

Práce se soubory

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.



Palacký University, Olomouc



- posloupnost bytů uložených v několika blocích – **proud dat** (stream)
- soubory:
 - textové
 - binární

Example

Číslo 65535 zabere v textovém souboru prostor 5 bytů, zatímco v binárním stačí byty 2.



- **buffer**

- **Základní datový typ:** ukazatel na objekt typu FILE

```
FILE *f;
```

- **Otevření souboru:** fopen()

```
f = fopen(soubor, rezim);
```

- **Uzavření souboru:** fclose()

```
fclose(f);
```

Požadavek	Režim otevření					
	''r''	''w''	''a''	''r+''	''w+''	''a+''
soubor musí existovat	×			×		
existující soubor bude vymazán		×			×	
existující soubor bude rozšiřován			×			×
neexistující soubor bude založen		×	×		×	×
data lze odkudkoliv číst	×		×	×	×	×
data lze kamkoliv zapisovat		×		×	×	
data lze zapisovat pouze na konec			×			×



■ formátovaně:

- `fprintf()`
- `fscanf()`

■ neformátovaně:

- po znacích – `getc()`, `putc()`, `fgetc()`, `fputc()`
- po řádcích – `fgets()`, `fputs()`
- po blocích – `fread()`, `fwrite()`

Konec souboru – EOF



„Vrací-li funkce chybový kód jako znamení potíží, budeš sledovat její návratovou hodnotu, ano i tehdy, když kontroly ztrojnásobí délku tvého kódu a způsobí puchýře na tvých prstech. Neboť jestliže se domníváš, že "se to nemůže stát", budeš zajisté potrestán za svou domýšlivost.“

(Henry Spencer)

■ Při chybě:

- `fopen()` vrací `NULL`
- `fclose()` vrací konstantu `EOF`

Example

```
#include <stdio.h>

int main(){
    FILE *fr;

    if((fr = fopen("soubor.txt", "r"))==NULL)
        printf("soubor.txt se nepodarilo otevrit \n");

    if(fclose(fr)==EOF)
        printf("soubor.txt se nepodarilo uzavrit \n");

    return 0;
}
```



Example

```
/* formatovane cteni ze souboru */  
fscanf(f, "format", argumenty);  
  
/* formatovany zapis do souboru */  
fprintf(f, "format", argumenty);
```

f proud dat typu FILE*

Example

```
#include <stdio.h>

int main()
{
    FILE *fw;
    int i;

    fw = fopen("priklad1.txt", "w");

    for(i = 1; i<=10; i++)
        fprintf(fw, "%i\n", i);

    fclose(fw);
    return 0;
}
```

Example

```
#include <stdio.h>

int main(){
    FILE *fr;
    float x,y,z;

    fr = fopen("priklad2.txt", "r");

    fscanf(fr, "%f %f %f", &x, &y, &z);
    printf("%f\n", x+y+z);

    fclose(fr);
    return 0;
}
```

Jak zjistíme, že soubor obsahuje 3 čísla?



Example

```
if ( fscanf(fr, "%f %f %f", &x, &y, &z)==3)
    printf("%f\n", x+y+z);
else
    printf("Soubor neobsahuje 3 realna cisla");
```

Example

```
/* cteni znaku ze souboru */  
c = getc(f);  
c = fgetc(f);  
  
/* zapis znaku c do souboru */  
putc(c, f);  
fputc(c, f);
```

f proud dat typu FILE*

c proměnná, z/do které budeme načítat znak

Příklad 3



Example

```
#include <stdio.h>

int main(){
    FILE *fr , *fw;
    int c;
    fr = fopen("priklad3.txt" , "r");
    fw = fopen("priklad3kopie.txt" , "w");

    c = getc(fr);    // s pomoci promenne c
    putc(c, fw);
    putc(getc(fr), fw);    // primo

    fclose(fr);
    fclose(fw);
    return 0;
}
```

Jak bychom zkopírovali celý soubor?

Example

```
#include <stdio.h>

int main()
{
    FILE *fr , *fw;
    int c;

    fr = fopen("priklad3.txt" , "r" );
    fw = fopen("priklad3kopie.txt" , "w" );

    while ((c = getc(fr)) != EOF)
        putc(c, fw);

    fclose(fr);
    fclose(fw);
    return 0;
}
```

Příklad 4



Example

```
#include <stdio.h>

int main()
{
    FILE *fr , *fw;
    int c;
    fr = fopen("priklad3.txt" , "r" );
    fw = fopen("priklad3kopie.txt" , "w" );

    c = getc(fr);
    while( feof(fr)==0){
        putc(c, fw);
        c = getc(fr);
    }
    fclose(fr);
    fclose(fw);
    return 0;
}
```



Example

```
/* cteni radku ze souboru */  
char *fgets(char *str, int max, FILE *fr);  
  
/* zapis radku c do souboru */  
char *fputs(char *str, FILE *fw);
```

`fr` proud dat typu `FILE*`

`str` řetězec, z/do kterého budeme načítat řádek

`max` maximální počet načítaných znaků

Example

```
#include <stdio.h>
#define MAX 5

int main(){
    FILE *fr;
    char str[MAX];
    fr = fopen("priklad5.txt", "r");

    while(fgets(str,MAX,fr) != NULL) // overeni konce souboru
        printf("%s\n", str);

    fclose(fr);
    return 0;
}
```

Jak bude vypadat výstup, pokud je v souboru jeden řádek obsahující čísla od 1 do 9?



Jak bude vypadat výstup, pokud je v souboru jeden řádek obsahující čísla od 1 do 9?

1234

5678

9

Jak upravit kód, aby se vypsalo celý řádek?

Example

```
/* cteni bloku */
int fread(void *kam, size_t rozmer, size_t pocet, FILE *fr);

/* zapis */
int fwrite(void *odkud, size_t rozmer, size_t pocet, FILE *fw);
```

`fr`, `fw` proud dat typu `FILE*`

`kam` ukazatel do paměti, kam budeme načítat

`odkud` ukazatel do paměti, odkud budeme načítat

`rozmer` velikost položky, kterou načítáme (v bytech)

`pocet` počet načítaných položek

`feof()` testuje konec souboru

`ferror()` testuje zda nedošlo k chybě

Example (Příklad použití fread())

```
#define VELIKOST_BLOKU 10

int data[VELIKOST_BLOKU];
FILE *fr = fopen("soubor.dat", "rb");
fread(data, sizeof(int), VELIKOST_BLOKU, fr);
```

Example (Příklad použití fwrite())

```
#define VELIKOST_BLOKU 10

int data[VELIKOST_BLOKU];
FILE *fw = fopen("soubor.dat", "wb");

fwrite(data, sizeof(int), VELIKOST_BLOKU, fw);
```

- **kurzor** – pozice v souboru
- **změna pozice** – `fseek()`

Example

```
int fseek(FILE *f, long posun, int odkud);
```

`fr`, `fw` proud dat typu `FILE*`
`posun` o kolik bytů se má kurzor posunout
`odkud` odkud se posun počítá

- **odkud:**
 - `SEEK_SET` – začátek souboru,
 - `SEEK_CUR` – aktuální pozice,
 - `SEEK_END` – konec souboru.



- `ftell()` – vrací aktuální pozici kurzoru v souboru `f`
- `rewind()` – posune kurzor na začátek souboru `f`
- `ungetc()` – vrácení znaku zpět do bufferu

Example

```
#include <stdio.h>

int main(void)
{
    FILE *soubor;
    size_t i;
    char pole[100];
    soubor = fopen("priklad1.txt", "rb");

    // posun na konec souboru
    fseek(soubor, 0L, SEEK_END);

    // %lu znaci, ze vystupem bude unsigned long cislo
    printf("Velikost souboru je %lu bytu.\n", ftell(soubor));

    fclose(soubor);
    return 0;
}
```



- knihovna `stdio` obsahuje ukazatele typu `FILE*`:
 - `stdin` – většinou vstup z klávesnice
 - `stdout` – většinou výpis do konzole
 - `stderr` – výpis chybových zpráv

Example

```
/* cteni jednoho znaku z klavesnice */
getc(stdin);
getchar();

/* vypis jednoho znaku na obrazovku */
putc(c, stdout);
putchar(c);
```




- 1** Napište program, který bude číst ze souboru desetinná čísla (libovolný počet) a vrátí jejich průměr.
- 2** Napište program tak, aby zkusil číst neexistující soubor. Zajistěte, aby program vhodně reagoval na tuto situaci.
- 3** Napište program, který spočítá celkový počet znaků v souboru.
- 4** Napište program, který uloží čísla od 0 do 9 do binárního i textového souboru. Zjistěte velikosti těchto souborů. Otevře si tyto soubory v textovém editoru.
- 5** Napište program, který uloží číslo 1234567 do binárního i textového souboru. Zjistěte velikosti těchto souborů. Proč je v minulém případě velikost binárního souboru větší a v tomto ne?
- 6** Napište program, který do binárního souboru uloží pole desetinných čísel (záleží na vás, jak velké).



- 7 Napište program, který z desetinných čísel uložených v binárním souboru (například vytvořeného v předchozí úloze) spočítá průměr. Měl by být napsán obecně tak, aby fungoval i když nebudeme vědět, kolik je v souboru čísel.
- 8 Napište program, který čte čísla ze vstupního souboru a ukládá je do jiného souboru. Pokud je čtené číslo větší než 100, číslo se neuloží do souboru, ale na obrazovku se vypíše chybová hláška Špatně zadané číslo CISLO (CISLO je konkrétní číslo).