

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

Vorlesung Unix-Praktikum

8. while und read

Carsten Gnörlich

Rechnerbetriebsgruppe
Technische Fakultät
Universität Bielefeld

07. Dezember 2015

Willkommen zur achten Vorlesung

Was gab es beim letzten Mal?

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

- ▶ for-Schleifen
- ▶ seq, basename
- ▶ CSV-Tabellen
- ▶ cut, tr, sed
- ▶ printf

Willkommen zur achten Vorlesung

Was machen wir heute?

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

Zeilenweises Arbeiten

```
while  
read
```

Arrays

read interaktiv

Shell-History

Zeilenweises Arbeiten

Motivation

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

Gegeben sei eine zweispaltige Datei `zahlen.txt`:

```
90  17
110 201
6   57
20  15
101 99
```

Wie addiert man die Datei zeilenweise?

```
90 + 17 =
```

```
110 + 201 =
```

```
usw.
```

Zeilenweises Arbeiten

for hilft nicht weiter

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
for i in $(cat zeilen.txt); do  
    echo $i  
done
```

90

17

110

201

...

- ▶ for-Schleifen arbeiten elementweise
- ▶ ⇒ hilft uns nicht weiter

Zeilenweises Arbeiten

Umwandlung in CSV-Tabelle wäre möglich

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
> tr -s " " <zahlen.txt | tr " " ";"  
90;17  
110;201  
6;57  
20;15  
101;99
```

- ▶ for-Schleife zerlegt Zeilen nicht mehr elementweise
- ▶ Weiterverarbeitung mit cut wie bei CSV-Tabellen gezeigt
- ▶ aber kein allgemeingültiger Weg
(z.B. wenn Elemente Leerzeichen enthalten dürfen)

Zeilenweises Arbeiten

while-Schleifen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
while steuerbefehl; do  
    Befehl1  
    Befehl2  
    . . .  
    Befehln  
done
```

Solange `steuerbefehl` wahr ist,
führe `Befehl1, . . . , Befehln` aus.

Zeilenweises Arbeiten

while-Schleifen: Beispiel

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
zaehler=1
```

```
while test $zaehler -le 3; do
```

```
    echo $zaehler
```

```
    zaehler=$((zaehler+1))
```

```
done
```

```
> ./while1.bash
```

```
1
```

```
2
```

```
3
```

Zeilenweises Arbeiten

read-Befehl

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

read: Eine Zeile aus der Eingabe lesen

```
read line
```

- ▶ liest eine Zeile in die Variable `line`
- ▶ nimmt den Wert "falsch" an wenn die Eingabe leer ist

Zeilenweises Arbeiten

read ist der perfekte Zuspeler zu while

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
while read line; do  
    echo "Zeile: $line"  
done < zahlen.txt
```

```
> ./while2.bash  
Zeile: 90 17  
Zeile: 110 201  
...
```

- ▶ `zahlen.txt` wird als Eingabe in die Schleife umgeleitet
- ▶ `read` stellt sie zeilenweise in der Var. `line` zur Verfügung

Zeilenweises Arbeiten

Lösung für das Addieren der Zahlen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
while read line; do
    line=$(echo $line | tr -s " ")
    a=$(echo $line | cut -d\  -f 1)
    b=$(echo $line | cut -d\  -f 2)
    echo $a + $b = $((a+b))
done < zahlen.txt
```

Arrays

Motivation: unterschiedlich lange Zeilen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

Verschärfte Bedingungen: zahlen2.txt

```
1 4 3
8 10 9 7
2 8 1
10 9 12 7 1
1 9
```

- ▶ unterschiedlich viele Elemente in den Zeilen!

Arrays

Variablen und Speicherbereiche

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

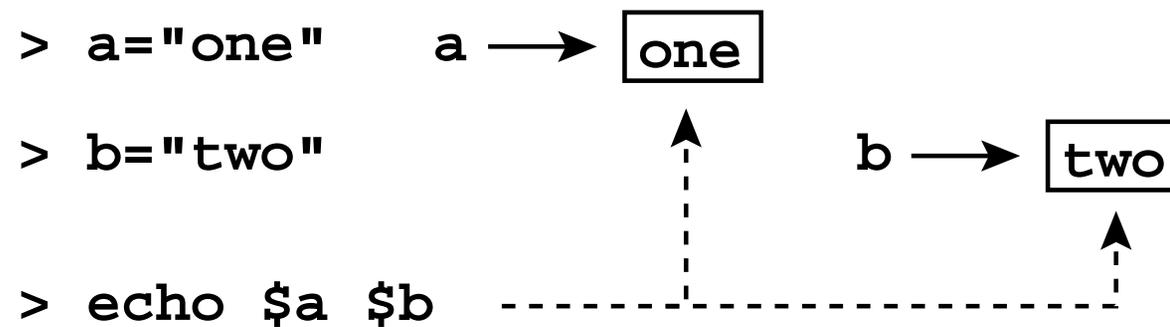
Arrays

read interaktiv

Shell-History

Standardfall:

- ▶ jeder Variablen ist ein Speicherbereich zugeordnet
- ▶ n Werte $\rightarrow n$ Variablennamen und Speicherbereiche



Arrays

Array-Variablen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

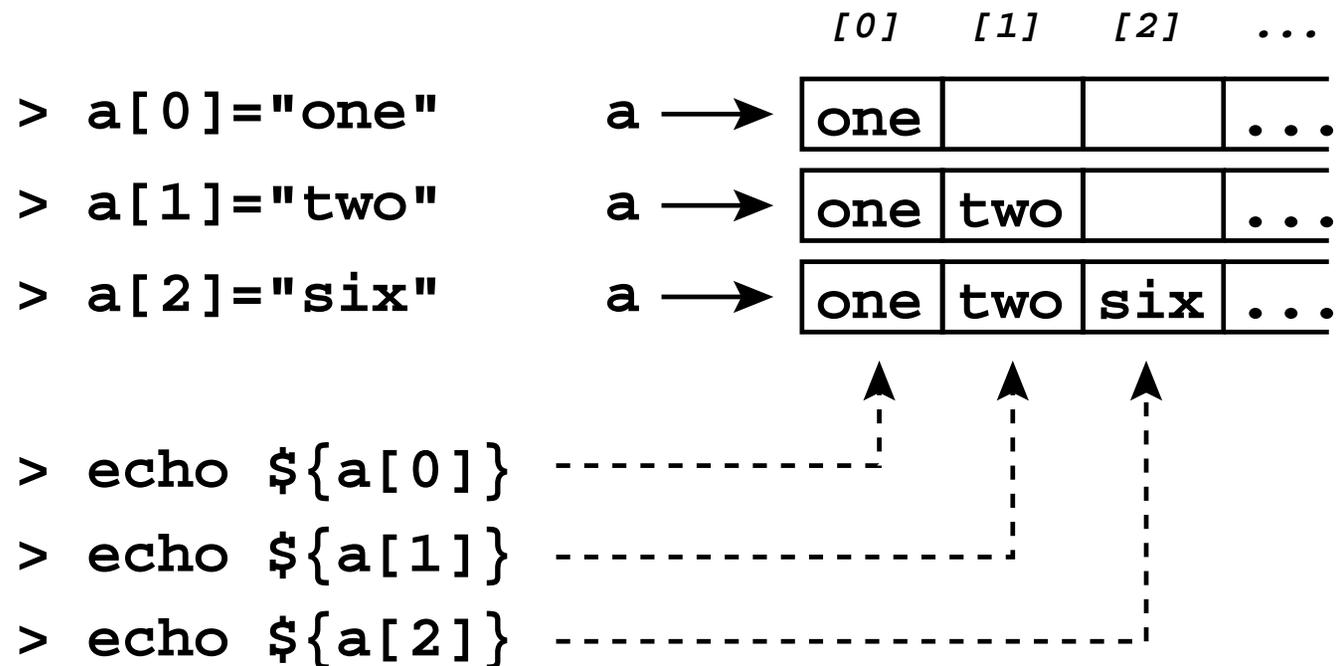
Arrays

read interaktiv

Shell-History

Array-Variablen (Feldvariablen):

- ▶ jeder Variablen sind n Speicherbereiche zugeordnet
- ▶ ein Name für n Speicherbereiche
- ▶ Zugriff/Unterscheidung der Speicherbereiche durch Index



Arrays

Übersicht: Länge feststellen, Elemente ausgeben

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

```
> a[0]="one"      a → [0] [1] [2] ...
> a[1]="two"      a → one |   |   | ...
> a[2]="six"      a → one | two | six | ...
```

Ein Element ausgeben:

```
> echo ${a[1]}
two
```

Alle Elemente ausgeben:

```
> echo ${a[*]}
one two six
```

Anzahl aller Elemente ermitteln:

```
> echo ${#a[*]}
3
```

Arrays

Arrays: Mehrere Elemente gleichzeitig setzen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
> b=(eins zwei drei vier fuenf)
```

```
> echo ${b[*]}
```

```
eins zwei drei vier fuenf
```

- ▶ Zuweisungsmechanismus `b=(...)`
hilft sehr beim Zerlegen von Zeilen!

Arrays

Vereinfachung des Additions-Skriptes durch Arrays

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
while read line; do
    line=$(echo $line | tr -s " ")
    a=$(echo $line | cut -d\  -f 1)
    b=$(echo $line | cut -d\  -f 2)
    echo $a + $b = $((a+b))
done < zahlen.txt
```

```
while read line; do
    z=( $line )
    echo ${z[0]} + ${z[1]} = $(( ${z[0]} + ${z[1]} ))
done < zahlen.txt
```

Arrays

Lösung: Datei zahlen2.txt bearbeiten

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
while read line; do
```

```
  a=($line)
```

```
  letztes=$(( ${#a[*]} - 1 ) )
```

```
  sum=${a[0]}
```

```
  echo -n "$sum "
```

```
  for i in $(seq 1 $letztes); do
```

```
    sum=$(( sum+${a[$i]} ) )
```

```
    echo -n "+ ${a[$i]} "
```

```
  done
```

```
  echo "= $sum"
```

```
done < zahlen2.txt
```

Array aus Zeile zusammenbauen

Index des letzten Elementes
= Länge des Arrays - 1

Ersten Summanden a[0] bearbeiten

Summanden a[1] ... a[letztes]
bearbeiten

Summe ausgeben

read

Shellskripte mit interaktiven Abfragen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
read -p "Geben Sie einen Dateinamen an: " name  
echo "Sie gaben ein: $name"
```

```
> ./skript.bash
```

```
Geben Sie einen Dateinamen an: brief.txt
```

```
Sie gaben ein: brief.txt
```

read

Abfrage mit vorbesetztem Wert

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
read -p "Dateiname: " -i neu.txt -e name  
echo "Sie gaben ein: $name"
```

```
> ./skript.bash
```

```
Geben Sie einen Dateinamen an: neu.txt
```

```
...
```

- ▶ Nutzer kann Voreinstellung mit der Eingabetaste übernehmen oder einen anderen Wert eingeben

read

Abfrage eines einzelnen Tastendrucks

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash
```

```
read -p "Drücken Sie eine beliebige Taste: " -n 1 taste  
echo -e "\nDie Taste war: $taste"
```

```
> ./skript.bash
```

```
Drücken Sie eine beliebige Taste: h
```

```
Die Taste war: h
```

read

Interaktiv laufende Programme

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
#!/bin/bash

taste="x"
while test "$taste" != "e"; do
    read -p "Würfeln oder Ende (w/e)? " -n 1 taste
    echo
    if test "$taste" == "w"; then
        echo $((1+RANDOM%6))
    fi
done

echo "Spiel beendet."
```

read

Interaktiv laufende Programme - Beispiellauf

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

```
> ./skript.bash
Würfeln oder Ende (w/e)? w
2
Würfeln oder Ende (w/e)? w
6
Würfeln oder Ende (w/e)? w
1
Würfeln oder Ende (w/e)? e
Spiel beendet.
```

Shell-History

history

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

history: Zuletzt eingegebene Befehle anzeigen

```
> history
```

```
...
```

```
1075  ls -l
```

```
1076  ./skript.bash
```

```
1077  history
```

Suchen in der History:

```
history | grep suchbegriff
```

Shell-History

Zuletzt ausgeführte Befehle erneut ausführen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

!*präfix*: führt letzte Eingabe erneut aus, die mit *präfix* beginnt

```
> history # nur zur Verdeutlichung
```

```
1078 gimp bild1.png
```

```
1079 gimp bild2.png
```

```
1080 gedit vergleich.bash
```

```
1081 ls -la
```

```
1082 cat vergleich.bash
```

```
> !gi
```

```
gimp bild2.png
```

(!g hätte nicht gereicht - dies würde gedit vergleich.bash ausführen)

Shell-History

Gezieltes Suchen nach Eingaben

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen
while
read

Arrays

read interaktiv

Shell-History

vgl. (Rückwärtssuche) im Emacs via CTRL-R:

- CTRL-R, Anfang des Suchbegriffes eingeben
- ▶ zeigt ersten Treffer

Suche weiter verfeinern:

- mehr Suchtext eingeben
- oder wiederholt CTRL-R zum weitersuchen

Eingabetaste: gefundene Eingabe wiederholen

Shell-History

Länge der History einstellen

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

Umgebungsvariable `HISTSIZE`:

```
> HISTSIZE=1000
```

- ▶ für permanente Einstellung: `.bash_aliases`

Ende der heutigen Vorlesung

Unix-
Praktikum

Carsten
Gnörlich

Zeilen lesen

while
read

Arrays

read interaktiv

Shell-History

Vielen Dank fürs Zuhören!

Bis nächste Woche!