

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

Vorlesung Unix-Praktikum

6. if-Verzweigungen und Variablen

Carsten Gnörlich

Rechnerbetriebsgruppe
Technische Fakultät
Universität Bielefeld

23. November 2015

Willkommen zur sechsten Vorlesung

Was gab es beim letzten Mal?

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

- ▶ Dateiverwaltung
- ▶ Aliase
- ▶ Shellskripte
- ▶ echo und grep

Willkommen zur sechsten Vorlesung

Was machen wir heute?

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

Shell-Skripte

Wiederholung: Shellskript

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

Prinzipieller Aufbau eines Shell-Skriptes

- ▶ Textdatei mit folgendem Inhalt:

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

Shell zum Ausführen des Skriptes



```
echo Hallo  
echo ich bin ein  
echo Shellskript
```

Aufrufe, wie Ihr sie auch
direkt eintippen würdet



Shell-Skripte

Wiederholung: Parameterübergabe

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

Beispiel zur Übergabe von Parametern:

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

```
echo "Erstes : $1"
```

```
echo "Zweites: $2"
```

```
echo "Drittes: $3"
```

```
echo "Anzahl : $#"
```

```
echo "Alle   : $*"
```

Shell-Skripte

Wiederholung: Datei mit Überschrift sortieren

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

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

```
head -2 $1 ; tail -n +3 $1 | sort -k $2 -n
```

```
> hsort2.bash planeten2.txt 2
```

Fallunterscheidungen

Motivation

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

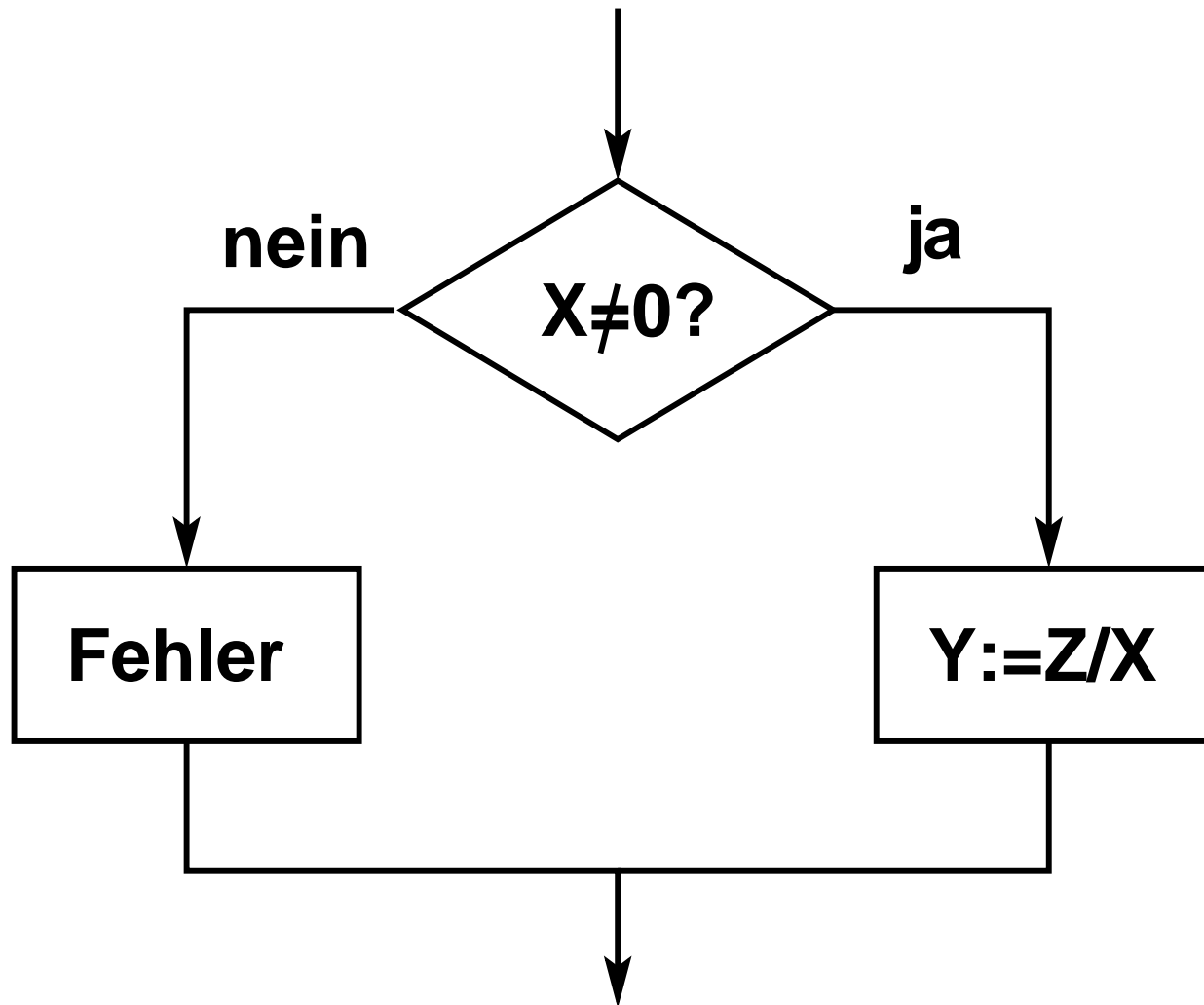
Verzweigungen

if

test

grep

Variablen



Fallunterscheidungen

Bedingte Ausführung

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

Bedingte Ausführung: if ... then ... else
oder auf Deutsch: wenn ... dann ... sonst

Wenn diese **Bedingung** erfüllt ist...

```
if test $1 = "eins"
then
    echo "$1 ist gleich eins"
else
    echo "$1 ist ungleich eins"
fi
```

dann mache dies
(Bedingung **erfüllt**)

sonst (**nicht erfüllt**) mache das

Fallunterscheidungen

Beispiel: Funktionalität von hsort und hsort2 zusammenfassen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

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

```
if test $# = 1 # 1 Argument?
```

```
then # ja, Aufruf wie hsort
```

```
head -2 $1 ; tail -n +3 $1 | sort
```

```
else # nein, Aufruf wie hsort2
```

```
head -2 $1 ; tail -n +3 $1 | sort -k $2 -n
```

```
fi
```

Fallunterscheidungen

Vorhandensein einer Datei als Bedingung

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
if test -r $1
then
    echo "Die Datei $1 ist vorhanden und lesbar!"
else
    echo "Schade, $1 kann nicht geöffnet werden!"
fi
```

Fallunterscheidungen

Fehlertolerante Skripte

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
if test ! -r $1      Negiert die Bedingung (" wenn nicht ...")
then
    echo "Die Datei $1 ist nicht vorhanden!"
    exit 1
fi
```

Bricht das Skript an dieser Stelle ab

```
if test $# = 1
then
    head -2 $1; tail +3 $1 | sort
else
    head -2 $1; tail +3 $1 | sort -k $2 -n
fi
```

Fallunterscheidungen

Schreibweisen; test und []

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

Gesamter if-Ausdruck in einer Zeile:

```
if test -r datei.txt; then echo da; else echo fehlt; fi
```

- ▶ Ausdruck durch Semikolon trennen!

test hat zwei äquivalente Schreibweisen:

```
if test -r datei.txt; then ...
```

```
if [-r datei.txt]; then ...
```

- ▶ bei der [...] -Variante auf die Leerzeichen `-` achten!

Fallunterscheidungen

(wichtigste) Testmöglichkeiten mit Zeichenketten

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
test "$1" = "hallo"    # Zeichenkette gleich "hallo"?
```

```
test "$1" != "hallo"  # Zeichenkette nicht "hallo"?
```

```
test -z "$1"          # $1 ist die leere Zeichenkette
```

```
test -n "$1"          # $1 ist nicht die leere Zeichenkette
```

(Nur) bei Zeichenketten: Variablen in "" setzen ("\$1")

- ▶ sonst Syntaxfehler wenn Variable = leere Zeichenkette

Fallunterscheidungen

Testmöglichkeiten mit Ganzzahlen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

```
test $1 -eq 42 # Zahl = 42?
```

```
test $1 -gt 42 # Zahl > 42?
```

```
test $1 -ge 42 # Zahl ≥ 42?
```

```
test $1 -lt 42 # Zahl < 42?
```

```
test $1 -le 42 # Zahl ≤ 42?
```

```
test $1 -ne 42 # Zahl ≠ 42?
```

Fallunterscheidungen

(wichtigste) Testmöglichkeiten bzgl. Dateien

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
test -f $1 # Datei $1 existiert und ist reguläre Datei
```

```
test -r $1 # Datei $1 existiert und ist lesbar
```

```
test -w $1 # Datei $1 existiert und ist schreibbar
```

```
test -x $1 # Datei $1 existiert und ist ausführbar
```

```
test -d $1 # $1 existiert und ist ein Verzeichnis
```

- ▶ es gibt noch mehr tests
- ▶ siehe man test

Fallunterscheidungen

logisches UND zwischen zwei Vergleichen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
if test "$1" = "rot" && test "$2" = "blau" ; then
    echo wahr
else
    echo falsch
fi
```

```
> ./skript.bash rot blau
```

```
wahr
```

```
> ./skript.bash gruen blau
```

```
falsch
```

```
> ./skript.bash rot gruen
```

```
falsch
```

```
> ./skript.bash blau rot
```

```
falsch
```

Wahrheitstabelle log. UND:

<u>wahr</u> \wedge <u>wahr</u>	=	<u>wahr</u>
<u>wahr</u> \wedge falsch	=	falsch
falsch \wedge <u>wahr</u>	=	falsch
falsch \wedge falsch	=	falsch

Fallunterscheidungen

logisches ODER zwischen zwei Vergleichen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
if test "$1" = "rot" || test "$2" = "blau" ; then
    echo wahr
else
    echo falsch
fi
```

```
> ./skript.bash rot blau
wahr
> ./skript.bash rot xxx
wahr
> ./skript.bash xxx blau
wahr
> ./skript.bash xxx xxx
falsch
```

Wahrheitstabelle log. ODER

<u>wahr</u> ∨ <u>wahr</u>	=	<u>wahr</u>
<u>wahr</u> ∨ falsch	=	<u>wahr</u>
falsch ∨ <u>wahr</u>	=	<u>wahr</u>
falsch ∨ falsch	=	falsch

Fallunterscheidungen

Komplexe Ausdrücke über UND und ODER

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
if test "$1" = "rot" &&  
    (test "$2" = "apfel" || test "$2" = "kirsche") ; then  
    echo wahr  
else  
    echo falsch  
fi
```

```
> ./skript.bash rot apfel  
wahr  
> ./skript.bash rot kirsche  
wahr  
> ./skript.bash rot banane  
falsch  
> ./skript.bash gruen apfel  
falsch
```

$\underline{W} \wedge (\underline{W} \vee \underline{W})$	=	\underline{W}
$\underline{W} \wedge (\underline{W} \vee F)$	=	\underline{W}
$\underline{W} \wedge (F \vee \underline{W})$	=	\underline{W}
$\underline{W} \wedge (F \vee F)$	=	F
$F \wedge (\underline{W} \vee \underline{W})$	=	F
$F \wedge (\underline{W} \vee F)$	=	F
$F \wedge (F \vee \underline{W})$	=	F
$F \wedge (F \vee F)$	=	F

Fallunterscheidungen

Wahrheitswerte von Kommandos

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

Kommandos wie `grep` und `diff` haben Wahrheitswerte:

`grep wort datei` `true` \Leftrightarrow `datei` enthält `wort`

`diff datei1 datei2` `true` \Leftrightarrow `datei1` und `datei2` sind gleich

Beispiele:

```
if grep hallo datei.txt; then ...
```

```
if echo $1 | grep hallo; then ...
```

u.s.w.

- ▶ bei *Pipes* gilt der Wert des letzten Befehls

Fallunterscheidungen

Wahrheitswerte von Kommandos

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

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

```
if grep -q -i $1 planeten.txt; then
```

```
    echo $1 ist ein Planet
```

```
else
```

```
    echo $1 ist kein Planet
```

```
fi
```

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

```
erde ist ein Planet
```

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

```
pluto ist kein Planet
```

```
grep -q: quiet; unterdrückt Ausgabe von grep
```

Variablen

Variablenzuweisungen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if


test

grep

Variablen

Wert an Variablen zuweisen:

keine Leerzeichen!


> **wort=eins**

Variablenwert benutzen / ausgeben:

> echo \$wort

eins

Variablen

Datentypen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

Variablen sind “schwach getypt”

- ▶ werden automatisch als Zeichenkette oder Zahl benutzt

Beispiel:

```
> a=1
```

```
> b="2"
```

```
> if test $a -lt $b; then echo wahr; fi  
wahr
```

```
> if test $a = "1"; then echo wahr; fi  
wahr
```

```
> if test $a -lt "zwei"; then echo true; fi  
bash: test: zwei: Ganzzahliger Ausdruck erwartet.
```

Variablen

Variablenzuweisungen aus Shell-Aufrufen

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

Zwischenspeichern von Programmausgaben:

```
> a=$(echo -n Linux | wc -m)
> echo $a
5
```

Auch eine komplette Zeile kann man sinnvoll speichern:

```
> a=$(ls -l eins.txt)
> echo $a
-rw-r--r-- 1 cg stud 4502 17. Nov 16:38 eins.txt
```

- ▶ Mehrzeilige Ausgaben besser nicht in Variablen packen!

Variablen

Variablen als Zeichenketten verarbeiten

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

```
> name=datei  
> verz=/home/juser  
> pfad=$verz/$name.jpg  
> echo $pfad  
/home/juser/datei.jpg
```


Variablen

Variablen als Zeichenketten: Sonderfälle

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

Variablennamen durch Klammern vom Text abtrennen:

```
> name=zeichen  
> echo ${name}kette  
zeichenkette
```

Leerzeichen durch Anführungszeichen ("...") erhalten:

```
> a=eins  
> b=zwei  
> c="$a $b"  
> echo $c  
eins zwei
```

Variablen

Variablen und Arithmetik

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

$\$(())$: Arithmetischen Ausdruck auswerten

```
> echo  $\$(3+5)$ 
```

```
8
```

Mit Variablen:

```
> a=9
```

```
> b=3
```

```
> echo  $\$(a*b)$ 
```

```
27
```

- ▶ Im arithm. Ausdruck darf man **a** statt **$\$a$** schreiben

Variablen

Arithmetische Operatoren

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen

if
test
grep

Variablen

+	Addition
-	Subtraktion
*	Multiplikation
/	ganzzahlige Division
%	Modulo (Rest der Division)

Beispiel:

```
> echo $((23 / 5))
```

```
4
```

```
> echo $((23 % 5))
```

```
3
```

- ▶ Shell kann nur Ganzzahlen verarbeiten
- ▶ Fließkommazahlen → später

Variablen

Zufallszahlen und Modulo

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte

Wiederholung

Verzweigungen

if

test

grep

Variablen

`$RANDOM`: liefert Zufallszahl zwischen 0...32767

Würfel mit 6 Seiten simulieren:

```
> echo $((1+RANDOM%6))
```

Ende der heutigen Vorlesung

Unix-
Praktikum

Carsten
Gnörlich

Shellskripte
Wiederholung

Verzweigungen
if
test
grep

Variablen

Vielen Dank fürs Zuhören!

Bis nächste Woche!