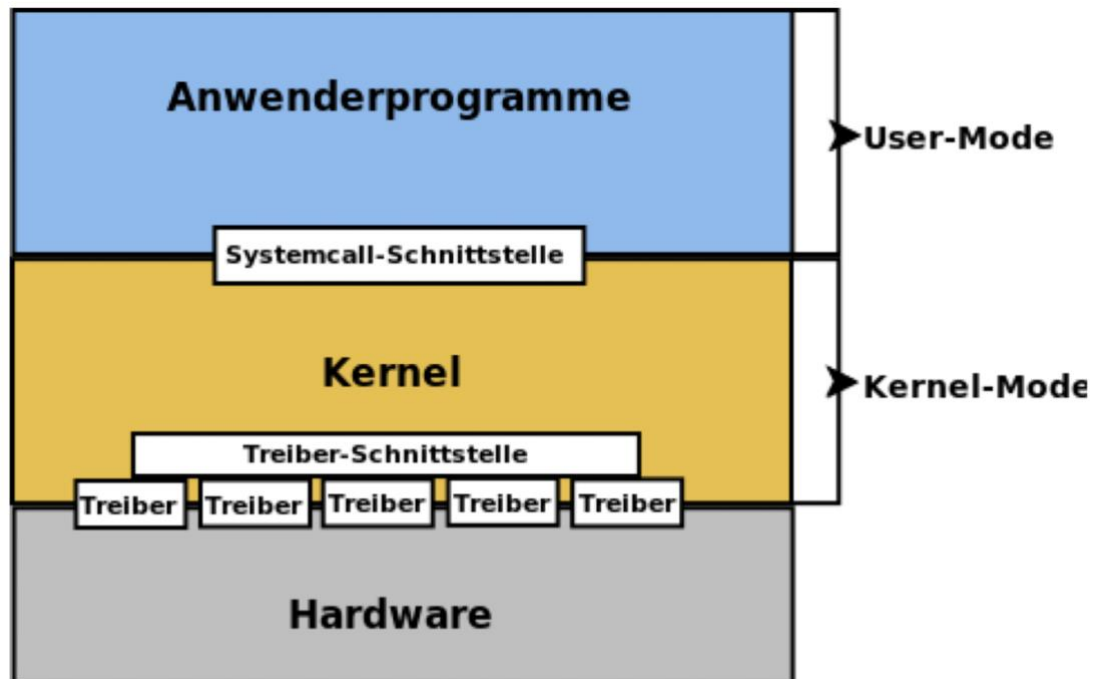
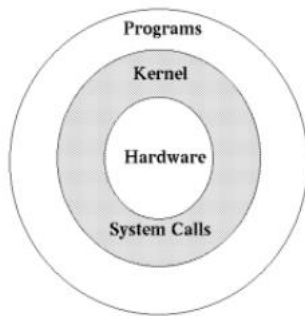


1. Einführung in Linux

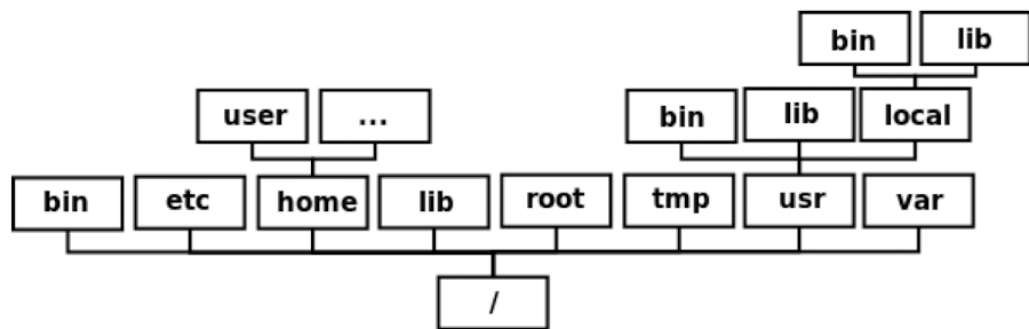
Linux ist ein Betriebssystem wie Windows, iOS, MacOS und Co. Die Entwicklung startete 1970 und basiert auf dem Betriebssystem Unix.

1.1 HW-Layers (HW, Kernel, User-Mode)



- Multi-user (mehrere Benutzer gleichzeitig), Multi-Process (mehrere Programme gleichzeitig)
- Kernel steuert mithilfe von Systemcalls zugriff zur Hardware. Für Abstraktion, Sicherheit und Übersicht
- Mithilfe von Kernelmodulen können Hardwarefeatures angesprochen werden. Z.B. Networking
- GUI gehört nicht zu Linux. Ist ein eigenes Programm das im User-Mode läuft.

1.2 Aufbau von Linux Verzeichnisstruktur



- Das **Wurzelverzeichnis** `/` - ist die root-Partition und beinhaltet alle vorhandenen Verzeichnisse auf dem System.
 - `/bin` - hier liegen die Binärdateien (ausführbare Dateien), die für Systemaufgaben benötigt werden
 - `/boot` - hier liegt der Linux-Kernel und alle zum Start benötigten Dateien
 - `/dev` - hier liegen die Gerätedateien, welche die Schnittstelle zwischen Kernel und Hardware sind (die Hardware ist über diese Dateien ansprechbar)
 - `/etc` - hier liegen Konfigurationsdateien
 - `/home` - hier liegen die Home-Verzeichnisse der einzelnen User
 - `/lib` - hier liegen Systembibliotheken
 - `/mnt` & `/media` - hier befinden sich Unterverzeichnisse eingehängter Geräte (`/media` meist Datenträger die nach dem Systemstart eingehängt wurden)
 - `/proc` - beinhaltet Dateien mit Informationen des laufenden Systems
 - `/root` - ist das Home-Verzeichnis des Users root (nicht im `/home`-Verzeichniss, damit das System gestartet werden kann ohne Partitionen einzuhängen - bei Fehlern)
 - `/sbin` - beinhaltet die System-Binärdateien
 - `/tmp` - zum Ablegen temporärer Dateien, wird beim Neustart gelöscht
 - `/usr` („Unix System Resources“) - wichtige Programme die das System anbietet
 - `/usr/bin`, `/usr/lib`, `/usr/sbin` - wie im Wurzelverzeichnis
 - `/usr/local` - dient zur Trennung von distributionseigenen, fremden und selbst kompilierte Programmen
 - `/usr/share` - beinhaltetet architekturunabhängige Dateien, die für den Betrieb diverser Programmen benötigt werden
 - `/usr/src` - Quelltexte für Programme des Standardsystems
 - `/var` - beinhaltet Dateien die häufig aktualisiert/neu beschrieben werden (z.B. Logdateien)
 - `/opt` - sollte verwendet werden um Programmpakete hier zu installieren und nicht mehr im `/usr`-Verzeichnis

1.3 Hilfreiche Programme:

- man oder <Programm> --help
- whoami (Eingeloggter User)
- id (Eingeloggter User + zusatzinfos)
- su (switch user)
- ps (Prozesse anzeigen)

1.4 Manövrieren in Linux:

- Pwd (print working directory)
- Cd (change directory)
- Ls (List → Files anzeigen)
- Mkdir (make directory → Ordner erstellen)
- Rm (remove → löschen von files)
- Rmdir (remove directory → löschen von ordnern)
- Cp (copy)
- Mv (move)
- Cat (ausgeben des inhalts einer datei)
- Less oder more (durchsehen des inhalts in einer datei)
- Touch (erstellen einer leeren datei)

1.5 unix command line Aufbau

Command options arguments

<i>ls</i>	<i>options and arguments are optional</i>
<i>ls -ls /etc /bin/*a*</i>	<i>usually options first</i>
<i>ls -ls -a</i>	<i>in one word or one by one</i>
<i>ls -- -la</i>	<i>-- ends the option list!! , -la' is an argument here</i>
<i>ls --all</i>	<i>--longFormatOption (user frindly version here an alternative to -a the GNU way to write options)</i>

1.6 Aufbau von Programmausführungen:

<Programmname> <Optionen> <Argrumente>

z.B.

ls → zeigt das den Inhalt des aktuellen Verzeichnisses

ls -l → zeigt den Inhalt in einer Liste

ls -l -a oder ls -la → zeigt den inhalt in einer Liste (-l) und zeigt auch versteckte dateien an (-a)

ls -la / → zeigt den vollständigen Inhalt in einer Liste vom Wurzelverzeichnis (Rootverzeichnis) an

1.7 Standard-Rechte / Default Permissions

Everything is a file! → We only need file permissions!

Permission	File	Directory	Device / Spec. File
r	read, copy	read (name + InodeNr)	read
w	write, change	write (create, rename, delete)	write
x	execute	use Inodes of dir-entries (what can you do without??)	---

Jede Datei und jedes Verzeichnis (ist auch eine Datei) hat eine kombination aus 3 Rechten: Read, Write, Execute.

Jede Datei hat einen Besitzer (Owner) und eine Gruppe (group)

Rechte können für drei Kategorien freigegeben werden

- User
- Gruppen
- Öffentlich

Rechte werden in Bits gespeichert:

user	group	other
r w x	r w -	r - -
1 1 1	1 1 0	1 0 0
7	6	4

1.8 Hilfreiche Befehle zur verwendung von Files:

- `ls -l <dateiname>` → (zeigt die Rechte einer bestimmten datei an)

```
patrickzivkovic@MacBook-Air ~ % ls -l datei
-rw-r--r-- 1 patrickzivkovic staff 0 Jul  8 11:44 datei
```

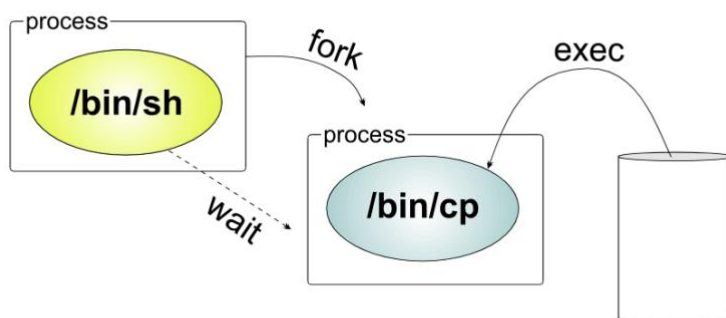
- `Chown <ursprungs-user>:<ursprungs-gruppe> <zieluser>:<zielgruppe>` → ändert den besitzer bzw. die gruppe einer datei
- `Chmod` → verschiedene formen, ändert die rechte einer datei
 - `Chmod u+rw <Dateiname>` → fügt dem Benutzer alle Rechte zu dieser Datei hinzu
 - `Chmod 700 <Dateiname>` → !Ändert! die Rechte der Datei damit NUR der Benutzer alles mit der Datei machen darf
 - `Chmod g-rw` → Entzieht der Gruppe alle Rechte der Datei
 - `Chmod o+rx` → jeder darf die Datei lesen und ausführen

1.9 Prozesse in Linux

Prozesse in Linux sind Hierarchisch aufgebaut. Der erste Prozess der mit dem Systemstart beginnt hat die Prozess-ID (PID) 1

Jeder weitere Prozess der gestartet wird stammt von diesem Prozess ab:

```
$ pstree -p | less
init(1)---atd(632)
    |-bdf flush(5)
    |-cardmgr(1201)
    |-cron(1540)
    |-kalar md(1967)
    |-kdeinit(1882)---artsd(1906)
    |               |-kdeinit(1914)
    |               |-kdeinit(1942)---bash(1949)
    |               |               `--bash(1951)---pstree(2586)
    |               |-kdeinit(1947)
    |               |-kdeinit(1959)
    |               |-kdeinit(2246)
```



1.10 Benutzer in Linux und UNIX

Benutzer werden unter Linux und UNIX in eine Benutzerdatenbank gespeichert. Es kann ausgelesen werden mit `cat /etc/passwd`:

```
(parallels@kali-gnu-linux-2023)-[~]  
$ cat /etc/passwd  
root:x:0:0:root:/root:/usr/bin/zsh  
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin  
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

Getrennt mit `:` sind Infos zu dem Benutzer gespeichert. Der Administrator und mächtigste Benutzer in Linux ist der `root` Benutzer. Dieser überschreibt alle Rechte und kann alles auf dem System machen.

Daher empfiehlt es sich für die tägliche Benutzung oder für spezielle Anwendungen eigene Benutzer anzulegen.

Mit `adduser <Username>` kann ein neuer Benutzer angelegt werden.

Um trotzdem Programme zu installieren oder auch um neue Benutzer anzulegen brauchen die Standard-Benutzer allerdings mehr Rechte. Um temporär mehr Rechte zu erhalten, kann vor einen Befehl das Schlüsselwort `sudo` verwendet werden. Z.B.: `sudo adduser parallels`

`<Benutzername>:<Passwort(wird-extra gespeichert):User-ID:Gruppen-ID:<Gruppen>:<heimlaufwerk>:<Standard-Start-Applikation>`

```
parallels:x:1000:1000:parallels,,,:/home/parallels:/usr/bin/zsh
```

Die Passwörter der Benutzer werden in einer eigenen Datei gehashed gespeichert:

`cat /etc/shadows:`

```
parallels:$y$j9T$ShgE00WU8Ae0RZwXSbfDq/$7w7MmaCl6vnS7D/KRxg8P6TB8Kqh3
```

Mit `su` kann der Benutzer gewechselt werden. Z.B.: `su root` um in den Root Benutzer zu wechseln und dauerhaft als der Administrator zu agieren.

2. Netzwerkkonfiguration in Linux:

2.1 Anzeigen der IP-Adresse:

Mit dem Befehl `ip address` oder `ip a` kann die Netzwerk Information angezeigt werden

```
(parallels@kali-gnu-linux-2023)-[~]  
$ ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host proto kernel_lo  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:1c:42:5e:bd:e4 brd ff:ff:ff:ff:ff:ff  
    inet 10.211.55.12/24 brd 10.211.55.255 scope global dynamic noprefixroute eth0  
        valid_lft 1401sec preferred_lft 1401sec  
    inet6 fdb2:2c26:f4e4:0:fcf1:8c23:ee15:6672/64 scope global temporary dynamic  
        valid_lft 604406sec preferred_lft 85558sec  
    inet6 fdb2:2c26:f4e4:0:21c:42ff:fe5e:bde4/64 scope global dynamic mngtmpaddr noprefixroute  
        valid_lft 2591963sec preferred_lft 604763sec  
    inet6 fe80::21c:42ff:fe5e:bde4/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

- `lo` und `eth0` sind die Schnittstellen (Interfaces) die mit an dem Computer angeschlossen sind. `lo` steht für Loopback und kümmert sich um Netzwerkverkehr auf dem Computer selbst. `eth0` ist die Ethernetschnittstelle des Computers und ist die Verbindung in Netzwerk
- `link/ether` gibt die MAC-Adresse an
- `inet` gibt die IPv4 Adresse an
- `inet6` gibt die IPv6 Adresse an

2.2 Ändern der IP-Adresse

Für eine leichtere Handhabung und dauerhafte Änderung der Netzwerkschnittstellen kann das Programm `nmcli` verwendet werden:

Einige hilfreiche Kommandos:

- `Nmcli dev status` → zeigt alle Schnittstellen und deren Status an
- `nmcli con add con-name "static-eth0" ifname eth0 type ethernet ip4 10.0.0.44/24 gw4 10.0.0.1` → erstellen einer neuen IP-Vorlage für die Schnittstelle `eth0`, mit der IP-Adresse `10.0.0.44/24` und dem Standardgateway `10.0.0.1` (!Achtung! die IP-Adresse ist noch nicht aktiv)
- `nmcli con up "static-eth0"` → Aktivieren der ursprünglich erstellten IP-Vorlage, jetzt ist die Verbindung hergestellt!
- `nmcli con down "static-eth0"` → Deaktivieren der IP-Vorlage