Obsah

24 Práce se soubory, struktury, pole

24.1 Práce se soubory

- typy souborů
  - textové soubory – soubory obsahující plain text
  - binární soubory – obsahují binární data nemožné otevřít v obyčejném editoru
- operace
  - vytvoření souboru
  - otevření existujícího souboru
  - zavření souboru
  - čtení a zápis dat do souboru
- proměnná souboru – pointer na souboru `FILE *fptr`

24.1.1 Otevírání souboru

- syntaxe `fptr = fopen("filepath","mode");`
  - `"filepath"` – cesta k souboru
  - `"mode"` – mód otevření souboru
- otevření datového streamu
- file open modes
  - `r, w, a` – čtení, zápis a připsání na konec
  - `r+, w+, a+` – čtení a zápis (soubor musí existovat), čtení a zápis (přepsání nebo vytvoření) a čtení a připsání (připsání nebo připsání)
  - `b` – binární soubor (`rb, rb+, wb, wb+, ab, ab+`)

24.1.2 Zavření souboru

- `fclose(fptr);`
- zavření souboru / streamu po dokončení operací, připsání EOF
- pointer zahozen

24.1.3 Čtení ze souboru

- `fscanf(fptr, "format", &variable)` – načtení textového souboru
  - `fptr` – file pointer
  - `"format"` – formát textu
  - `&variable` – adresa proměnné pro zapsání dat
- `fread(&variable, dataSize, dataCount, fptr)` – načtení textového souboru
  - `dataSize` – velikost dat v bytech (často `sizeof(datatype)` – `sizeof(int)`…)
  - `dataCount` – počet dat o velikosti `dataSize`, použito při načítání pole

24.1.4 Zápis do souboru

- `fprintf(fptr, "format", variable)` – vypsání textu do souboru
- `fwrite(&variable, dataSize, dataCount, fptr)` – zápis binárních dat
- připsání/zapsání/přepsání dat do souboru

```c
#include <stdio.h>

/* Calculate average of values in input.csv and write the output to
 output.csv*/

const char inputFilePath[] = "./input.csv";
const char outputFilePath[] = "./output.csv";

int main () {
  int values[3];

  FILE *inputFile = fopen(inputFilePath, "r");
  if (inputFile == NULL) {
    puts("Input file not found!");
    return(1);
  }

  fscanf(inputFile, "%d,%d,%d", &values[0], &values[1], &values[2]);
  fclose(inputFile);

  float avg = (values[0]+values[1]+values[2])/3.0f;

  FILE *outputFile = fopen(outputFilePath, "w");
  fprintf(outputFile, "Average: %f\n", avg);
  fclose(outputFile);

  return 0;
}
```

Kód 1: Načítání a zapisování do souboru

24.2 Struktury

- keyword `struct`
- proměnné/struktury uchovávající více dat/proměnných v sobě
- zjednodušení a seskupení dat
- příklad – struktura Kniha
  - název
  - autor
  - žánr
  - rok vydání

```c
struct book {
  char name[50];
  char author[50];
  char genre[50];
  int year;
};
```

Kód 2: Příklad struktury

- různé metody definovaní
  - `struct StructName {...};`
    - definuje `struct name`
    - inicializace proměnné – `struct StructName variableName;`
  - `struct StructName {...} variableName;`
    - definuje `variableName` typu `struct StructName`
    - zkrácení předchozí definice do jednoho příkazu, nevhodné pro opakované využití v kódu
  - `typedef struct {..} DatatypeName;` / `typedef struct StructName {..} DatatypeName;`
    - inicializace proměnné – `datatypeName variableName;`
    - definice „vlastního datatypu“
    - obchází potřebu při inicializaci psát `struct`
    - optional `structName`
- inicializace proměnné s hodnotami – `struct StructName structVar = insideVar1, insideVar2;`
- přístup k hodnotám – `structVar.insideVar1 = value;`

```
1  // -- Define struct Point --
2  struct Point {
3    int x,y;
4  };
5
6  struct Point p1 = {0,0}; // init p1 of type Point
7
8  // -- Declare p2 of type Point --
9  struct Point {
10   int x,y;
11 } p2 = {1,1};
12
13 // -- Define struct using typedef --
14 typedef struct {
15   int x,y;
16 } Point;
17
18 Point p3 = {-1,2}; // declare Point
```

Kód 3: Různé způsoby definování struktur

24.3 Pole

- jeden dlouhý list podobných hodnot o stejném data typu
- v paměti hodnoty uloženy hned po sobě
- možno udělat pole polí – 2D pole
- příklad – `int nums[] = {1,2,3,4,5};`, `char name[] = "David";`, `float values[10];`
- velikost pole předem daná – statické pole (v některý prog. jazycích i dynamická)
- začátek číslování od `0`

name[0]	name[1]	name[2]	name[3]	name[4]
'D'	'a'	'v'	'i'	'd'

Tab. 24.1: Obsah `char` pole

24.3.1 Hodnota položky

- syntax – `arrayName[index]`
- čtení – `foo = arrayName[index];`

- zápis – `arrayName[index] = foo;`

24.3.2 Pole struktur

- pole složené ze struktur
- hodnota pole `arrayName[index].insideVar`

```c
#include <stdio.h>

typedef struct {
  float x,y,z;
} Point;

Point points[] = {
  {1,4,2},
  {4,2,3},
  {2,3,6},
  {3,1,2}
};

int main () {
  for (int i=0; i<sizeof(points)/sizeof(points[0]); i++){
    printf("x: %.2f, y: %.2f, z: %.2f\n",
        points[i].x, points[i].y, points[i].z);
  }
}
```

Kód 4: Vypsání pole struktur