Cách write vào file text

#include <stdio.h>

#include <stdlib.h>

int main(){

FILE \*filetro; bắt đầu thao tác với file bằng con trỏ filetro

filetro = fopen("Vidu.txt", "w"); mở file với quyền write

if(filetro == NULL){ nếu không có quyền thao tác với file ở đây, filetro sẽ trả về NULL

printf("loi khong mo dc file\n");

exit(1); nếu không thao tác được thì đóng bằng cách này

}

int n; printf("nhap vao so n: ");

scanf("%d", &n); 2 câu lệnh này chỉ thao tác trên cpp

fprintf(filetro, "%d", n); câu lệnh này sẽ viết n vào file txt đang được trỏ đến

fclose(filetro); //dong file

}

-Cách đọc thông tin trong file text

int main(){

FILE \*filetro;

filetro = fopen("Vidu.txt", "r"); mở file với quyền read

if(filetro == NULL){

printf("loi khong mo dc file\n");

exit(1);

}

int n; phải khai báo n để có cái mà scanf

fscanf(filetro, "%d", &n); lệnh đọc file txt

printf("%d", n);

fclose(filetro);

}

- Cách đọc mảng 1 2 3 4 5 bằng fscanf. (Biết trước số phần tử muốn đọc)

int main(){

FILE \*fptr;

fptr = fopen("bai1.out", "r");

int n;

scanf("%d", &n);

int a[n];

for(int i = 0; i < n; i++){

fscanf(fptr, "%d", &a[i]);

}

for(int i = 0; i < n; i++){

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

}

fclose(fptr);

}

- Cách đọc file txt đến phần tử cuối cùng của nó mà không biết trước file có bao nhiêu phần tử.

int main(){

FILE \*fptr;

fptr = fopen("bai1.out", "r");

int n = 0;

int a[100];

while(feof(fptr) == 0){

fscanf(fptr, "%d", &a[n]);

n++;

}

for(int i = 0; i < n; i++){

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

}

fclose(fptr);

}

- Đọc cả file text(đọc cả dấu cách) bằng fgetc. Hàm fgetc() trong C được sử dụng để đọc từng ký tự một từ một file đã cho. Nó trả về EOF khi kết thúc file

#include<stdio.h>

int main() {

FILE \*fp;

char c;

fp=fopen("vidu.txt", "r");

while ((c = fgetc(fp)) != EOF) {

printf("%c", c);

}

fclose(fp);

}

- Đưa một kí tự vào file text bằng fputc.

#include <stdio.h>

main(){

FILE \*fp;

fp = fopen("vd.txt", "w");

fputc('k', fp);

fclose(fp);

}

KO biết trước số dòng phải scanf. Scanf từng dòng chứa thông tin đến khi hết file. Lưu số dòng có trong file txt.

#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include <ctype.h>

struct sinhvien{  
int stt;  
char mssv[100];  
char ten[100];  
char sdt[100];  
int diem;  
};  
typedef struct sinhvien sv;  
int main(){  
sv a[300];  
FILE \*f, \*f2;  
f = fopen("input.txt", "r");   
int i = 0;  
while(fscanf(f, "%d %s %s %s\n", &a[i].stt, a[i].mssv, a[i].ten, a[i].sdt) != EOF){ -Hàm fscanf sẽ trả về EOF khi hết file text.  
i++;  
}  
fclose(f);  
  
f2 = fopen("bangdiem.txt", "w");  
for(int j = 0; j < i; j++){  
printf("Nhap diem cua sinh vien so %d: ", a[j].stt);  
scanf("%d", &a[j].diem);  
fprintf(f2, "%d %s %s %s %d\n", a[j].stt, a[j].mssv, a[j].ten, a[j].sdt, a[j].diem);  
}  
fclose(f2);  
return 0;  
}

- Bài này phải scanf tên có chứa dấu cách. Bài này biết trước số dòng sẽ scanf

#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include <ctype.h>

struct sinhvien{  
char name[100];  
int toan, li, hoa;  
float tb;  
};  
typedef struct sinhvien sv;

int cmp(const void \*a, const void \*b){  
sv \*x = (sv\*)a;  
sv \*y = (sv\*)b;  
if(x->tb > y->tb){  
return -1;  
}  
return 1;  
}

int main(){  
sv a[300];  
FILE \*f, \*f2;  
f = fopen("student.txt", "r");  
int n; fscanf(f, "%d\n", &n);  
for(int i = 0; i < n; i++){  
fscanf(f, "%[^0-9] %d %d %d", a[i].name, &a[i].toan, &a[i].li, &a[i].hoa);  
a[i].tb = (float)(a[i].toan + a[i].li + a[i].hoa)/3;  
}  
fclose(f);  
f2 = fopen("student.dat", "w");  
for(int i = 0; i < n; i++){  
fprintf(f2, "%s %f", a[i].name, a[i].tb);  
}  
fclose(f2);  
qsort(a, n, sizeof(sv), cmp);  
for(int i = 0; i < n; i++){  
printf("%s %.2f", a[i].name, a[i].tb);  
}  
return 0;  
}

-Cách write vào file bin(file nhị phân)

struct ngaysinh{

int ngay, thang, nam;

};

typedef struct ngaysinh birth;

int main(){

FILE \*filetro;

filetro = fopen("Vidu.bin", "wb");

ngaysinh n1;

n1.ngay = 2;

n1.thang = 2;

n1.nam = 2004;

fwrite(&n1, sizeof(birth), 1, filetro); nhập biến n1 vào file nhị phân

fclose(filetro);

return 0;

}

- Cách đọc file nhị phân

B1: mở file dạng rb

B2: khai báo biến lưu dữ liệu đọc được vào biến đó

B3: đọc dữ liệu từ file bằng câu lệnh fread(&n1, sizeof(birth), 1, filetro);

B4: in biến khai báo ở bước 2.

struct ngaysinh{

int ngay, thang, nam;

};

typedef struct ngaysinh birth;

int main(){

FILE \*filetro;

filetro = fopen("Vidu.bin", "rb");

if(filetro == NULL){

printf("ko hop le");

exit(1);

}

birth n1;

fread(&n1, sizeof(birth), 1, filetro);

printf("%d %d %d", n1.ngay, n1.thang, n1.nam);

fclose(filetro);

return 0;

}