

Exemplo de criação de arquivos

```
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    // criando a variável ponteiro para o arquivo
    FILE *pont_arq;

    //abrindo o arquivo
    pont_arq = fopen("arquivo.txt", "a");

    // fechando arquivo
    fclose(pont_arq);

    //mensagem para o usuário
    printf("O arquivo foi criado com sucesso!");

    system("pause");
    return(0);
}
```

//Exemplo: Abrindo, gravando e fechando arquivo

```
#include <stdio.h>
#include <conio.h>

int main(void)
{
    FILE *pont_arq; // cria variável ponteiro para o
arquivo
    char palavra[20]; // variável do tipo string

    //abrindo o arquivo com tipo de abertura w
    pont_arq = fopen("arquivo_palavra.txt", "w");

    //testando se o arquivo foi realmente criado
    if(pont_arq == NULL)
    {
        printf("Erro na abertura do arquivo!");
        return 1;
    }
}
```

```

    printf("Escreva uma palavra para testar gravacao de
arquivo: ");
    scanf("%s", palavra);

    //usando fprintf para armazenar a string no arquivo
    fprintf(pont_arq, "%s", palavra);

    //usando fclose para fechar o arquivo
    fclose(pont_arq);

    printf("Dados gravados com sucesso!");

    getch();
    return(0);
}

//Leitura de arquivo
#include <stdio.h>
#include <conio.h>

int main(void)
{
    FILE *pont_arq;
    char texto_str[20];

    //abrindo o arquivo_frase em modo "somente leitura"
    pont_arq = fopen("arquivo_palavra.txt", "r");

    //enquanto não for fim de arquivo o looping será
    executado
    //e será impresso o texto
    while(fgets(texto_str, 20, pont_arq) != NULL)
        printf("%s", texto_str);

    //fechando o arquivo
    fclose(pont_arq);

    return(0);
}

```

```

/*
exemploArquivos1.c - Mostra como utilizar as funções
putc e getc
*/

#include <stdio.h>

int main() {
    // cria o arquivo se ele não existir
    FILE *fp = fopen("ExemploArquivos1.txt", "w");
    char *s = "Centro de Informatica";
    char c;
    int i;

    // se não houve erro
    if (fp != NULL) {
        // imprime a string caracter por caracter no
arquivo
        for (i = 0; s[i]; i++) {
            putc(s[i], fp);
        }
    } else {
        printf("Impossível abrir o arquivo.");
    }

    // fecha o arquivo
    fclose(fp);

    // abre o arquivo para leitura
    fp = fopen("ExemploArquivos1.txt", "r");

    // se não houve erro
    if (fp != NULL) {
        // lê todos os caracteres do arquivo
        c = getc(fp);
        while (c != EOF) {
            putchar(c);
            c = getc(fp);
        }
    } else {
        printf("Impossível abrir o arquivo.");
    }
}

```

```

    putchar('\n');

    // fecha o arquivo
    close(fp);

    return 0;
}

/*
exemploArquivos1.c - Mostra como utilizar as funções
ferror
*/

#include <stdio.h>

int main() {
    // cria o arquivo se ele não existir
    FILE *fp = fopen("ExemploArquivos2.txt", "w");
    char r[30], *s = "CIn\ncentro de Informatica";

    // se não houve erro
    if (fp != NULL) {
        fputs(s, fp);

        // verifica a ocorrência de erros
        if (ferror(fp)) {
            printf("Erro ao imprimir s.\n");
            exit(1); // sai do programa
        }
    } else {
        printf("Impossível abrir o arquivo.\n");
    }

    // fecha o arquivo
    fclose(fp);

    // abre o arquivo para leitura
    fp = fopen("ExemploArquivos2.txt", "r");

```

```

// se n, o houve erro
if (fp != NULL) {

    while (!feof(fp)) {
        fgets(r, 29, fp);

        // verifica a ocorrência de erros
        if (ferror(fp)) {
            printf("Erro ao ler do arquivo.\n");
            exit(1); // sai do programa
        }

        printf("%s", r);
    }

    putchar('\n');

} else {
    printf("Impossível abrir o arquivo.\n");
}

return 0;
}

/*
exemploArquivos3.c - Mostra como utilizar as funções
fwrite e fread
*/

#include <stdio.h>

int main() {

    FILE *pf = fopen("exemploArquivos3.txt", "w");
    int i, conteudo[4], mat[] = { 1, 2, 3, 4 };

    if (pf != NULL) {
        fwrite(mat, sizeof(int), 4, pf);

        if (ferror(pf)) {
            printf("Erro ao escrever no arquivo.\n");
        }
    }
}

```

```

        exit(1);
    }
} else {
    printf("Erro ao abrir o arquivo.\n");
}

fclose(pf);

pf = fopen("exemploArquivos3.txt", "r");

if (pf != NULL) {
    fread(conteudo, sizeof(int), 4, pf);

    if (ferror(pf)) {
        printf("Erro ao ler arquivo.\n");
        exit(1);
    }

    for (i = 0; i < 4; i++) {
        printf("%d ", conteudo[i]);
    }

    printf("\n");
} else {
    printf("Erro ao abrir o arquivo.\n");
}

fclose(pf);

return 0;
}

#include <stdio.h>

int main()
{
    FILE *fptr;
    char name[20];
    int age;
    float salary;

    /* open for writing */
    fptr = fopen("emp.txt", "w");

```

```

    if (fptr == NULL)
    {
        printf("File does not exists \n");
        return 0;
    }
    printf("Enter the name \n");
    scanf("%s", name);
    fprintf(fptr, "Name    = %s\n", name);
    printf("Enter the age\n");
    scanf("%d", &age);
    fprintf(fptr, "Age      = %d\n", age);
    printf("Enter the salary\n");
    scanf("%f", &salary);
    fprintf(fptr, "Salary   = %.2f\n", salary);
    fclose(fptr);

    return 0;
}

```

```

1.  /*
2.   * C program to create a file called emp.rec and store information
3.   * about a person, in terms of his name, age and salary.
4.   */
5.  #include <stdio.h>
6.
7.  void main()
8.  {
9.      FILE *fptr;
10.         char name[20];
11.         int age;
12.         float salary;
13.
14.         /* open for writing */
15.         fptr = fopen("emp.rec", "w");
16.
17.         if (fptr == NULL)
18.         {
19.             printf("File does not exists \n");
20.             return;
21.         }
22.         printf("Enter the name \n");
23.         scanf("%s", name);
24.         fprintf(fptr, "Name    = %s\n", name);
25.         printf("Enter the age\n");
26.         scanf("%d", &age);
27.         fprintf(fptr, "Age      = %d\n", age);
28.         printf("Enter the salary\n");
29.         scanf("%f", &salary);
30.         fprintf(fptr, "Salary   = %.2f\n", salary);

```

```
31.         fclose(fptr);
32.     }
```

```
1.  /*
2.   * C Program to Create Employee Record and Update it
3.   */
4.  #include <stdio.h>
5.  #include <stdlib.h>
6.  #include <string.h>
7.  #define size 200
8.
9.  struct emp
10.     {
11.         int id;
12.         char *name;
13.     }*emp1, *emp3;
14.
15.     void display();
16.     void create();
17.     void update();
18.
19.     FILE *fp, *fp1;
20.     int count = 0;
21.
22.     void main(int argc, char **argv)
23.     {
24.         int i, n, ch;
25.
26.         printf("1] Create a Record\n");
27.         printf("2] Display Records\n");
28.         printf("3] Update Records\n");
29.         printf("4] Exit");
30.         while (1)
31.         {
32.             printf("\nEnter your choice : ");
33.             scanf("%d", &ch);
34.             switch (ch)
35.             {
36.             case 1:
37.                 fp = fopen(argv[1], "a");
38.                 create();
39.                 break;
40.             case 2:
41.                 fp1 = fopen(argv[1], "rb");
```



```

42.         display();
43.         break;
44.     case 3:
45.         fp1 = fopen(argv[1], "r+");
46.         update();
47.         break;
48.     case 4:
49.         exit(0);
50.     }
51. }
52. }
53.
54. /* To create an employee record */
55. void create()
56. {
57.     int i;
58.     char *p;
59.
60.     emp1 = (struct emp *)malloc(sizeof(struct emp));
61.     emp1->name = (char *)malloc((size)*(sizeof(char)));
62.     printf("Enter name of employee : ");
63.     scanf(" %[^\n]s", emp1->name);
64.     printf("Enter emp id : ");
65.     scanf(" %d", &emp1->id);
66.     fwrite(&emp1->id, sizeof(emp1->id), 1, fp);
67.     fwrite(emp1->name, size, 1, fp);
68.     count++; // count to number of entries of records
69.     fclose(fp);
70. }
71.
72. /* Display the records in the file */
73. void display()
74. {
75.     emp3=(struct emp *)malloc(1*sizeof(struct emp));
76.     emp3->name=(char *)malloc(size*sizeof(char));
77.     int i = 1;
78.
79.     if (fp1 == NULL)
80.         printf("\nFile not opened for reading");
81.     while (i <= count)
82.     {
83.         fread(&emp3->id, sizeof(emp3->id), 1, fp1);
84.         fread(emp3->name, size, 1, fp1);
85.         printf("\n%d %s",emp3->id,emp3->name);
86.         i++;
87.     }

```

```

88.         fclose(fp1);
89.         free(emp3->name);
90.         free(emp3);
91.     }
92.
93.     void update()
94.     {
95.         int id, flag = 0, i = 1;
96.         char s[size];
97.
98.         if (fp1 == NULL)
99.         {
100.             printf("File cant be opened");
101.             return;
102.         }
103.         printf("Enter employee id to update : ");
104.         scanf("%d", &id);
105.         emp3 = (struct emp *)malloc(1*sizeof(struct emp));
106.         emp3->name=(char *)malloc(size*sizeof(char));
107.         while(i<=count)
108.         {
109.             fread(&emp3->id, sizeof(emp3->id), 1, fp1);
110.             fread(emp3->name,size,1,fp1);
111.             if (id == emp3->id)
112.             {
113.                 printf("Enter new name of employee to update : ");
114.
115.                 scanf(" %[^\n]s", s);
116.                 fseek(fp1, -204L, SEEK_CUR);
117.                 fwrite(&emp3->id, sizeof(emp3->id), 1, fp1);
118.                 fwrite(s, size, 1, fp1);
119.                 flag = 1;
120.                 break;
121.             }
122.             i++;
123.         }
124.         if (flag != 1)
125.         {
126.             printf("No employee record found");
127.             flag = 0;
128.         }
129.         fclose(fp1);
130.         free(emp3->name);          /* to free allocated memory */
131.         free(emp3);

```

<http://www.sanfoundry.com/c-programming-examples-file-handling/>