

HSLU  
Rotkreuz  
Information & Cyber Security

# Linux notes

Autor: David Jäggi

# Inhaltsverzeichnis

Inhaltsverzeichnis .....	I
1 General.....	1
1.1 Some commands .....	1
2 BASH basics .....	1
2.1 Shell expansion .....	1
2.2 Pipes.....	1
2.3 Common shortcuts .....	2
3 Scripting .....	3
3.1 Regex.....	4
4 Cronjobs.....	5
5 Archives.....	6
5.1 tar.....	6
5.2 rsync.....	6
6 File system .....	7
7 Links / Copies .....	8
8 VIM.....	9
9 SSH .....	10
10 User management .....	11
10.1 General.....	11
10.2 User types .....	11
10.3 /etc/passwd .....	11
10.4 Groups.....	12
10.5 Different sudos .....	12
10.6 User commands .....	13
10.7 Group commands .....	13
11 Permission management .....	14
11.1 Commands .....	14
12 Logging.....	16
12.1 systemd-journald .....	16
12.2 rsyslog .....	16
13 Process/Service .....	17
13.1 Process description .....	17
13.2 control processes .....	18
13.3 Systemd-daemon .....	19
13.4 system control commands.....	19
14 Network .....	20
14.1 Immer:.....	20
14.2 Testen .....	21
14.3 Konfigurationsdatei .....	21
14.4 IP route tool .....	20

14.5	Network Manager Client (Alternative zu IP route).....	20
14.6	IP commands.....	20
15	YUM .....	21
16	Partitions.....	22
16.1	Fragen .....	23
16.2	Tool parted.....	22
16.3	.....	23
16.4	Nice .....	24

# 1 BASH basics

## Recursive commands:

- mkdir -p(v)
- ls -R
- others: -r

## Variables in names:

- text\_\$(variable)\_text
- common ones are:
  - o \$(date +%F) → JJJJ-MM-DD
  - o \$(date +%s) → seconds since 01.01.1970

## 1.1 Shell expansion

Term	Meaning
*	Anything no matter how long (inclusive none)
?	Exactly one character
[ac]	a or c
{a, b, c, d}	Every character gets processed (a.txt, b.txt ...)
{1..4}	1, 2, 3, 4
{a..c}	a, b, c
file_{a, b}{1, 2}	file_a1, file_a2, file_b1, file_b2
file_{a{1, 2}, b, c}	file_a1, file_a2, file_b, file_c

## 1.2 Pipes

### command redirections:

- channel 0 > stdin
- channel 1 > stdout
- channel 2 > stderr
- 1> out.txt 2>&1 (both gets written into out.txt)

### to files:

> overwrites file  
>> appends to file

### 1.3 Common shortcuts

Shortcut	Output
Ctrl-U	Delete all chars
Ctrl-K	Delete all chars to the right of the cursor
Ctrl-A	Set cursor to the beginning
Ctrl-E	Set cursor to the end
Ctrl-W	Delete the word to the left
Ctrl-R	Search history
Ctrl-H	Like backspace
Ctrl-D	Like delete

## 2 Scripting

### Variables:

```
NAME="value"
echo $NAME
```

if the variable is in a string use:

```
"${VAR}"
```

### get input:

```
read VAR_NAME
```

### executing commands:

```
$(ls -la)
```

### logic (0 = true !!):

```
if [ condition ] ; then
    command
else
    command2
fi

for VARIABLE in 1 2 3 4 5          / for VARIABLE in file1 file2 file3
do
    command1 on $VARIABLE
    command2
done
```

### operators:

```
!=      -ne
==      -eq
```

### for global access:

/usr/local/bin

if scripts are used for programs:

/usr/bin

||:

if one fails the other gets executed

```
ls a.txt || touch a.txt // if file doesn't exists it creates it
```

&&:

both commands get executed

```
ls a.txt && cat a.txt
```

## 2.1 Regex

Sign	Output
<b>^</b>	Stands for beginning of line
<b>?</b>	The preceding item is optional and will match once at most
<b>\$</b>	Stands for end of line
<b>+</b>	The preceding item will be matched 1 or more times
<b>^(string)\$</b>	(String) Must be the only word in the text to match true
<b>.</b>	Placeholder for any character
<b>*</b>	The preceding character will match true 0 or more times
<b>.*</b>	Means none or anything

more:

### Übersicht

#### Regular Expressions

OPTION	DESCRIPTION
.	The period (.) matches any single character.
?	The preceding item is optional and will be matched at most once.
*	The preceding item will be matched zero or more times.
+	The preceding item will be matched one or more times.

OPTION	DESCRIPTION
{n}	The preceding item is matched exactly n times.
{n,}	The preceding item is matched n or more times.
{,m}	The preceding item is matched at most m times.
{n,m}	The preceding item is matched at least n times, but not more than m times.
[alnum:]	Alphanumeric characters: '[alpha:]' and '[digit:]' in the 'C' locale and ASCII character encoding, this is the same as '[0-9A-Za-z]'.
[alpha:]	Alphabetic characters: '[lower:]' and '[upper:]' in the 'C' locale and ASCII character encoding, this is the same as '[A-Za-z]'.
[blank:]	Blank characters: space and tab.
[cntrl:]	Control characters. In ASCII, these characters have octal codes 000 through 037, and 177 (DEL). In other character sets, these are the equivalent characters, if any.
[digit:]	Digits: 0 1 2 3 4 5 6 7 8 9.
[graph:]	Graphical characters: '[alnum:]' and '[punct:]'.
[lower:]	Lower-case letters; in the 'C' locale and ASCII character encoding, this is a b c d e f g h i j k l m n o p q r s t u v w x y z.

[print:]	Printable characters: '[alnum:]', '[punct:]', and space.
[punct:]	Punctuation characters; in the 'C' locale and ASCII character encoding, this is: " # \$ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` {   } ~. In other character sets, these are the equivalent characters, if any.
[space:]	Space characters: in the 'C' locale, this is tab, newline, vertical tab, form feed, carriage return, and space.
[upper:]	Upper-case letters: in the 'C' locale and ASCII character encoding, this is A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.
[xdigit:]	Hexadecimal digits: 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.
\b	Match the empty string at the edge of a word.
\B	Match the empty string provided it is not at the edge of a word.
\<	Match the empty string at the beginning of word.
\>	Match the empty string at the end of word.
\w	Match word constituent. Synonym for '[_[:alnum:]]'.
\W	Match non-word constituent. Synonym for '[^[:alnum:]]'.
\s	Match white space. Synonym for '[[:space:]]'.

### 3 Cronjobs

```
1 #M S T M W Befehl
2 #-----
3 5 9-20 * * * /home/BENUTZERNAME/script/script1.sh > /dev/null
4 */10 * * * * /usr/bin/script2.sh > /dev/null 2>&1
5 59 23 * * 0,4 cp QUELDATEI ZIELDATEI
6 * * * * * DISPLAY=:0 LANG=de_DE.UTF-8 zenity --info --text "
  Beispiel für das Starten eines Programms mit GUI"
7 0 0 * * * backup
8 0 1 8-14 1-12 * [ "$(date +%u)" == "6" ] && Befehl
9 #-----
```

HSLU 1. Dezember 2022

3. Fünf Minuten nach jeder vollen Stunde zwischen 9 und 20 Uhr (also 9:05, 10:05, ..., 20:05)
4. Alle 10 Minuten
5. Jeden Sonntag und Donnerstag um 23:59
6. Jede Minute ein Programm mit GUI, das die Display- und die Sprach-Variable benötigt
7. Jeden Tag Punkt Mitternacht 00:00 Uhr
8. Jeden 2. Samstag im Monat um 01:00 Uhr (Wichtig, das die entsprechende Shell-Variable (hier: /bin/bash) gesetzt ist)

Page «Nr.»

#### example combinations:

- now +5min
- teatime tomorrow (teatime is 16:00)
- noon +4days
- 5pm may 12 2022



## 4 Archives

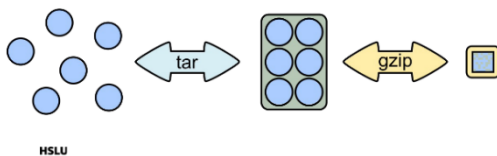
### 4.1 tar

#### Ausgewählte tar-Optionen

```
[root@host ~]# tar -czf /root/etcbackup.tar.gz /etc
tar: Removing leading '/' from member names
```

```
[root@host ~]$ tar -cjf /root/logbackup.tar.bz2 /var/log
tar: Removing leading '/' from member names
```

```
[root@host ~]$ tar -cJf /root/sshconfig.tar.xz /etc/ssh
tar: Removing leading '/' from member names
```



#### Übersicht über tar-Operationen

Option	Beschreibung
-c, --create	Neues Archiv erstellen.
-x, --extract	Aus vorhandenem Archiv extrahieren.
-t, --list	Inhaltsverzeichnis eines Archivs auflisten.

#### Ausgewählte, allgemeine tar-Optionen

Option	Beschreibung
-v, --verbose	Verbose. Zeigt, welche Dateien archiviert oder extrahiert werden.
-f, --file=	File name. Auf diese Option muss der Dateiname des zu verwendenden oder zu erstellenden Archivs folgen.
-p, --preserve-permissions	Berechtigungen für Dateien und Verzeichnisse beim Extrahieren eines Archivs beibehalten, ohne die <b>Aufhebung der Maskierung</b> zu entfernen.

#### Übersicht über tar Komprimierungsoptionen

Option	Beschreibung
-z, --gzip	Komprimierung gzip verwenden (.tar.gz).
-j, --bzip2	Komprimierung bzip2 verwenden (.tar.bz2). bzip2 erzielt in der Regel eine bessere Komprimierungsrate als gzip.
-J, --xz	Komprimierung xz verwenden (.tar.xz). Die Komprimierung xz erzielt in der Regel eine bessere Komprimierungsrate als bzip2.

### 4.2 rsync

#### Mit rsync -a aktivierte Optionen (Archivmodus)

Option	Beschreibung
-r, --recursive	zum rekursiven Synchronisieren der gesamten Verzeichnisstruktur
-l, --links	zum Synchronisieren von symbolischen Links
-p, --perms	zum Beibehalten von Berechtigungen
-t, --times	zum Beibehalten von Zeitstempeln
-g, --group	zum Beibehalten der Gruppeneigentümerschaft
-o, --owner	zum Beibehalten des Eigentümers der Dateien
-D, --devices	zum Synchronisieren von Gerätedateien

## 5 File system

**list partitions:**

```
lsblk (with UUID: -fp)
```

**list processes which use the partition:**

```
lsof /mnt/[partition]
```

New block devices / hard drives must be mounted. They are first visible under /dev.

```
mkdir /mnt/newspace
```

```
mount UUID="uuid" /mnt/newspace
```

## 6 Links / Copies

### **Symbolic or Softlinks:**

- Only a pointer to the original file.
- Is cross-filesystem possible.
- has different permissions / timestamps but get updated if chown is called with -h.
- doesn't work if file is renamed/moved.
- Works on folders.

### **Hardlinks:**

- Is basically a copy of the original file.
- inherits the permission / timestamps.
- Is not cross-filesystem possible.
- No problem if file gets renamed/moved.
- Doesn't work on folders.

### **Deeper explanation:**

Inode is basically the system pointer which points to a file on the filesystem. A hardlink points to the Inode and has therefore anytime the most current timestamps/location/permissions etc. A softlink "points" to the name of the file.

## 7 VIM

Input	Mode
(standard)	Normal mode. Only basic functions
I	Insert mode
:	Extended mode
V	Visual mode
P	Insert
Y	Copy
D	Delete
X	Delete in normal mode
:<x><cmd>	Does the command x times
:q	Quit
:w	save

## 8 SSH

```
ssh -arg serveraddr -l user
```

or

```
ssh user@serveraddr
```

-Y → graphical stuff gets sent.

### Connect with RSA keys:

```
ssh-keygen
```

```
ssh-copy-id user@serveraddr
```

```
ssh-agent (vault for different keys)
```

### Keys on server:

copy the key into ~/.ssh/authorized\_keys or use ssh-copy-id

### Options:

```
/etc/ssh/ssh_config
```

```
sudo systemctl restart ssh(d)
```

### Known hosts:

in ~/.ssh/known\_hosts or systemwide /etc/ssh/known\_hosts

## 9 User management

### 9.1 General

If you change the UID of a user manually he has no longer access to his files, because they are bound to a UID.

**see permissions:**

```
ls -l      -> with usernames
ls -ln     -> with UIDs
ls -ld     -> for a directory itself and not the content
```

### 9.2 User types

- root (UID 0)
- Systemuser (UID 100 – UID 200 -> fixed)
- Systemuser (UID 201-999 -> dynamic, after every installation)

**more (local) information with:**

```
id [username]
```

**includes external added users:**

```
getent [list]
getent passwd {startid..endid}
```

### 9.3 /etc/passwd

The screenshot shows a terminal window titled 'labstudent@lios-workstation-dev-01:~/Documents'. The terminal displays the contents of the `/etc/passwd` file. The entries are as follows:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
gnome-initial-setup:x:975:975::/run/gnome-initial-setup:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
florian:x:1003:1003::/home/florian:/bin/bash
labstudent:x:1004:1004::/home/labstudent:/bin/bash
labadmin:x:1005:1005::/home/labadmin:/bin/bash
```

Annotations with arrows point to specific parts of the file:

- /etc/passwd**: Points to the filename at the top right.
- Benutzername**: Points to the username 'labstudent' in the line 'labstudent:x:1004:1004::/home/labstudent:/bin/bash'.
- Passwort. Aus Sicherheitsgründen ausgeblendet → /etc/shadow**: Points to the 'x' character, indicating a placeholder for the password stored in `/etc/shadow`.
- Zugeordnete UID des Nutzers**: Points to the UID '1004'.
- Primäre Gruppe**: Points to the primary group '1004'.
- Home-Verzeichnis**: Points to the home directory '/home/labstudent'.
- Interpreter / Shell**: Points to the shell '/bin/bash'.

At the bottom of the terminal, there is a line: `-- INSERT --` followed by '6,32' and 'All'.

## 9.4 Groups

### Primary group:

- most of the time same name as username.
- gets assigned at user creation.
- Visible in `/etc/passwd`

### Other groups:

- Visible in `/etc/group`
- User can have multiples of it.

## 9.5 Different sudos

Die Unterschiede zwischen `su`, `sudo`, `sudo -i`, `sudo -s`, `sudo su...`

Empfohlen!

Befehl	Wirkung
<code>su</code>	Startet eine Shell (Interpreter) als Root im Kontext des aktuellen Nutzers (gleiche Umgebungsvariablen, gleiches aktuelles Verzeichnis)
<code>su -</code> , <code>su -l</code>	Startet eine <i>neue</i> Shell als Root (neuer Login). Arbeitsverzeichnis ändert sich. Neue Umgebungsvariablen. Shell des Root-Nutzers.
<code>su -l labadmin</code>	Startet eine <i>neue</i> Shell als Benutzer <i>labadmin</i> . Vollständige neue Shell / neuer Login. Neue Umgebungsvariablen etc.
<code>sudo -i</code>	Startet eine <i>neue</i> Shell (neuer Login). Arbeitsverzeichnis ändert sich... siehe „ <code>su -</code> “ oder „ <code>su -l</code> “
<code>sudo -s</code>	Startet eine Shell im Kontext des aktuellen Nutzers, siehe „ <code>su</code> “
<code>sudo -u labadmin -i</code>	Startet eine neue Shell als Benutzer <i>labadmin</i> , siehe „ <code>su -l labadmin</code> “

Die häufig verwendete **`sudo su`**-Kombination funktioniert wie folgt: Zuerst fragt `sudo` Sie nach Ihrem Passwort und ruft, wenn Sie dazu berechtigt sind, den nächsten Befehl (`su`) als Superuser auf. Da `su` von `root` aufgerufen wird (wegen `sudo`), müssen Sie das Kennwort des Zielbenutzers nicht eingeben. Mit **`sudo su`** können Sie also eine Shell als ein anderer Benutzer (einschließlich `root`) öffnen, wenn Ihnen Superuser-Zugriff durch die Datei `/etc/sudoers` gewährt wird. Generell: Es wird empfohlen `sudo -i` statt `sudo su` zu nutzen.

HSLU 20. Oktober 2022

## 9.6 User commands

### Basic user creation with options:

```
useradd username  
-g [primary group name]  
-m (creates home directory)  
-u [UID]  
-s [shell path]  
-c [Description]
```

### Change settings of a user:

```
usermod username
```

### Delete user (-r with deletion of user directories):

```
userdel -r username
```

### Basic user creation:

```
useradd username
```

### Basic user creation:

```
useradd username
```

```
[labadmin@lios-workstation-dev-01 ~]$ sudo useradd -u 1221 -g student /home/student/hslu_student -s /bin/ksh -c "HSLU Student Reto Meyer" reto
```

## 9.7 Group commands

```
sudo groupadd  
sudo groupdel
```



## 10 Permission management

### Directories:

read -> ls lists items.

write -> can move, rename and delete items in directory

execute -> can move into it with cd

### 10.1 Commands

#### Permission absolute:

Darstellung	Oktalschreibweise	Bedeutung
[r-----]	400	read (Leserecht für Eigentümer)
[-w-----]	200	write (Schreibrecht für Eigentümer)
[--x-----]	100	execute (Ausführrecht für Eigentümer)
[rwx-----]	700	read, write, execute (Lese-, Schreib- und Ausführungsrecht)
[---r-----]	040	read (Leserecht für Gruppe)
[----w-----]	020	write (Schreibrecht für Gruppe)
[-----x----	010	execute (Ausführrecht für Gruppe)
[---rwx----	070	read, write, execute für Gruppe
[-----r--]	004	read (Leserecht für alle anderen Benutzer)
[-----w-]	002	write (Schreibrecht für alle anderen Benutzer)
[-----x]	001	execute (Ausführrecht für alle anderen Benutzer)
[-----rwx]	007	read, write, execute für alle anderen Benutzer

```
chmod [number] file/directory
```

#### Permission relative:

**Wer:** u,g,o,a (user, group, other, all)

**Wie:** +, -, = (add, remove, set)

**Was:** r,w,x (read, write, execute)

```
chmod WerWieWas file/directory
```

#### Change owner:

```
chown owner:owner_group file/dir
```

-h -> softlinks permissions get updated

## Special bits:

Recht	Binärwert	Oktalwert
SUID	2 <sup>2</sup>	4
SGID	2 <sup>1</sup>	2
Sticky Bit	2 <sup>0</sup>	1

- **SUID Bit (4)** → temporäre Eigentümerschaft für den Aufrufer
  - **GUID Bit (2)** → Vererbung der Gruppeneigentümerschaft des Verzeichnisses auf neue Dateien und Verzeichnisse
  - **Sticky Bit (1)** → Sticky-Bit an einem Verzeichnis, nur noch Besitzer einer Datei kann diese löschen
- 
- **SUID Bit (4)** → temporäre Eigentümerschaft für den Aufrufer
    - Setzen des SUID-Bits ist **nur sinnvoll auf Binärdateien**, die ausführbar sind
    - Es beeinflusst die Eigentümerschaft beim Ausführen von Dateien
    - Beispiel:  
/etc/shadow → -rw-r----- → keine Schreibrechte für Nutzer  
-rwsr-xr-x /usr/bin/passwd → SUID für **passwd**-Befehl bewirkt, dass alle Nutzer /etc/shadow ändern dürfen – aber nur „über“ /usr/bin/passwd.
  - **GUID Bit (2)** → Vererbung der Gruppeneigentümerschaft des Verzeichnisses auf neue Dateien und Verzeichnisse
    - **SGID-Bit an Verzeichnis** → ab diesem Zeitpunkt wird jeder neue Eintrag unterhalb des Verzeichnisses immer der Gruppe gehören, die in diesem Verzeichnis als besitzende Gruppe eingetragen ist.
    - SGID-Bit vererbt sich auch auf alle neu erstellten Unterverzeichnisse
  - **Sticky Bit (1)** → **Sticky-Bit an einem Verzeichnis**, nur noch Besitzer einer Datei kann diese löschen
    - Mit dem Sticky-Bit an einem Verzeichnis, kann nur noch der Besitzer einer Datei in dem Verzeichnis diese auch löschen. Auch ein anderer Benutzer mit Rechten "r,w,x", ist er nicht berechtigt eine Datei zu löschen.
    - Das Sticky-Bit macht nur Sinn, wenn es auf Verzeichnissen gesetzt wird.

## 11 Logging

### 11.1 systemd-journald

- newer
- logs in binary format

read the binary logging file:

```
journalctl (-p err)
```

search for unit (e.g. ssh or cron)

```
journalctl-u [value] (short for journalctl _SYSTEMD_UNIT="value")
```

search last n entries:

```
journalctl -n
```

### 11.2 rsyslog

- writes the logs into /var/log in text format

/var/log/messages is /var/log/syslog in Debian/Ubuntu

Protokolldatei	Beschreibung und Inhalt
/var/log/messages	Zeigt allgemeine Meldungen und Informationen zum System an. Im Grunde ein Datenprotokoll aller Aktivitäten im gesamten System, ohne Meldungen zur Authentifizierung, für die E-Mail-Verarbeitung, zur Ausführung terminierter Jobs. Debian/Ubuntu: /var/log/syslog
/var/log/secure	Authentifizierungsprotokolle sowohl für erfolgreiche als auch für fehlgeschlagene Anmeldungen und Authentifizierungsprozesse. Die Speicherung hängt vom Systemtyp ab: Debian/Ubuntu in /var/log/auth.log. Redhat/CentOS in /var/log/secure.
/var/log/boot.log	Startmeldungen und Startinformationen. Konsolenmeldungen im Zusammenhang mit dem Systemstart, nicht von Syslog.
/var/log/cron	Speichert Nachrichten zu wiederkehrende Aufgaben und automatisierten Jobs: Start, Ende und Meldungen.
/var/log/maillog	Für Mailserver-Protokolle, praktisch für Postfix-, SMTPD- oder E-Mail-bezogene Dienstinformationen, die auf Ihrem Server ausgeführt werden.
/var/log/mysql oder /var/log/httpd	Einige Prozesse und Dienste legen eigene Unterverzeichnisse an und speichern ihre Logs in Unterverzeichnissen.

## 12 Process/Service

### list processes:

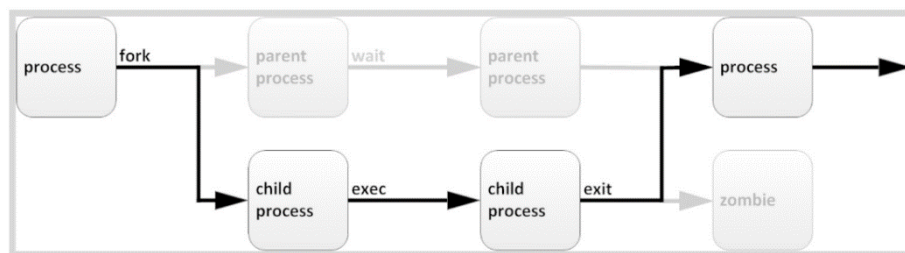
```
ps -aux  
top
```

### 12.1 Process description

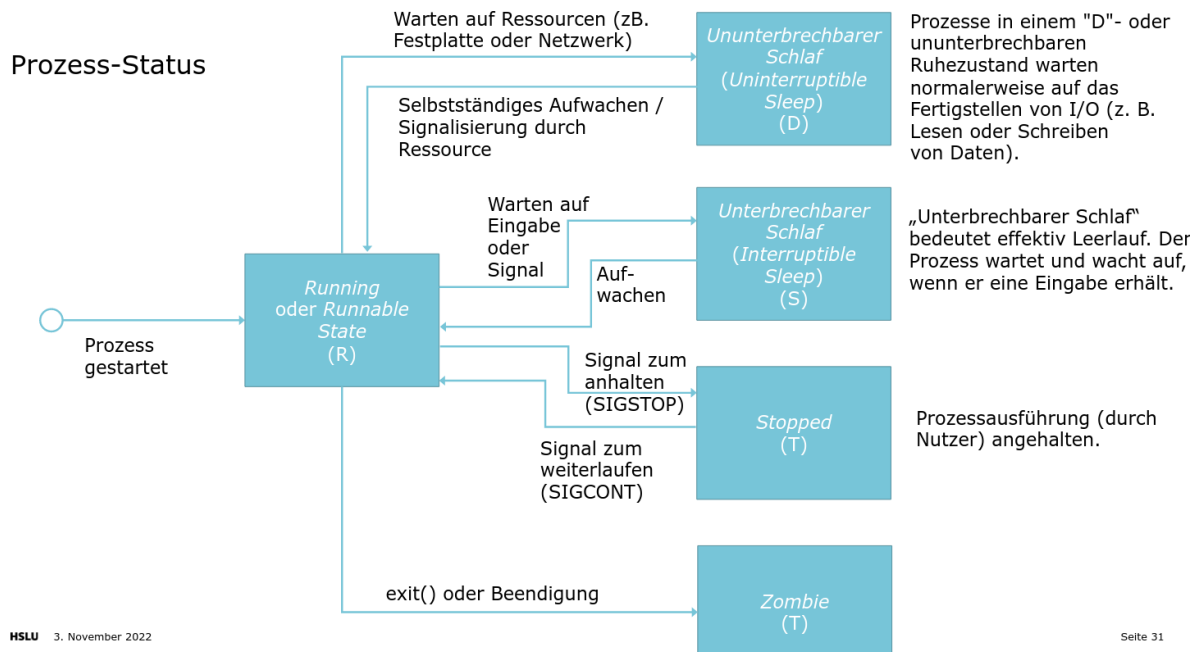
#### Definition eines Prozesses

- Ein *Prozess* ist ein Computerprogramm, das gerade ausgeführt wird.
  - Ein Computerprogramm bewirkt nichts, solange es nicht gestartet wurde.
  - Wird ein Programm gestartet/ausgeführt, wird eine Kopie des Programms in den Arbeitsspeicher geladen.
  - So wird diese Instanz zu einem (Betriebssystem-)Prozess, der zum Ablauf einem Prozessorkern zugeordnet werden muss.
- Ein Prozess besteht nicht nur aus dem laufenden Computerprogramm sondern zusätzlich aus Folgendem:
  - Ressourcenzuteilungen: Einem Adressraum von zugewiesenem Arbeitsspeicher
  - Sicherheitseigenschaften: wie Eigentümerinformationen und -berechtigungen
  - Verwaltungsinformationen: Prozessstatus
- Die *Umgebung* eines Prozesses beinhaltet Folgendes:
  - Lokale und globale Variablen
  - Einen aktuellen Umgebungskontext, z. B. Interpreter, Eltern-Prozess
  - Zugewiesene Systemressourcen wie Dateideskriptoren und Netzwerkports

#### Prozess-Lebenszyklus



- Über Routine *fork* erbt ein untergeordneter Prozess die Sicherheitsmerkmale, alte und aktuelle Dateideskriptoren, Port- und Ressourcenberechtigungen, Umgebungsvariablen und den Programmcode.
- Ein untergeordneter Prozess kann dann eigenen Programmcode ausführen (*exec*).
- Im Normalfall befindet sich ein übergeordneter Prozess im *Ruhezustand*, während der untergeordnete Prozess ausgeführt wird, und stellt die Anforderung (*wait*), bei Abschluss des untergeordneten Prozesses benachrichtigt zu werden.



## Grundlegende Prozessverwaltungssignale

Name	Nr.	Aktion	Standard	Bedeutung
SIGSTOP	19	Anhalten	POSIX	Der Prozess wurde angehalten (blockiert)
SIGTSTP	20	Anhalten	POSIX	Der Prozess wurde „von Hand“ durch den Benutzer angehalten (STRG-Z)
SIGCONT	18	Weiterlaufen	POSIX	Ein angehaltener Prozess soll weiterlaufen.
SIGINT	2	Ende	POSIX	Interrupt durch das Terminal oder den Benutzer (STRG-C)
SIGQUIT	3	Ende & Core	POSIX	Das Signal <i>quit</i> vom Terminal. Ähnlich zu SIGINT aber mit Speicherabbild (STRG-\)
SIGTERM	15	Ende	POSIX	Beendigung des Programms anfordern. Programme, die SIGTERM abfangen, bieten meistens einen »Soft Shutdown« an.
SIGKILL	9	Ende	POSIX	<i>Sofortiges Beenden</i> des Programmprozess
SIGHUP	1	Ende	POSIX	Das (virtuelle) Terminal oder der Interpreter wurde geschlossen oder beendet

## 12.2 control processes

```
kill [-option] <pid>
```

options are the numbers from the table above.

## 12.3 Systemd-daemon

- *systemd* führt das Konzept der Systemd-Units ein.

Typ	Dateierweiterung	
Systemd unit	-	Oberbegriff für alle Units in <i>systemd</i>
Service unit	.service	Systemdienst
Target unit	.target	Eine Gruppe von Systemd-Units
Mount unit	.automount	Automatischer Einhängpunkt (mount point) im Dateisystem
Socket unit	.socket	Kommunikation zwischen Prozessen: Wenn ein Client eine Verbindung zum Socket herstellt, startet <i>systemd</i> den entsprechenden Daemon und leitet die Verbindung weiter.

## 12.4 system control commands

### detailed information about a service:

```
systemctl stats (e.g., sshd.service)
start/stop [service]
restart [service]
reload [service]
list-dependencies [service]
is-active [service]
is-enabled [service]
is-failed [service]
--failed -all
daemon-reload (systemctl reload)
```

## 13 Network

### 13.1 Always do:

- Configure IP address & subnet mask.
- Configure router.

### 13.2 How to test

ping -c5 192.168.0.5

ss (socket statistics) [-ua]

ss -la (tcp ports)

tracpath <domain> (is like tracer)

### 13.3 Configuration file

/etc/sysconfig/network-scripts/ifcfg-ens192

### 13.4 IP route tool

Should be used to display Information:

```
ip
```

#### variations

```
ip route (routing informationen)
```

```
ip link (zeigt netzwerke von Netzwerkkarte an)
```

```
ip addr show dev ens192 (-s für Statistike)
```

```
ens192 (en -> ethernet; s192 Hot PCI Steckplatz 192)
```

```
ip addr show
```

### 13.5 Network Manager Client

Should be used for configurations:

nmcli

- device show
- device status
- connection show
- connection show System\ ens192 show

### 13.6 IP commands

```
nmcli connection add conn-name hslu ens256 type ethernet ifname  
ens256 ipv4.address 192.168.0.5/24 ipv4.gateway 192.168.0.254
```

## 14 YUM

```
yum info <package-name>
yum install <package-name>
yum remove <package-name>
yum history
yum history info <package-name>
yum history undo <number-in-history>
yum history redo <number-in-history>
yum history rollback <number-in-history>
yum history rollback last-1
```

### For component groups:

```
yum group <command>
```

### For modules:

```
yum module info (python36)
```

### External repos:

```
yum config-manager --add-repo <url>
yum repolist all
```



## 15 Partitions

### use GBT instead of MBR

GBT has more options, theoretically no limit of partitions and can handle bigger disks.

System max partitions = 128

msdos = MBR

persistent mount of file systems -> /etc/fstab/

### 15.1 Tool parted

parted can manage partitions.

```
1 [root@lios-server-dev-01 ~]# parted /dev/sdb
2 GNU Parted 3.2
3 Using /dev/sdb
4 Welcome to GNU Parted! Type 'help' to view a list of commands.
5 (parted) mkpart
6 Partition type? primary/extended? primary
7 File system type? [ext2]? xfs
8 Start? 2048s
9 End? 1000MB
10 (parted) quit
11 Information: You may need to update /etc/fstab.
12 [root@lios-server-dev-01 ~]# udevadm settle
```

Erstellen einer MBR-  
Festplattenpartition

is equivalent to:

```
1 [root@lios-server-dev-01 ~]# parted /dev/sdb mkpart primary xfs 2048s
1000MB
```

Erstellen einer GPT-  
Festplattenpartition

```
1 [root@lios-server-dev-01 ~]# parted /dev/sdb
2 GNU Parted 3.2
3 Using /dev/sdb
4 Welcome to GNU Parted! Type 'help' to view a list of commands.
5 (parted) mkpart
6 Partition name? []? backup
7 File system type? [ext2]? xfs
8 Start? 2048s
9 End? 1000MB
10 (parted) quit
11 Information: You may need to update /etc/fstab.
12 [root@lios-server-dev-01 ~]# udevadm settle
```

```
1 [root@lios-server-dev-01 ~]# parted /dev/sdb mkpart backup xfs 2048s
1000MB
```

222

udevadm settle -> wait until partitioning is finished.

## 15.2 Delete partitions

### Löschen von Partitionen

Die folgenden Schritte gelten sowohl für das MBR- als auch für das GPT-Partitionierungsschema.

```
1 [root@lios-server-01 ~]# parted /dev/sdb
2 GNU Parted 3.2
3 Using /dev/sdb
4 Welcome to GNU Parted! Type 'help' to view a list of commands.
5 (parted) print
6 Model: VMware Virtual disk (scsi)
7 Disk /dev/sdb: 5369MB
8 Sector size (logical/physical): 512B/512B
9 Partition Table: gpt
10 Disk Flags:
11
12 Number  Start   End     Size    File system  Name  Flags
13  2       1048kB  1049kB  512B
14  1       1049kB  1000MB  999MB   xfs          backup
15
16 (parted) rm 1
17 (parted) quit
18 Information: You may need to update /etc/fstab.
```