Prof. Dr. Franz Schweiggert   Institut für Angewandte Informationsverarbeitung   25. November 2008
Matthias Röhm   Blatt 6

Allgemeine Informatik III / Systemnahe Software I (WS 2008)

Abgabetermin: 02.12.2008

1 Aufgabe 1 (2 Punkte)

Welche Ausgabe erzeugt folgendes Programm?
Beschreiben sie die einzelnen Zugriffe auf die Vektoren $a1, a2, a3$ und b.

```
#include <stdio.h>

int main() {
        int i,j;
        int a1[3] = {1,2,3};
        int a2[3] = {4,5,6};
        int a3[3] = {7,8,9};
        int *b[9];
        int *c;

        for(i=0; i<9; i++) {
                if( i < 3 ) {
                        b[i] = &a1[i];
                }
                else if( i < 6 ) {
                        b[i] = &a2[i%3];
                }
                else {
                        b[i] = &a3[i%6];
                }
        }

        for(i=0; i<9; i++) {
                for(j=0; j<9; j++) {
                        if( *b[i] > *b[j] ) {
                                c = b[i];
                                b[i] = b[j];
                                b[j] = c;
                        }
                }
        }

        for(i=0; i<9; i++) {
                printf("%d ", *b[i]);
        }
        printf("\n");

        return 0;
}
```

2 Aufgabe 2 (8 Punkte)

In dieser Aufgabe sollen Sie einen einfachen Sortieralgorithmus ihrer Wahl implementieren. Ihre Funktion soll einen Vektor unabhängig von Datentyp und Länge aufsteigend sortieren können. Hierfür ist es erforderlich, dass für jeden Datentyp, den Sie sortieren wollen, zwei Funktionen vorhanden sind:

- **int compare(void *a, void *b)**. Diese Funktion vergleicht die als Zeiger übergebenen Werte. Sie gibt 1 zurück falls $*a > *b$, -1 falls $*a < *b$ und 0 falls $*a = *b$.
- **void swap(void *a, void *b)**. Diese Funktion vertauscht die als Zeiger übergebenen Werte.

Ihre Funktion erhält als Parameter diese beiden Funktionen (als Funktionszeiger), den zu sortierenden Vektor und eventuell weitere Parameter, die Sie für nötig erachten:

```
void sort( void *vector,
           int (*compare)(void *a, void *b),
           void (*swap)(void *a, void *b),
           ?
         ) {...}
```

Nach der Ausführung von *sort* soll der übergebene Vektor aufsteigend sortiert sein.

Nachdem Sie diese allgemeine Sortierfunktion implementiert haben, sollen Sie die folgenden Vektoren damit aufsteigend sortieren:

```
int vector_a = {123, 5, 36, 65, 13, 66, 21, 93, 3, 1, 7 , 8};
double vector_b = {0.1, 1.6, 56.2, 33.7, 0.5, 0.9, 6.3, 3.1, 5.2, 11.9};
```

Um diese Vektoren zu sortieren, müssen Sie natürlich die Funktionen *compare* und *swap* für *int* und *double* implementieren:

```
//Funktionen fuer int
int compare_int(void *a, void *b) {...}
```

```
void swap_int(void *a, void *b) {...}

//Funktionen fuer double
int compare_double(void *a, void *b) {...}
void swap_double(void *a, void *b) {...}
```

Hinweis:
Beachten Sie die besonderen Eigenschaften des *void** Zeigers. Da der Typ hier nicht bekannt ist, muss die Größe des zu verarbeitenden Typs bei der Zeigerarithmetik immer mit angegeben werden. Ebenso kann ein *void** Zeiger nicht ohne Angabe des Typs dereferenziert werden.

```
int a[3] = {1, 2, 3};
// v1 zeigt auf das erste Element von a.
void *v1 = (void *) a;

// Soll v2 auf das dritte Element von a Zeigen,
// muss v1 um "sizeof(int) * 2 = 8Byte" erhoeht werden.
void *v2 = v1 + sizeof(int) * 2;
printf("%d", *((int *)v2));
```

Viel Erfolg!