

# Bashscripting 106

„Nichts für die Ewigkeit“

von

Marius Schwarz

# Bashscripting 106

## Basic Bashscripting

# Bashscripting 106

Temporäre Dateinamen gehören zu den nützlichen Dingen, die man in BashScripten sehr häufig benötigt.

Um diese sorgfältig herzustellen, gibt es verschiedene Wege.

# Bashscripting 106

Eigenschaften eines temporärer Dateinamens

# Bashscripting 106

## Eigenschaften eines temporärer Dateinamens

1. Die Datei liegt im `/tmp/`

# Bashscripting 106

## Eigenschaften eines temporärer Dateinamens

1. Die Datei liegt im `/tmp/`
2. Der Dateiname hat ein Zufallselement

# Bashscripting 106

## Eigenschaften eines temporärer Dateinamens

1. Die Datei liegt im `/tmp/`
2. Der Dateiname hat ein Zufallselement
3. Der Dateiname enthält **keine** Sonderzeichen wie `$%&` etc.

# Bashscripting 106

## Eigenschaften eines temporärer Dateinamens

1. Die Datei liegt im `/tmp/`
2. Der Dateiname hat ein Zufallselement
3. Der Dateiname enthält **keine** Sonderzeichen wie `$%&` etc.
4. **Er wird am Ende des Script wieder gelöscht!**



# Bashscripting 106

## Die BASH Variable **\$RANDOM**

```
t[root ~]# echo $RANDOM  
14663  
[root ~]# echo $RANDOM  
30598  
[root ~]# echo $RANDOM  
10168
```

...erzeugt laufend andere Zufallszahlen.

# Bashscripting 106

aber wie macht man daraus einen Dateinamen?

# Bashscripting 106

Als erstes fügen wir vorn mal `/tmp/` ein,  
dann brauchen wir noch einen **Basisnamen** für unsere Datei.

z.B. „`/tmp/programmname...`“

# Bashscripting 106

Die \$RANDOM Variable kann man in andere Strings einfach einbauen:

```
[root ~]# echo "/tmp/.myprogname.$RANDOM"  
/tmp/.myprogname.18986  
[root ~]# echo "/tmp/.myprogname.$RANDOM$RANDOM"  
/tmp/.myprogname.890934248
```

doppelt geht auch.

# Bashscripting 106

Natürlich kann man den String auch in eine Variable schreiben:

```
[root ~]# TEMPNAME="/tmp/myprogsname.$RANDOM"  
[root ~]# echo $TEMPNAME  
/tmp/myprogsname.23930
```

# Bashscripting 106

Im Script sieht das dann z.b. so aus:

```
#!/bin/bash

TEMPNAME="/tmp/myprogsname.$RANDOM"

echo "Hagelsturm 2019" > $TEMPNAME
echo "Inhalt von Tempdatei $TEMPNAME:"

cat $TEMPNAME

rm -f $TEMPNAME
```

# Bashscripting 106

Es geht aber auch völlig anders

# Bashscripting 106

Es gibt einen Linuxbefehl namens *mktemp*:

```
[root ~]# mktemp /tmp/test-XXXX
/tmp/test-GEHi
[root ~]# ls -ls /tmp/test-GEHi
0 -rw----- 1 root root 0 16. Okt 10:14 /tmp/test-GEHi
[root ~]#
```



# Bashscripting 106

Mit `mktemp` lassen sich Dateien und Verzeichnisse sicher anlegen.

# Bashscripting 106

`mktemp` prüft,

**ob** die Datei oder das Verzeichnis **bereits existiert**.

Das ist etwas, was wir in unserem Script nicht getan haben.

# Bashscripting 106

In einem Script sieht das dann so aus:

```
#!/bin/bash

TEMPNAME="$(mktemp /tmp/myprogsname.XXXXXXX)"

echo "Hagelsturm 2019" > $TEMPNAME
echo "Inhalt von Tempdatei $TEMPNAME:"

cat $TEMPNAME

rm -f $TEMPNAME
```

# Bashscripting 106

Was passiert,  
wenn man das mit dem Löschen vergisst?

# Bashscripting 106

Das könnte dann so aussehen:

```
[root ~]# ll /tmp/  
insgesamt 180  
-rw----- 1 root root 57 31. Aug 00:00 dehydrated-05EXcq  
-rw----- 1 root root 176 26. Aug 00:00 dehydrated-2hU2ql  
-rw----- 1 root root 27 21. Sep 00:00 dehydrated-3ragJO  
-rw----- 1 root root 0 11. Okt 00:00 dehydrated-4f5Ntm  
-rw----- 1 root root 176 21. Aug 00:00 dehydrated-4qXI9V  
-rw----- 1 root root 57 11. Aug 00:00 dehydrated-5pGTdA  
-rw----- 1 root root 51 11. Sep 00:00 dehydrated-7rOFhp  
-rw----- 1 root root 0 11. Okt 00:00 dehydrated-8cl89p  
-rw----- 1 root root 51 11. Aug 00:00 dehydrated-Aoylsh  
-rw----- 1 root root 176 1. Sep 00:00 dehydrated-aUqfQU  
-rw----- 1 root root 51 6. Sep 00:00 dehydrated-CdqG0b  
-rw----- 1 root root 57 1. Sep 00:00 dehydrated-cMQISX  
-rw----- 1 root root 57 1. Aug 00:00 dehydrated-e0aOpY  
-rw----- 1 root root 330 26. Sep 00:00 dehydrated-f3eFxx  
-rw----- 1 root root 57 16. Aug 00:00 dehydrated-FGRjah  
-rw----- 1 root root 176 31. Aug 00:00 dehydrated-fqShhB  
-rw----- 1 root root 176 11. Sep 00:00 dehydrated-hDxMBX  
-rw----- 1 root root 176 6. Sep 00:00 dehydrated-iclVzi  
-rw----- 1 root root 330 28. Sep 23:16 dehydrated-j0zZgO  
-rw----- 1 root root 112 26. Sep 00:00 dehydrated-J4aKZD  
-rw----- 1 root root 51 1. Aug 00:00 dehydrated-kEiQQ7  
-rw----- 1 root root 27 28. Sep 23:16 dehydrated-LOVKQF  
-rw----- 1 root root 55 6. Okt 00:00 dehydrated-ME948G  
-rw----- 1 root root 57 11. Sep 00:00 dehydrated-MldsVL  
-rw----- 1 root root 57 21. Aug 00:00 dehydrated-NAS60Z  
-rw----- 1 root root 55 11. Okt 00:00 dehydrated-Nw6eEp  
-rw----- 1 root root 57 26. Aug 00:00 dehydrated-nxNel9  
-rw----- 1 root root 112 28. Sep 23:16 dehydrated-OdQwW9
```

```
-rw----- 1 root root 27 26. Sep 00:00 dehydrated-R3dZxX  
-rw----- 1 root root 55 16. Okt 00:00 dehydrated-rxI2Ub  
-rw----- 1 root root 51 26. Aug 00:00 dehydrated-TYDw01  
-rw----- 1 root root 51 21. Aug 00:00 dehydrated-ugqgUC  
-rw----- 1 root root 51 1. Sep 00:00 dehydrated-vaXNJa  
-rw----- 1 root root 57 6. Sep 00:00 dehydrated-VWxfR6  
-rw----- 1 root root 51 6. Aug 00:00 dehydrated-vzkkXg  
-rw----- 1 root root 55 30. Sep 12:50 dehydrated-Wngimu  
-rw----- 1 root root 55 1. Okt 00:00 dehydrated-XqRxKP  
-rw----- 1 root root 57 6. Aug 00:00 dehydrated-xWr76C  
-rw----- 1 root root 51 16. Aug 00:00 dehydrated-Ys0BKN  
-rw----- 1 root root 51 31. Aug 00:00 dehydrated-zW8J15
```

# Bashscripting 106

Ergo, nie das Aufräumen vergessen!