

TGI Spezifikation: Gruppe 40 - Assembler  
Aufgabe 1.30

Benjamin Frank, Jan Birke, Andreas Spurny

10. Juni 2004

# Inhaltsverzeichnis

<b>1</b>	<b>Aufgabenstellung:</b>	<b>3</b>
<b>2</b>	<b>Spezifikation</b>	<b>3</b>
2.1	Signatur: . . . . .	3
2.1.1	Variablen: . . . . .	3
2.1.2	Rückgabewert: . . . . .	3
2.2	Erklärung . . . . .	3

# 1 Aufgabenstellung:

Es soll ein Feld von Vektoren elementweise vertauscht werden.

## 2 Spezifikation

### 2.1 Signatur:

```
dswap (int n, int incx, int incy, double* vektor_x, double* vektor_y);
```

#### 2.1.1 Variablen:

- int n:  
Die Anzahl der zu vertauschenden Elemente
- int incx:  
Die Schrittweite, mit der Vektor X durchschritten wird
- int incy:  
Die Schrittweite, mit der Vektor Y durchschritten wird
- double\* vektor\_x:  
Der Pointer auf vektor\_x [ ], also die Startadresse des Vektorfeldes
- double\* vektor\_y:  
Der Pointer auf vektor\_y [ ], also die Startadresse des Vektorfeldes

#### 2.1.2 Rückgabewert:

Entfällt, da direkt im Speicher getauscht wird. Die Arrays werden praktisch by 'Reference' übergeben.

## 2.2 Erklärung

Die Assembleroutine `dswap()` soll bestimmte Elemente zweier Vektoren vertauschen. Die Vektoren sind dabei Felder aus `double` Variablen. Für die Implementierung werden vom C-Programm die Startadressen der Felder mit übergeben (siehe Signatur).

Dabei wird am Schleifenanfang als erstes überprüft, ob `n` größer als 0 ist. Wenn nicht, wird die Unteroutine abgebrochen. Dann wird das erste Element des `vector_x` in ein entsprechend großes (32-bit) Register (`eax`) geladen. Genau dasselbe wird mit `vector_y` und `ebx` gemacht. Danach wird der Inhalt von `eax` nach `vector_y` und `ebx` nach `vector_x` geschrieben. Dann wird `n` um 1 decremented und `vector_x` und `vector_y` um `incx` und `incy` erhöht und zum Schleifenanfang zurück gesprungen.

Gegebenenfalls sollte man überprüfen, ob *incx* und *incy* größer als 0 sind. Da Fehlerbehandlung in Assembler generell viel Arbeit ist, sollte das jedoch im C-Rahmenprogramm passieren.