

Základy programovania

Cvičenie 5

Dátové štruktúry

Cvičiaci: Ing. Magdaléna Ondrušková, (iondruskova)



15. října 2024

Dátové prúdy

- Základný mechanizmus pre prenos údajov medzi programom a napr. vonkajšími súbormi
- Funkcie podporujúce toto správanie sa nachádzajú v knižnici `stdio.h`

Tri štandardné prúdy:

- **`stdin`** - štandardný vstup, používa sa na čítanie vstupu, obvykle z klávesnice (napr. použitím funkcie `scanf`)
- **`stdout`** - štandardný výstup, používa sa na zápis výstupu, obvykle do konzoly (napr. použitím funkcie `printf`)
- **`stderr`** - štandardný chybový výstup, používa sa na zápis chybových výstupov

Smerovacie značky

- pri spustení z príkazového riadku vieme presmerovať štandardné prúdy na nejaký súbor (program) pomocou smerovacích značiek.

```
1 ./main < input.txt // vstup z klavesnice nahrad suborom input.  
   txt.  
2 ./main > output.txt // vystup do konzole nahrad suborom output.  
   txt.  
3 ./main >> output.txt // vystup pridaj na koniec suboru output.  
   txt.  
4 ./main 2> error.txt // chybovy vystup presmeruj do suboru  
   error.txt.  
5 ./main | ./main2 // vystup programu main presmeruj na vstup  
   main2
```

Návratová hodnota

- Scanf vracia počet úspešne načítaných vstupov
- Umožňuje overiť úspešnosť načítania vstupov

```
1 #include <stdio.h>
2
3 int main() {
4     int age;
5     float height;
6     char initial;
7     printf("Enter your age, heigh, and first initial: ");
8     int result = scanf("%d %f %c", &age, &height, &initial);
9
10    if (result == 3) {
11        printf("Entered: Age = %d, Height = %.2f, Initial = %c\n",
12            age, height, initial);
13    } else {
14        printf("Failed to read all values.\n");
15    }
16
17    return 0;
18 }
```

Návratová hodnota

- Vráti špeciálnu hodnotu EOF - end-of-file
- Znamená to, že narazil na koniec súboru
- Pomocou klávesnice: CTRL+Z (Windows) alebo CTRL+D (Linux)
- Vieme načítavať celý súbor bez toho, aby sme dopredu vedeli ako je tento súbor veľký (resp. koľko položiek obsahuje)

```
1 #include <stdio.h>
2
3 int main() {
4     char y[101]; //vytvarime si pole znaku
5     while (scanf("%100[^\n]\n", y) != EOF) //dokud nenarazime na
        koniec souboru
6     {
7         printf("%s\n", y); //vypisujeme nacteny retezec
8     }
9     return 0;
10 }
```

Príkazom `scanf("%100[^\n]\n", y)` načítavame vstup aj s medzerami

Príklad - načítanie vstupu zo súboru

- Vytvorte si súbor `cisla.txt`, ktorý obsahuje 5 rôznych čísiel. Postupne čísla načítajte a vypíšte ich súčet.

Príklad vstupu a výstupu:

```
1 // vstup cisla.txt
2 10 20 30 40 50
3
4 // výstup:
5 150
```

Príklad - načítanie vstupu zo súboru

- Vytvorte si súbor `znaky.txt`, ktorý obsahuje nahodné znaky. Spocitajte, koľkokrát sa prvý znak vyskytuje v subore (okrem 1. vyskytu).

Príklad vstupu a výstupu:

```
1 // vstup cisla.txt
2 a a a a b b a a b c
3
4 // pocitam teda pocet 'a' v zvyšnom subore
5
6 // výstup:
7 5
```

Dátový typ **FILE**

- Slúži na reprezentáciu súborového prúdu

```
1 FILE *file_pointer;
```

Otváranie súboru pomocou funkcie `fopen`

- `filename` - názov súboru
- `mode` - mód (režim) práce so súborom

Rôzne módy práce so súbormi:

- **r** - read, otvorí súbor iba pre čítanie
- **w** - write, otvorí súbor na zápis. Ak súbor existuje, vymaže jeho obsah. Ak neexistuje, súbor sa vytvorí.
- **a** - append, otvorí súbor na zápis. Zapisuje na koniec súboru (nezmaže predchádzajúci obsah súboru).

```
1 FILE *fopen(const char *filename, const char *mode);
```


Návratová hodnota funkcie `fopen`

- Ak sa súbor nepodarilo otvoriť, funkcia `fopen` vráti hodnotu `NULL`

Zatváranie súboru pomocou funkcie `fclose`

- Po skončení práce so súborom je dôležité súbor zavrieť
- Zatvorenie súboru uvoľňuje zdroje a zabezpečuje, že všetky písania sú zapísané do súboru.
- Nezavretie súboru môže viesť k poškodeniu súboru, ak je otvorený v režime zápisu.

```
1 int fclose(FILE *stream);
```

Príklad práce so súborom

```
1 #include <stdio.h>
2
3 int main() {
4     FILE *file = fopen("data.txt", "r"); // Otvorenie suboru na
        citanie
5     if (file == NULL) {
6         printf("Nepodarilo sa otvoriť subor.\n");
7         return 1; // Chyba pri otvorení
8     }
9
10    // Dalsie operacie so suborom...
11
12    fclose(file); // Zatvorenie suboru
13    return 0;
14 }
```

Čítanie zo súboru

- `fscanf()` - Číta formátované údaje zo súboru.
- `fgets()` - Číta jeden riadok textu zo súboru.
- `fgetc()` - Číta jeden znak zo súboru.

Príklad použitia `fgets()`:

```
1 #include <stdio.h>
2
3 int main() {
4     FILE *file = fopen("data.txt", "r");
5     char buffer[100];
6
7     if (file != NULL) {
8         while (fgets(buffer, sizeof(buffer), file) != NULL) {
9             printf("%s", buffer); // Vypise riadok zo suboru
10        }
11        fclose(file);
12    }
13    return 0;
14 }
```

Zápis do súboru

- `fprintf()` - Zapisuje formátované údaje do súboru.
- `fputs()` - Zapisuje reťazec do súboru.
- `fputc()` - Zapisuje jeden znak do súboru.

Príklad použitia `fprintf`

```
1 #include <stdio.h>
2
3 int main() {
4     FILE *file = fopen("output.txt", "w");
5
6     if (file != NULL) {
7         // Zapisuje text do suboru
8         fprintf(file, "Toto je testovací text.\n");
9         fclose(file);
10    }
11    return 0;
12 }
```

Úloha: Napíš program, ktorý načíta text zo súboru `input.txt`, prevedie všetky znaky na veľké písmená a uloží výsledok do nového súboru `output.txt`.

```
1 // Příklad vstupneho suboru input.txt:
2 a
3 ahoj
4 ahoj
5 a
6 b
7 Dnes je pekny den.
8 8.10.2024
9
10 // Příklad vystupneho suboru output.txt:
11 A
12 AHOJ
13 AHOJ
14 A
15 B
16 DNES JE PEKNY DEN.
17 8.10.2024
```

Definícia štruktúry

- Umožňujú spojenie položiek ľubovoľného dátového typu
- Uspodňujú organizáciu dát a predávanie parametrov do funkcie

Definujeme ju pomocou kľúčového slova `struct`.
Polozky oddelene pomocou bodkočiarky (stredníka).

```
1 struct nazov {  
2     typ1 premenna1;  
3     typ2 premenna2;  
4     // ... dalsie premenne  
5 };
```

Definovanie konkrétnej štruktúry pre osobu:

```
1 struct Osoba {  
2     char meno[50];  
3     int vek;  
4     float vyska;  
5 };
```

Po definovaní štruktúry môžeme vytvoriť premenné tohto dátového typu:

```
1 struct Osoba osoba1; // Deklaracia premennej typu Osoba
```

Príp. pomocou typedef pre zjednodušenie:

```
1 typedef struct Person { // definujeme strukturu menom Person  
2     char meno[50];      // položka meno  
3     int vek;            // položka vek  
4     float vyska;       // položka vyska  
5 } osoba; // deklarujeme jej datovy typ "osoba"  
6  
7 osoba Anna; // osoba "Anna" bez pociatocnych hodnot
```

Prístup k jednotlivým položkám:

```
1 #include <stdio.h>
2
3 typedef struct Person{
4     int age;
5     int height;
6 } osoba;
7
8 int main()
9 {
10     osoba person1;
11     person1.height = 167;
12
13     printf("Vyska osoby 1: %d\n", person1.height);
14     return 0;
15 }
```


Štruktúry môžeme použiť ako parametre funkcie aj ako ich návratový typ:

```
1 #include <stdio.h>
2
3 typedef struct Person{
4     int age;
5     int height;
6 } osoba;
7
8 void vypisOsobu(osoba o) {
9     printf("Vek: %d, Vyska: %d cm\n", o.age, o.height);
10 }
11
12
13 int main()
14 {
15     osoba person1;
16     person1.age = 25;
17     person1.height = 167;
18
19     vypisOsobu(person1);
20     return 0;
21 }
```

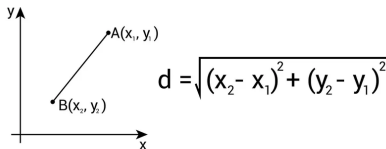
Definujte štruktúru, ktorá bude obsahovať informácie o kruhu:

- polomer
- stredová súradnica X
- stredová súradnica Y

Následne definujte funkcie, ktoré spočítajú:

- obvod kruhu $2 * \text{PI} * \text{polomer}$
- obsah kruhu $\text{PI} * \text{polomer} * \text{polomer}$
- vzdialenosť medzi dvoma kruhmi (ich stredmi)

Distance Formula



Obrázek: Příklad výpočtu vzdálenosti mezi dvěma body

Položkou štruktúry môže byť aj ďalšia štruktúra:

```
1 #include <stdio.h>
2 struct Datum {
3     int den;
4     int mesiac;
5     int rok;
6 };
7 struct Osoba {
8     int vek;
9     int vyska;
10    struct Datum datumNarodenia;
11 };
12 int main(){
13     struct Osoba osoba1;
14     osoba1.vek = 25;
15     osoba1.vyska = 1.80;
16     osoba1.datumNarodenia.den = 15;
17     osoba1.datumNarodenia.mesiac = 10;
18     osoba1.datumNarodenia.rok = 1995;
19     printf("Vek: %i, Datum narodenia: %i.%i.%i.", osoba1.vek,
20         osoba1.datumNarodenia.den, osoba1.datumNarodenia.mesiac,
21         osoba1.datumNarodenia.rok);
22     return 0;
23 }
```

Úloha: Definujte štruktúru Bod. Následne túto štruktúru použite pre definíciu tvaru Obdĺžnik (pomocou 4 bodov). Nad touto štruktúrou definujte a aplikujte (zavolajte) nasledujúce funkcie:

- Vzdialenosť dvoch bodov (pomocná funkcia)

```
float vzdialenostBodov(Bod a, Bod b)
```

- obvod obdĺžnika

```
float obvodObdlznika(Obdlznik o)
```

- plocha (obsah) obdĺžnika

```
float plochaObdlznika(Obdlznik o)
```

- Je zadaný obdĺžnik štvorec?

```
bool jeStvorec(Obdlznik o)
```

- Bod X, leží v obdĺžniku?

```
bool bodLeziVObdlzniku(Obdlznik o, Bod p)
```

- Zmeň veľkosť každej strany obdĺžnika o int x

```
Obdlznik zmenVelkostObdlznika(Obdlznik o, int x)
```