

## LV Rechnersysteme

### 5. Beleg

Seitenrahmenersetzungen:

- (a) Physikalischer Adressraum = 16KB  
Davon: 14 Seitenrahmen Programmtext  
2 Seitenrahmen Daten  
davon: 1 Seitenrahmen Konstanten  
1 Seitenrahmen Arrays  
Seitenrahmengröße = 1KB

Zahl der Seitenrahmenersetzungen:

- 3 Seitenrahmenersetzung für Laden von Main
  - 1 Seitenrahmenersetzung für Laden der Konstanten
  - 1. Schleife – beim 1. Schleifendurchlauf (Arrays außen vor):  
3 Seitenrahmenersetzung für Laden von F  
3 Seitenrahmenersetzung für Laden von FUN
  - 1. Schleife – pro Schleifendurchlauf:
    - i. A laden – 1 Ersetzung/Auslagerung des Array-Seitenrahmens
    - ii. B laden – 1 Ersetzung/Auslagerung des Array-Seitenrahmens
    - iii. C laden – 1 Ersetzung/Auslagerung des Array-Seitenrahmens
    - iv. Wieder A laden – 1 Ersetzung/Auslagerung des Array-Seitenrahmens
    - v. Wieder B laden – 1 Ersetzung/Auslagerung des Array-SeitenrahmensPro Schleifendurchlauf macht dies also 5 Ersetzungen.  
⇒ 5 Ersetzungen \* 1000 Schleifendurchläufe = 5000 Seitenrahmen-Ersetzungen
  - 2. Schleife – pro Schleifendurchlauf:
    - i. A laden – 1 Ersetzung/Auslagerung des Array-Seitenrahmens
    - ii. B laden – 1 Ersetzung/Auslagerung des Array-SeitenrahmensPro Schleifendurchlauf macht dies also 2 Ersetzungen.  
⇒ 2 Ersetzungen \* 1000 Schleifendurchläufe = 2000 Seitenrahmen-Ersetzungen
- ➔ Summa summarum werden 7010 Seitenrahmenersetzungen durchgeführt.

(b) Bezüglich Rechenzeit optimaler Programm-Ablauf:

```
X: =exp(az); T: =0; S: =0;

for I: =1 to 1000 do
begin
  A[I]: =F(K, I);
  S: =S+A[I];
end;

for I: =1000 downto 1 do { diese Schleife kann mit der vorigen nicht }
begin { verbunden werden, da S erst komplett }
  A[I]: =A[I]/S; { berechnet werden muss, um hier jedes }
end; { Element von A durch S zu dividieren... }

for I: =1 to 1000 do
begin
  B[I]: =FUN(X);
  T: =T+B[I];
end;

for I: =1000 downto 1 do { siehe oben, dasselbe gilt auch hier }
begin
  B[I]: =B[I]/T;
end;

for I: =1 to 1000 do
begin
  C[I]: =0;
end;
```

(c) Seitenrahmenersetzungen bei (b):

- 3 Seitenrahmenersetzungen für Laden von Main
  - 1 Seitenrahmenersetzung für Laden der Konstanten
  - 1. Schleife – beim 1. Schleifendurchlauf:  
3 Seitenrahmenersetzungen für Laden von F
  - 3. Schleife – beim 1. Schleifendurchlauf:  
3 Seitenrahmenersetzungen für Laden von FUN
  - 1.+2.+3.+4. Schleife – Ersetzungen für A und B:  
In den 1000 Iterationen einer der Schleifen 1 und 3 muss A bzw. B insgesamt 4 Seitenrahmenersetzungen unterzogen werden, da beide Arrays 4KB groß ist, aber nur 1 Seitenrahmen mit 1KB für Arrays zur Verfügung steht. In den Schleifen 2 und 4 benötigt man durch das Herunterzählen hingegen nur jeweils 3 Seitenrahmenersetzungen, da zu Beginn der Schleifen nicht ausgelagert wird.
  - 5. Schleife – Ersetzungen für C:  
In den 1000 Iterationen finden wie für A und B hier für C 4 Seitenrahmenersetzungen statt.
- Summa summarum werden nur 28 Seitenrahmenersetzungen durchgeführt.

(d) Dass so ein gewaltiger Unterschied an Seitenrahmenersetzungen zwischen (a) und (c) verbuchen lässt ist auf den Umstand zurück zu führen, dass bei (a) in jedem Schleifendurchlauf mehrere Seitenersetzungen durchzuführen waren. Da die Schleifen je 1000 mal aufgerufen werden, bedeutet dies eine hohe Anzahl an Seitenrahmenersetzungen. Bei (b) und (c) hingegen werden maximum 4 Ersetzungen für Arrays pro Schleife vorgenommen, nicht aber in jedem einzelnen Schleifendurchlauf. Dies wird durch eine Aufgliederung der vorigen Schleifen in viele kleinere ermöglicht, was ein wenig zu Kosten der Übersichtlichkeit geht, vom Prinzip der wenigsten Seitenersetzungen hier allerdings optimal ist.