

Programm- oder Funktionsrückgabe auswerten

In Bash-Skripten werden oft externe Programme und Tests ausgeführt deren Rückgabewerte man kontrollieren möchte, z.B. ob das Programm gelaufen ist oder ob es einen Fehler gab.

Folgendes wird gerne verwendet, ist aber für einen einfachen Test etwas umständlich:

```
# Beispiel: gibts einen bestimmten Benutzer?
grep -q username /etc/passwd
if (($? > 0)); then
    echo 'Benutzer wurde nicht gefunden' >&2
    exit 1
fi
```

Schneller und besser zu lesen ist dieser Code:

```
if grep -q username /etc/passwd; then
    # nix machen
    :
else
    echo 'Benutzer wurde nicht gefunden' >&2
    exit 1
fi

# oder noch kürzer mit Negation
if grep -q username /etc/passwd; then
    echo 'Benutzer wurde nicht gefunden' >&2
    exit 1
fi
```

Alternativ gehts auch mit einer || (ODER, wenn der Rückgabewert nicht 0 ist) oder && (UND, wenn der Aufruf eine 0 zurückgibt) Verknüpfung:

```
grep -q username /etc/passwd || {
    echo 'Benutzer wurde nicht gefunden' >&2
    exit 1
}
```

Die Auswertung von \$? ist damit eigentlich nur notwendig, wenn man tatsächlich auf einen ganz bestimmten Rückgabewert prüfen will. rsync etwa liefert für viele Fehler einen bestimmten Code:

- 0 Success
- 1 Syntax or usage error
- 2 Protocol incompatibility
- 3 Errors selecting input/output files, dirs
- 4 Requested action not supported: an attempt was made to manipulate 64-bit files on a platform that cannot support them; or an option was specified that is supported by the client and not by the server.
- 5 Error starting client-server protocol
- 6 Daemon unable to append to log-file
- 10 Error in socket I/O
- 11 Error in file I/O
- 12 Error in rsync protocol data stream
- 13 Errors with program diagnostics
- 14 Error in IPC code
- 20 Received SIGUSR1 or SIGINT
- 21 Some error returned by waitpid()
- 22 Error allocating core memory buffers
- 23 Partial transfer due to error
- 24 Partial transfer due to vanished source files
- 25 The --max-delete limit stopped deletions
- 30 Timeout in data send/receive
- 35 Timeout waiting for daemon connection

So kann z.B. nur der Fehler in ein Logfile geschrieben werden:

```
rsync -a --delete quelle ziel
if (($? == 30)); then
    echo "Timeout in data send/receive" >> $logfile
    exit 1
fi
```

Revision #1

Created 2021-07-27 10:22:28 UTC by magenbrot

Updated 2021-07-27 10:23:22 UTC by magenbrot