《数据结构与算法》

课程设计

|  |  |
| --- | --- |
| 项目名称： | 图书馆书籍分类 |
| 学 号： | 2017401320 |
| 姓 名： | 周文龙 |
| 班 级： | 17级5班 |
| 日 期： | 2018年12月28日 |

1. 项目概览
   1. 总体说明

|  |
| --- |
| 该系统主要是利用某本书的一些信息，通过其中的任意一个信息，找出我们所需要的查找的该本书的所有信息，采用基数排序法对一组具有结构特点的书籍编号进行排序，利用二分查找法对排序好的书籍记录按书籍编号实现快速查找，并按其他关键字的查找可以采用最简单的顺序查找方法进行。 |

* 1. 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 功能名称 | 功能描述 | 实现函数原型 |
| 1 | 程序主函数 | 系统入口，监控程序运行 | int main() |
| 2 | 系统界面要求 | 接收键盘输入，根据跳转相应界面或输出错误信息 | void menu() |
| 3 | 书籍添加 | 根据键盘输入建立书籍信息等 | BookList \*add\_book(BookList \*BL) |
| 4 | 书籍修改 | 可以根据书籍名称进行模糊查询的操作 | BookList \*modify\_book(BookList \*BL) |
| 5 | 书籍删除 | 可以根据书籍编号删除书籍 | BookList \*del\_book(BookList \*BL) |
| 6 | 书籍查找 | 根据书籍编号、作者查找、出版社、出版日期 | void search\_book\_num(BookList \*BL) |
| 7 | 文件操作功能 | 使用文件保存信息，需要时读取 | fopen() fclose() fscanf() fprintf() |
| 8 | 退出功能 | 在分支服务界面选择退出时返回上一级界面，在主界面退出时直接退出程序 | if() |

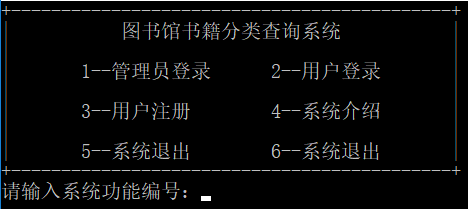
1. 主要函数说明

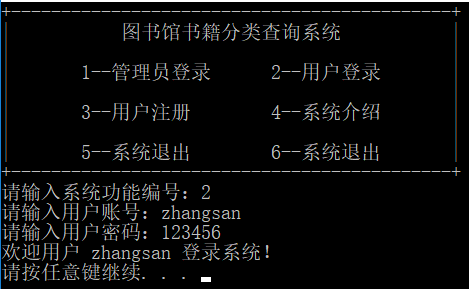
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 主要算法说明 | 涉及的课程知识 | 该设计的优点 | 该设计的缺点 |
| 1 | 基数排序，对一组具有结构特点的书籍编号进行排序 | 链表、数组 | 排序速度快 | 仅仅限于整数排序 |
| 2 | 二分查找，对排序好的书籍记录按书籍编号实现快速查找 | 数组 | 查找速度快 | 不适用于数据量小的查找 |
|  |  |  |  |  |
|  |  |  |  |  |

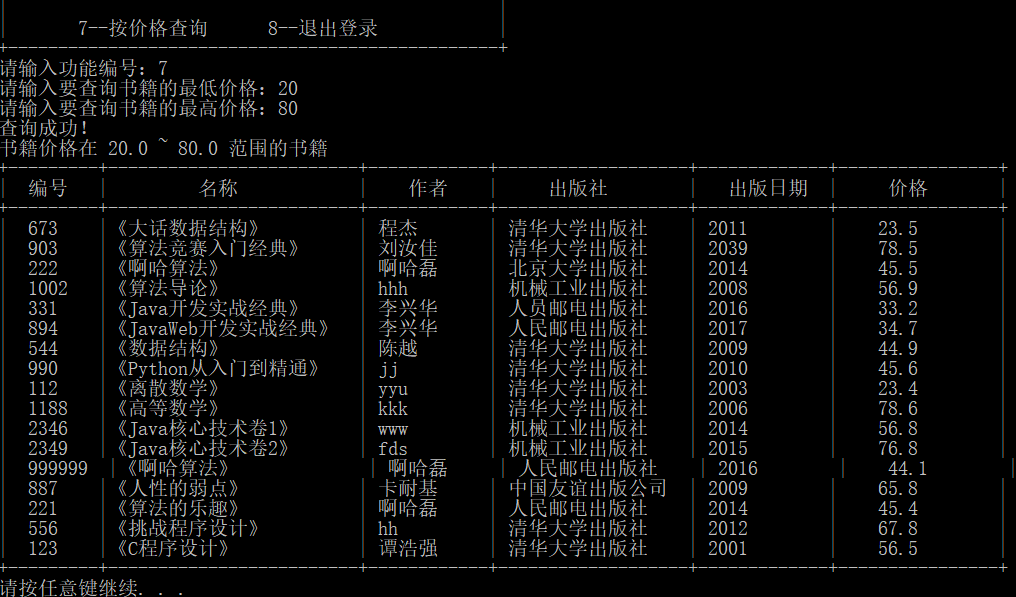
1. 项目代码

|  |
| --- |
| 1. #include<stdio.h> 2. #include<stdlib.h> 3. #include<string.h> 4. //管理员 5. typedef struct 6. { 7. char admin\_name[30]; 8. char admin\_pass[30]; 9. }Admin; 10. typedef struct Node1 11. { 12. Admin admin; 13. struct Node1 \*next; 14. }AdminList; 15. //用户 16. typedef struct 17. { 18. char user\_name[30]; 19. char user\_pass[30]; 20. }User; 21. typedef struct Node2 22. { 23. User user; 24. struct Node2 \*next; 25. }UserList; 26. //图书 27. typedef struct 28. { 29. int num; 30. char name[30]; 31. char author[30]; 32. char publish[30]; 33. char date[30]; 34. double price; 35. }Book; 36. typedef struct Node3 37. { 38. Book book; 39. struct Node3 \*next; 40. }BookList; 41. AdminList \*admin\_list; 42. UserList \*user\_list; 43. BookList \*book\_list; 44. void main\_menu() 45. { 46. printf("+--------------------------------------------+\n"); 47. printf("| 图书馆书籍分类查询系统 |\n"); 48. printf("| |\n"); 49. printf("| 1--管理员登录 2--用户登录 |\n"); 50. printf("| |\n"); 51. printf("| 3--用户注册 4--系统介绍 |\n"); 52. printf("| |\n"); 53. printf("| 5--系统退出 6--系统退出 |\n"); 54. printf("+--------------------------------------------+\n"); 55. } 56. void admin\_menu() 57. { 58. printf("+--------------------------------------------+\n"); 59. printf("| 管理员界面 |\n"); 60. printf("| |\n"); 61. printf("| 1--书籍管理 2--用户管理 |\n"); 62. printf("| |\n"); 63. printf("| 3--管理员显示 4--退出登录 |\n"); 64. printf("+--------------------------------------------+\n"); 65. } 66. void admin\_book\_menu() 67. { 68. printf("+--------------------------------------------+\n"); 69. printf("| 书籍管理界面 |\n"); 70. printf("| |\n"); 71. printf("| 1--书籍增加 2--书籍删除 |\n"); 72. printf("| |\n"); 73. printf("| 3--书籍查询 4--书籍修改 |\n"); 74. printf("| |\n"); 75. printf("| 5--书籍显示 6--退回上一级 |\n"); 76. printf("+--------------------------------------------+\n"); 77. } 78. void admin\_book\_search\_menu() 79. { 80. printf("+-------------------------------------------------+\n"); 81. printf("| 书籍查询界面 |\n"); 82. printf("| |\n"); 83. printf("| 1--书籍显示 2--按书籍编号查询 |\n"); 84. printf("| |\n"); 85. printf("| 3--按书名查询 4--按书籍作者查询 |\n"); 86. printf("| |\n"); 87. printf("| 5--按出版社查询 6--按出版日期查询 |\n"); 88. printf("| |\n"); 89. printf("| 7--按价格查询 8--退回上一级 |\n"); 90. printf("+-------------------------------------------------+\n"); 91. } 92. void admin\_book\_modify\_menu() 93. { 94. printf("+-----------------------------------------------------+\n"); 95. printf("| 书籍修改界面 |\n"); 96. printf("| |\n"); 97. printf("| 1--修改书籍编号 2--修改书籍名称 |\n"); 98. printf("| |\n"); 99. printf("| 3--修改书籍作者 4--修改出版社名称 |\n"); 100. printf("| |\n"); 101. printf("| 5--修改出版日期 6--修改书籍价格 |\n"); 102. printf("+-----------------------------------------------------+\n"); 103. } 104. void admin\_user\_menu() 105. { 106. printf("+--------------------------------------------+\n"); 107. printf("| 用户管理界面 |\n"); 108. printf("| |\n"); 109. printf("| 1--删除用户 2--修改用户 |\n"); 110. printf("| |\n"); 111. printf("| 3--查询用户 4--显示所有用户 |\n"); 112. printf("| |\n"); 113. printf("| 5--退回上一级 6--退回上一级 |\n"); 114. printf("+--------------------------------------------+\n"); 115. } 116. void user\_menu() 117. { 118. printf("+-------------------------------------------------+\n"); 119. printf("| 用户界面 |\n"); 120. printf("| |\n"); 121. printf("| 1--书籍显示 2--按书籍编号查询 |\n"); 122. printf("| |\n"); 123. printf("| 3--按书名查询 4--按书籍作者查询 |\n"); 124. printf("| |\n"); 125. printf("| 5--按出版社查询 6--按出版日期查询 |\n"); 126. printf("| |\n"); 127. printf("| 7--按价格查询 8--退出登录 |\n"); 128. printf("+-------------------------------------------------+\n"); 129. } 130. void book\_show\_menu() 131. { 132. printf("+------------------------------------------------------------+\n"); 133. printf("| 书籍显示界面 |\n"); 134. printf("| |\n"); 135. printf("| 1--价格升序显示 2--价格降序显示 |\n"); 136. printf("| |\n"); 137. printf("| 3--出版时间升序显示 4--出版时间降序显示 |\n"); 138. printf("| |\n"); 139. printf("| 5--普通显示 6--退回上一级 |\n"); 140. printf("+------------------------------------------------------------+\n"); 141. } 142. void admin\_init() 143. { 144. Admin admin; 145. AdminList \*head,\*rear,\*p; 146. head = NULL; 147. rear = NULL; 148. FILE \*fp; 149. if((fp=fopen("admin.txt","r"))==NULL) 150. { 151. printf("cannot open admin.txt file!"); 152. exit(0); 153. } 154. while(fscanf(fp,"%s %s",admin.admin\_name,admin.admin\_pass)!=EOF) 155. { 156. p = (AdminList\*)malloc(sizeof(AdminList)); 157. strcpy(p->admin.admin\_name,admin.admin\_name); 158. strcpy(p->admin.admin\_pass,admin.admin\_pass); 159. if(head == NULL) 160. { 161. head = p; 162. rear = p; 163. p->next = NULL; 164. } 165. else 166. { 167. rear->next = p; 168. rear = p; 169. } 170. } 171. fclose(fp); 172. rear->next = NULL; 173. admin\_list = head; 174. } 175. void print\_admin(AdminList \*AL) 176. { 177. AdminList \*p = AL; 178. printf("+-------------+--------------+\n"); 179. printf("| 账号 | 密码 |\n"); 180. printf("+-------------+--------------+\n"); 181. while(p) 182. { 183. printf("| %-10s |%10s | \n",p->admin.admin\_name,p->admin.admin\_pass); 184. p = p->next; 185. } 186. printf("+-------------+--------------+\n"); 187. } 188. int admin\_validate() 189. { 190. char name[30]; 191. char pass[30]; 192. printf("请输入管理员账号："); 193. scanf("%s",name); 194. printf("请输入管理员密码："); 195. scanf("%s",pass); 196. AdminList \*p = admin\_list; 197. while(p) 198. { 199. if(strcmp(p->admin.admin\_name,name)==0&&(strcmp(p->admin.admin\_pass,pass)==0)) 200. { 201. printf("欢迎管理员 %s 登录系统！\n",name); 202. system("pause"); 203. return 1; 204. } 205. p = p->next; 206. } 207. printf("管理员账号或密码有误！\n"); 208. return 0; 209. } 210. void user\_init() 211. { 212. User user; 213. UserList \*head,\*rear,\*p; 214. head = NULL; 215. rear = NULL; 216. FILE \*fp; 217. if((fp=fopen("user.txt","r"))==NULL) 218. { 219. printf("cannot open user.txt file!"); 220. exit(0); 221. } 222. while(fscanf(fp,"%s %s",user.user\_name,user.user\_pass)!=EOF) 223. { 224. p = (UserList\*)malloc(sizeof(UserList)); 225. strcpy(p->user.user\_name,user.user\_name); 226. strcpy(p->user.user\_pass,user.user\_pass); 227. if(head == NULL) 228. { 229. head = p; 230. rear = p; 231. p->next = NULL; 232. } 233. else 234. { 235. rear->next = p; 236. rear = p; 237. } 238. } 239. fclose(fp); 240. rear->next = NULL; 241. user\_list = head; 242. } 243. void print\_user(UserList \*UL) 244. { 245. UserList \*p = UL; 246. printf("+-------------+--------------+\n"); 247. printf("| 账号 | 密码 |\n"); 248. printf("+-------------+--------------+\n"); 249. while(p) 250. { 251. printf("| %-10s | %10s |\n",p->user.user\_name,p->user.user\_pass); 252. p = p->next; 253. } 254. printf("+-------------+--------------+\n"); 255. } 256. int user\_validate() 257. { 258. char name[30]; 259. char pass[30]; 260. printf("请输入用户账号："); 261. scanf("%s",name); 262. printf("请输入用户密码："); 263. scanf("%s",pass); 264. UserList \*p = user\_list; 265. while(p) 266. { 267. if(strcmp(p->user.user\_name,name)==0&&(strcmp(p->user.user\_pass,pass)==0)) 268. { 269. printf("欢迎用户 %s 登录系统！\n",name); 270. system("pause"); 271. return 1; 272. } 273. p = p->next; 274. } 275. printf("用户账号或密码有误！\n"); 276. return 0; 277. } 278. UserList\* search\_user(UserList \*UL,char name[]) 279. { 280. UserList \*p = UL; 281. while(p) 282. { 283. if(strcmp(p->user.user\_name,name)==0) 284. break; 285. p = p->next; 286. } 287. return p; 288. } 289. UserList\* add\_user(UserList \*UL) 290. { 291. char name[30]; 292. char pass[30]; 293. printf("请输入要注册的用户账号："); 294. scanf("%s",name); 295. printf("请输入要注册的用户密码："); 296. scanf("%s",pass); 297. if(search\_user(UL,name)!=NULL) 298. { 299. printf("该用户已经存在！\n"); 300. return UL; 301. } 302. else 303. { 304. UserList \*p = UL,\*q; 305. User user; 306. FILE \*fp; 307. q = (UserList\*)malloc(sizeof(UserList)); 308. strcpy(q->user.user\_name,name); 309. strcpy(q->user.user\_pass,pass); 310. while(p) 311. { 312. if(p->next==NULL) 313. break; 314. p = p->next; 315. } 316. q->next = p->next; 317. p->next = q; 318. if((fp=fopen("user.txt","a"))==NULL) 319. { 320. printf("cannot open user.txt file!"); 321. exit(0); 322. } 323. strcpy(user.user\_name,name); 324. strcpy(user.user\_pass,pass); 325. fprintf(fp,"\n %-20s %-8s",user.user\_name,user.user\_pass); 326. fclose(fp); 327. printf("注册用户成功！\n"); 328. return UL; 329. } 330. } 331. UserList\* del\_user(UserList \*UL) 332. { 333. char name[30]; 334. printf("请输入要删除的用户名："); 335. scanf("%s",name); 336. if(search\_user(UL,name)==NULL) 337. { 338. printf("该用户不存在！\n"); 339. return UL; 340. } 341. else 342. { 343. UserList \*p = UL,\*q; 344. if(strcmp(p->user.user\_name,name)==0) 345. { 346. q = p; 347. p = p->next; 348. free(q); 349. printf("删除用户成功！\n"); 350. return p; 351. } 352. else 353. { 354. while(p) 355. { 356. if(strcmp(p->next->user.user\_name,name)==0) 357. break; 358. p = p->next; 359. } 360. q = p->next; 361. p->next = q->next; 362. free(q); 363. printf("删除用户成功！\n"); 364. return UL; 365. } 366. } 367. } 368. UserList\* modify\_user(UserList \*UL) 369. { 370. char name[30],pass[30]; 371. printf("请输入要修改的用户名："); 372. scanf("%s",name); 373. if(search\_user(UL,name)==NULL) 374. { 375. printf("该用户不存在！\n"); 376. return UL; 377. } 378. else 379. { 380. UserList \*p = UL; 381. while(p) 382. { 383. if(strcmp(p->user.user\_name,name)==0) 384. break; 385. p = p->next; 386. } 387. printf("请输入修改后的用户名："); 388. scanf("%s",name); 389. printf("请输入修改后的用户密码："); 390. scanf("%s",pass); 391. strcpy(p->user.user\_name,name); 392. strcpy(p->user.user\_pass,pass); 393. printf("修改用户成功！\n"); 394. return UL; 395. } 396. } 397. void update\_user(UserList \*UL) 398. { 399. FILE \*fp; 400. User user; 401. UserList \*p = UL; 402. if((fp=fopen("user.txt","w"))==NULL) 403. { 404. printf("cannot open user.txt file!"); 405. exit(0); 406. } 407. while(p) 408. { 409. strcpy(user.user\_name,p->user.user\_name); 410. strcpy(user.user\_pass,p->user.user\_pass); 411. fprintf(fp," %-20s %-8s\n",user.user\_name,user.user\_pass); 412. p = p->next; 413. } 414. fclose(fp); 415. } 416. void book\_init() 417. { 418. Book book; 419. BookList \*head,\*rear,\*p; 420. head = NULL; 421. rear = NULL; 422. FILE \*fp; 423. if((fp=fopen("book.txt","r"))==NULL) 424. { 425. printf("cannot open book.txt file!"); 426. exit(0); 427. } 428. while(fscanf(fp,"%d %s %s %s %s %lf",&book.num,book.name,book.author, 429. book.publish,book.date,&book.price)!=EOF) 430. { 431. p = (BookList\*)malloc(sizeof(BookList)); 432. p->book.num = book.num; 433. strcpy(p->book.name,book.name); 434. strcpy(p->book.author,book.author); 435. strcpy(p->book.publish,book.publish); 436. strcpy(p->book.date,book.date); 437. p->book.price = book.price; 438. if(head == NULL) 439. { 440. head = p; 441. rear = p; 442. p->next = NULL; 443. } 444. else 445. { 446. rear->next = p; 447. rear = p; 448. } 449. } 450. fclose(fp); 451. rear->next = NULL; 452. book\_list = head; 453. } 454. void print\_book(BookList \*BL) 455. { 456. BookList \*p = BL; 457. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 458. printf("| 编号 | 名称 | 作者 | 出版社 | 出版日期 | 价格 |\n"); 459. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 460. while(p) 461. { 462. printf("| %-5d |%-25s| %-10s | %-17s | %-10s | %-10.1lf |\n",p->book.num,p->book.name, 463. p->book.author,p->book.publish,p->book.date,p->book.price); 464. p = p->next; 465. } 466. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 467. } 468. int findMax(Book book[],int L,int R) 469. { 470. if(L==R) 471. return book[L].num; 472. else 473. { 474. int a = book[L].num; 475. int b = findMax(book,L+1,R); 476. if(a>b) 477. return a; 478. else 479. return b; 480. } 481. } 482. void radixSort(Book book[],int n) 483. { 484. int max = findMax(book,0,n-1); 485. int i,j,k,num,index; 486. for(i=1;max/i>0;i=i\*10) 487. { 488. Book buckets[n][10]; 489. for(j=0;j<n;j++) 490. { 491. for(k=0;k<10;k++) 492. { 493. buckets[j][k].num = -1; 494. } 495. } 496. for(j=0;j<n;j++) 497. { 498. num = (book[j].num/i)%10; 499. buckets[j][num].num = book[j].num; 500. strcpy(buckets[j][num].name,book[j].name); 501. strcpy(buckets[j][num].author,book[j].author); 502. strcpy(buckets[j][num].publish,book[j].publish); 503. strcpy(buckets[j][num].date,book[j].date); 504. buckets[j][num].price = book[j].price; 505. } 506. index = 0; 507. for(j=0;j<10;j++) 508. { 509. for(k=0;k<n;k++) 510. { 511. if(buckets[k][j].num!=-1) 512. { 513. book[index].num = buckets[k][j].num; 514. strcpy(book[index].name,buckets[k][j].name); 515. strcpy(book[index].author,buckets[k][j].author); 516. strcpy(book[index].publish,buckets[k][j].publish); 517. strcpy(book[index].date,buckets[k][j].date); 518. book[index].price = buckets[k][j].price; 519. index++; 520. } 521. } 522. } 523. } 524. } 525. int binary\_search(Book book[],int len,int book\_num) 526. { 527. int low = 0; 528. int hight = len - 1; 529. int midder; 530. while(low <= hight) 531. { 532. midder = (low + hight) / 2; 533. if(book[midder].num == book\_num) 534. { 535. return midder; 536. } 537. else if(book[midder].num > book\_num) 538. { 539. hight = midder - 1; 540. } 541. else if(book[midder].num < book\_num) 542. { 543. low = midder + 1; 544. } 545. } 546. return -1; 547. } 548. void search\_book\_num(BookList \*BL) 549. { 550. int num,k=0; 551. BookList \*p = BL; 552. Book book[100]; 553. while(p) 554. { 555. book[k].num = p->book.num; 556. strcpy(book[k].name,p->book.name); 557. strcpy(book[k].author,p->book.author); 558. strcpy(book[k].publish,p->book.publish); 559. strcpy(book[k].date,p->book.date); 560. book[k].price = p->book.price; 561. k++; 562. p = p->next; 563. } 564. printf("请输入要查询的书籍编号："); 565. scanf("%d",&num); 566. radixSort(book,k);//基数排序 567. int flag = binary\_search(book,k,num); 568. if(flag==-1) 569. { 570. printf("没有查询到编号为%d的书籍！\n",num); 571. } 572. else 573. { 574. printf("查找成功！\n"); 575. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 576. printf("| 编号 | 名称 | 作者 | 出版社 | 出版日期 | 价格 |\n"); 577. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 578. printf("| %-5d |%-25s| %-10s | %-17s | %-10s | %-10.1lf |\n",book[flag].num,book[flag].name, 579. book[flag].author,book[flag].publish,book[flag].date,book[flag].price); 580. printf("+---------+-------------------------+------------+-------------------+-------------+----------------+\n"); 581. } 582. } 583. int search\_book\_num2(BookList \*BL,int book\_num) 584. { 585. int num,k=0; 586. BookList \*p = BL; 587. Book book[100]; 588. while(p) 589. { 590. book[k].num = p->book.num; 591. strcpy(book[k].name,p->book.name); 592. strcpy(book[k].author,p->book.author); 593. strcpy(book[k].publish,p->book.publish); 594. strcpy(book[k].date,p->book.date); 595. book[k].price = p->book.price; 596. k++; 597. p = p->next; 598. } 599. radixSort(book,k);//基数排序 600. int flag = binary\_search(book,k,book\_num); 601. if(flag==-1) 602. return 0; 603. else 604. return 1; 605. } 606. void search\_book\_name(BookList \*BL) 607. { 608. char name[30]; 609. printf("请输入要查询的书籍名称："); 610. scanf("%s",name); 611. BookList \*p = BL; 612. while(p) 613. { 614. if(strcmp(p->book.name,name)==0) 615. break; 616. p = p->next; 617. } 618. if(p==NULL) 619. printf("没有查询到书名为%s的书籍！\n",name); 620. else 621. { 622. printf("查询成功！\n"); 623. print\_book(p); 624. } 625. } 626. void search\_book\_author(BookList \*BL) 627. { 628. char author[30]; 629. printf("请输入要查询的书籍作者名称："); 630. scanf("%s",author); 631. BookList \*p = BL; 632. BookList \*head,\*rear,\*q; 633. head = NULL; 634. rear = NULL; 635. while(p) 636. { 637. if(strcmp(p->book.author,author)==0) 638. { 639. q = (BookList\*)malloc(sizeof(BookList)); 640. q->book.num = p->book.num; 641. strcpy(q->book.name,p->book.name); 642. strcpy(q->book.author,p->book.author); 643. strcpy(q->book.publish,p->book.publish); 644. strcpy(q->book.date,p->book.date); 645. q->book.price = p->book.price; 646. if(head == NULL) 647. { 648. head = q; 649. rear = q; 650. q->next = NULL; 651. } 652. else 653. { 654. rear->next = q; 655. rear = q; 656. } 657. } 658. p = p->next; 659. } 660. if(head == NULL) 661. printf("没有查询到书籍作者为%s的书籍！\n",author); 662. else 663. { 664. rear->next = NULL; 665. printf("查询成功！\n"); 666. BookList \*p = head; 667. print\_book(p); 668. } 669. } 670. void search\_book\_publish(BookList \*BL) 671. { 672. char publish[30]; 673. printf("请输入要查询的出版社名称："); 674. scanf("%s",publish); 675. BookList \*p = BL; 676. BookList \*head,\*rear,\*q; 677. head = NULL; 678. rear = NULL; 679. while(p) 680. { 681. if(strcmp(p->book.publish,publish)==0) 682. { 683. q = (BookList\*)malloc(sizeof(BookList)); 684. q->book.num = p->book.num; 685. strcpy(q->book.name,p->book.name); 686. strcpy(q->book.author,p->book.author); 687. strcpy(q->book.publish,p->book.publish); 688. strcpy(q->book.date,p->book.date); 689. q->book.price = p->book.price; 690. if(head == NULL) 691. { 692. head = q; 693. rear = q; 694. q->next = NULL; 695. } 696. else 697. { 698. rear->next = q; 699. rear = q; 700. } 701. } 702. p = p->next; 703. } 704. if(head == NULL) 705. printf("没有查询到出版社为%s的书籍！\n",publish); 706. else 707. { 708. rear->next = NULL; 709. printf("查询成功！\n"); 710. BookList \*p = head; 711. print\_book(p); 712. } 713. } 714. void search\_book\_date(BookList \*BL) 715. { 716. char date[30]; 717. printf("请输入要查询的书籍出版日期："); 718. scanf("%s",date); 719. BookList \*p = BL; 720. BookList \*head,\*rear,\*q; 721. head = NULL; 722. rear = NULL; 723. while(p) 724. { 725. if(strcmp(p->book.date,date)==0) 726. { 727. q = (BookList\*)malloc(sizeof(BookList)); 728. q->book.num = p->book.num; 729. strcpy(q->book.name,p->book.name); 730. strcpy(q->book.author,p->book.author); 731. strcpy(q->book.publish,p->book.publish); 732. strcpy(q->book.date,p->book.date); 733. q->book.price = p->book.price; 734. if(head == NULL) 735. { 736. head = q; 737. rear = q; 738. q->next = NULL; 739. } 740. else 741. { 742. rear->next = q; 743. rear = q; 744. } 745. } 746. p = p->next; 747. } 748. if(head == NULL) 749. printf("没有查询到书籍出版日期为%s的书籍！\n",date); 750. else 751. { 752. rear->next = NULL; 753. printf("查询成功！\n"); 754. BookList \*p = head; 755. print\_book(p); 756. } 757. } 758. void search\_book\_price(BookList \*BL) 759. { 760. double start,end; 761. printf("请输入要查询书籍的最低价格："); 762. scanf("%lf",&start); 763. printf("请输入要查询书籍的最高价格："); 764. scanf("%lf",&end); 765. BookList \*p = BL; 766. BookList \*head,\*rear,\*q; 767. head = NULL; 768. rear = NULL; 769. while(p) 770. { 771. if(p->book.price>=start&&p->book.price<=end) 772. { 773. q = (BookList\*)malloc(sizeof(BookList)); 774. q->book.num = p->book.num; 775. strcpy(q->book.name,p->book.name); 776. strcpy(q->book.author,p->book.author); 777. strcpy(q->book.publish,p->book.publish); 778. strcpy(q->book.date,p->book.date); 779. q->book.price = p->book.price; 780. if(head == NULL) 781. { 782. head = q; 783. rear = q; 784. q->next = NULL; 785. } 786. else 787. { 788. rear->next = q; 789. rear = q; 790. } 791. } 792. p = p->next; 793. } 794. if(head == NULL) 795. printf("没有查询到书籍价格在 %.1lf ~ %.1lf 范围的书籍！\n",start,end); 796. else 797. { 798. rear->next = NULL; 799. printf("查询成功！\n"); 800. printf("书籍价格在 %.1lf ~ %.1lf 范围的书籍\n",start,end); 801. BookList \*p = head; 802. print\_book(p); 803. } 804. } 805. BookList \*add\_book(BookList \*BL) 806. { 807. Book book; 808. printf("请输入要增加的书籍编号："); 809. scanf("%d",&book.num); 810. printf("请输入要增加的书籍名称："); 811. scanf("%s",book.name); 812. printf("请输入要增加的书籍作者："); 813. scanf("%s",book.author); 814. printf("请输入要增加的书籍出版社名称："); 815. scanf("%s",book.publish); 816. printf("请输入要增加的书籍出版时间："); 817. scanf("%s",book.date); 818. printf("请输入要增加的书籍价格："); 819. scanf("%lf",&book.price); 820. if(search\_book\_num2(BL,book.num)==1) 821. { 822. printf("该书籍已经存在！\n"); 823. return BL; 824. } 825. else 826. { 827. BookList \*p = BL,\*q; 828. FILE \*fp; 829. q = (BookList\*)malloc(sizeof(BookList)); 830. q->book.num = book.num; 831. strcpy(q->book.name,book.name); 832. strcpy(q->book.author,book.author); 833. strcpy(q->book.publish,book.publish); 834. strcpy(q->book.date,book.date); 835. q->book.price = book.price; 836. while(p) 837. { 838. if(p->next==NULL) 839. break; 840. p = p->next; 841. } 842. q->next = p->next; 843. p->next = q; 844. if((fp=fopen("book.txt","a"))==NULL) 845. { 846. printf("cannot open book.txt file!"); 847. exit(0); 848. } 849. fprintf(fp,"\n %-5d %-26s %-10s %-20s %-10s %-10.1lf ",book.num,book.name,book.author,book.publish,book.date, 850. book.price); 851. fclose(fp); 852. printf("增加书籍成功！\n"); 853. return BL; 854. } 855. } 856. BookList \*del\_book(BookList \*BL) 857. { 858. int num; 859. printf("请输入要删除的书籍编号："); 860. scanf("%d",&num); 861. if(search\_book\_num2(BL,num)==0) 862. { 863. printf("该书籍不存在！\n"); 864. return BL; 865. } 866. else 867. { 868. BookList \*p = BL,\*q; 869. if(p->book.num==num) 870. { 871. q = p; 872. p = p->next; 873. free(q); 874. printf("删除书籍成功！\n"); 875. return p; 876. } 877. else 878. { 879. while(p) 880. { 881. if(p->next->book.num==num) 882. break; 883. p = p->next; 884. } 885. q = p->next; 886. p->next = q->next; 887. free(q); 888. printf("删除书籍成功！\n"); 889. return BL; 890. } 891. } 892. } 893. BookList \*modify\_book(BookList \*BL) 894. { 895. char book\_name[30]; 896. printf("请通过关键字查询要修改的书籍的名称："); 897. scanf("%s",book\_name); 898. BookList \*p = BL; 899. BookList \*head,\*rear,\*q; 900. head = NULL; 901. rear = NULL; 902. while(p) 903. { 904. if(strstr(p->book.name,book\_name)) 905. { 906. q = (BookList\*)malloc(sizeof(BookList)); 907. q->book.num = p->book.num; 908. strcpy(q->book.name,p->book.name); 909. strcpy(q->book.author,p->book.author); 910. strcpy(q->book.publish,p->book.publish); 911. strcpy(q->book.date,p->book.date); 912. q->book.price = p->book.price; 913. if(head == NULL) 914. { 915. head = q; 916. rear = q; 917. q->next = NULL; 918. } 919. else 920. { 921. rear->next = q; 922. rear = q; 923. } 924. } 925. p = p->next; 926. } 927. if(head==NULL) 928. { 929. printf("没有查询到书名包含“%s”关键字的书籍！\n",book\_name); 930. return BL; 931. } 932. else 933. { 934. rear->next = NULL; 935. system("cls"); 936. printf("通过关键字“%s”查询书籍成功！\n",book\_name); 937. print\_book(head); 938. int book\_num; 939. printf("请输入你要修改的书籍编号："); 940. scanf("%d",&book\_num); 941. while(head) 942. { 943. if(head->book.num==book\_num) 944. break; 945. head = head->next; 946. } 947. if(head==NULL) 948. { 949. printf("书籍编号为 %d 的书籍不存在！\n",book\_num); 950. return BL; 951. } 952. else 953. { 954. system("cls"); 955. int book\_modify; 956. admin\_book\_modify\_menu(); 957. printf("请输入功能编号："); 958. scanf("%d",&book\_modify); 959. if(book\_modify==1) 960. { 961. int num; 962. printf("请输入修改后的书籍编号："); 963. scanf("%d",&num); 964. if(search\_book\_num2(BL,num)==1) 965. { 966. printf("书籍编号重复，不能修改！\n"); 967. return BL; 968. } 969. else 970. { 971. BookList \*p = BL; 972. while(p) 973. { 974. if(p->book.num==book\_num) 975. break; 976. p = p->next; 977. } 978. p->book.num = num; 979. printf("修改书籍编号成功！\n"); 980. return BL; 981. } 982. } 983. else if(book\_modify==2) 984. { 985. char name[30]; 986. printf("请输入修改后的书籍名称："); 987. scanf("%s",name); 988. BookList \*p = BL; 989. while(p) 990. { 991. if(p->book.num==book\_num) 992. break; 993. p = p->next; 994. } 995. strcpy(p->book.name,name); 996. printf("修改书籍名称成功！\n"); 997. return BL; 998. } 999. else if(book\_modify==3) 1000. { 1001. char author[30]; 1002. printf("请输入修改后的书籍作者名称："); 1003. scanf("%s",author); 1004. BookList \*p = BL; 1005. while(p) 1006. { 1007. if(p->book.num==book\_num) 1008. break; 1009. p = p->next; 1010. } 1011. strcpy(p->book.author,author); 1012. printf("修改书籍作者成功！\n"); 1013. return BL; 1014. } 1015. else if(book\_modify==4) 1016. { 1017. char publish[30]; 1018. printf("请输入修改后的出版社名称："); 1019. scanf("%s",publish); 1020. BookList \*p = BL; 1021. while(p) 1022. { 1023. if(p->book.num==book\_num) 1024. break; 1025. p = p->next; 1026. } 1027. strcpy(p->book.publish,publish); 1028. printf("修改书籍出版社成功！\n"); 1029. return BL; 1030. } 1031. else if(book\_modify==5) 1032. { 1033. char date[30]; 1034. printf("请输入修改后的书籍出版时间："); 1035. scanf("%s",date); 1036. BookList \*p = BL; 1037. while(p) 1038. { 1039. if(p->book.num==book\_num) 1040. break; 1041. p = p->next; 1042. } 1043. strcpy(p->book.date,date); 1044. printf("修改书籍出版时间成功！\n"); 1045. return BL; 1046. } 1047. else if(book\_modify==6) 1048. { 1049. double price; 1050. printf("请输入修改后的书籍价格："); 1051. scanf("%lf",price); 1052. BookList \*p = BL; 1053. while(p) 1054. { 1055. if(p->book.num==book\_num) 1056. break; 1057. p = p->next; 1058. } 1059. p->book.price = price; 1060. printf("修改书籍价格成功！\n"); 1061. return BL; 1062. } 1063. } 1064. } 1065. } 1066. void update\_book(BookList \*BL) 1067. { 1068. FILE \*fp; 1069. Book book; 1070. BookList \*p = BL; 1071. if((fp=fopen("book.txt","w"))==NULL) 1072. { 1073. printf("cannot open book.txt file!"); 1074. exit(0); 1075. } 1076. while(p) 1077. { 1078. book.num = p->book.num; 1079. strcpy(book.name,p->book.name); 1080. strcpy(book.author,p->book.author); 1081. strcpy(book.publish,p->book.publish); 1082. strcpy(book.date,p->book.date); 1083. book.price = p->book.price; 1084. fprintf(fp," %-5d %-26s %-10s %-20s %-10s %-10.1lf \n",book.num,book.name,book.author,book.publish, 1085. book.date,book.price); 1086. p = p->next; 1087. } 1088. fclose(fp); 1089. } 1090. int book\_amount(BookList \*BL) 1091. { 1092. BookList \*p = BL; 1093. int index = 0; 1094. while(p) 1095. { 1096. index++; 1097. p = p->next; 1098. } 1099. return index; 1100. } 1101. void show\_book\_price\_up(BookList \*BL) 1102. { 1103. int i,j,n = book\_amount(BL); 1104. BookList \*p,\*q; 1105. Book book; 1106. for(i=0;i<n-1;i++) 1107. { 1108. p = BL; 1109. for(j=0;j<n-1-i;j++) 1110. { 1111. q = p->next; 1112. if(p->book.price>q->book.price) 1113. { 1114. book.num = p->book.num; 1115. strcpy(book.name,p->book.name); 1116. strcpy(book.author,p->book.author); 1117. strcpy(book.publish,p->book.publish); 1118. strcpy(book.date,p->book.date); 1119. book.price = p->book.price; 1120. p->book.num = q->book.num; 1121. strcpy(p->book.name,q->book.name); 1122. strcpy(p->book.author,q->book.author); 1123. strcpy(p->book.publish,q->book.publish); 1124. strcpy(p->book.date,q->book.date); 1125. p->book.price = q->book.price; 1126. q->book.num = book.num; 1127. strcpy(q->book.name,book.name); 1128. strcpy(q->book.author,book.author); 1129. strcpy(q->book.publish,book.publish); 1130. strcpy(q->book.date,book.date); 1131. q->book.price = book.price; 1132. } 1133. p = p->next; 1134. } 1135. } 1136. print\_book(BL); 1137. } 1138. void show\_book\_price\_down(BookList \*BL) 1139. { 1140. int i,j,n = book\_amount(BL); 1141. BookList \*p,\*q; 1142. Book book; 1143. for(i=0;i<n-1;i++) 1144. { 1145. p = BL; 1146. for(j=0;j<n-1-i;j++) 1147. { 1148. q = p->next; 1149. if(p->book.price<q->book.price) 1150. { 1151. book.num = p->book.num; 1152. strcpy(book.name,p->book.name); 1153. strcpy(book.author,p->book.author); 1154. strcpy(book.publish,p->book.publish); 1155. strcpy(book.date,p->book.date); 1156. book.price = p->book.price; 1157. p->book.num = q->book.num; 1158. strcpy(p->book.name,q->book.name); 1159. strcpy(p->book.author,q->book.author); 1160. strcpy(p->book.publish,q->book.publish); 1161. strcpy(p->book.date,q->book.date); 1162. p->book.price = q->book.price; 1163. q->book.num = book.num; 1164. strcpy(q->book.name,book.name); 1165. strcpy(q->book.author,book.author); 1166. strcpy(q->book.publish,book.publish); 1167. strcpy(q->book.date,book.date); 1168. q->book.price = book.price; 1169. } 1170. p = p->next; 1171. } 1172. } 1173. print\_book(BL); 1174. } 1175. void show\_book\_date\_up(BookList \*BL) 1176. { 1177. int i,j,n = book\_amount(BL); 1178. BookList \*p,\*q; 1179. Book book; 1180. for(i=0;i<n-1;i++) 1181. { 1182. p = BL; 1183. for(j=0;j<n-1-i;j++) 1184. { 1185. q = p->next; 1186. if(strcmp(p->book.date,q->book.date)>0) 1187. { 1188. book.num = p->book.num; 1189. strcpy(book.name,p->book.name); 1190. strcpy(book.author,p->book.author); 1191. strcpy(book.publish,p->book.publish); 1192. strcpy(book.date,p->book.date); 1193. book.price = p->book.price; 1194. p->book.num = q->book.num; 1195. strcpy(p->book.name,q->book.name); 1196. strcpy(p->book.author,q->book.author); 1197. strcpy(p->book.publish,q->book.publish); 1198. strcpy(p->book.date,q->book.date); 1199. p->book.price = q->book.price; 1200. q->book.num = book.num; 1201. strcpy(q->book.name,book.name); 1202. strcpy(q->book.author,book.author); 1203. strcpy(q->book.publish,book.publish); 1204. strcpy(q->book.date,book.date); 1205. q->book.price = book.price; 1206. } 1207. p = p->next; 1208. } 1209. } 1210. print\_book(BL); 1211. } 1212. void show\_book\_date\_down(BookList \*BL) 1213. { 1214. int i,j,n = book\_amount(BL); 1215. BookList \*p,\*q; 1216. Book book; 1217. for(i=0;i<n-1;i++) 1218. { 1219. p = BL; 1220. for(j=0;j<n-1-i;j++) 1221. { 1222. q = p->next; 1223. if(strcmp(p->book.date,q->book.date)<0) 1224. { 1225. book.num = p->book.num; 1226. strcpy(book.name,p->book.name); 1227. strcpy(book.author,p->book.author); 1228. strcpy(book.publish,p->book.publish); 1229. strcpy(book.date,p->book.date); 1230. book.price = p->book.price; 1231. p->book.num = q->book.num; 1232. strcpy(p->book.name,q->book.name); 1233. strcpy(p->book.author,q->book.author); 1234. strcpy(p->book.publish,q->book.publish); 1235. strcpy(p->book.date,q->book.date); 1236. p->book.price = q->book.price; 1237. q->book.num = book.num; 1238. strcpy(q->book.name,book.name); 1239. strcpy(q->book.author,book.author); 1240. strcpy(q->book.publish,book.publish); 1241. strcpy(q->book.date,book.date); 1242. q->book.price = book.price; 1243. } 1244. p = p->next; 1245. } 1246. } 1247. print\_book(BL); 1248. } 1249. void show\_book(BookList \*BL) 1250. { 1251. system("cls"); 1252. while(1) 1253. { 1254. book\_show\_menu(); 1255. int book\_show; 1256. printf("请输入功能编号："); 1257. scanf("%d",&book\_show); 1258. if(book\_show==1) 1259. { 1260. show\_book\_price\_up(BL); 1261. book\_init(); 1262. } 1263. else if(book\_show==2) 1264. { 1265. show\_book\_price\_down(BL); 1266. book\_init(); 1267. } 1268. else if(book\_show==3) 1269. { 1270. show\_book\_date\_up(BL); 1271. book\_init(); 1272. } 1273. else if(book\_show==4) 1274. { 1275. show\_book\_date\_down(BL); 1276. book\_init(); 1277. } 1278. else if(book\_show==5) 1279. print\_book(BL); 1280. else if(book\_show==6) 1281. break; 1282. system("pause"); 1283. system("cls"); 1284. } 1285. } 1286. int main() 1287. { 1288. admin\_init(); 1289. user\_init(); 1290. book\_init(); 1291. while(1) 1292. { 1293. int main\_num; 1294. main\_menu(); 1295. printf("请输入系统功能编号："); 1296. scanf("%d",&main\_num); 1297. if(main\_num==1) 1298. { 1299. if(admin\_validate()==1) 1300. { 1301. system("cls"); 1302. while(1) 1303. { 1304. int admin\_num; 1305. admin\_menu(); 1306. printf("请输入功能编号："); 1307. scanf("%d",&admin\_num); 1308. if(admin\_num==1) 1309. { 1310. system("cls"); 1311. while(1) //书籍管理 1312. { 1313. admin\_book\_menu(); 1314. int book; 1315. printf("请输入功能编号："); 1316. scanf("%d",&book); 1317. if(book==1) //增加书籍 1318. book\_list = add\_book(book\_list); 1319. else if(book==2) //删除书籍 1320. { 1321. book\_list = del\_book(book\_list); 1322. update\_book(book\_list); 1323. } 1324. else if(book==3) //书籍查询 1325. { 1326. while(1) 1327. { 1328. admin\_book\_search\_menu(); 1329. int book\_search; 1330. printf("请输入功能编号："); 1331. scanf("%d",&book\_search); 1332. if(book\_search==1) 1333. print\_book(book\_list); 1334. else if(book\_search==2) 1335. search\_book\_num(book\_list); 1336. else if(book\_search==3) 1337. search\_book\_name(book\_list); 1338. else if(book\_search==4) 1339. search\_book\_author(book\_list); 1340. else if(book\_search==5) 1341. search\_book\_publish(book\_list); 1342. else if(book\_search==6) 1343. search\_book\_date(book\_list); 1344. else if(book\_search==7) 1345. search\_book\_price(book\_list); 1346. else if(book\_search==8) 1347. break; 1348. system("pause"); 1349. system("cls"); 1350. } 1351. } 1352. else if(book==4) //书籍修改 1353. { 1354. book\_list = modify\_book(book\_list); 1355. update\_book(book\_list); 1356. } 1357. else if(book==5) //书籍显示 1358. show\_book(book\_list); 1359. else if(book==6) //退回上一级 1360. break; 1361. system("pause"); 1362. system("cls"); 1363. } 1364. } 1365. else if(admin\_num==2) 1366. { 1367. system("cls"); 1368. while(1) //用户管理 1369. { 1370. admin\_user\_menu(); 1371. int user; 1372. printf("请输入功能编号："); 1373. scanf("%d",&user); 1374. if(user==1) 1375. { 1376. user\_list = del\_user(user\_list); 1377. update\_user(user\_list); 1378. } 1379. else if(user==2) 1380. { 1381. user\_list = modify\_user(user\_list); 1382. update\_user(user\_list); 1383. } 1384. else if(user==3) 1385. { 1386. char name[30]; 1387. printf("请输入要查询的用户名："); 1388. scanf("%s",name); 1389. UserList \*p = search\_user(user\_list,name); 1390. if(p == NULL) 1391. printf("用户%s不存在！\n",name); 1392. else 1393. { 1394. printf("查询成功！\n"); 1395. print\_user(p); 1396. } 1397. } 1398. else if(user==4) 1399. print\_user(user\_list); 1400. else if(user==5||user==6) 1401. break; 1402. system("pause"); 1403. system("cls"); 1404. } 1405. } 1406. else if(admin\_num==3) 1407. print\_admin(admin\_list); 1408. else if(admin\_num==4) 1409. break; 1410. system("pause"); 1411. system("cls"); 1412. } 1413. } 1414. } 1415. else if(main\_num==2) 1416. { 1417. if(user\_validate()==1) 1418. { 1419. system("cls"); 1420. while(1) 1421. { 1422. int num3; 1423. user\_menu(); 1424. printf("请输入功能编号："); 1425. scanf("%d",&num3); 1426. if(num3==1) 1427. show\_book(book\_list); 1428. else if(num3==2) 1429. search\_book\_num(book\_list); 1430. else if(num3==3) 1431. search\_book\_name(book\_list); 1432. else if(num3==4) 1433. search\_book\_author(book\_list); 1434. else if(num3==5) 1435. search\_book\_publish(book\_list); 1436. else if(num3==6) 1437. search\_book\_date(book\_list); 1438. else if(num3==7) 1439. search\_book\_price(book\_list); 1440. else if(num3==8) 1441. break; 1442. system("pause"); 1443. system("cls"); 1444. } 1445. } 1446. } 1447. else if(main\_num==3) 1448. user\_list = add\_user(user\_list); 1449. else if(main\_num==4) 1450. { 1451. printf("+----------------------------------------------+\n"); 1452. printf("|该系统主要是利用某本书的一些信息，通过其中的任|\n"); 1453. printf("|意一个信息，找出我们所需要的查找的该本书的所有|\n"); 1454. printf("|信息，采用基数排序法对一组具有结构特点的书籍编|\n"); 1455. printf("|号进行排序，利用二分查找法对排序好的书籍记录按|\n"); 1456. printf("|书籍编号实现快速查找，并按其他关键字的查找可以|\n"); 1457. printf("|采用最简单的顺序查找方法进行。 |\n"); 1458. printf("+----------------------------------------------+\n"); 1459. } 1460. else if(main\_num==5||main\_num==6) 1461. break; 1462. system("pause"); 1463. system("cls"); 1464. } 1465. return 0; 1466. } |

1. 主要数据及结果







1. 项目总结

这学期的数据结构课程设计我选的是一个图书馆书籍分类查询系统，主要采用单链表、键盘操作和文件知识来完成，涉及的算法有基数排序和二分查找，基本上实现了一个管理系统的增删查改功能。这个项目最难的地方应该是基数排序算法的实现和文件的操作，链表课堂上都已经学过了，就不绝得难，文件在大一就没掌握好，这次可以当作一次复习了，总之，在每次课程设计之前，一定要有计划，只要有了计划才能更好的去完成整个项目。