



gramm bis zur Schleife sieht dann folgendermaßen aus.

```
#!/bin/sh
# Anzeige aller Dateien laut übergebenem
Muster
# geordnet nach der Änderungszeit, die
ältesten zuerst
#
ls -ltr -l tmp.$$ $* | tee tmp.$$
#
summe=0
anzahl=0
```

Aus der Datei `tmp.$$` wird nun die Information, wie groß die einzelnen Dateien sind mit Hilfe des Kommandos `cut` herausgeschnitten.

Die Information über die Größe der einzelnen Dateien steht an den Position 33 bis 41. Mit Hilfe von `cut` kann aus der Datei `tmp.$$` eine Liste von Zahlen erstellt werden, die der Größe der vorkommenden Dateien entspricht. (die notwendige Option `-b` steht in der `man-page`).

```
$(cut tmp.$$ -b 33-41)
```

Das führende `$` teilt der Shell mit, dass der Inhalt zwischen den beiden runden Klammern auszuwerten ist, und mit diesem Ergebnis der angegebene Ausdruck zu ersetzen ist. Die so erhaltene Liste der Zahlen wird nun mit einer `for`-Schleife verarbeitet. Für jedes Element der in der `for`-Anweisung angegebenen Liste werden die nachfolgenden Befehle verarbeitet. (Nachfolgende Befehle sind jene, die zwischen dem `for`-Anweisung folgen.

dem `do` und dem entsprechendem `done` stehen.)

Die Zeile lautet :

```
for i in $(cut tmp.$$ -b 33-41);
do
....
done
```

Leider produziert der `ls`-Befehl an den Positionen 33-41 nicht nur Zahlen, sondern manchmal auch unerwünschten Text. Um etwaige Fehlermeldungen vorzubeugen, ist es sinnvoll, mit Hilfe der Parametersubstitution etwaigen Text zu ersetzen. Die Syntax dafür lautet: `${var_muster}` (zu finden ist dies in der (sehr langen) `man-page` der `bash` (`man bash`)). Benötigt wird: falls ein Zeichen vorkommt, das keine Ziffer ist, oder falls ein Leerzeichen vorkommt, so soll der gesamte Ausdruck gelöscht werden. `${i##*[!0-9, ']*}` erfüllt diese Forderung. Die Programmzeile lautet:

```
i=${i##*[!0-9, ']*}
```

Nur wenn `i` nun irgendwelche Zeichen erhält, soll aufsummiert werden.

```
if [ i ]; then
summe = ${summe + $1}
anzahl = ${anzahl + 1}
fi
done
```

Jedes `if` endet mit `fi`, nach dem Ende der Bedingung kommt ein `;` gefolgt vom Schlüsselwort `then`. Um in der `bash` auf den Wert einer Variablen zugreifen zu

können, muss vor dem Variablennamen ein `$` vorangesetzt werden. Für die Zuweisung selbst, genügt die Angabe des Variablennamens. Das abschließende `done` gehört zum `for`.

Nun folgt die formatierte Ausgabe der beiden Werte. Da große Werte nicht leicht lesbar sind, sollen ein Punkt, bzw. ein Beistrich nach jeweils drei Ziffern geschrieben werden. `printf` wird verwendet, damit auch führende Nullen in Dreier-Gruppen angezeigt werden.

```
if [ $summe -ge 1000000 ]; then
#Speicherverbrauch ist größer als
1.000.000
echo -n "$summe / 1000000".
printf '%.3d.' $[(($summe %
1000000)/1000)
printf '%.3d' $[(($summe % 1000)]
elif [ $summe -ge 1000 ]; then
#Speicherverbrauch ist größer als 1.000
echo -n "$summe / 1000".
printf '%.3d' $[(($summe % 1000)]
else
```

```
#normal anzeigen
echo -n $summe
fi
echo " Byte"
#und nun die Anzahl der Dateien ausgeben
echo "das sind $anzahl dateien"
```

Damit die temporäre Datei nicht zu einer permanenten wird, muss sie nun abschließend gelöscht werden.

```
rm tmp.$$
```

```
#!/bin/sh
# Anzeige aller Dateien laut übergebenem Muster
# geordnet nach der Änderungszeit, die ältesten zuerst
#
ls -ltr -l tmp.$$ $* | tee tmp.$$
#
summe=0
anzahl=0
```

```
#nun eine Liste des Speicherverbrauches erzeugen
for i in $(cut tmp.$$ -b 33-41);
do
# für jedes Element prüfen ob es nur Ziffern enthält
i=${i##*[!0-9, ']*}

#falls das Listenelement nicht leer ist, summiere auf
if [ i ]; then
summe=${summe+$i}
anzahl=${anzahl+1}
fi
done
```

```
#und nun ausgeben
echo -n "gesamter Speicherverbrauch: "

if [ $summe -ge 1000000 ]; then
```

```
#Speicherverbrauch ist größer als 1.000.000
```

```
echo -n "$summe / 1000000".
printf '%.3d.' $[(($summe % 1000000)/1000)
printf '%.3d' $[(($summe % 1000)]
```

```
elif [ $summe -ge 1000 ]; then
```

```
#Speicherverbrauch ist größer als 1.000
```

```
echo -n "$summe / 1000".
printf '%.3d' $[(($summe % 1000)]
```

```
else
```

```
#normal anzeigen
```

```
echo -n $summe
```

```
fi
echo " Byte"
```

```
#die temporäre Datei entfernen
rm tmp.$$
```

*Nichts auf der Welt ist so gerecht verteilt wie der Verstand:
jeder glaubt, genug bekommen zu haben.*

Jacques Tati