

Soubory

Knihovna `stdio.h` nabízí funkce `fscanf` a `fprintf` pro čtení a zápis se soubory. Chovají se podobně jako funkce `scanf` a `sprintf`, pouze první parametr je zdroj souboru. Ten je reprezentován strukturou `FILE` a odkaz na něj vrátí funkce `fopen(jméno,mód)`. Po skončení práce se souborem funkce `fclose(FILE*)` uvolní zdroje souboru¹.

Módy práce se souborem

- `"r"` (*read*) otevře soubor pro čtení a nastaví ukazatel na začátek; pokud soubor neexistuje, `fopen` vrátí `NULL`
- `"w"` (*write*) **vytvoří soubor nulové délky**; pokud soubor existoval, byl smazán!
- `"a"` (*append*) otevře soubor pro zápis a nastaví ukazatel na konec

Za tyto základní módy se mohou do řetězce přidat další doplňující znaky:

- `+` (*update*) soubor je otevřen pro čtení i zápis
- `b` (*binary*) se souborem se pracuje binárně: znaky `\r`, `\n` a `\r\n` jsou u znakového otevření považovány za jeden a týž znak konce řádky, při zápisu `fputs` se vloží konec řádku tak, jak je běžné v daném OS
- `x` (pouze po `w`, od verze C11) pokusí se vytvořit soubor nulové délky; pokud existoval, není smazán a `fopen` vrátí `NULL`

Funkce `scanf`, `sscanf` a `fscanf` vrací počet argumentů, které se podařilo načíst. Funkce `printf`, `sprintf` a `fprintf` vrací počet vytisknutých znaků.

```
printf("Zadejte tri cisla oddelena mezerou: ");
int i[3] {}; // vynulovani pole
int inlength = scanf("%d %d %d", &i[0], &i[1], &i[2]);
printf("uspesne nactena cisla: %d\n", inlength);
FILE* fd = fopen("data.txt", "w"); // fd = file descriptor
int outlength = fprintf(fd, "%d %d %d", i[0], i[1], i[2]);
printf("zapsano %d znaku\n", outlength);
fclose(fd);
puts("Nyni soubor muzete precist, editovat i smazat.");
getchar(); // aplikace běží do stisku klávesy
```

Zjištění velikosti souboru

Velikost souboru zjistíme jeho otevřením pro čtení, nastavením ukazatele na konec (`fseek`) a zjištěním pozice ukazatele (`ftell`):

```
FILE* fd = fopen("data.txt", "rb");
fseek(fd, 0, SEEK_END); // nastav ukazatel na pozici 0 od konce
unsigned long int filesize = ftell(fd);
// fseek(fd, 0, SEEK_SET); // jít na začátek
// fclose(fd); // nezapomenout zavřít
```

¹ také jiná aplikace s ním bude moci pracovat; OS obvykle uvolní zdroje automaticky po skončení aplikace

Ostatní funkce pro práci se soubory

- `remove` smazání souboru
- `rewind` posunutí ukazatele na začátek souboru
- `rename` přejmenování souboru
- `tmpfile` vytvoření dočasného souboru (s unikátním jménem), smazaného po jeho uzavření
- funkce ze souboru `stdio.h` začínající na `f` mající ekvivalenty v konzolových funkcích. Více viz na cplusplus.com.

Adresáře

Vytvářet adresáře lze funkcí `mkdir` a mazat je funkcí `rmdir` z hlavičkového souboru `sys/stat.h`. Mazat lze pouze prázdné adresáře a situace se zkomplikovala:

- Windows nepodporuje druhý parametr funkce `mkdir` (oprávnění souboru). Ve Visual Studiu je v souladu s ISO v souboru `direct.h` definována funkce `_mkdir` s jedním parametrem, nicméně GCC pod Windows definuje funkci `mkdir` bez podtržítka
- ve Windows lze alternativně použít `CreateDirectory` funkce z winAPI (hlavičkový soubor `windows.h`)
- lze též použít funkci `system`, která vykoná příkaz příkazové řádky OS

```
system("mkdir data");
```