DataStructureHW3

2013011372박병욱

1. Assignment

순환 큐를 배열을 이용해서 구현한다.

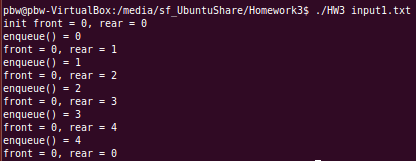
Enqueue, Dequeue와 기타 peek, is\_empty, is\_full함수를 구현한다.

1. Result

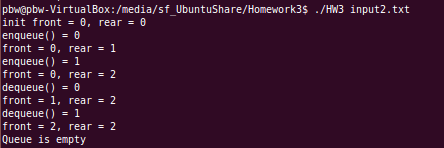
$./HW3



$./HW3 input1.txt



$./HW3 input2.txt



1. 소스코드
2. #include <stdio.h>
3. #include <stdlib.h>
4. #define MAX\_QUEUE\_SIZE 5
5. typedef int element;
6. typedef struct {
7. element queue[MAX\_QUEUE\_SIZE ];
8. int front, rear;
9. } QueueType;
10. void init(QueueType \*q);
11. void error(char \*message);
12. int is\_empty(QueueType \*q);
13. int is\_full (QueueType \*q);
14. void enqueue( QueueType \*q, element item );
15. element dequeue(QueueType \*q);
16. element peek(QueueType \*q);
17. int main(int argc, char \*argv[])
18. {
19. char command;
20. int key,temp;
21. FILE \*input;
22. QueueType \*q;
23. QueueType t;
24. q = &t;
25. init(q);
26. printf("init front = 0, rear = 0\n");
27. //입력 파일을 지정하지 않았을 경우
28. if(argc == 1)
29. input = fopen("input.txt", "r");
30. //입력 파일을 지정했을 경우
31. else if(argc == 2)
32. input = fopen(argv[1], "r");
33. //아규먼트가 많을 경우
34. else error("Usage : HW3 arg\n");
35. //파일을 찾을 수 없을경우
36. if(input == NULL)
37. error("Can't find the file\n");
38. //while문 안에서 input파일 내용을 fgetc로 얻어온다
39. while(1) {
40. command = fgetc(input);
41. if(feof(input)) break;
42. switch(command) {
43. case 'e':
44. fscanf(input, "%d", &key);//enqueue할 key값을 얻어온다.
45. if(!is\_full(q)){
46. enqueue(q,key);
47. printf("enqueue() = %d\n",key);
48. printf("front = %d, rear = %d\n",q->front, q->rear);
49. }
50. break;
51. case 'd'://dequeue 일 경우
52. if(!is\_empty(q)){
53. temp = dequeue(q);
54. printf("dequeue() = %d\n",temp);
55. printf("front = %d, rear = %d\n",q->front, q->rear);
56. }
57. break;
58. case 'p'://peek일 경우
59. if(!is\_empty(q)){
60. temp = peek(q);
61. printf("peek() = %d\n",temp);
62. printf("front = %d, rear = %d\n",q->front, q->rear);
63. }
64. break;
65. default :
66. ;
67. }
68. }
69. fclose(input);
70. return 0;
71. }
72. void error(char \*message)
73. {
74. fprintf(stderr,"%s\n",message);
75. exit(1);
76. }
77. void init(QueueType \*q)
78. {
79. q->front =0;
80. q->rear = 0;
81. }
82. //front와 rear가 같을 경우 큐는 empty이다
83. int is\_empty(QueueType \*q)
84. {
85. if(q->front == q->rear){
86. printf("Queue is empty\n");
87. return 1;
88. }else return 0;
89. }
90. //front와 rear+1가 같을 경우 큐는 full이다
91. int is\_full (QueueType \*q)
92. {
93. if((q->rear + 1)%MAX\_QUEUE\_SIZE == q->front){
94. printf("Queue is full\n");
95. return 1;
96. }
97. else return 0;
98. }
99. // q->rear에 1을 증가 하고 MAX\_QUEUE\_SIZE로 나눈 값을 q->rear에 다시 넣고
100. // q->rear에 item을 넣는다.
101. void enqueue( QueueType \*q, element item )
102. {
103. q->rear = (q->rear + 1)% MAX\_QUEUE\_SIZE;
104. q->queue[q->rear] = item;
105. }
106. // q->front에 1을 증가 하고 MAX\_QUEUE\_SIZE로 나눈 값을 q->front에 다시 넣고
107. // q->front의 값을 얻어온다.
108. element dequeue(QueueType \*q)
109. {
110. element ret;
111. q->front = (q->front + 1)% MAX\_QUEUE\_SIZE;
112. ret = q->queue[q->front];
113. return ret;
114. }
115. // q->front+1을 MAX\_QUEUE\_SIZE로 나눈 값을 얻어온다.
116. element peek(QueueType \*q)
117. {
118. element ret;
119. ret = q->queue[(q->front+1) % MAX\_QUEUE\_SIZE];
120. return ret;
121. }

* 소스코드 설명

주어진 given..c를 이용하여 구현하였다.

arg[1]를 받아 인풋파일로 이용하였으며,

아규먼트가 다수일 경우 error함수로 처리하였다.

파일을 열 수 없을 경우 error함수로 처리하였다.

While문 안에서 fgetc로 input파일을 읽었고, 줄의 시작을 구분하였다.

줄의 시작에 따라 enqueue, dequeue, peek함수를 구현하였고,

큐가 full일경우 enqueue를 하지 않고, 큐가 empty일경우 dequeue, peek을 하지 않는다.

Queue의 front와 rear는 각각 0에서 시작하며, enqueue 시 rear를 1증가한 후 값을 넣고,

Dequeue 시 front를 1증가한 후 값을 얻어왔다.

Queue가 full인지, empty인지는 Queue에서 한 칸을 비워두고, front와 rear가 같을경우 큐는 empty, (rear+1)%MAX\_QUEUE\_SIZE와 front가 같을경우 큐는 full로 처리하였다.