

Übungen Mikrocomputertechnik III, Blatt 1

Christian Ege

Wintersemester 2012

1 C-Programmierung

1.1

Schreiben Sie ein Programm welches folgende Operation (Multiply-Accumulate MAC) durchführt:

$$S_n = S_{n-1} + A \times B$$

Die Parameter A und B lesen Sie bitte aus der Datei foobar.txt ein und speichern die Werte in zwei Arrays (operand_a, operand_b). Die Datei foobar.txt finden Sie unter:

P:\\ce\\mc3\\aufgabe1

Sonderfälle bitte entsprechend kommentieren. Speichern Sie das Ergebnis in ein entsprechendes Array (result) ab.

Sie können zum Einlesen der Daten folgende Systemaufrufe verwenden:

```
FILE * fopen ( const char * filename , const char * mode );  
int fclose ( FILE * stream );  
size_t fread ( void * ptr , size_t size , size_t count , FILE * stream );  
char * fgets ( char * str , int num , FILE * stream );
```

Hilfe zu diesen Systemaufrufen finden Sie bei:

<http://cplusplus.com>

1.2

Schreiben Sie das Ergebnis Array (*result*) aus Aufgabe 1.1 in eine Datei (result.txt) bitte wie folgt formatiert:

$$S_n = S_{n-1} + A \times B$$

1.3

Machen Sie sich mit dem erzeugten Assembler-Listing vertraut. Das Assembler-Listing können Sie in Visual Studio wie folgt erzeugen:

Projekt-Options → C/C++ → Output Files → Assembler Output

Wählen Sie bitte die gemischte Darstellung von C/C++ Quelltext und Assembler-Listing.

- /FA: Assembly–Only Listing
- /FAcs: Assembly, Machine Code and Source
- /FAc: Assembly With Machine Code
- /FAs: Assembly With Source Code

1.4

Schreiben Sie Ihr Programm so um, dass es die Berechnung in einer Funktion `calculate_mac_operation` kapselt. Analysieren Sie auch hier Ihr Ergebnis mittels des Assembler Listings.

1.5

Messen Sie mittels des Funktionsaufrufs „`stop_stop_watch`“ aus „`stopwatch.h`“ die benötigte Zeit für die unterschiedlichen Phasen Ihres Programms.

```
/**
 * @brief Initializes the stop watch.
 *
 * This function checks also if everything is available
 * to work as expected.
 * The init function also starts the stop watch.
 *
 * @return EXIT_SUCCESS on success otherwise EXIT_FAILURE
 */
int init_stop_watch(void);

/**
 * @brief Starts the stop watch
 */
void start_stop_watch(void);

/**
 * @brief Stops the measurement and return the elapsed time in
 *        micro (10-6) Seconds.
 *
 * @return the ellapsed time in micro (10-6) Seconds since the
 *        last start.
 */
uint64_t stop_stop_watch(void);
```