

- **int fflush(FILE \*stream)**

- Parâmetro:
  - Stream: ponteiro para objeto FILE
- Retorno: 0 para sucesso, EOF para erro.

- **int fscanf(FILE \*stream, const char \*formato, ...)**

- Parâmetro:
  - Stream: ponteiro para objeto FILE,
  - formato: é um string de formatação como aqueles usados com printf()
- Retorno: 0 para sucesso, EOF para erro.

- **int fprintf(FILE \*stream, const char \*formato, ...)**

- Parâmetros:
  - Stream: ponteiro para objeto FILE
  - formato: é um string de formatação como aqueles usados com printf()
- Retorno: 0 para sucesso, EOF para erro.

- **int remove(const char \*nomeDoArquivo)**

- Parâmetros:
  - nomeDoArquivo: arquivo a ser removido.
- Retorno: 0 para sucesso, outro valor caso contrário.

- **int rename(const char \*nomeAtual, const char \*nomeNovo)**

- Parâmetros:
  - nomeAtual: nome do arquivo a ser modificado.
  - nomeNovo: nome novo do arquivo.
- Retorno: 0 para sucesso, outro valor caso contrário.

- **char \*fgets(char \*ar, int n, FILE \*stream)**

- Parâmetros:
  - ar: array que armazenará o string lido.
  - n: tamanho desse array.
  - stream: ponteiro para objeto FILE.
- Retorno: retorna o endereço do array recebido como parâmetro quando consegue cumprir sua missão ou NULL, quando ocorre erro ou tentativa de leitura além do final do arquivo sem que ela tenha conseguido ler nenhum caractere.

- **int fgetc(FILE \*stream)**

- Parâmetro:
  - Stream: ponteiro para objeto FILE
- Retorno: 0 para sucesso, EOF para erro.

- **int fputs(const char \*s, FILE \*stream)**

- Parâmetros:
    - s: é o endereço de um string.
    - stream: ponteiro para objeto FILE
  - Retorno: retorna um valor não negativo quando obtém sucesso, caso contrário, retorna EOF.
- **int fputc(int byte, FILE \*stream)**
    - Parâmetros:
      - byte: Este é o caracter a ser escrito.
      - stream: ponteiro para objeto FILE
    - Retorno: Se não houver erros, o mesmo caractere que foi escrito é retornado. Se ocorrer um erro, EOF é retornado e o indicador de erro está configurado.
  - **size\_t fread(void \*ar, size\_t tamanho, size\_t n, FILE \*stream)**
    - Parâmetros:
      - \*ar: é o endereço do array de bytes no qual o bloco lido será armazenado. O tipo void \* utilizado na declaração deste parâmetro permite que ele seja compatível com ponteiros e endereços de variáveis de quaisquer tipos.
      - tamanho: é o tamanho de cada elemento do array.
      - Stream: ponteiro para objeto FILE.
    - Retorno: retorna o número de elementos que foram realmente lidos.
  - **size\_t fwrite(const void \*ar, size\_t tamanho, size\_t n, FILE \*stream)**
    - Parâmetros:
      - ar: ar é o endereço do array que armazena os bytes que serão escritos no stream. O tipo void \* utilizado na declaração deste parâmetro permite que ele seja compatível com ponteiros e endereços de variáveis de quaisquer tipos.
      - tamanho: tamanho de cada elemento do array.
      - n : número de elementos do array que serão escritos no stream.
      - Stream: ponteiro para objeto FILE
    - Retorno: retorna o número de elementos que foram realmente lidos.
  - **int fseek(FILE \*stream, int distancia, int deOnde)**
    - Parâmetros:
      - Stream: ponteiro para objeto FILE
      - distancia: é um deslocamento (positivo ou negativo), medido a partir do terceiro parâmetro, que indica para onde o indicador de posição do arquivo será movido.
      - deOnde: é o local a partir de onde o deslocamento (segundo parâmetro) será determinado.
    - Retorno:
  - **void rewind(FILE \*stream)**

- Parâmetro:
  - Stream: ponteiro para objeto FILE
- Retorno: 0 para sucesso, EOF para erro.

- **int ftell(FILE \*stream)**

- Parâmetro:
  - Stream: ponteiro para objeto FILE
- Retorno: 0 para sucesso, EOF para erro.