



KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

Práce se soubory

Základy programování 2

Mgr. Markéta Trnečková, Ph.D.

Soubor

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

Příklad

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

Práce se soubory

- **buffer**

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

```
FILE *f;
```

- **Otevření souboru:** fopen()

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

- **Uzavření souboru:** fclose()

```
fclose(f);
```

Režim otevření

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			×			×

- t – textový
- b – binární

Práce s daty

■ formátovaně:

- fprintf()
- fscanf()

■ neformátovaně:

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

Konec souboru – EOF

Testování

„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`

Testování

Příklad

```
#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;
}
```

Čtení a zápis dat

Příklad (pr10-příklad2.c)

```
#include <stdio.h>

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

    fr = fopen("příklad2.txt", "r");

    fscanf(fr, "%f %f %f", &x, &y, &z)==3);
    printf("%f\n", x+y+z);

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

Příklad

Jak zjistíme, že soubor obsahuje 3 čísla?

Čtení a zápis dat

Čtení a zápis jednoho znaku

Příklad

```
/* ctení 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

Čtení a zápis dat

Příklad (pr10-příklad3.c)

```
#include <stdio.h>
int main(){
    FILE *fr , *fw;
    int c;
    fr = fopen("příklad3.txt", "r");
    fw = fopen("příklad3kopie.txt", "w");
    // s pomoci promenne c
    c = getc(fr);
    putc(c, fw);
    // primo
    putc(getc(fr), fw);
    fclose(fr); fclose(fw);
    return 0;
}
```

Příklad

Jak bychom zkopírovali celý soubor?

Čtení a zápis dat

Příklad

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

Čtení a zápis dat

Čtení a zápis jednoho řádku

Příklad

```
/* 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

Čtení a zápis dat

Příklad (pr10-příklad5.c)

```
#include <stdio.h>
#define MAX 5
int main(){
    FILE *fr;
    char str[MAX];
    fr = fopen("příklad5a.txt", "r");
    while(fgets(str,MAX, fr) != NULL) // overeni konce souboru
    {
        printf("%s", str);
    }
    fclose(fr);
    return 0;
}
```

Příklad

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

Čtení a zápis dat

Příklad

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

1234

5678

9

Příklad

Jak upravit kód, aby se vypsalo celé číslo?

Čtení a zápis dat

Čtení a zápis jednoho bloku

Příklad

```
/* 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

Čtení a zápis jednoho bloku

Příklad (Příklad použití fread())

```
#define VELIKOST_BLOKU 10

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

Příklad (Příklad použití fwrite())

```
#define VELIKOST_BLOKU 10

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

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


Čtení a zápis dat

Pozice čtení a zápisu

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

Příklad

```
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.

Čtení a zápis dat

Další funkce

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

Příklad

Jak zjistíme velikost souboru?

Čtení a zápis dat

Příklad

```
#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;
}
```

Standardní vstup a výstup

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

Příklad

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

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

Standardní vstup a výstup

Přesměrování

Operační systém:

```
C:\Umisteni_souboru\program.exe > output.txt
```

```
C:\Umisteni_souboru\program.exe < input.txt
```

```
C:\Umisteni_souboru\program1.exe | program2.exe
```

Bufferování

- **Řádkové bufferování** – `stdin`
- **Blokové bufferování** – práce se soubory
- **Žádné bufferování** – `stdout` a `stderr`.

Cvičení

- 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).

Cvičení

- 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 znaky ze vstupního souboru a opisuje je buď na obrazovku, nebo do souboru "novy.txt". To, kam se má provést výstup, zvolí uživatel.
- 9 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). Použijte vhodně proud stderr.
- 10 Předpokládejme, že existuje soubor zprava.txt, obsahující následující text (bez uvozovek, včetně mezer):
"jakuzyp k sec stje v cajooecl"
Máme následující program tajemstvi:

Cvičení

Příklad (tajemstvi.c)

```
#include <stdio.h>
int main(){
    char c;
    int i=0;
    while((c=getchar()) != EOF){
        if((i % 4) > 1)
            putc(c, stdout);
        else
            putc(c, stderr);
        i++;
    }
    return 0;
}
```

Co bude obsahem souborů zprava1.txt a zprava2.txt, pokud program spustíme následujícím způsobem?

```
tajemstvi.exe < zprava.txt > zprava1.txt 2> zprava2.txt
```

Svou domněnku ověřte.