# Operating Systems

Van Marcel Bos – HBO deeltijd jaar 4  
Practicum 2

### De code

De code is als bestand in de repository beschikbaar.

### Uitleg van de code:

Zie het filmpje voor een uitleg en de werking van het script.

Ik zal hieronder uitleggen wat elke regel doet.

#!/bin/bash

# De eerste parameter is de directory waar de foto’s staan.

# De tweede parameter bestaat uit de aanduiding “maand” of “week”.

Hieronder wordt een check gedaan of parameter 2 goed gevuld is met week of maand, anders wordt er een tekst geprind.

if [ $2 != "week" ] && [ $2 != "maand" ]; then

echo "Parameter 2 is niet of niet goed ingevuld"

else

Als parameter 2 goed gevuld is, dan wordt er gekeken of parameter 1 daadwerkelijk een folder is.

if [ -d "$1" ]; then

Er wordt een teller gezet om straks de progressie te tonen

TELLER=0

Vervolgens worden afbeeldingen gezocht binnen de $1 folder als week dan 7 dagen als maand dan 30 dagen.

if [ $2 == "week" ]; then

AFBEELDINGEN=$(find "$1" -iname "\*.jpg" -o -iname "\*.png" -atime +7)

fi

if [ $2 == "maand" ]; then

AFBEELDINGEN=$(find "$1" -iname "\*.jpg" -o -iname "\*.png" -atime +30)

fi

Het aantal afbeeldingen (regels) wordt geteld om later te gebruiken en de progressie te tonen.

AANTAL=$(echo "$AFBEELDINGEN" | wc -l)

Er wordt een door de afbeelding geloopt

for X in $AFBEELDINGEN; do

De teller voor de progressie wordt opgehoogd

TELLER=$((TELLER + 1))

Ik maak het scherm leeg om duidelijk te laten zien hoe ver het script is.

clear

# week of maand nummers mappen aanmaken

Hier haal ik de datums op van elk bestand om de datum variabele te vullen die ik ga gebruiken voor de folders.

if [ $2 == "week" ]; then

DATUM=$(date -r "$X" +%W)

fi

if [ $2 == "maand" ]; then

DATUM=$(date -r "$X" +"%m%y")

Ik doe hier de afbeelding printen

echo "$X"

Ik doe hier de actie printen die gedaan wordt

echo "Kopiëren van afbeelding $X naar fotos/$DATUM"

Ik toon hier de voortgang aan de gebruiker.

echo "Voortgang: $TELLER van $AANTAL"

# Als map nog niet bestaat dan aanmaken

Op het moment dat de folder nog niet bestaat wordt de map in fotos aangemaakt

mkdir -p "fotos/$DATUM"

Daarna wordt de afbeelding gekopieerd naar de folder

cp "$X" "fotos/$DATUM"

Vervolgens wordt de md5sum van de oorspronkelijke afbeelding gepakt en dan alleen de hash gepakt

FILE1=$(md5sum $X | awk '{ print $1 }')

Ik heb de basename nodig om vervolgens de foto te zoeken in de nieuwe map waar hij net geplaatst is. Ik wist niet hoe ik anders de net gekopieerde foto terug moest halen. Maar dit werkt goed.

BESTAND=$(basename $X)

BESTAND2=$(find fotos/$DATUM -name "$BESTAND")

Zodra ik het bestand heb gevonden pak ik ook van bestand 2 de md5sum op dezelfde manier als de eerste

FILE2=$(md5sum $BESTAND2 | awk '{ print $1 }')

Ik print vervolgens de beide md5 sums..

echo "Zijn bestanden gelijk?> $FILE1 en $FILE2"

en ik doe de check

if [[ $FILE2 == $FILE1 ]]; then

echo "$BESTAND = Bestanden $FILE1 en $FILE2 zijn gelijk . Bestand 1 wordt nu verwijderd."

Als ze gelijk zijn verwijder ik vervolgens de foto in de oorspronkelijke locatie

rm $X

fi

Ik doe even een sleep tussendoor, zodat het niet te snel gaat voor de eindgebruiker. Vermoedelijk gaat het ook prima zonder sleep :P Maar dan kon ik het niet laten zien in de demonstratie film

sleep 0.1

done

else

Mocht de folder geen folder zijn dan houdt hij hier al op. De check op folder werkt overigens ook als je een symbolisch link in parameter $1 gebruikt.

echo "Parameter 1: $1 is geen folder of symbolische link"

fi

fi

Einde van practicum deel 2 😊