造体を使って書籍の情報を格納する例。

Listing 4: 構造体の利用

```
1     // Book information struct
2     struct bookInfo
3     {
4         int id;
5         char title[64];
6         char author[64];
7         int enable; // can be borrowed? or not?
8     };
```

3.6 ファイル操作

ファイルの操作は、書籍情報の読み込みおよび保存に使われている。その操作の関数は、以下に示す。

Listing 5: ファイル操作

```
1 // save the information of books to file
2 void saveInfoToFile(const char *fileName, struct Node *headNode)
3 {
4     FILE *fp = fopen(fileName, "w");
5     struct Node *pMove = headNode->next;
6     while (pMove)
7     {
8         fprintf(fp, "%d\t%s\t%s\t%d\t\n", pMove->data.id,
9             pMove->data.title, pMove->data.author,
10             pMove->data.enable);
11         pMove = pMove->next;
12     }
13     fclose(fp);
14 }
15
16 // read the information of books from file
17 void readInfoFromFile(const char *fileName, struct Node *headNode)
18 {
19     FILE *fp = fopen(fileName, "r");
20     // if the file does not exist, creat one
21     if (fp == NULL)
22     {
23         fp = fopen(fileName, "w+");
24     }
25     struct bookInfo temp;
26     while (fscanf(fp, "%d\t%s\t%s\t%d\t\n", &temp.id,
27         temp.title, temp.author, &temp.enable) != EOF)
28     {
29         insertNodeByTail(headNode, temp);
30     }
31
32     fclose(fp);
33 }
```

4 感想

4.1 今回の課題

今回の課題は、今まで学んだことを活かして、書籍管理システムを実現できた。作成の途中は、いろいろ苦労したが、楽しかった。また、最後の出来上がりの瞬間は、物凄く達成感を感じられた。

その上に、今回の課題を通して、プログラミング能力をある程度身につけたと思う。特にポインタへの理解も深まった。

4.2 今後の課題

作成されたシステムは、まだ不完全なところがある。例えば、データの変動のたびに、全てのデータに対して、書き込みが行われる。とても効率が悪いとも言える。

今後、もし機会や余裕があれば、システムを改善してみたいと思う。

5 プログラム全体図

書籍管理システムの全体を渡って、330 行もある。プログラム全体の詳細は以下に示す。組み版の問題で、見づらいところもある。

Listing 6: プログラムの全体

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define TRUE 1 // boolean value true
6  #define FALSE 0 // boolean value false
7  #define clear() printf("\033[H\033[J") // clear the screen
8  #define FILENAME "./bookInfo.dat"
9
10 struct Node *list;
11
12 // Book information struct
13 struct bookInfo
14 {
15     int id;
16     char title[64];
17     char author[64];
18     int enable; // can be borrowed? or not?
19 };
20
21 // Linked list node struct
22 struct Node
23 {
24     struct bookInfo data;
25     struct Node *next;
26 };
27
28 // method: create the head of node
```

```
29     struct Node *createHead()
30     {
31         // dynamic application of memory space
32         struct Node *headNode = (struct Node *)malloc(sizeof(struct Node))
33         // initialization
34         headNode->next = NULL;
35         return headNode;
36     };
37
38     // method: create the node to save the data
39     struct Node *createNode(struct bookInfo data)
40     {
41         struct Node *newNode = (struct Node *)malloc(sizeof(struct Node))
42         newNode->data = data;
43         newNode->next = NULL;
44         return newNode;
45     }
46
47     // insert a new node after the first element
48     void insertNodeByHead(struct Node *headNode, struct bookInfo data)
49     {
50         struct Node *newNode = createNode(data);
51         newNode->next = headNode->next;
52         headNode->next = newNode;
53     }
54
55     // insert a new node after the last element
56     void insertNodeByTail(struct Node *headNode, struct bookInfo data)
57     {
58         struct Node *pMove = headNode;
59         while (pMove->next)
60         {
61             pMove = pMove->next;
62         }
63         struct Node *newNode = createNode(data);
64         pMove->next = newNode;
65     }
66
67     // delete the node
68     int deletaNodeById(struct Node *headNode, int id)
69     {
70         struct Node *desLeftNode = headNode;
71         struct Node *desNode = headNode->next;
```

```
72     while (desNode != NULL && (desNode->data).id != id)
73     {
74         desLeftNode = desNode;
75         desNode = desLeftNode->next;
76     }
77     if (desNode == NULL)
78     {
79         return FALSE;
80     }
81     else
82     {
83         desLeftNode->next = desNode->next;
84         free(desNode);
85         desNode = NULL;
86         return TRUE;
87     }
88 }
89
90 // print the Linked list
91 void printList(struct Node *headNode)
92 {
93     struct Node *pMove = headNode->next;
94     puts("id\図書t\著者t\貸出可能t\n");
95     while (pMove)
96     {
97         printf("%d\t%s\t%s\t%d\t\n", pMove->data.id, pMove->data.title, pMove->data.author, pMove->data.available);
98         pMove = pMove->next;
99     }
100 }
101
102 // save the information of books to file
103 void saveInfoToFile(const char *fileName, struct Node *headNode)
104 {
105     FILE *fp = fopen(fileName, "w");
106     struct Node *pMove = headNode->next;
107     while (pMove)
108     {
109         fprintf(fp, "%d\t%s\t%s\t%d\t\n", pMove->data.id, pMove->data.title, pMove->data.author, pMove->data.available);
110         pMove = pMove->next;
111     }
112     fclose(fp);
113 }
114
```

```
115     // read the information of books from file
116     void readInfoFromFile(const char *fileName, struct Node *headNode)
117     {
118         FILE *fp = fopen(fileName, "r");
119         // if the file does not exist, creat one
120         if (fp == NULL)
121         {
122             fp = fopen(fileName, "w+");
123         }
124         struct bookInfo temp;
125         while (fscanf(fp, "%d\t%s\t%s\t%d\t\n", &temp.id, temp.title, temp.author, &temp.price))
126         {
127             insertNodeByTail(headNode, temp);
128         }
129
130         fclose(fp);
131     }
132
133     // find the book by name
134     struct Node *findByTitle(struct Node *headNode, char *title)
135     {
136         struct Node *desNode = headNode->next;
137         while (desNode != NULL && strcmp(desNode->data.title, title))
138         {
139             desNode = desNode->next;
140         }
141         return desNode;
142     }
143
144     // find the book by id
145     struct Node *findById(struct Node *headNode, int id)
146     {
147         struct Node *desNode = headNode->next;
148         while (desNode != NULL && desNode->data.id != id)
149         {
150             desNode = desNode->next;
151         }
152         return desNode;
153     }
154
155     // 書籍登録メソッド
156     void bookRegister(struct Node *headNode)
157     {
```

```
158         struct bookInfo temp;
159         printf("書籍ID数字(): ");
160         scanf("%d", &temp.id);
161         printf("書名: ");
162         scanf("%s", temp.title);
163         printf("著者: ");
164         scanf("%s", temp.author);
165         temp.enable = 1;
166         insertNodeByTail(headNode, temp);
167         saveInfoToFile(FILENAME, headNode);
168         puts("登録完了!");
169     }
170
171     //書籍削除メソッド
172     void bookDelete(struct Node *headNode)
173     {
174         int id;
175         printf("削除の書籍ID: ");
176         scanf("%d", &id);
177         deleteNodeById(headNode, id);
178         saveInfoToFile(FILENAME, headNode);
179         puts("削除完了!");
180     }
181
182     //検索メソッド
183     void findBook(struct Node *headNode)
184     {
185         struct Node *result;
186         struct bookInfo temp;
187         printf("検索の書名: ");
188         scanf("%s", temp.title);
189         result = findByTitle(headNode, temp.title);
190         if (!result)
191         {
192             puts("該当の書籍は見つかりませんでした。");
193         }
194         else
195         {
196             puts("id\\図書t\\著者t\\貸出可能t\\n");
197             printf("%d\\t%s\\t%s\\t%d\\t\\n", result->data.id, result->data.t
198         }
199     }
200
```

```
201 //貸出メソッド
202 void borrowBook(struct Node *headNode)
203 {
204     struct Node *desBook;
205     int id;
206     printf("貸出書籍のid:");
207     scanf("%d", &id);
208     desBook = findById(headNode, id);
209     if (!desBook)
210     {
211         puts("該当の書籍は見つかりませんでした。");
212     }else if(desBook->data.enable == 0) {
213         puts("該当の書籍は既に貸出されています。");
214     } else {
215         desBook->data.enable = 0;
216         saveInfoToFile(FILENAME, headNode);
217         puts("貸出完了!");
218     }
219 }
220
221 //返還メソッド
222 void returnBook(struct Node *headNode)
223 {
224     struct Node *desBook;
225     int id;
226     printf("貸出書籍のid:");
227     scanf("%d", &id);
228     desBook = findById(headNode, id);
229     if (desBook)
230     {
231         desBook->data.enable = 1;
232         saveInfoToFile(FILENAME, headNode);
233     }
234
235     puts("返還完了!");
236 }
237
238 // Navigation Bar
239 void makeNav()
240 {
241     puts("-----\n");
242     puts("書籍管理システム\n");
243     puts("\t0. システムログアウト\n");
```

```
244         puts("\t1. 書籍登録\n");
245         puts("\t2. 書籍削除\n");
246         puts("\t3. 書籍閲覧\n");
247         puts("\t4. 検索\n");
248         puts("\t5. 貸出\n");
249         puts("\t6. 返還\n");
250         puts("-----\n");
251         printf("動作番号を入力してください(0-6): ");
252     }
253
254     // Interaction
255     void interaction()
256     {
257         int actKey = 0;
258         scanf("%d", &actKey);
259         clear();
260         switch (actKey)
261         {
262             case 0:
263                 puts("「システムログアウト」 -----");
264                 puts("ログアウトできました!");
265                 exit(0);
266                 break;
267
268             case 1:
269                 puts("「書籍登録」 -----");
270                 bookRegister(list);
271                 break;
272
273             case 2:
274                 puts("「書籍削除」 -----");
275                 bookDelete(list);
276                 break;
277
278             case 3:
279                 puts("「書籍閲覧」 -----");
280                 printList(list);
281                 break;
282
283             case 4:
284                 puts("「検索」 -----");
285                 findBook(list);
286                 break;
```

```
287
288         case 5:
289             puts("「貸出」 -----");
290             borrowBook(list);
291             break;
292
293         case 6:
294             puts("「返還」 -----");
295             returnBook(list);
296             break;
297
298         default:
299             puts("入力番号エラー: 正しい番号を入力してください\n");
300             break;
301     }
302 }
303
304 // Pause
305 void pauseContro(int *FLAGE)
306 {
307     int temp = 0;
308     printf("メインメニューに戻りますか? /yes...1 no...0: ");
309     scanf("%d", &temp);
310     clear();
311     *FLAGE = temp;
312 }
313
314 int main()
315 {
316     // create the list of books
317     list = createHead();
318     // read the info from file
319     readInfoFromFile(FILENAME, list);
320     int FLAGE = 1;
321     while (FLAGE)
322     {
323         makeNav();
324         interaction();
325         pauseContro(&FLAGE);
326     }
327
328     return 0;
329 }nable; // can be borrowed? or not?
```


330	};
-----	----