Você pode escrever os dados em um vetor ou em uma Struct, depende da implementação que você vai fazer, coloquei algumas referencias no fim da resposta.

veja um exemplo de como escrever um vetor de inteiros no arquivo

#include <stdio.h>

main() {

FILE \*arq;

// Esses dados vão ser gravados !

int ret, vet[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};

// arquivo alvo

char nomearq[] = "vet.dat";

// arquivo tem que ter permissão w para escrita e b para abrir como binario

arq = fopen(nomearq, "wb");

if (arq != NULL) {

// aqui é feita a escrita !!

ret = fwrite(vet, sizeof(int), 10, arq);

if (ret == 10)

printf("Gravacao com sucesso\n");

else

printf("Foram gravados apenas %d elementos\n", ret);

fclose(arq);

}

else

puts("Erro: criacao do arquivo");

}

Você pode recuperar os dados escritos no primeiro exemplo dessa forma

#include <stdio.h>

main() {

FILE \*arq;

int i, ret, vet[10];

char nomearq[] = "vet.dat";

arq = fopen(nomearq, "rb");

if (arq != NULL) {

// estou recuperando AQUI

ret = fread(vet, sizeof(int), 10, arq);

if (ret == 10) {

printf("Elementos: ");

for (i = 0; i < 10; i++)

printf("%d ", vet[i]);

}

else

printf("Foram lidos apenas %d elementos\n", ret);

fclose(arq);

}

else

puts("Erro: abertura do arquivo");

}

Você pode guardar estruturas mais complexas usando struct, exemplo

#include <stdio.h>

const int na = 6;

typedef struct {

char nome[10];

int nota;

} tp\_aluno;

main() {

tp\_aluno alunos[] = {{"Luiz", 5}, {"Paulo", 5}, {"Maria", 3},

{"Luiza", 4}, {"Felipe", 8}, {"Fabiana", 6}};

int ret;

FILE \*arq;

char nomearq[] = "turma.dat";

arq = fopen(nomearq, "wb");

if (arq != NULL) {

ret = fwrite(alunos, sizeof(tp\_aluno), na, arq);

if (ret == na)

printf("Gravacao %d registros com sucesso\n", ret);

else

printf("Foram gravados apenas %d elementos\n", ret);

fclose(arq);

}

else

puts("Erro: abertura do arquivo");

}

Para recuperar os dados você pode fazer assim

#include <stdio.h>

const int na = 6;

typedef struct {

char nome[10];

int nota;

} tp\_aluno;

main() {

tp\_aluno alunos[na];

int i, ret;

FILE \*arq;

char nomearq[] = "turma.dat";

arq = fopen(nomearq, "rb");

if (arq != NULL) {

ret = fread(alunos, sizeof(tp\_aluno), na, arq);

if (ret == na) {

printf("Lidos %d registros com sucesso\n", ret);

for (i = 0; i < ret; i++)

printf("%s %d\n", alunos[i].nome, alunos[i].nota);

}

else

printf("Foram lidos apenas %d elementos\n", ret);

fclose(arq);

}

else

puts("Erro: abertura do arquivo");

}

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63 | #include <cstdlib>  #include <iostream>  #include <stdio.h>      typedef struct tlista{          int n;          tlista \*prox;  };    tlista \*lu\_insereinicio(tlista \*plst, int pn)  {         tlista \*aux;         aux = (tlista\*)malloc(sizeof(tlista));         aux -> n = pn;         aux -> prox = plst;         return aux;  }    void lu\_exibelista(tlista \*plst)  {       tlista \*aux = plst;         while (aux != NULL)       {             printf("%d, ", aux -> n);             aux = aux -> prox;       }  }    int main()  {      int num; // ESTE É O NUMERO QUE SERÁ LIDO DO ARQUIVO E PASSADO PARA A LISTA      tlista \*list; // ESTOU CRIANDO A LISTA, NADA DE NOVO      list = NULL;      FILE \*arquivo; // REFERENCIA DO ARQUIVO, SEMPRE QUE FOR UTILIZAR O ARQUIVO                     // SERA REFERENDIDO-SE PELO arquivo      arquivo = fopen("c:\\arq.txt","r"); // O FOPEN ABRE O ARQUIVO, NO CASO                                          // É PASSADO POR PARAMETRO O ENDEREÇO                                          // DO ARQUIVO, E O "r" SIGNIFICA QUE                                          // O ARQUIVO ESTÁ SENDO ABERTO APENAS                                          // PARA LEITURA        if (arquivo) // CASO NAO TENHA DADO ERRO AO ABRIR O ARQUIVO         {             do               {                  fscanf(arquivo,"%d",&num ); // O FSCANF É IDENTICO AO SCANF QUE                                              // TODOS CONHECEM, PORÉM ELE É                                              // UTILIZADO PARA LER DADOS DE                                              // ARQUIVOS, E É PASSADO COMO                                              // PARAMETRO, O PROPRIO ARQUIVO                                              // QUE ESTA SENDO LIDO                  list = lu\_insereinicio(list, num); // INSERE O NUMERO NA LISTA               }while(!feof(arquivo)); // ELE PERCORRERÁ TODAS AS LINHAS DO ARQUIVO                                       // ENQUANTO O ARQUIVO NAO CHEGAR AO FIM                                       // FEOF = END OF FILE           }        lu\_exibelista(list); // EXIBE A LISTA      fclose(arquivo); // FECHA O ARQUIVO      system("pause");  } |