

## Shell Skripte unter Linux

### Teil 1: Grundlagen

Shell Skripte sind Kommandoabfolgen, vergleichbar von Batch-Dateien unter Windows. So wurde z.B. früher beim Systemstart von Win9x die Datei autoexec.bat automatisch ausgeführt.

Allerdings bieten Linux-/Unix-Shell Skripte deutlich mehr Möglichkeiten. Unter Ubuntu können Sie z.B. für jeden User, im Home Verzeichnis in der Datei .bash\_profile, ein Startskript hinterlegen. Die Leistungsfähigkeit der Shell Skripte ist stark von der verwendeten Shell (z.B. Bourne again-Shell) abhängig.

Ein Shell Skript ist eine Textdatei, in der Kommandoabfolgen gespeichert sind und das mit Parametern aufgerufen werden kann. Ziel von Shell Skripten ist die Arbeitserleichterung durch Zusammenfassung vieler Befehle in einem oder auch die Automatisierung von Befehlsabfolgen zu bestimmten Zeitpunkten (z.B. cron-Jobs).

**Aufruf:** `bash myscript`

Wenn der Skript-Datei das Execute-Recht zugewiesen wird (`chmod +x myscript`), kann das Skript wie ein normaler Befehl durch einen Benutzeraufgerufen werden. Als root: `./myscript`

Damit das Betriebssystem für das Skript auch die richtige Shell verwendet, steht in der ersten Zeile eines Skriptes folgende Zeichenkette:

`#!/bin/sh` oder `#!/bin/bash`

... je nachdem, welche Shell zum Einsatz kommen soll.

Zu Beginn ein einfaches Beispiel:

```
#!/bin/bash
# Testbeispiel
echo Hallo Welt!
echo Datum, Uhrzeit und Arbeitsverzeichnis:
date
pwd
echo Kommando: $0
echo Übergabe - Parameter: $*
echo Übergabe - Parameter1: $1
echo Übergabe - Parameter2: $2
echo Summe - Parameter: $#
echo Alle-Parameter: $@
```

## Übung 1:

Öffnen Sie einen Editor und erstellen Sie das obige Skript in der Datei my1script.  
Anschließend rufen Sie das Programm mit `bash my1script pinguin tux` auf!  
Notieren Sie die Bildschirmausgabe des Shellskriptes:


Probieren Sie den Aufruf des Skriptes mit dem -x Parameter `bash -x myscript`.  
Was beobachten Sie?


### Weiterführende Übungen:

1. Schreiben Sie ein Shell-Skript, das testet, ob eine Datei gelesen, beschrieben und/oder ausgeführt werden kann. Ferner ob es sich um ein Verzeichnis oder normale Datei handelt.
2. Schreiben Sie ein Shell-Skript, das überprüft, ob die Datei mit dem angegebenen Namen leer ist oder nicht. Wenn nicht, dann zeigen Sie sie an, andernfalls geben Sie eine Fehlermeldung aus.
3. Schreiben Sie ein Shell-Skript, das einen Anwendernamen als Parameter erhält und dann den Login-tty-Port des angegebenen Anwenders ausgibt.