Wir verfassen ein GPT-Backup-Skript

1. Aufgabe

Wichtige BASH-Befehle, die im Skript verwendet werden nachfolgend in

Stichworten kurz erläutert:

lshw: listet die Hardware, die in einem PC verbaut ist auf! Die Parameter -class und -short grenzen die Ausgabe ein. Hier z.B. auf die Hardwaretype „disk".

grep: greift aus einem Datenstrom bzw. Datei nur Zeilen heraus, die einen bestimmten Ausdruck bzw. Ein bestimmtes Text- und Zahlenmuster (Pattern) enthalten.

sed: schneidet z. B. ein Textmuster aus und ersetzt diesen Bereich durch einen neuen Inhalt. Mit **"s/** wird die Stringbearbeitung eingeleitet. Mit **.*\/** werden alle beliebigen Zeichen **(.)** in beliebiger Anzahl **(*)** einer Zeichenkette bis zu einem Schrägstrich **(/)** ausgeschnitten und durch ein „Nichts" ersetzt **(//)**

|: Den senkrechten Strich-Befehl nennt man „Pipe" (=englisch Leitung). Damit wird gemeint, dass ein Datenstrom z. B. eine Zeichen- bzw. Textausgabe als „Input-Stream" dem nachfolgenden Befehl als Eingabestrom zugeführt wird.

$$: Ist die Prozess-Nr des gerade laufenden Skriptes. Jedes Skript erhält eine eigene Prozess-Nr. Verwendet man diese Nr im Dateinamen im Verzeichnis /tmp, so wird dadurch verhindert, dass zwei Prozesse die gleiche Datei aus dem /tmp-Verzeichnis zur Datenzwischenspeicherung verwenden.

echo: gibt einen Text bzw. Meldung aus! Mit dem Parameter **-e** können Erweiterungen genutzt werden. So z. B. „\n" für einen Zeilenumbruch oder „\c" für eine Unterdrückung des Zeilenumbruchs.

cat: kann eine oder mehrere Dateien zusammenfügen und anschließend an die Standardausgabe senden z. B. einfach auf dem Bildschirm ausgeben. Der Bildschirm/Konsole ist die Standardausgabe!!

rm:

gdisk:

Die ***if-***Anweisung

read:

>

((...)):

parted -ms .... print:

wc:

`**.....Befehl ....**`

IFS:

2. Aufgabe

Erläutern Sie die Funktion des nachfolgenden Skriptes!

```
#!/bin/bash
lshw -class disk -short | grep "/dev/" | sed "s/.*\///g" | sed "s/disk//" >
/tmp/platten$$
echo "Sie haben hardwaretechnisch auf nachfolgende Fesplatten Zugriff:"
echo "------------------------------"
cat /tmp/platten$$
echo "------------------------------"
```

```
echo "Geben Sie bitte die Festplatte, von der der GPT gesichert werden soll, an!"
echo "Antworten Sie mit der Kurzbezeichnung aus obiger Auflistung wie z.B.: \"sda\" "
echo "Von welcher Platte soll der GPT in einer Backup-Datei gesichert werden?"
echo -e "Von der Festplatte: \c"
read platte
rm /tmp/platten$$
if [[ ! -a /dev/$platte ]]
then
   echo "Du Döddele sollst doch eine vorhandene Platte mit GPT auswählen!"
   exit 5
fi
if gdisk -l /dev/$platte | grep "MBR: MBR only" > /dev/null
then
   echo "Die Festplatte /dev/$platte enthält keinen GPT, sondern nur einen MBR!"
   exit 7
fi
 # Wir berechnen die Anzahl „a“ der gpt Partitionen:
 ((a=`parted -ms /dev/$platte print | wc -l`-2))
 echo "Die Festplatte /dev/sda hat $a Partitionen"
 echo "Wir berechnen jetzt die Größe des GPTs"
 ((Tabellengroesse=128*$a))
 ((gpt_groesse=1024+$Tabellengroesse))
 echo "GPT-Größe=$gpt_groesse"
 echo -e "Geben Sie hier den GPT-Speicher-Backup-Pfad an: \c"
 read backuppfad
 IFS="/"
 set $backuppfad
 IFS=" "
 echo "$@"
 pfad1=""
 while [[ $# -gt 1 ]]
   do
      pfad1="${pfad1}$1/"
      shift
   done
 echo "Der Pfad zu Backup-Verzeichnis lautet: $pfad1"
 backupdatei=$1
 if [[ ! -d $pfad1 ]]
 then
    ### echo -e "Der Pfad zu Backup-Verzeichnis\n \"$pfad1\" existiert
    ## else
    echo "Der angegebene Backuppfad existiert nicht!"
    echo "Legen Sie zuerst den Backuppfad an!"
    exit
 fi
 if [[ -d $backuppfad ]]
 then
    echo "Sie müssen einen vollständige Angabe in der Form:"
    echo "\"/Verzeichnis1/\""
    exit
 fi
 if [[ -f $backuppfad ]]
 then
    echo -e "Soll die vorhandene Datei $backuppfad \nüberschrieben werden (Y/N)? \c"
    read X
 else
    X="Y"
 fi
 if [[ "$X" == "Y" || "$x" == "y" ]]
    then
      # echo -e "Vergleiche mit:\n$backuppfad"
      echo -e "Die Backupdatei: \"${backuplfad}\" wird neu erzeugt!"
      dd if=/dev/sda count=1 bs=$gpt_groesse of=$backuppfad
 else
    echo "Das Backup kann ohne gültige Pfadangabe nicht erstellt werden!"
 fi
```

Wir verfassen ein GPT-Backup-Skript

1. Aufgabe

Wichtige BASH-Befehle, die im Skript verwendet werden nachfolgend in

Stichworten kurz erläutert:

lshw: listet die Hardware, die in einem PC verbaut ist auf! Die Parameter -class und -short grenzen die Ausgabe ein. Hier z.B. auf die Hardwaretype „disk".

grep: reift aus einem Datenstrom bzw. Datei nur Zeilen heraus, die einen bestimmten Ausdruck bzw. Ein bestimmtes Text- und Zahlenmuster (Pattern) enthalten.

sed: schneidet z. B. ein Textmuster aus und ersetzt diesen Bereich durch einen neuen Inhalt. Mit **"s/** wird die Stringbearbeitung eingeleitet. Mit **.*\/** werden alle beliebigen Zeichen **(.)** in beliebiger Anzahl **(*)** einer Zeichenkette bis zu einem Schrägstrich **(/)** ausgeschnitten und durch ein „Nichts" ersetzt **(//)**

|: Den senkrechten Strich-Befehl nennt man „Pipe" (=englisch Leitung). Damit wird gemeint, dass ein Datenstrom z. B. eine Zeichen- bzw. Textausgabe als „Input-Stream" dem nachfolgenden Befehl als Eingabestrom zugeführt wird.

$$: st die Prozess-Nr des gerade laufenden Skriptes. Jedes Skript erhält eine eigene Prozess-Nr. Verwendet man diese Nr im Dateinamen im Verzeichnis /tmp, so wird dadurch verhindert, dass zwei Prozesse die gleiche Datei aus dem /tmp-Verzeichnis zur Datenzwischenspeicherung verwenden.

echo: gibt einen Text bzw. Meldung aus! Mit dem Parameter -e können Erweiterungen genutzt werden. So z. B. „\n" für einen Zeilenumbruch oder „\c" für eine Unterdrückung des Zeilenumbruchs.

cat: kann eine oder mehrere Dateien zusammenfügen und anschließend an die Standardausgabe ausgeben.

rm: löscht eine Datei oder mit der Option -r ein Verzeichnis rekursiv.

gdisk: ist ein Partidionierungstool, das mit dem "-l" Parameter die Festplattenpartitionierung ausgibt!

Die ***if-***Anweisung leitet eine Verzweigung ein, die von der Bedingung in den Klammern [[abhängig ist. Das "!"-Zeichen invertiert das Ergebnis der logischen Abfrage. "-a" fragt, ob ein Zugang zur Datei oder einem Datenstrom entstehen würde.

read: fordert den Bediener auf eine Eingabe vorzunehmen, diese wird dann in die nachfolgende Variable gespeichert.

> Der Pfeil ">" leitet einen Datenstrom (Ausgabe) z. B. in eine Datei oder einen TCP-Socket etc. um.

((...)): Innerhalb doppelter runder Klammern sind einfache Integer-Rechnungen möglich!

parted -ms .... print: gibt Auskunft über die Partitionen einer Festplatte.

wc: ählt die Wörter einer Datei bzw. Dateiausgabe. Mit dem Parameter "-l" werden die Zeilen gezählt!

`.....Befehl ....` Die Back-Ticks stellen eine Substitution dar, d. h. der Text zwischen den `Back-Ticks` wird als Befehl ausgeführt und das Ergebnis der Verarbeitung wird an die Stelle des "Back-Tick-Strings" zur weiteren Verarbeitung verwendet.

IFS: Der IFS "Internal Field Separator" enthält das Trennzeichen, das der "set"-Befehl zum aufsplitten eines übergebenen Zeichen-Strings in einzelne Positionsparameter anwendet.