Úvod do programování v C

Vlastimil Křápek

6. prosince 2004

1 Editor a překladač

Pro psaní programů potřebujete editor, k jejich překladu do spustitelné podoby překladač. Obojí je k dispozici na monoceru a na studentských počítačích v podkrovní PC učebně.

Editor: na monoceru příkazem `mc` spustíte Midnight Commander (obdoba známého NC, WC nebo M602). Současným stiskem `Shift` a `F4` vytvoříte nový soubor, do nějž můžete psát zdrojový text programu. Soubor ukládáte pomocí `<F2>`, editor ukončíte pomocí `<F10>`. Před prvním uložením budete dotázáni na jméno souboru, pojmenujte jej např. `projekt.c`. V PC učebně můžete využít další nainstalované editory (nedit, Kwrite, Kate, gVim).

Překladač: je výhodné spouštět z druhého okna. Napište `gcc projekt.c`, v případě chyby se vypíšou na obrazovku chybová hlášení, v případě úspěchu se vytvoří soubor `a.out`, který je možné spustit.

Můžete si také stáhnout vlastní překladač (obvykle se zabudovaným editorem). Lze doporučit např. Bloodshed Dev C++ (`http://www.bloodshed.net/devcpp.html`). Použijte verzi 4, beta verze 5 obsahuje chyby.

2 Pravidla práce na monoceru

Velmi snadno napíšete program, který spotřebuje všechny dostupné systémové prostředky. Protože na monoceru běží mnoho aplikací různých uživatelů, je nutné chovat se tak, aby nedošlo k omezení provozu. Proto spouštějte programy výhradně se sníženou prioritou pomocí `nice -19 a.out`. Pokud běh programu neskončí zhruba do minuty, je pravděpodobně něco špatně. V takovém případě ukončete běh programu současným stiskem `Ctrl` a `c`.

Pokud je to možné, použijte místo monocera jiný stroj, případně požádejte o zřízení účtu na mirsamu, který je pro výpočty určen.

3 Ahoj, svete

První program v C může vypadat takto:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
  printf("ahoj, svete");
  getc(stdin);
  return 0;
}
```

Rozebereme si podrobně funkci jednotlivých částí programu.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

načítá potřebné knihovny s deklaracemi standardních funkcí. Obvykle vystačíte se třemi výše zmíněnými.

```
int main()
{
  ...
  return 0;
}
```

je definice funkce `main`, která má návratový typ `int` a po provedení příkazů `...` vrací hodnotu `0`. Tato funkce je nezbytnou součástí každého programu v C. Po jeho spuštění se provedou příkazy uvedené v těle funkce (mezi složenými závorkami). V našem případě jsou to

```
  printf("ahoj, svete");
  getc(stdin);
```

Příkaz `printf` vytiskne na obrazovku text obsažený v uvozovkách. Příkaz `getc` přečte jeden znak z datového proudu `stdin` (tedy z klávesnice). Pokud by chyběl, tak by běh programu skončil bezprostředně po výpisu `ahoj, svete`, což by v některých prostředích znemožnilo si vypsanou zprávu přečíst. Každý příkaz musí být ukončen středníkem.

4 Proměnné

Důležitou součástí jazyka jsou proměnné, které mohou obsahovat různé hodnoty. Podle typu hodnoty rozlišujeme proměnné číselné, znakové, složené a další.

Následující program přečte z klávesnice dvě celá čísla a vypíše jejich součet. Kvůli stručnosti zápisu uvádíme pouze funkci `main`, které ale vždy musí předcházet načtení knihoven pomocí `#include`.

```
int main()
{
  int i,j,k;
  printf("Zadejte dve cela cisla: \n");
  scanf("%d",&i);
  scanf("%d",&j);
  getc(stdin);
  k=i+j;
  printf("Souctem cisel %d a %d je %d\n",i,j,k);
  getc(stdin);
  return(0);
}
```

Hned za složenou závorkou se deklarují proměnné, které budeme v programu používat. Deklarace

```
int i,j,k;
```

znamená, že budeme používat tři proměnné pojmenované `i`, `j`, `k`, které jsou typu `int`, tedy celočíselné. Reálné proměnné bychom definovali na dalším řádku pomocí typu `double`, znakové proměnné pomocí `char`, další datové typy potřebovat nebudeme.

Bezprostředně po deklaraci je v proměnné uložena náhodná hodnota. Předat hodnotu proměnné lze dvěma způsoby, pomocí přiřazení a pomocí načtení.

Příkaz `scanf("%d",&i)` přečte z klávesnice celé číslo a uloží jej do proměnné `i`. Všimněte si, že před proměnnou `i` je napsán znak `&`, nemusíte vědět, proč tam je, ale musíte ho tam vždy

při volání `scanf` napsat. To, že se čte celé číslo, je dáno textem v uvozovkách, pokud by v nich stálo `"%c"`, přečetl by se znak, v případě `"%lf"` by se četlo reálné číslo v desetinném tvaru (např. `-0.0027`), v případě `"%le"` reálné číslo v exponenciálním tvaru tvaru (např. `-7.2e+2` ($-7{,}2 \times 10^2$), `0.02e-20`). Po zadání čísla je nutné zmáčknout `Enter`.

Přiřazovací příkaz `k=i+j` uloží do proměnné `k` součet hodnot uložených v proměnných `i` a `j`. Lze přiřazovat i další aritmetické výrazy (podíl se bude chápat jako celočíselný, jsou-li dělenec i dělitel celočíselné).

```
k=i-(j+2);
k=3*i+j/i;
i=i+1;
```

Hodnoty uložené v proměnných vypíšeme pomocí printf, kde do uvozovek na příslušné místo uvedeme znak `%` následovaný identifikací typu (`d` pro celé, `e` nebo `f` pro reálné – pozor, na rozdíl od `scanf` píšeme bez `l`). Za uvozovkami následuje seznam proměnných, které chceme vytisknout (právě jedna pro každé `%` v řetězci). Řetězec `\n` ukončuje řádek.

5 Výstup do souboru

Následující program demonstruje vytvoření souboru a zápis do něj.

```
int main()
{
  int i=1,j=3;
  double x=5.2;
  FILE *F;

  F=fopen("vystup.txt","w");
  fprintf(F,"vystupni soubor\n");
  fprintf(F,"dve cela cisla: %d %d\n",i,j);
  fprintf(F,"jejich soucet: %d\n",i+j);
  fprintf(F,"realne cislo v desetinnem (%f) a exponencialnim (%e) tvaru\n",x,x);
  fclose(F);
  return 0;
}
```

Se soubory se pracuje prostřednictvím popisovače souboru – proměnné typu `FILE *`. Prostřednictvím příkazu

```
F=fopen("vystup.txt","w");
```

je popisovači přidělen soubor se jménem `vystup.txt` (bude umístěn v aktuálním adresáři programu), který je otevřen pro čtení (druhý argument je `"w"`). Pokud soubor neexistuje, je vytvořen, pokud existuje, je přepsán.

Poté je možné do souboru zapisovat příkazem `fprintf`, který se od dříve popsaného `printf` liší pouze tím, že jeho prvním argumentem je popisovač souboru, do nějž má být výstup zapsán.

Po skončení práce se souborem se soubor uzavře příkazem `fclose(F);`.

6 Cykly

Cykly slouží k opakovanému provádění sekvence příkazů. Následující program vypíše do souboru průběh funkce sinus v intervalu `xmin`, `xmax` s krokem `dx`

```
int main()
{
```

```
  double xmin=0,xmax=3.14,dx=0.01,x;
  FILE *F;

  x=xmin;
  F=fopen("sin.txt","w");
  while (x<xmax) {
    fprintf(F,"%f %f\n",x,sin(x));
    x=x+dx;
  }
  fclose(F);
  return 0;
}
```

Obecně je cyklus zapsán ve tvaru

```
  while (podmínka) {
    blok
  }
```

přičemž význam je takový, že je-li splněna podmínka uvedená za klíčovým slovem `while`, pak se provede blok uzavřený ve složených závorkách následujících podmínku, jinak pokračuje běh programu za blokem.

Reference

[1] http://programovani.wz.cz/rs/view.php?cisloclanku=2004101501