Rekurze

Jiří Zacpal

DEPARTMENT OF COMPUTER SCIENCE
PALACKÝ UNIVERSITY, OLOMOUC

KMI/ZP2 Základy programování 2

Doporučená literatura

1. Pavel Herout: Učebnice Jazyka C. Kopp, 2007.
2. Reek Kenneth: Pointers on C. Addison Wesley, 1997.
3. Robert Sedgewick: Algorithms in C.
   Addison-Wesley Professional, 2001.
4. Jeri R. Hanly, Elliot B. Koffman: Problem Solving and Program Design in C. Addison Wesley, 2006.
5. Eric S. Roberts: Programming Abstractions in C.
   Addison Wesley, 1997.
6. Eric S. Roberts: The Art and Science of C.
   Addison Wesley, 1994.
7. Libovolné další učebnice jazyka C

Požadavky na zápočet

1. pro zápočet je potřeba získat **20 bodů**:
   - 1 bod na každém cvičení za příklad
   - 0 až 5 bodů za úkoly vyhlašované v semestru (celkem 2 úkoly)
   - 0, 3, 5 bodů za písemné práce v semestru (celkem 2 za semestr)
2. individuální domluva možná

Konzultace

1. v pracovně 5.044
2. každou středu 9.30 – 11.30
3. jindy po vzájemné domluvě
4. email: jiri.zacpal@upol.cz
5. web: edis.upol.cz
   - podmínky zápočtu
   - příklady
   - body

Funkce - opakování

1. Co je to funkce?
2. Co je parametr funkce?
3. Jaký je rozdíl mezi předáním parametru hodnotou a odkazem (ukazatelem, adresou)?
4. Jaké rozlišujeme rozsahy platnosti proměnných (resp. identifikátorů)? Jaké jsou mezi nimi rozdíly?

Příklad 1

```
main()
{
	unsigned int c=12345;
	unsigned int *p=NULL;
	int pocet=mincovka(c,&p);
	printf("castka: %d\npouzita platidla: ",c);
	for(int i=0;i<pocet;i++)
	{
		printf("%d,",p[i]);
	}
}

int mincovka(unsigned int castka, unsigned int **platidla)
{
	int mince[]={5000,2000,1000,500,200,100,50,20,10,5,2,1},p=0,m=0;
	while (castka>0)
	{
		if(castka>=mince[m])
		{
			p++;
			if (p==1)
				(*platidla)=(unsigned int *)malloc(sizeof(int));
			else
				(*platidla)=(unsigned int *)realloc((*platidla),p*sizeof(int));
			(*platidla)[p-1]=mince[m];
			castka-=mince[m];
		}
	else
	m++;
	}
	return p;
}
```

Rekurzivní funkce

1. funkce, která ve svém těle volá sama sebe
2. postupným voláním funkce je problém zjednodušován → dekompozice → metoda rozděl a panuj
3. dvě části rekurze:
   - **limitní podmínka rekurze** je podmínka, po jejímž splnění je vyhodnocen výraz, jež nezpůsobí další aplikaci samotné rekurzivní funkce
   - **předpis rekurze** je část těla funkce, při jejímž vyhodnocení dochází k rekurzivní aplikaci funkce

Průběh výpočetního procesu rekurzivní funkce

1. výpočetní proces se skládá ze dvou fází:
   - **fáze navíjení**
     - je fáze, ve které dochází k postupné rekurzivní aplikaci
   - **fáze odvíjení**
     - nastává po dosažení limitní podmínky rekurze
     - během této fáze dochází k dokončení vyhodnocenı těla procedury
2. příklady: funkce pro výpočet faktoriálu, funkce pro výpočet fibonacciho čísla, quicksort, ...

Příklad 2

```
long double factrec(unsigned int n)
{
    if (n==0) return 1;
    return n * factrec(n-1);
}

long double factiter(unsigned int n)
{
    long double out = 1;
    unsigned int i;
    for (i=2; i<=n; i++) out *= i;
    return out;
}
```

Příklad 3

Fibonacciho čísla:

Fib(0) = 0
Fib(1) = 1
Fib(n) = Fib(n-1) + Fib(n-2) pro n > 1

```
double fibonaccirec(int n)
 {
    if (n<2) return n;
           else
    return fibonaccirec(n-1)+fibonaccirec(n-2);
 }

double fibonacciiter(int n)
 {
    int i;
    double fib=0,Nmin1,Nmin2,temp;
    if(n<2) return n;
    Nmin1=1;
    Nmin2=0;
    for (i=2;i<n;i++)
    {
         temp=Nmin1;
         Nmin1=Nmin1+Nmin2;
         Nmin2=temp;
    }
    return (Nmin1+Nmin2);
 }
```

Bodovaný úkol

Napište v jazyku C rekurzivní funkci

```
int puleni(int cisla[], int a, int b, int hledane);
```

která pomocí metody půlení intervalu najde v zadaném setříděném poli **`cisla`** hodnotu **`hledane`** a vrátí její index v tomto poli.

Připomínám, že metoda půlení intervalu je založena na porovnání hodnoty hledaného čísla s číslem „uprostřed“ právě prohledávaného intervalu (v našem případě intervalu mezi prvky s indexy **`a`** a **`b`**). Pokud se hodnoty rovnají, našli jsme hledané číslo a můžeme tedy přímo vrátit jeho index. Pokud se nerovnají, stačí (rekurzivním voláním) prohledávat pouze jeden z intervalů prvek s indexem **`a`** až „prostřední prvek" nebo „prostřední prvek“ až prvek s indexem **`b`**.

Zdrojový soubor: zp2_01_rekurze_ukol.c