

– Praktikumsaufgabe 6 –

Thema: *Numerische Berechnungen im Shellskript, Brace Expansion*

Zielstellung: Erlernen des Einsatzes einfacher numerischer Ausdrücke und Berechnungen in der Bash, Verwendung von Feldern, Beherrschung des Mechanismus der Ersetzung geschweifeter Klammern (*Brace Expansion*)

1. Schreiben Sie ein Shellskript, das die Grundrechenarten mit 2 Operanden beherrscht. Der Aufruf soll so erfolgen:

```
robge@ipaetz2:~$ ./calc.sh 12345679 \* 72
8888888888
```

Das Skript soll die Gültigkeit des Operatorsymbols testen, d. h., Lösungen der Art

```
#!/bin/bash
echo $(( $1$2$3 ))
```

sind etwas zu simpel. **Hinweise:**

- Variablen in der Bash sind ungetypt, sie nehmen je nach initialer Zuweisung Zeichenketten oder Integer-Werte auf.
 - Arithmetische Operationen (auch Zuweisungen!) müssen entweder in `(())` stehen, oder ihnen muss das Schlüsselwort `let` vorangestellt sein. `let sum=$a+$b` ist also äquivalent zu `((sum=$a+$b))`.
 - Achtung: Bei `let` dürfen keine Leerzeichen zwischen Operanden und Operator stehen
 - Mechanismen zum Datenaustausch zwischen ‘Hauptprogramm’ und Funktion haben wir im vorigen Praktikum erlernt.
2. a) Schreiben Sie ein Shellskript, das die Reihe der Fibonacci-Zahlen von 1 bis 1.000.000 ausgibt. Diese sind folgendermaßen definiert:

$$\text{fib}(n) = \begin{cases} 0 & : n = 0 \\ 1 & : n = 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & : n \geq 2 \end{cases}$$

Literatur: <http://de.wikipedia.org/wiki/Fibonacci-Reihe>

- b)* Implementieren Sie ein Shellskript, das zu einem gegebenen n die zugehörige Fibonacci-Zahl ermittelt. Nutzen Sie eine Funktion, die *rekursiv* gerufen wird.

- 3.* Schreiben Sie ein Shellskript, das das Sieb des Eratosthenes implementiert. Schreiben Sie dann ein C-Programm und ermitteln Sie den Geschwindigkeitsgewinn, indem Sie die Zeit für das Durchsuchen der Zahlen von 1 bis 1.000.000 für beide Versionen ermitteln.

Hinweis: Sie benötigen für den Algorithmus ein eindimensionales Array, was die Bash interessanterweise bietet.

Literatur: http://de.wikipedia.org/wiki/Sieb_des_Eratosthenes

4. Zur flexiblen Generierung von Listen kennt die Shell den Mechanismus der Brace Expansion. Innerhalb von geschweiften Klammern angegebene und durch Komma getrennte Elemente werden permutiert. Der Mechanismus erklärt sich am besten mit einigen Beispielen:

```
robge@ipaetz2:~$ echo {a,b,c}rgl
argl brgl crgl
robge@ipaetz2:~$ echo {a,b,c}{d,e,f}
ad ae af bd be bf cd ce cf
robge@ipaetz2:~$ cp foo.{txt,bak}
robge@ipaetz2:~$ ls foo*
foo.bak  foo.txt
robge@ipaetz2:~$ echo {a..d}{1..4}
a1 a2 a3 a4 b1 b2 b3 b4 c1 c2 c3 c4 d1 d2 d3 d4
```

Schreiben Sie (kurze) Brace-Ausdrücke zur Generierung

- a) aller Zeichenketten, die aus vier Kleinbuchstaben bestehen
- b) aller Zeichenketten, die aus 1-3 Buchstaben bestehen
- c) aller Rechner der Labore 136c und 146a (isys101 ... isys121, isys6, isys8, isys10 ... isys29),
- c)* bb bbbb aa aabb aabbbb aaaa aaaabb aaaabbbb
- d) Was generiert der Ausdruck `b{a{d,n,k},ed}s?`