Linux basic

die shell (1)

Version 2008-02-04

Dipl.Inf. (FH) Bernd Schwägerl
Dipl.Inf. (FH) Stefan Edenhofer

Hilfefunktionen

man [Sektion] [Befehl]	Handbuchseite der Sektion des Befehls anzeigen.
man -k [Begriff]	Die Kurzbeschreibungen nach diesem Begriff durchsuchen.
man -a [Befehl]	Zeigt nacheinander die Manpages **aller** Sektionen zu diesem Befehl an.
man -f [Befehl]	Zeigt jedes Vorkommen des Befehls mit dessen Kurzbeschreibung an
whatis [Befehl]	Anzeige der Sektionen, die sich auf Befehl beziehen.

Dateien

file [Datei|Verzeichnis]	Anzeige des Dateityps
cp [Quelle] [Ziel]	Kopieren der Datei Quelle nach Ziel
scp [user1]@[host1]:[absPfad] [user2]@[host2]:[absPfad]	
rm [Optionen] [Datei]	Löschen einer Datei
-R	Unterverzeichnisse
-f	nicht nachfragen
mv [Quelle] [Ziel]	Verschieben von Quelle nach Ziel
ln -s [Datei] [Link]	Symbolischer Link file/dir
cat [Datei]	Datei anzeigen
cat >[Datei]	Datei interaktiv erzeugen (Ende:<strg><d>)
touch [Datei]	Leere Datei erzeugen
tar -xfzp [Datei].tar.gz	.tar.gz. Dateien entpacken
find [Pfad] [Opt] [Datei]	Suchen einer Datei mit den Eigenschaften Opt unterhalb von Pfad
-name	Opt: Dateiname
-size +/-n; -mtime +/-n	Größer/Kleiner n; jünger/älter n Tage

Verzeichnisse

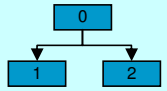
ls -la	Anzeigen des Verzeichnisinhalts ("long, all")
mkdir	Erzeugen eines Verzeichnisses (Löschen mit rm -Rf)
du	Belegten HD-Platz abfragen
-s	Summe anzeigen
-h	"menschenlesbar"
-a	all - auch Dateien
pwd	aktuelles Verzeichnis anzeigen
cd [Verzeichnis]	Verzeichniswechsel...nach...
cd /	...Wurzel des Systems
cd ~	...ins Homedir des aktuellen Benutzers
cd ..	...eine Ebene höher

"." ist das aktuelle Verzeichnis. Dieses befindet sich beim Systemadministrator für ausführbare Programme aus Sicherheitsgründen niemals im Suchpfad!

Benutzerinformationen

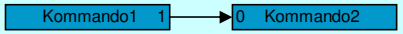
whoami / who am i	Wer bin ich im Moment (Multiuser-OS!)
who	Wer ist derzeit angemeldet?
An welchen Devices: ps aux | grep bash | cut -b37-43 | sort | uniq	

Ein- und Ausgabeumleitung



0	Standardeingabe (Tastatur)
	stdin
1	Standardausgabe (Bildschirm)
	stdout
2	Standardfehler (Bildschirm)
	stderr
[cmd] 1> [file]	Umleitung der Standardausgabe
[cmd] 2> [file]	Umleitung des Standardfehlers
[cmd] 1>[file1] 2>[file2]	Umleitung in jeweils eine Datei
[cmd] 1>[file] 2>&1	stderr -> stdout; stdout -> file
[cmd] >> [file]	An Datei [file] anhängen
[cmd] < [file]	stdin ist die Datei [file]

Ein- und Ausgabeumleitung



[cmd1] | [cmd2]	Ausgabe cmd1 ist Eingabe für cmd2
[cmd1] | tee [file]	Ausgabe auf Bildschirm und Datei [file]
Hinweis: Siehe Filtern mit "grep"	

Shell- und Umgebungsvariablen

Shellvariable	gilt ausschliesslich in der aktuellen Shell. Da manche Befehle eine eigene Shell nutzen, ist die Variable dort nicht gültig.
Umgebungs-/ Environmentvariable	Eine Umgebungsvariable wird an Subshells weitervererbt. Für Parentshells gilt sie nicht!
[VARNAME]=["Wert"]	Setzt eine Shellvariable
$[VARNAME]	greift auf den Wert der Variable zurück.
export [VARNAME]	Wandelt Shellvariable in Umgebungsvariable
unset [VARNAME]	Löscht die Variable

Systemressourcen

free	Anzeige freier Speicher
df [Devicefile]	Speicherplatz Festplatten/CDs
	Erste IDE-HD: /dev/hda
	Meist erste SCSI-HD: /dev/sda
top	Auslastungsanzeige
-n 1 -b	eine Iteration im Batchmode

Dateirechte / Siehe auch Sonderrechte

ls -la	Anzeigen von Dateirechten
chmod 0421 [Datei|Dir]	absolute Änderung der Rechte
chmod o+r g-x u+x [Datei|Dir]	relative Änderung der Rechte
chown [user]:[group] [datei]	Besitzer und Besitzergruppe ändern
umask 022	"Defaultrechte" (Komplementärvektor)

"-rwxrw--r-- 1 user group ..."

Die Datei gehört dem Benutzer user und der Gruppe group. user darf schreiben (w = write = 2), lesen (r = read = 4) und ausführen (x = execute = 1), Die Gruppenmitglieder von group dürfen "r und w". ALLE Anderen "r".

Filtern/Vergleichen von Dateien und Streams

sort	Sortieren der Zeilen
cut	Ausschneiden von Feldern pro Zeile
cut -f n -d " "	delimiter Leerzeichen, Feld n
uniq	aufeinanderf. doppelte Zeilen weglassen
diff [-u|-dtw] [datei1] [datei2]	Inh. Vergleich zweier Dateien
wc -l	Word count -l = lines
cmp [Datei1] [Datei2]	Zeilenweiser Vergleich
grep [Muster]	Zeilenausgabe in der Muster ist
grep -v [Muster]	...nicht ist
tail [Datei]	Ende der Datei ausgeben
head [Datei]	Anfang der Datei ausgeben
split --bytes=1024 [Datei]	Zerteilen in 1KB-Dateien
cat [Dat1] [Dat2] > [Datei]	Zusammensetzen von Dateien

Reguläre Ausdrücke / Muster

Zeichen und Zeichenklassen:	
.	irgendein Zeichen
^	Zeilenanfang
$	Zeilenende
[0-9]	Zeichenklasse (0,1,2,3...9)
[abxyz]	Zeichenklasse (a,b,x,y,z)
[[:blank:]]	Alle Whitespaces
[[:alnum:]]	Alle Buchstaben und Zahlen
[[:alpha:]]	Alle Buchstaben
Wiederholungen / Vorkommen:	
?	ein oder kein
*	kein oder viele
+	ein oder mehrmaliges
{n}; {n,}; {n,m}	genau n mal, n oder öfter; mindestens n mal, höchstens m mal

Sonderrechte

Datei		Verzeichnis
	1	Enthaltene Dateien dürfen nur vom Besitzer der Datei gelöscht werden.
wird mit den Rechten des Besitzers ausgeführt.	2	
wird mit den Rechten der Besitzergruppe ausgeführt.	4	Neue Files in diesem Verzeichnis erhalten die Gruppe des Verzeichnisses

Die Sonderrechte werden bei einem chmod einfach vorne angestellt.
(Bei absolutem setzen!) Beispiel: chmod 4000 [Datei|Dir].
Angezeigt (lsmod) werden die Sonderrechte als
s beim Besitzer (4), s bei der Gruppe (2) und t (1).

Linux basic

die shell (2)

Version 2008-02-04

Dipl.Inf. (FH) Bernd Schwägerl
Dipl.Inf. (FH) Stefan Edenhofer

Partitionierung

Partitionen, die auf "/"-Partition müssen	/bin (wichtige Befehle) /sbin (wichtige root-Befehle) /dev (Zugriffshandles Devices) /etc (alle Konfigurationen) /lib (Libraries) /proc (virt.!! Kernelkommunikation)
Partitionen, die auf "/"-Partition sein sollten	/root (Homedirectory für den Systemverwalter)
Partitionen, die ausgelagert werden sollten.	/usr (Unix System Ressource) /var (variable Daten, Log & Spool) /home (Homedirectories User) /tmp (Temporäre Dateien) /boot (Bootloader, Kernel) /swap ("Auslagerungspartition")

Runlevels

Konfiguration	/etc/inittab /etc/init.d
init [runlevel]	Wechsel in "runlevel"
runlevel	Ausgabe vorheriger und jetziger runlevel
runlevels:	
0	Stoppen des Computers
1	Singleusermode
2	Multiuser // ohne Netz
3	Multiuser mit Netz
4	(unbenutzt)
5	Graphische Benutzeroberfläche
6	Neustart

Links (Verknüpfungen)

Es gibt 2 unterschiedliche Arten von Links (Verknüpfungen). Soft- und Hardlinks. Hardlinks haben exakt die gleiche Verwaltungsinformation, wie die Datei selbst. Softlinks haben eigene Verwaltungsinformation. (inode)

ln [Ziel] [Linkname]	Hardlink erstellen
ln -s [Ziel] [Linkname]	Softlink erstellen
ls -i	Directory mit Inode-Nummern anzeigen.

Mit Softlinks geht fast alles - und dies sind auch die bessere Wahl!

Schnelle Dateisuche

Ausschliesslich den Suchpfad ($PATH) durchsuchen:

which [Dateiname]	Suche nach ausführbaren Dateien (Ergebnis: Erste Fundstelle!)
whereis [Dateiname]	Ausgabe aller Dateien von PATH (viele!)
PATH="$PATH:[/neuesDir]"	Erweitern der Shellvariable PATH
export PATH	Den Pfad vererbbar machen

Über eine Datenbank suchen. Gefunden werden nur Dateien, die in der Datenbank bereits erfasst wurden.

updatedb	Die Suchdatenbank aktualisieren.
locate [Datei]	Finden von erfassten Dateien
slocate [Datei]	dito, aber nur auf die Zugriff besteht.

Sicherheitsaspekt: Geben Sie Usern nur slocate als Suchfunktion.

Hostinformationen

uptime	Zeit, seitdem das System läuft
hostname	Ausgabe des Hostnamens
hostname -d	Ausgabe der Domain
date	Datum / Uhrzeit des Hosts

VI - der Standardeditor

Es gibt 2 wichtige Modi (Befehlsmodus <ESC> und Insert <ESC><I>)

d[n][d|w|c]	n Zeilen, Wörter, Zeichen löschen
y[n][y|w|c]	n Zeilen, Wörter, Zeichen kopieren
p	einfügen was gelöscht ODER kopiert wurde
x	ein Zeichen löschen
r	ein Zeichen ersetzen
:[n]	In Zeile n gehen
:$	ans Dateiende gehen
$	ans Zeilenende gehen
0	an den Zeilenanfang gehen
/muster	nach "muster" vorwärts suchen
?muster	nach "muster" rückwärt suchen
:1,$s/m1/m2/ig	Zwischen Zeile 1 und Dateiende ($) - auch mehrmaligen Vorkommens pro Zeile (g=global) und egal ob Gross/Klein-schreibung (i=ignore) - ersetzen (s=substitute) von Muster m1 durch Muster m2
:wq!	schreiben (w=write), verlasen (q=quit) - ohne nachzufragen (wenn schreibgeschützt, wenn Änderungen etc.) Nur verlassen des VIs: q!
J	Füge eine Zeile drunter an
<strg>g	Zeige Zeilenstatistik
u	undo - geht bis zu dem Zeitpunkt, an dem der VI gestartet wurde
k h l j	Wenn keine Cursortasten auf der Tastatur vorhanden, oder Terminal falsch eingestellt: Navigation

Tipps und Tricks

Alle Benutzer des Systems anzeigen. (Auch Daemon-User!):

```
cat /etc/passwd | cut -d":" -f1
```

Prozessänderungen der letzten [SEKUNDEN] anzeigen:

```
diff <(ps ax) <(sleep [SEKUNDEN]; ps ax)
```

Alle Prozess-IDs anzeigen, die in der ps-Ausgabe MUSTER enthalten:

```
ps aux | grep [MUSTER] | grep -v grep | cut -b10-14
```

Mail an EMAIL mit Subject SUBJECT - Mailtext: Dateinamen von [DIR] - Anhang der zusammengefasste (tar) Inhalt v. [DIR] in dem File demo.tar

```
tar cvf demo.tar [DIR] | elm -s [SUBJECT] [EMAIL] -A demo.tar
```

Hostübergreifende Verzeichnissynchronisation (Backup!)

Verschlüsselt (-e ssh) ; archivieren (a), recursive (r), verbose (v)

```
rsync -avr -e ssh [user@host:]:/srcpath [user@host:]/destpath
```

Prozesse

ps aux	Anzeige aller Prozesse
pstree	-"- in Abhängigkeit
kill [signal] [PID]	Beendigungssignal senden
killall [Muster]	Alle Prozesse die Muster enthalten
jobs	Anzeige von Jobs und JOBID
bg [cmd1] [cmd1] &	Kommando im Hintergrund ausführen
fg [JOBID]	Job in den Vordergrund holen
<strg><z>	Job in den Hintergrund schicken. Achtung, mit bg [JOBID] erst wieder starten!
nohup [cmd1]	Kommando nach dem Abmelden weiterlaufen lassen

Prozessprioritäten:

Prozesse können mit unterschiedlicher Priorität gestartet werden. Hierfür gibt es einen Nice-Wert, welcher von -20 (höchste Priorität) bis 19 (niedrigste) geht. Defaultwert ist 0.

nice -n[WERT] [Befehl]	Befehl mit der Nice-Zeit von WERT starten. (Normale User nur >-1)
renice +/-[DELTA] [PID]	Ändern des Nicewertes. (Normale User nur >-1!!!)
renice +/-[DELTA] -u root	Alle dem User "root gehörenden Prozesse ändern.

Netzwerk

IPv4-Adresse setzen:

```
ifconfig eth[n] netmask [Maske] broadcast [BC-Adr] [IP]
```

ifconfig [NIC] up	NIC hochfahren
ifconfig [NIC] down	NIC runterfahren

Route zu Netz oder Host ZIEL (mit Maske) über Rechner gateway:

```
route add –net/host [ZIEL] netmask [netmask] gw [gateway]
```

Route entfernen:

```
route del –net [ZIEL] netmask [netmask]
```

route	Alle Routes anzeigen

Abhängigkeit von Kommandos

cmd1; cmd2	Erst cmd1, dann cmd2 ausführen
cmd1 && cmd2	Wenn cmd1 erfolgreich,auch cmd2
cmd1 || cmd2	cmd2 nur ausführen, wenn cmd1 nicht erfolgreich ausgeführt wurde.
cmd1 & cmd2	cmd1 beginnen (Hintergrund) und zeitgleich cmd2 im Vordergrund

Hilfreich z.B. beim Kernelkompilieren (da zeitaufwendig): &&

Linux basic

netzwerk

Version 2008-02-04

Dipl.Inf. (FH) Bernd Schwägerl
Dipl.Inf. (FH) Stefan Edenhofer

Netzwerkeinstellungen abfragen und einstellen (IPv4)	
Netzwerk-Interface-Card (NIC):	
ifconfig	Abfrage der aktuellen Einstellungen für alle Schnittstellen
ifconfig [schnittstelle] [adresse] [parameter]	
ifconfig eth0 172.17.21.11 netmask 255.255.255.0 broadcast 172.17.21.255	
Die wichtigsten Parameter	
broadcast [BC-Adresse]	Broadcastadresse setzen
down	Schnittstelle deaktivieren
mtu [Zahl]	Maximale Transfer Unit setzen
netmask [MASK]	Netzmaske setzen
up	Schnittstelle aktivieren
Routing:	
arp -n	ARP-Tabelle (MAC-IP-Zuweisung) anzeigen ohne Namensauflösung (-n).
route -n	Abfrage der aktuellen Routinginformationen ohne Namensauflösung (-n).
route [add|del] [-net|-host] [Ziel] netmask [NM] gw [Gateway] dev [IF]	
route add -net 172.16.1.0 netmask 255.255.255.0 gw 172.17.21.1 dev eth0	
Aktivieren des Routings :	
echo "1" > /proc/sys/net/ipv4/ip_forward	
Deaktivieren des Routings:	
echo "0" > /proc/sys/net/ipv4/ip_forward	
Anzeigen, ob Routing aktiviert wurde (0=nein/1=ja):	
cat /proc/sys/net/ipv4/ip_forward	
Weitere wichtige Optionen für den Befehl route:	
metric [METRIK]	Mit der Metrik werden die "Kosten" der Strecke festgelegt. Je höher die Metrik, umso weniger "gern" wird diese Strecke (route) genutzt.
mss [MAXSIZE]	Die maximale Fragmentgröße (Paketgröße) - wird eventuell benötigt zwischen unterschiedlichen Topologien/Medien. Dies ist auch für einige (schlechte) Firewalls eventuell notwendig.
ACHTUNG! Jegliche Information, die Sie mit diesen Tools setzen, wird nach einem Reboot nicht mehr zur Verfügung stehen! Daher müssen diese in ein Startskript (siehe Shellprogrammierung)	
SuSE-Konfigurationsfiles	/etc/sysconfig/network/ifcfg....

Wichtige Tools, die man kennen sollte	
ping	Testen der Erreichbarkeit
-c [Anzahl]	(count) Anzahl der Pings
-n	(nummerisch) Keine Namensauflösung
traceroute -n	Weg eines Pakets - im Internet meist sinnlos, da keine Rückantwort (*) kommt. (-n ohne Namensauflösung)
tcpdump -i [NIC]	Ist ein sehr spärlicher Netzwerksniffer.
ssh	SSL verschlüsselter Remotezugang
scp [user1]@[host1]:[absPfad] [user2]@[host2]:[absPfad]	
	verschlüsseltes Remotekopieren (siehe shell)

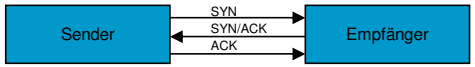
Namensauflösung (nslookup / dig)	
Achtung! nslookup wird durch dig abgelöst!	
nslookup	Öffnet nslookup interaktiv und konnektiert an den im System angegebenen Nameserver
set q=[TYP]	(siehe Ressource Records)
server [IP/NAME]	Diesen Server befragen
dig -t [TYP] @[NServer]	Den Server NServer nach dem Ressource Record TYP befragen.

Ressource Records (TYP) für nslookup und dig	
SOA	Verwaltungsinfos
A	Adresseintrag (NAME .-> IP)
NS	Nameserver
CNAME	Aliasnamen
PTR	Adresseintrag (IP -> NAME)
MX	MaileXchanger - Mailserver
HINFO	Hostinfos (falls gesetzt)
TXT	Kommentare
ALL	(fast) alles - (QueryType - aber eigentlich KEIN Ressource Record)

Anzeige Netzwerkrelevanter Infos (netstat) (ausführlich -v)	
netstat -s	Statistische Protokollinformationen
netstat -c	Fortlaufende Anzeige
netstat -l	Empfangsbereite Sockets
netstat -a	Sende- und Empfangssckets
netstat -p	Mit Programmen

TCP-Header		
1. Byte	2. Byte	
Source Port (Dienstkennung)		
Destination Port (Dienstkennung)		
Sequenz Nummer (4 Byte)		
Sequenz Nummer (4 Byte)		
Acknowledge Nummer (4 Byte)		
Acknowledge Nummer (4 Byte)		
Daten Offset	Reserviert (6 Bit) |	Flags (6 Bit)
Window (Puffergröße des Empfängers)		
Checksumme (Prüfsumme Header UND Daten)		
Urgent Pointer (Zeigt auf "vorrangig zu bearbeitende Daten")		
Options (Immer 32 Bit-Vielfaches)		
Options + Padding ("Füllung")		
Dateninhalt		

Erfolgreiche TCP-Verbindung (3-way-handshake)



TCP-Flags:	
URG	1 = Urgent Pointer-Feld muß gelesen werden („vorrangig" zu bearbeitende Daten)
ACK	1 = Acknowledge Number-Feld muß gelesen werden
PSH	1 = Daten müssen unverzüglich an den entsprechenden Dienst weitergeleitet werden
RST	1 = Verbindung soll beendet werden
SYN	1 = Verbindung soll aufgebaut werden
FIN	1 = Zeigt endgültigen Verbindungsabbau

netcat / nc
netcat, bzw. nc ermöglicht die Kommunikation mit einer tcp- oder udp-Verbindung über stdin/out, beispielsweise aus einem Shellscript heraus.
Abfrage von Klartextprotokollen (Beispiel für HTTP)
printf 'GET / HTTP/1.0\n\n' | netcat -w [TIMEOUT] [SERVER] [PORT]
Umleiten von lokalen Ports auf Remoterechner (siehe auch rinetd)
Eintrag in /etc/inetd.conf (Port <-> Programmzuordnung!): [L-Port] stream tcp nowait [User] nc nc -w 5 [R-host] [R-port]
Mit netcat / nc lassen sich Shellprogramme als Daemon nutzen. netcat muss mit der Option (-DGAPING_SECURITY_HOLE compiliert sein.
nc -l -p [PORT] -e [Pfad zu einem Shellskript]

Ihre Zusätze	

Gängige TCP-Ports	
(TCP) 20	Datenkanal FileTransferProtocol (FTP)
(TCP) 21	Kontrollkanal FileTransferProtocol (FTP)
(TCP) 22	Secure Login
(TCP) 23	telnet
(TCP) 25	Simple Mail Transfer Protocol (SMTP)
(TCP/UDP) 53	DomainNameService (DNS)
(TCP) 80	HyperTextTransferProtocol (HTTP)
(TCP) 110	PostOfficeProtocol 3 (POP3)
(TCP) 443	https
Die bekannten Dienst / Port - Zuordnung finden sich in /etc/services	

Siehe hierzu auch "Linux basic - die shell"!
(Variablen, Pipes, Verkettung von Befehlen....)

Variablen	
$* oder "$@"	Alle übergebenen Variablen als Liste
$* oder $@	Alle übergebenen Variablen als ein Wort
$#	Anzahl der übergebenen Werte
$?	Rückgabewert des zuletzt ausgeführten Kommandos
$$	Prozess-ID der aktuellen Shell
$!	Prozess-ID des zuletzt gestarteten Subprozesses
$0	Der Programmname
$1 - $9	Die ersten 9 Übergabeparameter

shift verschiebt die Übergabeparameter um eins nach links. Der linkste Wert fällt nach "Lemmingmethode einfach über die Klippen". D.h. der ehemalige Wert von $2 ist nun der Wert von $1. Auf den Wert von $1 können Sie zu diesem Zeitpunkt nicht mehr zugreifen!

[VARIABLE]="[WERT]"	Eine Variable VARIABLE auf einen WERT setzen
$[VARIABLE]	Zugriff auf den Wert der Variable. z.B. echo $VARIABLE.

Rechenoperationen mit dem Befehl expr	
Addition	ERGEBNIS=`expr 1 + 2`
Subtraktion	ERGEBNIS=`expr 1 - 2`
Multiplikation	ERGEBNIS=`expr 2 * 2`
Ganzzahlige Division	ERGEBNIS=`expr 8 / 2`
Rest einer Division	ERGEBNIS=`expr 7 % 2`
Binäres Oder	ERGEBNIS=`expr 1 | 0`
Binäres Und	ERGEBNIS=`expr 1 & 1`
Vergleichsoperationen	< <= = == != >= >

Die Beispiele zeigen auch, wie man eine Variable mit dem Wert des Ergebnisses eines Befehls setzt. Auf diesem Weg können Sie Ergebnisse anderer Befehle, wie z.B. "date", "ifconfig" etc. in einer Variable auffangen. Das Ergebnis ist dann der Wert der Variable, auf den Sie mit einem vorangestellten $ zugriff haben.

Beispiel: echo $ERGEBNIS - gibt das Ergebnis auf den Bildschirn aus

Ergebnisse von Vergleichsoperationen sind 1=wahr 0=unwahr

Umsetzungen für "Standard-For-Schleifen"

```
for ZAEHLER in `seq 1 100`
do
echo "Durchlauf $ZAEHLER"
done
```

`[BEFEHL]` beeinhaltet den Rückgabewert des Kommandos!!!

```
ZAEHLER=1
while [ $ZAEHLER -ne 101 ]
do
ZAEHLER=`expr $ZAEHLER + 1`
echo "Durchlauf: $ZAEHLER"
done
```

for-Schleife

Die For-Schleife ist nicht klassisch als "Zählwerkzeug" belegt, wie in anderen Programmiersprachen, sondern durchläuft eine Liste. Wenn Sie eine Zählschleife wünschen, müssen Sie mit einer while-Schleife und einer arithmetischen Operation (expr) arbeiten.

```
for [VARIABLE] in [LISTE]
do
....
done
```

while-Schleife

Eine While-Schleife wird nur durchlaufen, wenn die Bedingung wahr ist

```
while [Bedingung]
do
....
done
```

until-Schleife

Eine until-Schleife wird mind. einmal durchlaufen.
Mehrmals, solange die Bedingung wahr ist

```
until [Bedingung]
do
....
done
```

Startskripten (Ziel des Programmierexkurses)

```
#! /bin/sh
# 2004-12-20-bs-0001 bs@muekno.de
start ()
{
Datum=`date`
echo "Starte den Dienst um $Datum"
}
stop ()
{
Datum=`date`
echo "Stoppe den Dienst um $Datum"
}
case $1 in
        status) PIDS=`ps aux | grep sdjadhagh | grep -v grep | cut -b9-14`
            if [ "$PIDS" = "" ]
            then echo "Is not running!"
            else echo "Is running with PID(s): $PIDS"
            fi
            }
        start)  start ;;
        stop) stop ;;
        restart) stop; start ;;
        *) echo "Usage $0 [start|stop|restart|status]
esac
```

Startskripten müssen mindestens auf die Übergabewerte ($1) start und stop reagieren. Sie liegen in der Regel in /etc/init.d. Starten (Stoppen = K99startskript) in Runlevel 3:
ln -s /etc/init.d/startskript.sh /etc/rc.d/rc3.d/S99startskript

Linux basic programmierung

Version 2008-02-04

Dipl.Inf. (FH) Bernd Schwägerl
Dipl.Inf. (FH) Stefan Edenhofer

Abfragen

if-Statement:

```
if [Bedingung]
then
 echo „Bedingung war wahr=0“
else
 echo „Bedingung war unwahr=1“
fi
```

case-Statement:

```
case [VARIABLENWERT] in
                    1) echo „war der Wert 1“ ;;
                    2) echo „war der Wert 2“ ;;
                    *) echo „war irgendwas anderes ;;
esac
```

Unterrroutinen (Funktionen)

Funktionen definieren:

```
[funktionsname] ()
{
# Ausgabe des Übergabeparameters
echo $1
}
```

Funktionen aufrufen:

[funktionsname] UEBERGABEPARAMETER

Teststatements (Gebräuchlichste Art der Bedingungen)	
[$A = $B]	$A gleich $B (Chars, Strings)
[$A != $B]	$A nicht $B (Chars, Strings)
[$A -eq $B]	$A = $B (nummerisch)
[$A -ne $B]	$A nicht $B (nummerisch)
[$A -lt $B]	$A < $B (nummerisch)
[$A -gt $B]	$A > $B (nummerisch)
[$A -le $B]	$A <= $B (nummerisch)
[$A -ge $B]	$A=> $B (nummerisch)
[-r file]	file existiert und ist lesbar
[-w file]	file existiert und ist beschreibbar
[-x file]	file existiert und ist ausführbar
[-f file]	file ist ein FILE und existiert
[-d file]	file ist ein DIR und existiert
Verknüpfen von Teststatements (AND und OR):	
[$A > $B -o $C < $B]	OR-Verknüpfung
[$A > $B -a $C < $B]	AND-Verknüpfung

Da jeder Shellskriptaufruf einen Aufruf einer eigenen Shell darstellt, sind Variablen in oder aus "Subskripten" nicht verfügbar. (Abhilfe: ". /skript")