

1.1/MSE.h

```
1  #ifndef MSE_H
2  #define MSE_H
3
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <time.h>
7
8  #define N 100
9
10 typedef struct sArq{
11     FILE *f;
12     int pos, max, *buffer;
13 }Arquivo;
14
15 void criaArquivoTeste(char* nome, int ini, int fim){
16     int i;
17     FILE *f = fopen(nome, "w");
18     srand(time(NULL));
19     for(i=1; i<1000; i++){
20         fprintf(f, "%d\n", ini + rand() % (fim-ini + 1));
21         fprintf(f, "%d", ini + rand() % (fim-ini + 1));
22     }
23     fclose(f);
24 }
25
26 //QuickSort
27 void troca(int* a, int *b){
28     int aux = *a;
29     *a = *b;
30     *b = aux;
31 }
32
33 int particao(int *v, int ini, int fim){
34     int i = ini, j = fim;
35     int pivo = v[(ini+fim)/2];
36     while (1) {
37         while(v[i] < pivo) i++; //procura algum >= pivo do lado esquerdo
38         while(v[j] > pivo) j--; //procura algum <= pivo do lado direito
39
40         if(i<j){
41             troca(&v[i], &v[j]); //troca os elementos encontrados
42             i++;
43             j--;
44         }else
45             return j; //retorna o local onde foi feita a particao
46     }
47 }
48
49 void QuickSort(int *v, int ini, int fim){
50     if(ini < fim){
51         int q = particao(v, ini, fim);
52         QuickSort(v, ini, q);
53         QuickSort(v, q+1, fim);
54     }
55 }
56
57 //-----
```

```
58
59
60 //Cria Arquivos Ordenados
61 void salvaArquivo(char *nome, int *v, int tam, int mudaLinhaFinal){
62     int i;
63     FILE *f = fopen(nome, "a");
64     for(i=0; i<tam-1; i++)
65         fprintf(f, "%d\n", v[i]);
66     if(mudaLinhaFinal == 0)
67         fprintf(f, "%d", v[tam-1]);
68     else
69         fprintf(f, "%d\n", v[tam-1]);
70     fclose(f);
71 }
72
73 int criaArquivosOrdenados(char *nome){
74     int *v = (int*) malloc (N*sizeof(int));
75     char novo[20];
76     int K = 0, total = 0;
77     FILE *f = fopen(nome, "r");
78     while(!feof(f)){
79         fscanf(f, "%d", &v[total]);
80         total++;
81         if(total == N){
82             K++;
83             sprintf(novo, "Temp%d.txt", K);
84             QuickSort(v, 0, N-1);
85             salvaArquivo(novo, v, total, 0);
86             total = 0;
87         }
88     }
89     if(total > 0){
90         K++;
91         sprintf(novo, "Temp%d.txt", K);
92         QuickSort(v, 0, total-1);
93         salvaArquivo(novo, v, total, 0);
94     }
95     fclose(f);
96     free(v);
97     return K;
98 }
99
100
101 //-----
102
103 //Multiway Merging
104 void preencheBuffer(Arquivo* arq, int T){
105     int i;
106     if(arq->f == NULL) return;
107     arq->pos = 0;
108     arq->max = 0;
109     for(i=0; i<T; i++){
110         if(!feof(arq->f)){
111             fscanf(arq->f, "%d", &arq->buffer[arq->max]);
112             arq->max++;
113         }else{
114             fclose(arq->f);
115             arq->f = NULL;
116             break;
117         }
118     }
```

```
118     }
119 }
120
121 int procuraMenor(Arquivo* arq, int K, int T, int* menor){
122     int i, idx = -1;
123     for(i=0; i<K; i++){
124         if(arq[i].pos < arq[i].max){
125             if(idx == -1)
126                 idx = i;
127             else{
128                 if(arq[i].buffer[arq[i].pos] < arq[idx].buffer[arq[idx].pos])
129                     idx = i;
130             }
131         }
132     }
133     if(idx != -1){
134         *menor = arq[idx].buffer[arq[idx].pos];
135         arq[idx].pos++;
136         if(arq[idx].pos == arq[idx].max)
137             preencheBuffer(&arq[idx], T);
138         return 1;
139     }else
140         return 0;
141 }
142
143
144 void multiWayMerge(char *nome, int K, int T){
145     char novo[20];
146     int i;
147     int *saida = (int*) malloc (T*sizeof(int));
148     Arquivo *arq;
149     arq = (Arquivo*) malloc (K*sizeof(Arquivo));
150     for(i=0; i<K; i++){
151         sprintf(novo, "Temp%d.txt", i+1);
152         arq[i].f = fopen(novo, "r");
153         arq[i].buffer = (int*) malloc (T*sizeof(int));
154         preencheBuffer(&arq[i], T);
155     }
156     int menor, qtdSaida = 0;
157     while(procuraMenor(arq, K, T, &menor) == 1){
158         saida[qtdSaida] = menor;
159         qtdSaida++;
160         if(qtdSaida == T){
161             salvaArquivo(nome, saida, T, 1);
162             qtdSaida = 0;
163         }
164     }
165     if(qtdSaida != 0)
166         salvaArquivo(nome, saida, qtdSaida, 1);
167
168     for(i=0; i<K; i++) free(arq[i].buffer);
169     free(arq);
170     free(saida);
171 }
172
173 //-----
174
175 void MergeSortExterno(char* nome){
176     char novo[20];
177     int K = criaArquivosOrdenados(nome);
```

```
178 |     int i, T = N / (K + 1);  
179 |     remove(nome);  
180 |     multiWayMerge(nome, K, T);  
181 |     for(i=0; i<K; i++){  
182 |         sprintf(novo, "Temp%d.txt", i+1);  
183 |         remove(novo);  
184 |     }  
185 | }  
186 |  
187 | #endif
```