

# Betriebssysteme

## 7. Systemadministration

Lehrveranstaltung im Studienschwerpunkt Verwaltungsinformatik

erstellt durch:

Name: Karl Wohlrab  
Telefon: 09281 / 409-279  
Fax: 09281 / 409-55279  
Email: [mailto: Karl.Wohlrab@fhvr-aiv.de](mailto:Karl.Wohlrab@fhvr-aiv.de)

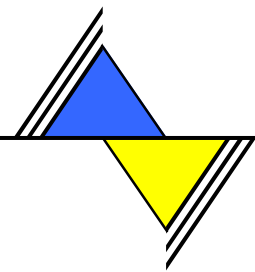
Der Inhalt dieses Dokumentes darf ohne vorherige schriftliche Erlaubnis des Autors nicht (ganz oder teilweise) reproduziert, benutzt oder veröffentlicht werden.

Das Copyright gilt für alle Formen der Speicherung und Reproduktion, in denen die vorliegenden Informationen eingeflossen sind, einschließlich und zwar ohne Begrenzung Magnetspeicher, Computerausdrucke und visuelle Anzeigen.

### Anmerkungen

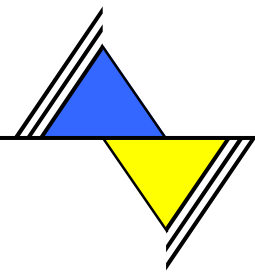
Bei dem vorliegenden Skriptum handelt es sich um ein Rohmanuskript im Entwurfsstadium. Das Skript wird begleitend zur Lehrveranstaltung fortgeschrieben und überarbeitet.

Es erhebt keinen Anspruch auf Vollständigkeit und Korrektheit. Inhaltliche Fehler oder Ungenauigkeiten können ebenso wenig ausgeschlossen werden wie Rechtschreibfehler.



# Inhalt

<b>1</b>	<b>Systemadministration allgemein</b>	<b>3</b>
1.1	Benutzerverwaltung.....	3
1.1.1	Autorisierung.....	5
1.1.1.1	Arten der Autorisierung.....	5
1.1.1.2	Möglichkeiten der Benutzerverwaltung im Netzwerk.....	8
1.1.2	Dateizugriffsschutz.....	9
1.1.2.1	Rechte und Attribute auf Dateien im Vergleich .....	10
1.1.2.2	Rechte und Attribute auf Verzeichnissen .....	10
1.1.3	Verteilte Dateisysteme.....	11
1.2	Backup und Recovery .....	12
1.3	Boot-Konzept .....	12
<b>2</b>	<b>Systemadministration unter LINUX</b>	<b>13</b>
2.1	Benutzerverwaltung.....	13
2.1.1	Dateien der Benutzerverwaltung .....	13
2.1.2	Der Login-Ablauf .....	14
2.1.3	Die Datei /etc/passwd .....	15
2.1.4	Gültigkeitsdauer des Passworts .....	16
2.1.5	Die Datei /etc/group .....	17
2.1.6	Die Datei /etc/shadow .....	18
2.1.7	Die Datei /etc/default/passwd .....	19
2.1.8	Anlegen eines Benutzers .....	20
2.1.9	Sperren eines Benutzers .....	20
2.1.10	Löschen eines Benutzers.....	21
2.2	Backup und Recovery .....	22
2.2.1	Sicherungsmedien.....	22
2.2.1.1	Magnetbänder und Streamer.....	22
2.2.1.2	Mehrere Dateien auf einem Magnetband.....	24
2.2.1.3	Dateien (Archive) auf mehreren Magnetbändern .....	24
2.2.1.4	Floppystreamer.....	25
2.2.1.5	QIC-Streamer .....	25
2.2.1.6	SCSI-Streamer .....	26
2.2.1.7	Disketten.....	26
2.2.2	Methoden der Datensicherung .....	27
2.2.2.1	Gesamtsicherung (Full-Backup) ##### .....	27
2.2.2.2	Inkrementelles Backup .....	27
2.2.3	Backup-Software.....	29
2.2.4	Datensicherung mit tar .....	30
2.2.5	Datensicherung mit afio und cpio .....	33
2.3	Boot-Konzept .....	34
2.3.1	Das Programm init .....	35
2.3.2	Runlevel .....	36
2.3.3	Die init-Scripten.....	37
<b>3</b>	<b>Systemadministration unter WINDOWS-XP</b>	<b>38</b>
3.1	Benutzerverwaltung.....	38
3.2	Backup und Recovery .....	38
3.3	Boot-Konzept .....	38



# 1 Systemadministration allgemein

## 1.1 Benutzerverwaltung

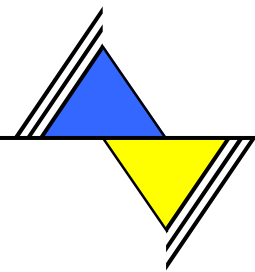
Quelle: Plate-Script: Betriebssysteme

In Mehrbenutzersystemen ist es üblich, jedem Einzelbenutzer eine fest eingegrenzte Arbeitsumgebung zu bieten, die ihn von anderen Benutzern abschirmt:

- ☒ Prozesse eines Benutzers können gleichzeitig laufende Prozesse anderer Benutzer nicht beeinflussen (Speicherschutz, etc.)
- ☒ Auf Dateien eines Benutzers können andere Benutzer nur mit dessen ausdrücklicher Erlaubnis zugreifen
- ☒ Für gemeinsam genutzte Ressourcen und Dateien können definierte Zugriffsrechte vergeben werden
- ☒ Die Nutzungsdauer (Rechenzeit), der Zeitraum der Nutzung (z. B. nur von 8 ... 17 Uhr) oder die maximal zu beanspruchende Plattenkapazität können festgelegt werden
- ☒ Zu Beginn der Arbeit am Rechner (login) muss sich der Benutzer identifizieren (Benutzerauthentizität)

### Aufgaben der Benutzerverwaltung

- ➡ **Prozessschutz**
- ➡ **Dateischutz**
- ➡ **Schutzmaßnahmen für bestimmte Geräte oder Anwendungen**
- ➡ **Nutzungsrechte für bestimmte Geräte oder Anwendungen**
- ➡ **Abrechnungsinformationen (Accounting)**
  - ★ **Login-Zeit**
  - ★ **verbrauchte CPU-Zeit / Rechenzeit**
  - ★ **Belegter Speicherplatz (RAM, Festplatte,...)**
  - ★ **Ressourcenbedarf (gedruckte Seiten, ...)**
  - ★ **Online-Zeiten (z.B.: Internetnutzung)**
  - ★ **Übertragenes Datenvolumen**
  - • •

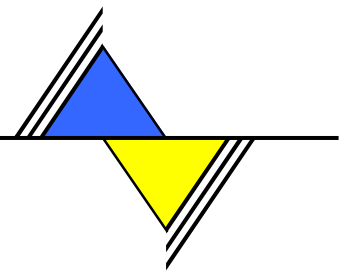


Bei besonders kritischer Rechnerumgebung kann es sogar wünschenswert erscheinen, dass sich die Benutzer beim Aufruf bestimmter Programme nochmals identifizieren müssen.

Die o. g. Schutzmaßnahmen sind ins Betriebssystem integriert (z. B. Prozess-Schutz oder Dateischutz). Die Benutzeridentifikation wird durch ein eigenes Login-Programm vollzogen. Spezielle Geräte erfordern besondere Schutzmaßnahmen; so ist es inzwischen üblich, bei Benutzern, die sich über ein Modem (also per Telefon) anmelden, diesen nicht sofort Zugang zum Rechner zu gewähren, sondern den Anschluss des Benutzers vom Rechner aus zurückzurufen (Call-Back-Verfahren). Zusätzliche Aufgaben der Benutzerverwaltung betreffen Abrechnungsinformationen (Accounting):

- ⊗ die Zeit, die der Benutzer angemeldet ist
- ⊗ die verbrauchte Rechenzeit
- ⊗ die Verweilzeit von Prozessen im Rechner
- ⊗ Zugriff und Nutzung der Ressourcen (Platte, Drucker, etc.)





Derartige Informationen werden in bestimmten Dateien des Betriebssystems protokolliert (so genannte 'Logfiles'), deren Information statistisch ausgewertet werden kann.



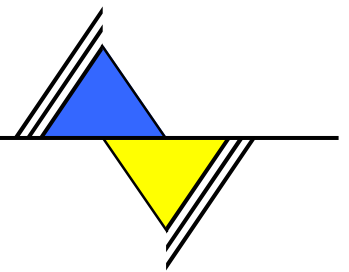
## 1.1.1 Autorisierung

### 1.1.1.1 Arten der Autorisierung

#### Arten der Authorisierung

-  **Knowledge**
-  **Challenge**
-  **Token**
-  **Biometrie**

- ⊗ **Knowledge:** Passwort, PIN  
Dies ist der derzeit häufigste Fall. Das Verfahren ist einfach zu implementieren, Passwörter und PINs können geändert werden. Sie lassen sich jedoch auch unbemerkt und einfach kopieren. Sichere Passwörter müssen lang sein und sollten oft geändert werden. Sie sind daher für die Benutzer schwer zu handhaben. Passwörter/PINs können vergessen werden.
- ⊗ **Challenge:**  
Es wird eine Frage gestellt, deren Antwort überprüft wird und die nur die autorisierte Person wissen sollte. Das Frage-Antwort-Protokoll kann auch auf kryptographischen Verfahren basieren (Public Key Kryptosysteme). Auch eine Liste von Einmal-TANs gehört in diese Rubrik.
- ⊗ **Token:** Schlüssel, Smartcard, SecureID Card, usw.  
Mit 'Token' wird als etwas 'Greifbares' bezeichnet. Tokens sind einfach in der Anwendung. Sie lassen sich zudem mit anderen Sicherungssystemen/Datenerfassungssystemen z.B. Zeiterfassung oder Passwort kombinieren. Unbemerkte Kopien sind schwierig bis unmöglich. Ein Nachteil ist der Kostenaufwand durch zusätzliche Hardware. Karten oder Schlüssel können vergessen oder verloren werden. Die unbefugte Benutzung verlorener Tokens ist möglich.
- ⊗ **Biometrie:** Fingerabdruck, Gesichtserkennung, Irisscan, Schriftodynamik, Tippverhalten, DNA-Analyse, usw.  
Auch Biometrieverfahren sind einfach in der Anwendung. Biometrische Merkmale können nicht vergessen werden. Derzeit (2002/2003) sind die angebotenen Verfahren jedoch noch nicht für allgemeine Anwendung geeignet. Einfache Sensoren lassen sich relativ leicht austricksen, z.B. durch Kopie eines Fingerabdrucks oder Fotos. Der Kostenaufwand durch zusätzliche Hardware ist noch recht hoch. Biometrische Informationen lassen sich schwer oder gar nicht ändern.



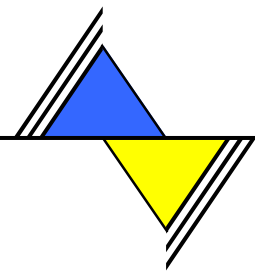
Das Passwort ist der kritische Punkt in jeder Benutzerauthentisierung. Man hat deshalb die Sicherheit von Passwortprüfungen ständig verbessert.

- ☒ Passworte werden einwegverschlüsselt gespeichert. Sie sind nicht entschlüsselbar. Die Prüfung erfolgt durch Einwegverschlüsselung der Tastatureingabe und Vergleich der beiden einwegverschlüsselten Bitmuster.
- ☒ Passworte müssen eine definierte Mindest- und Maximallänge haben.
- ☒ In einer Liste festgelegte Worte oder Wortbestandteile dürfen nicht verwendet werden, um leicht zu erratende "weiche" Passworte wie "System", "Test" usw. zu verhindern.
- ☒ Passworte müssen nach Ablauf einer definierten Frist geändert werden.
- ☒ Passworte dürfen nicht oder erst nach n-maliger Änderung wieder benutzt werden.

## Passworte

- ➔ **verschlüsselte Speicherung (Einwegverschlüsselung)**
- ➔ **definierte Mindest- und Maximallänge**
- ➔ **Bestandteile eines Passworts (Buchstaben, Ziffern, Sonderzeichen, ...)**
- ➔ **bestimmte Worte (oder Wortbestandteile) dürfen nicht verwendet werden**
- ➔ **begrenzte Gültigkeitsdauer**
- ➔ **Wiederverwendung erst nach n-maliger Änderung**
- ➔ **Protokollierung mißglückter Login-Versuche**

Trotzdem ist das Passwort in Verruf geraten, weil seine Sicherheit ausschließlich vom sorgfältigen Umgang des Benutzers abhängt. Befragungen von Computerbenutzern haben immer wieder ergeben, dass sie das Passwort nur allzu sorglos behandeln.



Eine vernünftige Passwortpolitik trägt in sehr wesentlichem Maße zur Sicherheit bei. Dies setzt voraus, dass die Benutzer die Bedeutung von Passwörtern und deren Verwendung respektieren. Es gibt jedoch von Unternehmen zu Unternehmen und von Mitarbeiter zu Mitarbeiter große Unterschiede im Grad der Passwortdisziplin. Die wenigsten Mitarbeiter würden auch nur davon träumen, die ID-Karte ihrer Firma irgendwo liegenzulassen, wohingegen dieselbe Person weit größere Nachlässigkeit im Zusammenhang mit der Wahl oder Weitergabe von Passwörtern zeigt. Nutzlose Passwörter, wie z. B. der Vorname des Benutzers oder gleich die Benutzer-ID, kommen auch häufig vor.

Damit die Passwörter effektiv angewendet werden, ist es notwendig, eine Reihe von Richtlinien in Form einer entsprechenden Passwortpolitik festzulegen. Die Passwortpolitik des lokalen Netzes sollte Richtlinien für folgende Punkte enthalten:

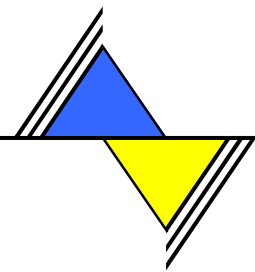
- ☒ Mindestlänge des Passworts (Mindestanzahl von Zeichen)
- ☒ Bestandteile eines Passworts (Buchstaben, numerische Zeichen und Sonderzeichen)
- ☒ Maximaler Gültigkeitszeitraum eines Passworts und Regeln für die obligatorische Erneuerung sowie Begrenzungen für die Wiederverwendung von Passwörtern.
- ☒ Protokollierung der Benutzeraktionen nach einer vorgegebenen Anzahl missglückter Login-Versuche.
- ☒ Regeln für die Übertragung von Passwörtern an andere.

Im Gegensatz zu einem physischen Gegenstand, wie z. B. ein Schlüssel, dessen Verlust man bemerkt, kann ein Passwort entschlüsselt werden, ohne dass sich der Benutzer dessen unmittelbar bewusst wird.

- ☒ Dagegen hilft nur die zeitliche Begrenzung des Gültigkeitszeitraums von Passwörtern.
- ☒ Die Regel sollte streng gehandhabt werden, d. h. Sperrung von Benutzer-IDs, die es unterlassen, ihr Passwort zu ändern.
- ☒ Müssen Passwörter zu häufig aktualisiert werden, tendieren die Benutzer dazu, Passwörter aufzuschreiben.
- ☒ Bei 30-tägiger Passwortlebensdauer neigen viele Benutzer dazu, den Monatsnamen als Passwort zu verwenden.
- ☒ Benutzer-ID und Passwort dürfen nie von mehreren Benutzern gleichzeitig verwendet werden.
- ☒ Passwörter müssen streng vertraulich und persönlich sein.
- ☒ Ein Benutzer darf unter keinen Umständen sein Passwort an andere weitergeben.

Die Passwortpolitik des Netzes sollte aus den Benutzungsrichtlinien deutlich hervorgehen und konsequent gehandhabt werden

- ☒ Eine starke Authentisierung, wie sie zunehmend gefordert wird, kann jedoch nicht mehr auf dem Passwort basieren.
- ☒ Sie kann beispielsweise auf dem "Challenge-Response"-Prinzip beruhen. Voraussetzung hierfür ist neben der Software ein so genannter Token, ein spezieller Taschenrechner im Kreditkartenformat mit Tastatur, Verschlüsselungsprozessor und persönlichem Schlüssel (z.B. die SecureID-Card).
- ☒ Zunächst muss sich der Benutzer des Tokens durch Eingabe seiner PIN als berechtigter Besitzer ausweisen.
- ☒ Wenn der Benutzer sich am Computersystem anmelden möchte, generiert dieser eine Zufallszahl (Challenge), die am Bildschirm erscheint.
- ☒ Der Benutzer tippt die Challenge in seinen Token ein, der durch Verschlüsselung die Antwort (Response) ermittelt.
- ☒ Diese wiederum wird als Passwort in die Computertastatur eingegeben und vom Computersystem verifiziert.
- ☒ Manche Tokens, wie z.B. SecureID, generieren die Zufallszahl zeitabhängig selbst, so dass eine Eingabe entfällt.



Challenge-Response hat folgende Vorteile:

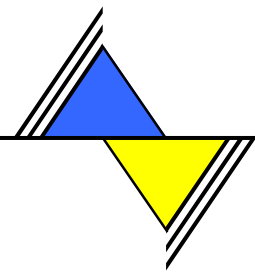
- ⊗ Ein Passwort besteht immer nur für eine Sitzung. Ein Hacker, der das Passwort abfangen würde, könnte damit nichts anfangen, weil bei der nächsten Anmeldung ein neues Passwort generiert wird.
- ⊗ Es wird das Prinzip "Besitz und Wissen" angewendet. Ein Benutzer muss über einen Token verfügen und die richtige PIN kennen. Ein verlorener Token ist für den Finder solange wertlos, solange die PIN unbekannt bleibt.

SmartCards:

- ⊗ Die Verwendung von SmartCards als Token anstelle des speziellen Rechners macht die starke Authentisierung benutzerfreundlicher und sehr sicher.
- ⊗ Das Verfahren ist ähnlich, läuft aber im Wesentlichen automatisch ab.
- ⊗ Die einzige Aktion des Benutzers besteht aus seiner Authentisierung gegenüber der SmartCard durch Eingabe einer PIN oder durch ein biometrisches Merkmal.
- ⊗ Danach fordert der Host die SmartCard durch Übermittlung einer Random-Zahl heraus (Challenge).
- ⊗ Die SmartCard errechnet die Antwort und sendet sie an den Host zurück, der sie verifiziert (Response).
- ⊗ Nachdem die Anwendung von SmartCards nicht mehr so teuer ist, hat sie nur noch Vorteile gegenüber den auf Taschenrechnern basierenden Tokens.

### **1.1.1.2 Möglichkeiten der Benutzerverwaltung im Netzwerk**

Ist nicht Thema der vorliegenden Lehrveranstaltung sondern vielmehr Thema der Lehrveranstaltung Netzwerke.



## 1.1.2 Dateizugriffsschutz

Der Dateieintrag im Verzeichnis wird um einen Schutzeintrag erweitert, das die Zugriffsrechte festlegt. Diese Rechte können je nach System feiner oder gröber gestaffelt sein. Am einfachsten ist die Vergabe von Lese- oder Schreibrecht für alle Benutzer. Die Zugriffsrechte selbst lassen sich weiter unterteilen und auch für verschiedene Benutzer unterschiedlich gestalten - bis hin zur Zuteilung dedizierter Rechte an einen oder mehrere bestimmten Benutzer.

Einen Mittelweg beschreitet UNIX, das hier als einfaches Beispiel dienen soll. Für eine Datei gibt es unter UNIX drei Zugriffsarten:

- R (read): Lesen aus der Datei
- W (write): Schreiben in die Datei (Ändern/Verkürzen/Erweitern)
- X (execute): Ausführen der Datei (bei Programmen)

Da es durchaus möglich sein sollte, anderen Benutzern das Lesen (nicht aber das Schreiben) einer Datei zu gestatten; es u. U. auch Dateien mit wichtigen Informationen gibt, die man vor eigenen Fehlern schützen will, gibt es drei Gruppen der Zugriffsberechtigung:

- Zugriffsrechte für den Datei-Eigentümer
- Zugriffsrechte für die Arbeitsgruppe des Eigentümers
- Zugriffsrechte für alle anderen

Es ergeben sich also neun Zugriffsrechte:

S	I	S	G	I	S	T	I	R	W	X	R	W	X	R	W	X
Spezial				Benutzer			Gruppe			Andere						

Die im Bild links angegebenen weiteren drei Zugriffsrechte dienen besonderen Aufgaben. Ein Benutzer kann durchaus mehreren Arbeitsgruppen angehören. Durch die Zuordnung von Gerätedateien zu bestimmten Gruppen, kann man die Verwendung der Geräte auf eine bestimmte Gruppe von Programmen beschränken.

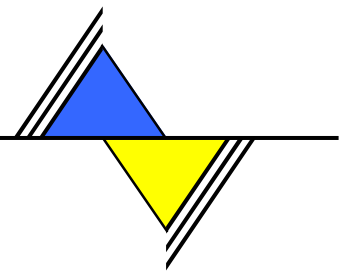
Beispiele:

Der Zugriff auf die Druckerschnittstellen wird nur einem User gewährt. In diesem Fall stehen die Drucker nur den Druckerprozess zur Verfügung und sind für andere Prozesse gesperrt.

Die seriellen Schnittstellen, an denen Modems angeschlossen sind, werden der Gruppe "Modem" zugeordnet. Um Benutzern den Zugriff auf die Modems zu gestatten, muss der Systemverwalter nur diese Benutzer der Gruppe "Modem" zuordnen.

Bei einigen Betriebssystemen werden die Zugriffsrechte noch verfeinert, so lässt sich beispielsweise die Schreiberlaubnis noch weiter unterteilen:

- beliebiges Schreiben auf die Datei
- Ändern von bestimmten Teilen der Datei
- Anhängen von Daten an die Datei
- Löschen der Datei



### 1.1.2.1 Rechte und Attribute auf Dateien im Vergleich

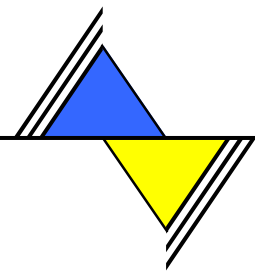
Attribut/Recht	Windows NT/2000	Unix	Netware
Lesen	R	r	R
Schreiben	W	w	W
Löschen	D	w	E
Ausführen	X	x	X
Rechte setzen	P	w (Verzeichnis)	A
Besitzer ändern	O	w (Verzeichnis)	A
Attribute ändern	W	w (Verzeichnis)	M
Super-Rechte		root-User	S

### 1.1.2.2 Rechte und Attribute auf Verzeichnissen

Attribut/Recht	Windows NT/2000	Unix	Netware
Schreiben	W	w	W
Löschen	D	w	E
Verzeichnis anzeigen	X	x	F
Rechte setzen	P	w	A
Besitzer ändern	O	w	A
Attribute ändern	W	w	M
Super-Rechte	û	root-User	S
Dateien anlegen	C	w	C

#### Anmerkungen:

- Unter Windows NT/2000 können mit jeder Freigabe (Share) allgemeine rechte für Benutzer und Gruppen eingestellt werden. Dies ist eine Art von "Ersatz" für die fehlende Vererbung. Das "x" unter Unix kennzeichnet ausführbare Dateien.
- Die Namenserweiterung ist bei UNIX vollkommen egal.
- Das "X" unter Windows NT/2000 und Netware ist ein 'eXecute only'-Attribut.
- Unter Unix gibt es noch die Attribute "l" für symbolische Links (Zeiger auf eine andere Datei, 'Verknüpfung'), "s" beim Eigentümer für SUID (Ausführbare Dateien behalten die User-ID des Dateieigentümers, denn normalerweise laufen alle Programme unter der User-ID des Aufrufenden.) und "s" bei der Gruppe für SGID (Bei Dateien wie SUID, jedoch gruppenbezogen, bei Verzeichnissen wird die Gruppenzugehörigkeit neuer Dateien automatisch auf die des Verzeichnisses gesetzt).



Die Dateifreigabe erfolgt mit den jeweiligen Kommandos des Betriebssystems. Bei UNIX ist dies meist das Shell-Kommando 'chmod' (Change Modus), mit dem für Dateien, Verzeichnisse oder ganze Verzeichnisbäume die Zugriffsrechte gesetzt werden können. Die Zuordnung der Dateien und Verzeichnisse zu bestimmten Benutzern und Gruppen erfolgt über die Kommandos 'chown' (Change Owner) und 'chgrp' (Change Group), die aber nur vom Superuser 'root' verwendet werden können.

Bei Windows 95/98 können Sie Ordner und Drucker freigeben, indem Sie mit der rechten Maustaste darauf klicken und Freigabe auswählen. Dabei ist es möglich, nur Lese- oder Lese- und Schreibzugriff zu gestatten und dafür jeweils Passwörter einsetzen. Die Option Freigabe auf Benutzerebene kann nur in Verbindung mit einem Windows NT- oder Windows 2000-Server verwendet werden. Die Freigabe der Ordner und Drucker erfolgt ebenfalls durch unter 'Freigabe' im jeweiligen Kontextmenü (rechte Maustaste).

Bei NTFS-Laufwerken (Windows NT/2000) können Sie detaillierte Rechte für das Verzeichnis, alle Unterverzeichnisse oder auch einzelne Dateien vergeben. Im Kontextmenü unter Sicherheit-Berechtigungen legen Sie fest, welche Benutzer welche Aktionen ausführen dürfen. Sind Windows 95/98-Rechner im Netzwerk aktiv, dürfen Sie für die Freigabennamen nicht mehr als zwölf Zeichen verwenden. Ansonsten erscheint die Freigabe auf diesen Rechnern nicht. Windows NT/2000 versucht zunächst, Sie mit dem Benutzernamen und Kennwort anzumelden, das Sie bei der Windows-Anmeldung verwendet haben. Ist damit keine Anmeldung möglich, können Sie Kenn- und Passwort neu eingeben. Datei- und Druckerfreigaben sollten immer mit Passwort geschützt werden.

Viele Betriebssysteme kennen weitere Einschränkungen, z. B.:

☒ **Account Restrictions**

Den Benutzern kann eine Reihe von Einschränkungen auferlegt werden, unter anderem die Höchstanzahl gleichzeitiger Logins, eine Mindestlänge für Passwörter und eine Geltungsdauer für Passwörter (z. B.: UNIX, Novell).

☒ **Time Restrictions**

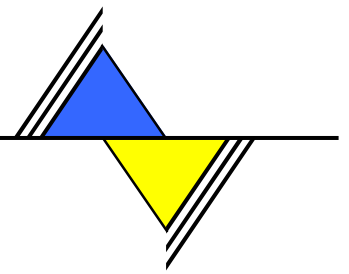
Der Login im Netz kann auf bestimmte Zeiten begrenzt werden, z. B. nur innerhalb der normalen Arbeitszeit. Die Zeitbeschränkung kann sowohl auf Default- als auch auf individueller Ebene angegeben werden (z.B.: Novell).

☒ **Station Restrictions**

Außer der Einschränkung der Anzahl sämtlicher Logins für die einzelne Benutzer-ID ist es auch möglich, das Login auf eine bestimmte Rechner beschränken. Die Station kann durch die MAC- oder IP-Adresse identifiziert werden.

### 1.1.3 Verteilte Dateisysteme

Ist nicht Thema der vorliegenden Lehrveranstaltung.

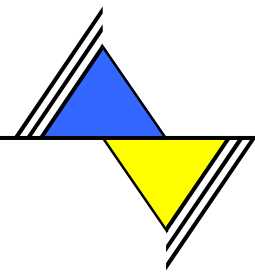


## 1.2 Backup und Recovery

Siehe Backup und Recovery unter LINUX

## 1.3 Boot-Konzept

Siehe Boot-Konzept unter LINUX



## 2 Systemadministration unter LINUX

### 2.1 Benutzerverwaltung

#### 2.1.1 Dateien der Benutzerverwaltung

Die Benutzerverwaltung ist bei UNIX ebenso offen aufgebaut, wie der Rest des Systems. Es existieren zwei Text-Dateien, die sämtliche Benutzerinformation aufnehmen:

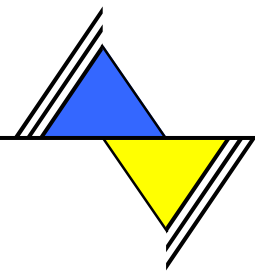
- ⊗ `/etc/passwd`  
nimmt Benutzerdaten und Passwort auf  
so war es jedenfalls bei den früheren Systemen. Deshalb gibt es seit einiger Zeit:
- ⊗ `/etc/shadow`  
nimmt das Passwort auf und ist nur von root lesbar
- ⊗ `/etc/group`  
nimmt die Gruppenzugehörigkeit auf

#### Dateien zur Benutzerverwaltung unter LINUX

<code>/etc/passwd</code>	Benutzerdaten und Passwort
<code>/etc/shadow</code>	Passwort und Zusatzinformationen
<code>/etc/group</code>	Gruppenzugehörigkeit
<code>/etc/skel</code>	Kopiervorlage für Home-Directory
<code>/etc/profile</code> <code>/etc/profile.local</code> <code>\$HOME/.profile</code>	Scripten, die beim Login abgearbeitet werden

Beim Login (und beim Wechsel des Benutzerkennzeichens während einer Terminalsitzung) wird auf diese Dateien zugegriffen - sie sind übrigens für alle Benutzer lesbar. Schreiben darf jedoch nur der Superuser und das Programm `passwd`.

Da `passwd` von jedem Benutzer aufgerufen werden kann, ergibt sich hier eigentlich ein Widerspruch. Gelöst wird das Problem durch das UID-Bit von `passwd`: während das Programm läuft, nimmt der Prozess die Identität seines Eigentümers an - und der darf schreiben.



### Anmerkung:

Ab der UNIX-Version System V, Version 3 steht das Passwort nicht mehr in `/etc/passwd`, sondern in einer eigenen Datei `/etc/shadow`.

Da `/etc/passwd` für alle lesbar sein muss (z. B. für die Anzeige des Benutzernamens im `ls`-Kommando), kann man mit entsprechenden Programmen versuchen, die Passwörter zu 'knacken'. Durch die Verlagerung der Passwortinfo in `/etc/shadow`, die nur von Superuser-Prozessen gelesen werden kann, wird diese Sicherheitslücke geschlossen.

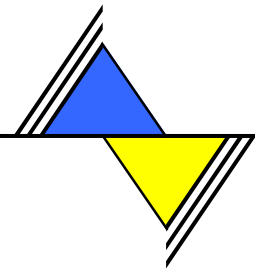
## 2.1.2 Der Login-Ablauf

Der Login-Prozess läuft immer in den folgenden Schritten ab:

- ⊗ Das Programm `login` erhält vom `getty`-Prozeß den eingegeben Benutzernamen (Login-Namen) und sucht nun in der Datei `/etc/passwd` nach einer passenden Zeile. Wird nichts gefunden wird der Benutzer nach der Passwortabfrage abgewiesen.
- ⊗ Ist der Benutzer gefunden, wird überprüft, ob ein Passwort eingetragen ist. Falls vorhanden, wird das Passwort abgefragt und bei fehlerhafter Eingabe der Benutzer abgewiesen.
- ⊗ Danach wird das im Benutzereintrag spezifizierte Programm gestartet. In der Regel ist dies die Shell.

### Wichtig:

- Das Paßwort ist in der Datei `/etc/passwd` (bzw. `/etc/shadow`) verschlüsselt gespeichert.
- Die Verschlüsselung erfolgt beim Ändern des Passworts mit dem Programm `passwd` oder bei der Eingabe im Login-Programm.
- Verglichen werden immer nur die verschlüsselten Passwörter.
- Auch der Superuser kann das Paßwort nicht entschlüsseln.
- Wenn Sie Ihr Paßwort vergessen haben, kann der Superuser nur ihr altes Paßwort löschen, damit Sie dann ein neues eintragen können.
- Um das "knacken" von Login-Name und Paßwort zu erschweren, fragt das Login-Programm auch dann das Paßwort ab, wenn schon der Benutzername falsch war.
- Außerdem wird beim Eingeben des Passworts nichts auf dem Bildschirm angezeigt.



### 2.1.3 Die Datei /etc/passwd

Die Datei ist eine normale Textdatei und sie enthält für jeden Benutzer genau eine Zeile mit 7 Feldern, die jeweils durch einen Doppelpunkt voneinander getrennt sind. Die Zeilenlänge darf 511 Zeichen nicht überschreiten, wobei die Länge der einzelnen Felder variabel ist.

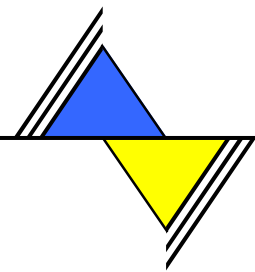
#### **/etc/passwd**

Der Aufbau der Zeile ist:

**Login-Name:Paßwort:UID:GID:Kommentar:Home-Directory:Programm**

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
...
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
irc:x:39:65534:IRC daemon:/usr/lib/ircd:/bin/bash
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
pop:x:67:100:POP admin:/var/lib/pop:/bin/false
mailman:x:72:67:GNU mailing list manager:/var/lib/mailman:/bin/bash
karl:x:500:100:./home/karl:/bin/bash
wohlab:x:501:100:./home/wohlab:/bin/bash
...
```

- Der Login-Name (3 .. 6 Zeichen) ist der Name, unter dem der Benutzer dem Betriebssystem bekannt ist.
- Er kann Großbuchstaben enthalten. Besteht er vollständig aus Großbuchstaben, schaltet UNIX auf Großschreibung um (historisch bedingt).
- Das Passwort ist verschlüsselt und es ist stets 13 Zeichen lang.  
ist das Feld leer muss kein Passwort eingegeben werden (nur RETURN-Taste).  
Der Superuser kann zusätzlich ein Komma und zwei weitere Zeichen eintragen, die die Gültigkeitsdauer festlegen (später mehr).  
Anmerkung: Ab Version 5.3 steht hier nur ein "x" und das Passwort in /etc/shadow.  
Das Passwort darf nicht mit dem Login-Namen übereinstimmen, muss mindestens 6-8 Zeichen lang sein und muss mind. zwei Buchstaben und eine Ziffer enthalten.
- UID = User Ident:  
In diesem Feld wird die Benutzernummer festgehalten (Wertebereich 0 bis 50000).  
Jeder Benutzer muss eine individuelle UID besitzen.  
Die 0 ist für den Superuser reserviert, die UIDs 1 - 99 für interne Zwecke.  
Reguläre Benutzer beginnen ab UID 100.



- **GID = Group ID:**  
In diesem Feld wird festgehalten, zu welcher Gruppe der Benutzer gehört (Wertebereich 0 bis 50000).  
Im Allgemeinen wird die GID 100 als Sammelgruppe verwendet (für alle Benutzer die keiner anderen Gruppe zugeordnet werden).
- **Kommentar:**  
In diesem Feld (max. Länge 30 Stellen) werden allgemeine Hinweise zum Benutzer eingetragen (normalerweise der vollständige Name des Benutzers, Abteilung, etc).  
Dieses Feld wird beispielsweise von Mail- und News-Programmen abgefragt, um automatisch den Absender einzutragen.
- **Home Directory:**  
Beim Eintragen eines Benutzers wird ihm auch ein Arbeitsverzeichnis, das Home Directory, eingerichtet. In dieses Verzeichnis wird beim Login verzweigt.  
Der Pfad muss vollständig eingegeben werden.
- **Programm:**  
Hier wird das Programm angegeben (vollständiger Pfadname, max. 256 Zeichen), das nach Beendigung von login gestartet werden soll.  
Bei normalen Benutzern ist dies der Kommandointerpreter, die Shell.  
Sie wird auch gestartet, wenn dieser Eintrag fehlt.  
Je nach Anwendung des Rechners können hier aber auch andere Programme eingetragen werden, z. B. für die Buchhaltung gleich das Buchhaltungsprogramm.  
Es gibt auch die Auswahl zwischen den verschiedenen Shells (C-Shell, Bourne-Shell, Korn-Shell) oder der restricted Shell, die einen eingeschränkten Befehlsumfang besitzt.

## 2.1.4 Gültigkeitsdauer des Passworts

Die Gültigkeitsdauer des Passworts wird im Passwortfeld der /etc/profile durch Anfügen von Komma und zwei Zeichen vom Superuser eingeschränkt. Bei Verwendung der Shadow-Datei stehen diese Angaben dort:

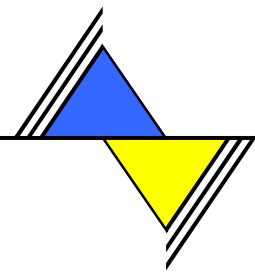
- Das erste Zeichen legt die minimale Gültigkeitsdauer fest, d.h. in dieser Zeit kann das Passwort nicht geändert werden.
- Das zweite Zeichen legt die maximale Gültigkeitsdauer fest,
- d. h. nach Ablauf dieser Zeit muss das Passwort geändert werden.

Die Zeitdauer wird in Wochen gezählt; der Punkt "." bedeutet 0 Wochen, der Schrägstrich "/" 1 Woche, dann folgen Ziffern und Buchstaben:

0..9:	2 bis 11 Wochen
A..Z:	12 bis 37 Wochen
a..z:	38 bis 63 Wochen

Sonderfälle:

- .. (zwei Punkte)  
Der Benutzer muss beim nächsten Login sein Passwort ändern - danach lebt es ewig (für neu eingerichtete Benutzer).
- ./(Punkt, Schrägstrich)  
Der Benutzer kann sein Passwort nicht mehr ändern.  
Sinnvoll bei einer "Gast"-Kennung, denn der Gast soll das Passwort nicht ändern können.



## 2.1.5 Die Datei /etc/group

Diese Datei kann von den Benutzern nur gelesen werden, das Schreiben ist dem Superuser vorbehalten. Sie legt die Gruppenzugehörigkeit der Benutzer fest. Die Datei besteht aus Textzeilen mit 4 Feldern, die durch Doppelpunkte getrennt sind.

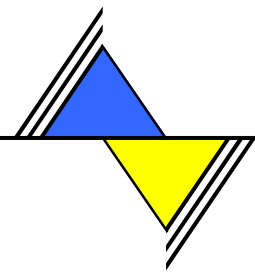
### **/etc/group**

Aufbau der Datei:

**Gruppenname:Paßwort:GID:Benutzernamen**

```
root:x:0:
bin:x:1:daemon
daemon:x:2:
sys:x:3:
tty:x:5:
disk:x:6:
lp:x:7:
www:x:8:
kmem:x:9:
wheel:x:10:
mail:x:12:
news:x:13:
uucp:x:14:karl,wohlab,ada
shadow:x:15:
dialout:x:16:karl,wohlab
audio:x:17:karl,wohlab
floppy:x:19:
cdrom:x:20:
console:x:21:
utmp:x:22:
public:x:32:
video:x:33:karl,wohlab
games:x:40:
xok:x:41:
trusted:x:42:
modem:x:43:
named:x:44:
ftp:x:49:
man:x:62:
users:x:100:
nobody:x:65533:nobody
nogroup:x:65534:nobody
at:x:25:
ntadmin:x:71:
ldap:x:70:
sshd:x:65:
postfix:x:51:
maildrop:x:59:
mailman:x:67:
```

- Gruppenname:  
Maximal 8 Buchstaben/Ziffern, die den Namen der Gruppe festlegen.
- Passwort:  
Zwar ursprünglich konzipiert, bleibt dieses Feld leer, denn es gibt keine Möglichkeit, das Passwort verschlüsselt einzutragen.
- GID = Group ID:  
In diesem Feld wird festgehalten, zu welcher Gruppe der Benutzer gehört (Wertebereich 0 bis 50000). Die Nummern 0 bis 99 sind für interne Zwecke reserviert, GID 100 wird als Sammelgruppe verwendet.
- Benutzernamen:  
Hier werden alle Login-Namen der zur Gruppe gehörenden Benutzer, getrennt durch Komma, eingetragen. Einträge für die Standardgruppe, die in der Datei /etc/passwd festgelegt ist, sind nicht nötig.



## 2.1.6 Die Datei /etc/shadow

Diese Datei kann von den Benutzern nicht gelesen werden, das Lesen und Schreiben ist dem Superuser vorbehalten. Sie enthält das Benutzerpasswort und Angaben über die Gültigkeitsdauer von Passwort und Benutzeraccount. Die Datei besteht aus Textzeilen mit 9 Feldern, die durch Doppelpunkte getrennt sind:

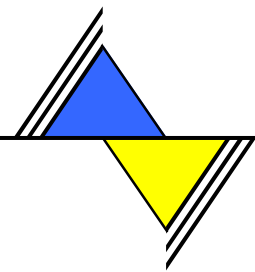
### **/etc/shadow**

Aufbau der Datei:

**Name:Paßwort:letzte Änderung:Min:Max:Vorwarnzeit  
Inaktiv:Verfall:Kennzeichen**

```
root:wJGEGCpWlcaak:12230:0:10000:::::
bin:*:8902:0:10000:::::
daemon:*:8902:0:10000:::::
lp:*:8902:0:10000:::::
mail:*:8902:0:10000:::::
news:*:8902:0:10000:::::
uucp:*:8902:0:10000:::::
games:*:8902:0:10000:::::
man:*:8902:0:10000:::::
wwwrun:*:8902:0:10000:::::
ftp:*:8902:0:10000:::::
named:*:8902:0:10000:::::
nobody:*:8902:0:10000:::::
at:!:12230:0:99999:7:::
irc:!:12230:0:99999:7:::
ldap:!:12230:0:99999:7:::
sshd:!:12230:0:99999:7:::
postfix:!:12230:0:99999:7:::
pop:!:12230:0:99999:7:::
squid:!:12230:0:99999:7:::
ntp:!:12230:0:99999:7:::
mailman:!:12230:0:99999:7:::
karl:BF1NhU8wU2qbg:12230:0:99999:7:::
wohrlab:3trfZYk4.SXNk:12230:0:99999:7:::
```

- Name:  
Derselbe Benutzername, wie er in /etc/passwd steht
- Passwort:  
Normalerweise 13-stelliges, verschlüsseltes Passwort, in dem die ersten beiden Stellen die Verschlüsselungsmethode (eine aus 4096) kennzeichnen. Anstelle des Passworts kann hier auch ein Sperrvermerk stehen.
- letzte Änderung:  
Der Tag der letzten Änderung des Passworts. Gezählt wird in Tagen ab dem 1.1.1970 (offizieller Entstehungstag von UNIX).
- Min:  
Minimale Gültigkeitsdauer des Passworts in Tagen (vorher ist kein Ändern möglich).



- **Max:**  
Maximale Gültigkeitsdauer des Passworts in Tagen.  
Der Benutzer muss vor Ablauf dieser Frist sein Passwort ändern.
- **Vorwarnzeit:**  
Anzahl der Tage, wie lange der Benutzer vor Verfall seines Passworts auf die notwendige Änderung hingewiesen wird.
- **Inaktiv:**  
Anzahl der Tage, die es dem Benutzer gestattet ist, seinen Account unbenutzt zu lassen.
- **Verfall:**  
Absolutes Datum, an dem die Verwendung des Accounts gesperrt wird.
- **Kennzeichen:**  
Reserviert für künftige Verwendung.  
Derzeit auf 0 gesetzt. Damit nun nicht alle diese Daten beim Anlegen eines Benutzers von Hand eingegeben werden müssen, wird - so vorhanden - die Datei `/etc/default/passwd` herangezogen, in der Standardwerte gespeichert sind.

## 2.1.7 Die Datei `/etc/default/passwd`

Im Verzeichnis `/etc/default` werden unter UNIX V.4 die Standardwerte für verschiedene Programme abgelegt. Das Verzeichnis enthält Dateien, deren Namen sich auf das zugehörige Programm oder die zugehörige Funktion des Betriebssystems beziehen, z. B. "passwd", "tar", "boot", "init", etc. Die Einträge in den Daten bestehen in der Regel aus Zeilen mit jeweils einer Zuweisung der Form

```
'Schlüsselwort'='Wert'.
```

Dazu ein Beispiel: `/etc/default/passwd`:

Standardwerte für das Passwort und seine Gültigkeitsdauer.

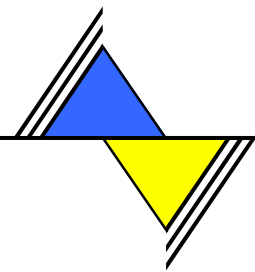
Einige wichtige Werte sind:

MAXWEEKS='maximale Zeit, bis das PW geändert werden muss'

MINWEEKS='minimale Zeit, in der das PW nicht geändert wird'

WARNWEEKS='Vorwarnzeit'

PASSLENGTH='minimale Passwortlänge'



## 2.1.8 Anlegen eines Benutzers

Erstaunlicherweise gab es ursprünglich kein Programm zum Anlegen von Benutzern - bei den meisten Systemen hat sich jedoch der Systemverwalter (= Superuser) ein Shell-script dafür angelegt. Bei vielen Systemen wird heute ein Standard-Tool mitgeliefert ('useradd' oder 'adduser').

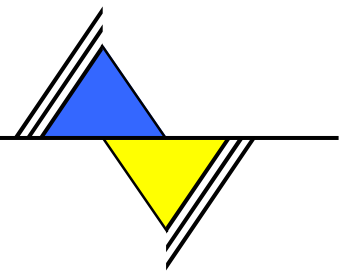
Um einen Benutzer neu einzutragen sind folgende Schritte notwendig:

- ⊗ Benutzer in der Datei /etc/passwd eintragen.  
Dabei ist darauf zu achten, dass der Login-Name und die UID eindeutig, d. h. nur einmal vorhanden, sind - sonst könnte es zu Schwierigkeiten bei der Zuordnung des Homedirectory und anderer Dateien.
- ⊗ In der Datei /etc/group wird der neue Login-Name im entsprechenden Gruppeneintrag ergänzt.
- ⊗ Nun wird das Home-Directory des neuen Benutzers eingerichtet und - falls erwünscht - eine Standardversion der Datei .profile (bei C-Shell .login) dorthin kopiert.  
Als Kopiervorlage für das Home-Directory dient (falls vorhanden) das Verzeichnis /etc/skel
- ⊗ Home-Directory und .profile sind immer noch "Eigentum" des Superusers, mit dem chown-Befehl (Change Owner) werden beide an den neuen Benutzer übergeben.
- ⊗ An sich reicht das aus. Jetzt können noch von der Arbeitsumgebung abhängende Maßnahmen vorgenommen werden.

In der Regel wird das Anlegen und Löschen eines Users durch passende Systemprogramme oder - Skripten erledigt.

## 2.1.9 Sperren eines Benutzers

Soll ein Benutzer zeitweise (z. B. bei längerem Urlaub) oder auf Dauer gesperrt werden, seine Dateien aber noch erhalten bleiben, trägt der Superuser im Passwortfeld eine Zeichenfolge ein, die als Ergebnis der Verschlüsselung nicht vorkommen kann, z. B. '\*LCK\*'. Überprüfung von /etc/passwd und /etc/group.



## 2.1.10 Löschen eines Benutzers

Die hierfür notwendigen Schritte sind wesentlich gefährlicher, da hier u. U. wichtige Informationen gelöscht werden.

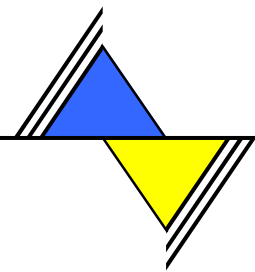
In der Regel wird der Benutzer zunächst deaktiviert (er kann sich also nicht mehr einloggen) und erst nach einer gewissen Zeit vollständig gelöscht. Dazu sind dann folgende Schritte nötig:

- ⊗ Löschen des Home-Directories mit allen darunter liegenden Dateien und Verzeichnissen (rm-Kommando)
- ⊗ Löschen des mail-Directory und aller Spool-Dateien
- ⊗ Löschen der zum Benutzer gehörenden Zeile aus /etc/passwd
- ⊗ Löschen des Login-Namens aus /etc/group Anmerkung:

Um alle Dateien (nicht Directories) eines Benutzers zu löschen, kann das Kommando `find` verwendet werden (für UID wird die User-ID des Benutzers eingesetzt):

```
find / -user UID -type f -exec rm -f {} ";"
```

Probleme können durch Dateien des zu löschenden Benutzers verursacht werden, auf die andere Benutzer ein Link gesetzt haben. Man kann diese Dateien z. B. den betroffenen Benutzern zuordnen.



## 2.2 Backup und Recovery

Quelle: LINUX-Anwender-Handbuch 7.0 / Datensicherung

Die regelmäßige Datensicherung (Backup) ist eine der wichtigsten Aufgaben des Systemverwalters. Sei es durch versehentliches Löschen, sei es durch Hardwarefehler oder durch einen Fehler des Betriebssystems, früher oder später macht jeder Computerbenutzer die Erfahrung eines Datenverlustes.

Ein Backup ist in der Regel die einzige Möglichkeit, wenigstens einen Teil der Daten zu restaurieren.

Auch wenn ein regelmäßiges Backup einigen Arbeitsaufwand bedeutet, steht diese Mühe meist in keinem Verhältnis zu einer manuellen Rekonstruktion der Daten. Auf professionellen Systemen, bei denen möglicherweise sogar mehrere Benutzer auf einem Datenbestand arbeiten, ist wenigstens eine wöchentliche, besser eine tägliche Datensicherung erforderlich

### 2.2.1 Sicherungsmedien

Datensicherung kann auf sehr verschiedene Weisen und auf verschiedenen Medien erfolgen. Es besteht zum Beispiel die Möglichkeit, Kopien von wichtigen Dateien in einem anderen Teil des Dateisystems (auf einer anderen Partition oder Festplatte) anzulegen. Natürlich können auch Disketten zur Datensicherung verwendet werden. Im Allgemeinen werden aber Magnetbänder benutzt, um Backups vom System zu machen.

#### 2.2.1.1 Magnetbänder und Streamer

Magnetbänder haben gegenüber allen anderen Medien mehrere Vorteile:

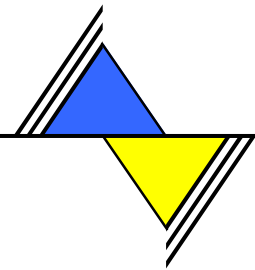
- + Sie bieten eine hohe Kapazität zu einem niedrigen Preis.
- + Sie lassen sich leicht vom Rechner entfernen und an einem sicheren Ort aufbewahren.
- + Sie eignen sich gut zur unbeaufsichtigten Datensicherung.

Der Nachteil von Magnetbändern besteht im sehr langsamen Zugriff auf einzelne Dateien. Die Topologie des Magnetbandes erzwingt eine rein sequentielle Form der Datenspeicherung. Um an Daten in der Mitte des Bandes heranzukommen, muss das gesamte Band bis zur gesuchten Stelle gelesen werden.

Die Vorteile überwiegen diesen Nachteil bei der Datensicherung aber so stark, dass auf allen Systemen, wo ernsthafte Datensicherung betrieben wird, Magnetbänder eingesetzt werden.

Magnetbandlaufwerke werden als zeichenorientierte Geräte betrieben. Trotzdem sind die Daten auf dem Medium in Blöcken organisiert. Das schafft einige Verwirrung und führt in manchen Grenzfällen zu Komplikationen. Von den echten Blockgeräten unterscheiden sich die Bandlaufwerke, weil nicht ein bestimmter einzelner Block gelesen oder geschrieben werden kann.

Im Unterschied zu den Magnetplattenspeichern werden die Magnetbänder in der Regel nicht formatiert. Deshalb gibt es auf Magnetbändern keine physikalisch nummerierten Datenblöcke, es gibt keine Dateisysteme und kein Inhaltsverzeichnis und das Einbinden eines Magnetbandes in das Dateisystem mit dem mount-Kommando ist unmöglich. Der Zugriff auf die Daten findet immer sequentiell statt. Um die Daten eines bestimmten Blockes zu erhalten, müssen alle vorhergehenden Blöcke gelesen werden.



Bei den alten 9-Spur Industrielaufwerken konnte (oder musste) das Band im Start/Stop Betrieb zwischen zwei Datenblöcken immer angehalten werden. Bei den Magnetbandgeräten für PC geht das in der Regel nicht, weil der für das Anhalten nötige Zwischenraum zugunsten der Datenkapazität praktisch weggefallen ist. Die Daten werden auf den Magnetbändern als kontinuierlicher Datenstrom gespeichert. Wegen dieser Art des Datentransfers werden die Bandlaufwerke auch als **Streamer** bezeichnet.

Für alle Magnetbandgeräte existieren zwei verschiedene Betriebsarten:

☒ "Rewind on Close".

In dieser Betriebsart wird das Band nach dem Schließen der Gerätedatei, also nach Beendigung der Lese- oder Schreiboperation, automatisch zurückgespult.

☒ "No Rewind on Close".

In dieser Betriebsart wird das Band nach Beendigung einer Operation angehalten und bleibt so stehen, bis die nächste Operation durchgeführt wird.

Die Auswahl einer Betriebsart findet unter UNIX/LINUX durch die Gerätedatei statt, über die das Laufwerk angesprochen wird.

Viele Bandgeräte können mit Magnetbändern unterschiedlicher Kapazität arbeiten. Dazu müssen verschiedene Aufzeichnungsparameter, namentlich die Anzahl der Spuren, die Schreibgeschwindigkeit und die Schreibdichte, eingestellt werden.

Obwohl die meisten Laufwerke diese Parameter automatisch einstellen, erlauben einige Gerätetreiber die Vorauswahl eines Bandtyps durch die Benutzung einer bestimmten Gerätedatei.

Alle von Linux unterstützten Bandlaufwerke arbeiten mit Magnetbandkassetten (Cartridges). Die Bänder der am weitesten verbreiteten Streamer sind 1/4 Zoll breit, deshalb werden sie auch als **Quarter-Inch-Cartridges** bezeichnet. Sowohl das Aufzeichnungsformat für solche Bänder, als auch die Ansteuerung der mit diesen Bändern arbeitenden Laufwerke ist in den verschiedenen Standards des Quarter Inch Comitee (**QIC**) definiert.

Die für Linux wichtigsten QIC-Normen (ohne Anspruch auf Vollständigkeit) sind:

☒ **QIC-02**

In der QIC-02 Spezifikation ist die Ansteuerung von DC6XXX Streamern mit eigenen Controllerkarten beschrieben.

☒ **QIC-24**

In QIC-24 ist das Aufzeichnungsformat für DC600A Bänder mit 60MB Kapazität (9Spuren, 10.000FTPI) definiert.

☒ **QIC-40/-80/-3010/-3020**

In den Standards QIC-40, QIC-80 QIC-3010 und QIC-3020 sind die Aufzeichnungsformate für DC2XXX und Travan TR2/TR3 Magnetbänder festgelegt, wie sie in Floppystreamern verwendet werden.

☒ **QIC-117**

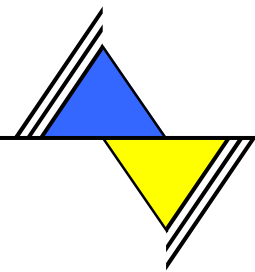
In QIC-117 ist die Ansteuerung der Floppystreamer beschrieben.

☒ **QIC-120/-150/-525...**

Die Spezifikationen QIC-120, QIC-150, QIC-525 usw. beschreiben das Aufzeichnungsformat der DC6XXX Magnetbänder mit den entsprechenden Kapazitäten. Das Aufzeichnungsverfahren ist bei diesen Formaten identisch mit QIC-24, lediglich die Aufzeichnungsdichte und die Bandqualität sind unterschiedlich.

Neben den QIC-konformen Bandgeräten kann Linux auch mit allen SCSI-Streamern umgehen.

Alle Streamer, die am Druckerport betrieben werden, können unter Linux (noch) nicht betrieben werden.



### 2.2.1.2 Mehrere Dateien auf einem Magnetband

Wenn ein kontinuierlicher Datenstrom, sprich eine Datei, abgeschlossen ist, wird auf dem Band eine Markierung für das Dateiende (**EOF, End Of File**) geschrieben.

Wenn Sie ein Bandarchiv mit tar oder cpio erzeugen, wird diese Markierung immer ans Ende des gesamten Archivs gesetzt. Die einzelnen Dateien in diesem Archiv werden nur durch das Archivierungsprogramm unterschieden.

Wenn nach dem Dateiende noch freier Speicherplatz auf dem Band vorhanden ist, kann eine weitere Datei oder ein Archiv an die bereits geschriebenen Daten angehängt werden. Mit dem mt-Kommando müssen Sie dazu das Band hinter die Endmarkierung positionieren. Damit das Band nach dem mt-Kommando nicht automatisch zurückgespult wird, muss das Bandgerät im "No Rewind on Close" Modus betrieben werden.

Wenn Sie beispielsweise bereits eine Datei (ein Archiv) auf einem SCSI-Streamerband gespeichert haben und eine zweite Datei auf das gleiche Band schreiben wollen, positionieren Sie das Band mit dem folgenden Kommando hinter die erste Datei:

```
$ mt -f /dev/nst0 fsf 1  
$ _
```

- Mit der -f Option wird der erste SCSI-Streamer im "No Rewind On Close" Modus ausgewählt.
- Die Operation fsf 1 (forward skip file) spult das Band bis zur ersten Dateiendemarke vor.
- Wenn das Kommando abgeschlossen ist, hält der Bandmotor an und der Schreib/Lesekopf steht hinter der ersten Datei.

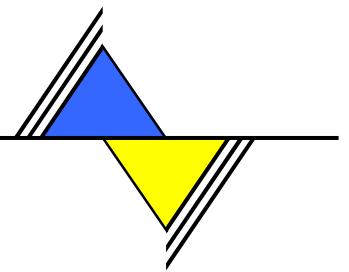
Wenn Sie jetzt mit einem der Archivierungsprogramme ein Schreib- oder Lesekommando ausführen, läuft der Bandmotor wieder an und führt die Aktion an dieser Stelle aus.

Wenn mehrere Dateien (Archive) auf einem Band gespeichert sind, können Sie keine Datei am Anfang oder in der Mitte löschen/überschreiben. Eine Schreiboperation in der Mitte des Bandes würde automatisch alle hinter der Dateiende-Markierung liegenden Daten unzugänglich machen.

### 2.2.1.3 Dateien (Archive) auf mehreren Magnetbändern

Obwohl die Magnetbänder große Datenmengen speichern können, kommt es manchmal vor, dass ein Linux-System oder eine Linux-Partition, nicht einmal komprimiert, auf ein einzelnes Magnetband passt. In diesem Fall kann, die geeignete Software vorausgesetzt, das Backup auch auf mehrere Bänder verteilt werden.

Durch spezielle Markierungen auf den Magnetbändern, die so genannten Early Warning Marks, erkennt das Bandgerät das Bandende im Voraus. Bei den Bandgeräten mit fester Blockgröße (QIC-02 und Floppystreamer) reicht die Kapazität immer aus, um den aktuellen Datenblock und einen abschließenden Block vom Archivierungsprogramm unterzubringen. Bei SCSI-Streamern, die variable Blockgrößen verwenden können und die einen Teil der Daten verzögert schreiben, kann es zu Problemen kommen, wenn das Band die verzögerten Daten nicht mehr fassen kann.



#### 2.2.1.4 Floppystreamer

Wegen des günstigen Anschaffungspreises sind die Floppystreamer für den Einsatz in kleinen bis mittleren Linux-Systemen besonders interessant. Diese Bandlaufwerke werden wie ein drittes Diskettenlaufwerk an den normalen Floppycontroller angeschlossen.

Die im Handel befindlichen kleinen Floppystreamer verwenden in der Regel das QIC-80 Format. In diesen Streamern werden Mini-Cartridges vom Format DC2080 oder DC2120 verwendet, die eine Kapazität von 80 bzw. 120 Megabyte unkomprimierter Daten haben, durch extra lange Bänder kann die Kapazität auf 170 MB gesteigert werden.

Die moderneren Floppystreamer mit höheren Kapazitäten arbeiten mit Traven TR2/TR3-Bändern in den Formaten QIC-3010 und QIC-3020. Diese Bänder können 400MB und mehr speichern.

Die Datentransferrate aller Floppystreamer wird vom Floppycontroller bestimmt. Im schlechtesten Fall lassen sich 126 Megabyte pro Stunde schreiben. Mit dem Controllerbaustein 82078-1 kann die Leistung des gleichen Laufwerks vervierfacht werden.

Im Unterschied zu den SCSI-Streamern, die durch die sehr genaue Spezifikation der Geräteschnittstelle alle in der gleichen Weise angesteuert werden können, lassen sich die QIC-117 Streamer nicht alle exakt gleich betreiben.

Um Daten auf einem Floppytape speichern zu können, muss das Medium zuerst formatiert werden. Ähnlich wie beim Formatieren von Disketten werden leere Blöcke mit einer Größe von 512 Bytes auf das leere Band geschrieben. Diese Blöcke werden dann beim Beschreiben des Bandes mit Daten gefüllt.

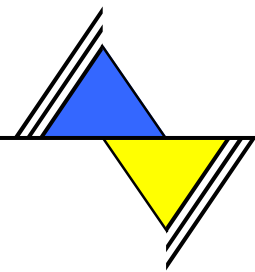
#### 2.2.1.5 QIC-Streamer

Die Bandlaufwerke mit einem eigenen Controller nach dem QIC-02 Standard verwenden 1/4 Zoll Cartridges normaler Größe (15x10 cm). Je nach Streamertyp können die QIC-02 Geräte Bänder in den Formaten QIC-24, QIC-120, QIC-150 usw. schreiben und/oder lesen.

Der QIC-02 Treiber wurde für den Wangtek-5150 Streamer geschrieben. Es gibt keine offizielle Liste aller unterstützten QIC-02 Streamer. Bei Everex und Archive Laufwerken sind die Chancen sehr gut, dass der Streamer sofort erkannt wird.

Je nach Typ des Bandgerätes können Bänder mit verschiedener Aufzeichnungsdichte verwendet werden. Die Einstellung einer bestimmten Dichte geschieht durch die Wahl der Gerätedatei für das Bandgerät.

Die QIC-02 Streamer arbeiten mit einer festen Blockgröße von 512 Bytes.



### 2.2.1.6 SCSI-Streamer

Wie alle SCSI Geräte haben auch die SCSI Bandlaufwerke eine sehr umfangreiche Steuerlogik "an Bord". Das Betriebssystem kommuniziert mit dem Laufwerk über den Hostadapter auf einem sehr hohen Abstraktionsniveau. Das Magnetbandtyp oder das physikalische Aufzeichnungsformat spielen hier keine Rolle mehr. Aus diesem Grund können praktisch alle SCSI Bandgeräte unter Linux eingesetzt werden. Lediglich die Größe der physikalischen Blöcke darf die maximale Puffergröße von 32kB nicht überschreiten.

Weite Verbreitung haben die 1/4 Zoll Streamer, die mit den gleichen Cartridges arbeiten wie die QIC-02 Bandgeräte und die DAT-Streamer, die auf nur 4mm breiten Bändern enorme Datenmengen speichern können.

Weil der SCSI-Standard sehr generell gefasst ist, werden die physikalischen Aufzeichnungsparameter nicht fest vorgegeben.

Die Aufzeichnungsdichte, die Blockgröße und die Datenpufferung können durch Systemaufrufe verändert werden, vorausgesetzt die Kombination von Bandgerät und Band erlauben die gewünschten Werte.

Während die Aufzeichnungsdichte automatisch erkannt wird und in der Regel nicht verändert werden sollte, kann es bei manchen Bandgeräten nötig oder vorteilhaft sein, die Datenpufferung einzuschalten, um ein gleichmäßigeres Strömen der Daten zu erreichen.

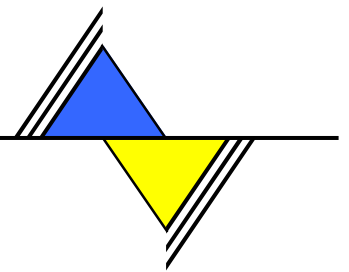
Wenn Sie Magnetbänder zwischen verschiedenen Betriebssystemen tauschen wollen, muss die Einstellung für die Größe der physikalischen Datenblöcke beim schreibenden und dem lesenden System übereinstimmen. Bei den SCSI-Geräten können verschiedene feste oder während der Aufzeichnung variierende Blockgrößen benutzt werden, wenn das Bandformat und das Gerät mitspielen.

Die Datenübertragungsrate der SCSI-Streamer hängt stark vom verwendeten Bandgerät, der Aufzeichnungsdichte (also dem Bandtyp) und auch vom Hostadapter ab. Sie ist aber in jedem Fall deutlich höher als bei den Floppystreamern.

### 2.2.1.7 Disketten

Anstelle von Magnetbändern können auch Disketten als Medium zur Datensicherung eingesetzt werden. Dabei werden die formatierten Disketten direkt, roh beschrieben. Ein Dateisystem ist ebenso unnötig wie das Mounten der Diskette. Stattdessen wird die rohe Diskette direkt über die Gerätedatei für das Diskettenlaufwerk angesprochen.

Im Unterschied zu den zeichenorientierten Bandlaufwerken arbeiten die Diskettenlaufwerke blockorientiert. Dadurch sind Rückschritte und Positionierung vor das Dateieinde kein Problem.



## 2.2.2 Methoden der Datensicherung

Das Ziel jeder Datensicherung ist es, alle System- und Benutzerdaten einer Linux-Installation vor einem ungewollten Verlust zu schützen.

Weil sich bestimmte Daten, vor allem die Daten der Systembenutzer, kontinuierlich ändern, ist eine hohe Frequenz der Datensicherung erforderlich. Für Systeme im professionellen Einsatz ist ein tägliches Backup angebracht.

### 2.2.2.1 Gesamtsicherung (Full-Backup) #####

In regelmäßigen, größeren Abständen wird eine vollständige Sicherung aller Daten auf einem oder mehreren Bändern großer Kapazität gemacht. Relativ zu dieser Vollsicherung werden dann inkrementelle Sicherungen durchgeführt.

### 2.2.2.2 Inkrementelles Backup

Bei einer sehr hohen Backupfrequenz ist es weder erforderlich, noch wünschenswert, jedes Mal sämtliche Daten erneut zu sichern. Im Prinzip reicht es aus, immer nur die seit dem letzten Backup veränderten Dateien zu sichern, um den aktuellen Zustand restaurieren zu können. Diese Methode des inkrementellen (aufsteigenden) Backups hat mehrere Vorteile:

- ☒ Jedes einzelne Backup lässt sich sehr schnell durchführen, weil die Menge der veränderten Daten verglichen mit dem gesamten System sehr klein ist.
- ☒ Die Methode ist kostengünstig, weil keine unveränderten Daten doppelt gespeichert werden.
- ☒ Die Wahl des Mediums ist maximal flexibel. Bei vielen Systemen kann für die inkrementellen Backupsschritte sogar eine Diskette verwendet werden.

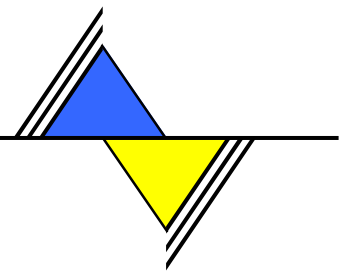
Bei inkrementellen Backups können noch verschiedene Level unterschieden werden, indem mehrere Sicherungen als Bezugspunkt gewählt werden.

- Einem vollständigen Backup wird der Level 0 zugeordnet.
- In den Backups mit Levels größer als 0 werden nur die Daten gespeichert, die seit der letzten Sicherung mit einem Level kleiner oder gleich dem aktuellen Level verändert worden sind.
- In einem Backup Level 1 werden also alle Daten gespeichert, die seit dem letzten Vollbackup verändert worden sind und die noch nicht in einem anderen Backup vom Level 1 enthalten sind.
- Im Level 2 werden dann nur die Daten gespeichert, die seit dem letzten Backup mit Level 0, 1 oder 2 verändert worden sind und so weiter.

Als Abwandlung des oben beschriebenen Modells können die Levels auch so definiert werden, dass immer nur die Veränderungen seit einem Backup mit streng niedrigerem Level gespeichert werden. Bei dieser Methode werden bei wiederholten Sicherungen eines Levels viele Daten mehrfach gesichert.

Sie sollten in jedem Fall so viele Bänder zur Datensicherung verwenden, dass Sie bei jedem Sicherungsschritt das letzte Backup vollständig behalten können. Nur so sind Sie in der Lage, einen Festplattencrash während des Sicherungslaufes zu restaurieren.

Die optimale Methode der Datensicherung hängt von der Beschaffenheit Ihres Systems ab. In der Regel besteht sie aus zwei oder dreistufigen Kombinationen von vollständigen und inkrementellen Backups.



Beispiel 1:

- An jedem ersten Sonntag im Monat wird ein Voll-Backup durchgeführt. Gehen wir davon aus, dass hierfür zwei Bänder benötigt werden.
- Auf einem dritten Band werden jeden weiteren Sonntag alle Änderungen der vergangenen Woche als inkrementelle Sicherungen (Level 1) gespeichert.
- Um die Bandkapazität besser auszunutzen, können alle Level 1 Archive eines Monats hintereinander auf ein Band geschrieben werden.

Beispiel 2:

- Ein anderes Modell ergibt sich, wenn die inkrementellen Sicherungen immer relativ zum letzten Vollbackup angelegt werden.
- Dann werden die Level 1 Sicherungen von Mal zu Mal größer.
- Man macht das nächste Vollbackup, wenn die Menge der veränderten Daten ein vorher bestimmtes Maß überschreitet, spätestens wenn sie nicht mehr auf einem einzigen Band Platz findet.

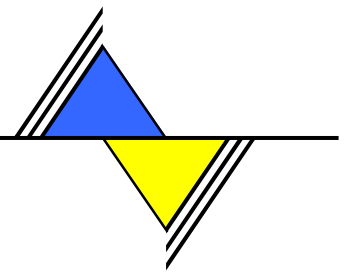
Der Vorteil des zweiten Modells besteht darin, dass das Magnetband nicht hinter die bereits geschriebenen Daten der letzten Sicherungen positioniert werden muss, weil die ja noch einmal geschrieben werden.

Der Nachteil ist, dass die Sicherungen von Mal zu Mal länger dauern.

Wenn Sie besonders wichtige Daten, beispielsweise die Heimatverzeichnisse, häufiger sichern wollen, bieten sich auf einem kleinen System Disketten als Sicherungsmedium an.

Weil Sie den größten Teil des Systems auf einem Installationsmedium vorliegen haben, ist im Extremfall gar keine Sicherung des kompletten Dateisystems notwendig. Sie können der "rohen" Linux-Installation den Level 0 zuordnen und alle Veränderungen als inkrementelle Sicherungen relativ zu Ihrer Distribution speichern.

Selbstverständlich stehen Ihnen Kommandos zur Verfügung, die Sie bei der Erstellung inkrementeller Backups unterstützen. Vor allem das `find`-Kommando ist hervorragend zur Erstellung von Dateilisten geeignet. Das `tar`-Programm bietet Ihnen Optionen, mit denen Sie sehr einfach inkrementelle Backups machen können. Es ist leicht möglich, die Datensicherung automatisch durch ein Script ausführen zu lassen.

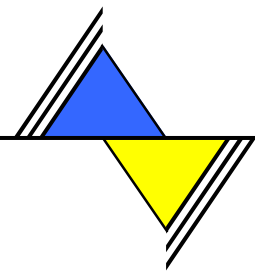


### 2.2.3 Backup-Software

Auf Magnetbändern können keine Dateisysteme eingerichtet werden, sie enthalten keine Verzeichnisse und können nicht in den Dateisystembaum eingebunden werden. Es ist zwar möglich, mehrere Dateien auf ein Band zu schreiben, die Dateinamen, Eigentümer, Zugriffsrechte und alle Zeitmarken gehen aber verloren, weil diese Daten nicht Teil der Datei selbst sind, sondern in der Inode der Datei gespeichert werden.

Deshalb werden zur Sicherung der Daten auf Band spezielle Programme verwendet, die eine Vielzahl von Dateien zu einem einzigen Datenstrom, also einer einzigen Datei, zusammenfassen und mit allen dazugehörigen Systemdaten verwalten können:

- ☒ Die am häufigsten zu diesem Zweck herangezogenen Programme sind **tar**, **afio** und **cpio**.
- ☒ Die Programme **dump** und **restore**, die bei BSD-Unix zur Datensicherung verwendet werden, sind unter Linux noch nicht erhältlich.
- ☒ In besonderen Fällen kommt noch das **dd**-Kommando als Sicherungssoftware in Frage.



## 2.2.4 Datensicherung mit tar

Eine der Grundaufgaben von Backupsoftware, nämlich die Zusammenfassung mehrerer Dateien zu einer einzigen, sowie alle zum Management dieses Datenpaketes gehörenden Aufgaben werden nicht nur zur Datensicherung auf Magnetbändern benötigt. Die gleiche Funktionalität wird auch zur Verwaltung der C-Funktionsbibliotheken gebraucht. Diese Bibliotheken, auch als Archive bezeichnet, werden von dem zum Entwicklungssystem gehörenden `ar`-Kommando erzeugt und verwaltet.

Speziell auf die Verwendung mit Bandarchiven weiterentwickelt gibt es auf praktisch allen Unix-Systemen den Tape-Archiver `tar`.

Dank des sehr flexiblen Dateikonzeptes von Linux kann `tar` auch Archive im Dateisystem als neue Dateien erzeugen und sie verwalten. In dieser Form wird fast alle Freie Software im Internet verteilt. Die Softwarepakete aller Linux-Distributionen sind mit `tar` archiviert.

Mit den Magnetbandgeräten, die unter Linux Verwendung finden, können nicht alle Funktionen von `tar` uneingeschränkt genutzt werden.

- ☒ Insbesondere kann ein Bandarchiv nicht erweitert werden. Das bedeutet, dass die Optionen `-A`, `-r` und `-u` nicht funktionieren.
- ☒ Außerdem ist das Löschen oder Ersetzen einzelner Dateien in einem Bandarchiv mit `-delete` nicht möglich.
- ☒ Trotzdem ist das GNU-`tar` ausgezeichnet zur Sicherung/Archivierung großer Datenmengen auf Magnetbänder geeignet.

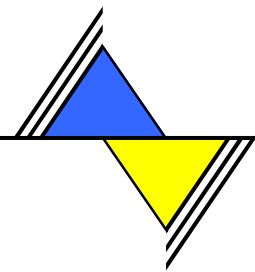
Beispiel:

Sie erzeugen ein neues `tar`-Archiv mit der Option `-c`.

Um beispielsweise ein vollständiges Backup auf das SCSI-Bandlaufwerk zu schreiben, geben Sie die folgende Kommandozeile ein:

```
# tar -c -M -b 126 -V "Level 0" -f /dev/rmt0 -g /var/adm/Backup/Dir /
tar: Removing leading / from absolute path names in the archive.
tar: Removing leading / from absolute links
Prepare volume #2 for /dev/rmt0 and hit return:
# _
```

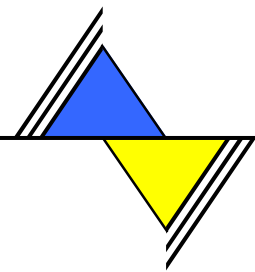
- Wie Sie sehen, macht `tar` automatisch aus allen absoluten Pfadnamen relative. Das geschieht, damit Sie das Archiv relativ zu jedem beliebigen Verzeichnis auspacken können. Wenn Sie die absoluten Namen behalten wollen, müssen Sie den Schalter `-P` setzen.
- Die `-M` Option zeigt `tar` an, dass Sie ein Archiv auf mehreren Bändern anlegen wollen, weil Ihre Daten nicht auf ein einziges Band passen. Wenn Ihre Daten nicht mehr als zwei Bänder füllen, können Sie durch die `-z` Option die Daten durch den `gzip`-Kompressor filtern. Wenn nicht ein großer Teil der Daten bereits komprimiert im Dateisystem vorliegt, erreichen Sie eine Verdichtung der Daten mindestens um den Faktor 2.
- Es ist allerdings nicht möglich, komprimierte `tar` Archive auf mehrere Bänder zu verteilen.
- In dem Beispiel oben wurde die Blockgröße mit der Option `-b 126` gegenüber dem Standardwert von 20 deutlich erhöht. Das Optionsargument 126 bedeutet, dass `tar` die Daten in Portionen zu 126x512 Bytes, also 64 Kilobyte, auf das Band schreibt. Die Vergrößerung des Wertes sorgt für einen flüssigeren Datenstrom und damit zu einer Beschleunigung des Sicherungslaufes. Sie müssen bei jeder Veränderung der Blockgröße aber darauf achten, dass Sie beim Lesen den gleichen Wert einstellen.



- Mit der Option `-V` wird dem Archiv ein Name gegeben. Damit können Sie die Sicherungsbänder identifizieren, auch wenn Ihnen die externe Beschriftung verloren gehen sollte.
- Nach der `-f` Option ist in dem Beispiel die Gerätedatei für den ersten SCSI-Streamer angegeben worden. Das Gerät wird im "Rewind On Close" Modus betrieben, das Band also nach Beendigung des Kommandos automatisch zurückgespult. Wenn Sie kein Gerät angeben, versucht tar auf das bei der Übersetzung des Programms Voreingestellte Gerät zuzugreifen. Wenn Sie ein anderes Gerät für mehrere Aufrufe von tar oder mt voreinstellen möchten, können Sie das über die Umgebungsvariable `TAPE` machen.
- Mit der `-g` Option wird eine Datei bestimmt, die gleichzeitig Zeitmarke des Backups und Inhaltsverzeichnis des Archives ist. Hier werden nur die Verzeichnisse, nicht die einzelnen Dateien eingetragen. Mit Hilfe dieser Datei ist es besonders einfach, inkrementelle Backups zu machen. Wenn Sie beispielsweise an einem anderen Tag alle Veränderungen relativ zu dem vollständigen Backup sichern wollen, können Sie folgendes Kommando eingeben:

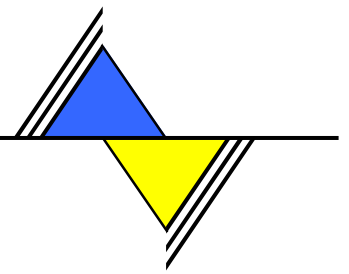
```
# export TAPE=/dev/fd0
# tar -c -z -V "Backup Level 1 vom 28.08.94" -g /var/adm/Backup/Dir /
tar: Removing leading / from absolute path names in the archive.
# _
```
- Jetzt werden automatisch alle Dateien gespeichert, deren Änderungszeit neuer als das letzte Backup ist. Hierbei werden Dateien, die mit `mv` umbenannt oder verschoben worden sind ebenso wenig gesichert wie zusätzlich installierte, ältere Pakete, deren Erzeugungszeit beim Auspacken normalerweise nicht verändert wird.
- Je nachdem wie das Linux-System genutzt wird, reicht für ein inkrementelles Backup auch eine Diskette als Speichermedium aus. In diesem Beispiel werden die Daten komprimiert, so dass mehr als drei Megabyte Textdaten auf einer rohen Diskette gespeichert werden können. Weil ein fehlerhafter Block auf der Diskette das gesamte Archiv ab diesem Block unbrauchbar macht, sollten Sie besonders zur komprimierten Datensicherung nur geprüfte Markendisketten verwenden.
- Wenn bei den einzelnen Sicherungsschritten mehr Daten anfallen, als auf einer Diskette Platz finden, werden Sie wahrscheinlich auch für die inkrementellen Backups Magnetbänder verwenden wollen. Um die Kapazität der Bänder auszunutzen, können Sie mehrere separate Archive hintereinander auf einem Band speichern. Zum Positionieren des Bandes müssen Sie das separate Kommando `mt` benutzen.

```
# export TAPE=/dev/nftape
# mt eom
# tar -c -V "Backup Level 1 vom 30.08.94" -g /var/adm/Backup/Dir /
tar: Removing leading / from absolute path names in the archive.
# mt rewind
```
- Mit dem ersten `mt`-Kommando wird das Band bis an das Ende der zuletzt geschriebenen Daten vorgespult. Dabei ist es gleichgültig, wie viele Dateien/Archive bereits auf das Band geschrieben worden sind.
- Weil das Bandgerät im "No Rewind On Close" Modus betrieben wird (der Modus wurde durch die Gerätedatei gewählt), schreibt das tar-Kommando seine Daten von dieser Stelle an. Um das Magnetband zu schonen, ist es sinnvoll, das Band vor dem Entfernen des Cartridges aus dem Laufwerk zurückzuspulen.



- Wenn Sie mehrere Archive auf einem Band gespeichert haben, kommen Sie an ein bestimmtes Archiv nur mit dem `mt`-Kommando heran. Magnetbänder haben kein Inhaltsverzeichnis, deshalb müssen Sie sich selbst merken, wie viele Archive auf dem Band gespeichert sind. Um beispielsweise das vierte Archiv auf dem Band zu erreichen, müssen Sie drei Archive überspringen:

```
# export TAPE=/dev/nftape
# mt fsf 3
# cd /
# tar -x
# mt rewind
# _
```
- Mit der `-x`-Option veranlassen Sie `tar`, alle Dateien aus dem Archiv zu extrahieren. Durch dieses Kommando restaurieren Sie den Zustand des Dateisystems zum Zeitpunkt der Sicherung. Um ein vollständig zerstörtes System zurückzusichern, müssen Sie alle Backups in der Reihenfolge ihres Entstehens auf die leere Festplatte einspielen.
- Das `tar`-Programm bietet noch eine Vielzahl weiterer Varianten der Datensicherung, die Sie beispielsweise in der Online-Dokumentation zu `tar` nachlesen können.



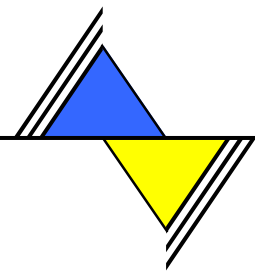
## 2.2.5 Datensicherung mit `afio` und `cpio`

Das `cpio`-Kommando und die modernere Variante `afio` bilden die Grundlage für ein sehr flexibles Backupsystem. Die Flexibilität wird erreicht, indem sich die beiden Kommandos auf einen Teilaspekt der Sicherungsarbeit konzentrieren: sie erzeugen und verwalten ein Archiv, einen Datenstrom aus bestimmten Dateien. Die Namen der zu archivierenden Dateien müssen den Kommandos im Standardeingabekanal, beispielsweise als Ausgabe von `find`, übergeben werden. Damit sind `cpio` und `afio` zur Verwendung in Shellscripten prädestiniert.

Während `tar` bei den inkrementellen Sicherungen einfache und feste Kriterien zur Unterscheidung alter und neuer Dateien anlegt, kann durch das Zusammenspiel mehrerer auf bestimmte Teilaufgaben spezialisierter Werkzeuge eine sehr genaue Liste der veränderten Dateien erzeugt werden.

`afio` bietet zusätzlich die Möglichkeit, während der Archivierung die einzelnen Dateien zu komprimieren, ohne das gesamte Archiv durch den Kompressor zu leiten. Das hat den Vorteil, dass bei kleineren Fehlern im Archiv nur einzelne Dateien betroffen sind. Außerdem können so komprimierte Archive auf mehrere Bänder verteilt werden.

Wenn Sie die Vorteile von `afio` oder `cpio` nutzen wollen, brauchen Sie sich das dazu notwendige Script nicht selbst zu schreiben. Besonders empfehlenswert ist das `backup-1.03` Paket für `afio` von Karel Kubat, das auf verschiedenen FTP-Sites zu finden ist.



## 2.3 Boot-Konzept

Bei einem Desktop PC befindet sich der Bootblock meistens auf der ersten IDE Festplatte. Ihr erster Sektor wird auch als MBR oder Master Boot Record bezeichnet. Auch die einzelnen Partitionen der Festplatte besitzen in ihren ersten Sektoren jeweils Bootsektoren. In diesen legen die Betriebssysteme, die auf diesen Partitionen installiert wurden ihre Bootprogramme ab.

Der MBR wird sofort in den Speicher an Adresse 0000:7C00 kopiert und ausgeführt. Dieser Programmcode heißt Bootprogrammcode:

Das Bootprogramm kopiert sich selbst an Stelle 0000:0600 und fährt dort mit der Ausführung fort. Hier beginnt nun spezifischer Code der verschiedenen Bootmanager. Dazu stehen dem Programm insgesamt 446 Bytes zur Verfügung. Im Anschluss daran befindet sich die Partitionstabelle:

Der Bereich der Partitionstabelle beginnt ab der Adresse 0000:07BE im Speicher, im MBR im ersten Sektor der Festplatte am Offset 01BE.

Die Partitionstabelle selber besteht aus vier Einträgen von jeweils 16 Byte Länge. Jeder dieser Einträge ist wie folgt aufgeteilt:

Bytes	Bedeutung
00	Aktivbit
01-03	Start-CHS
04	Partitionstyp
05-07	End-CHS
08-0B	Start LBA
0C-0F	End LBA

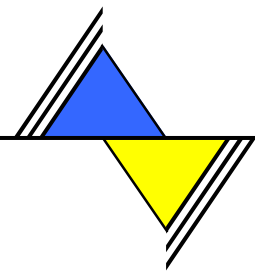
### Struktur eines Eintrags in der Partitionstabelle

Die folgende Grafik zeigt ein Beispiel für eine Beispielpartitionstabelle. Sie weist eine 300MB große Primäre DOS FAT16 Partition mit einer 105,8MB großen Erweiterten Partition sowie eine Linux Partition von 3,9GB und eine Auslagerungsdatei von 256MB.

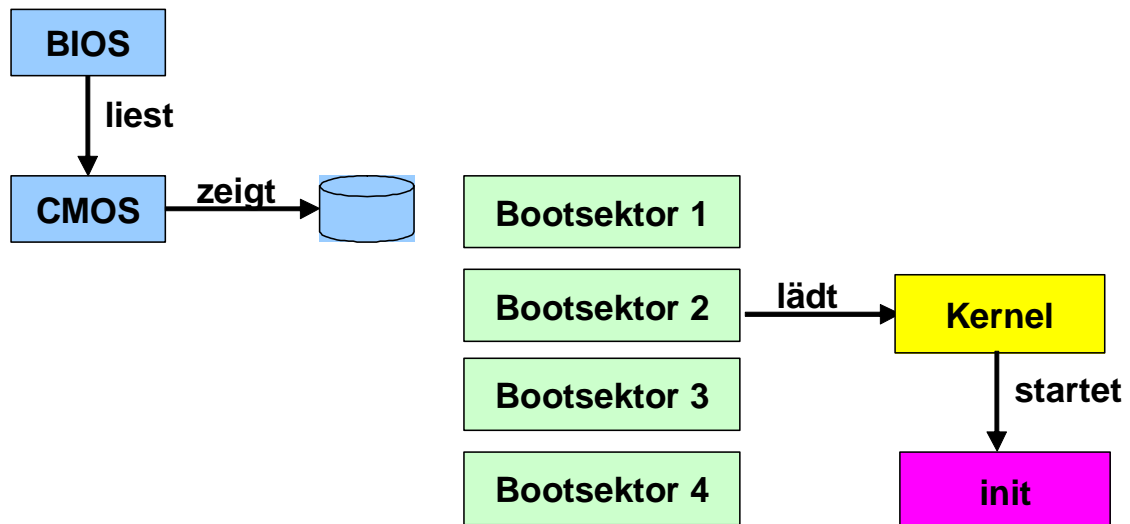
```

Adresse 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00001BE 80 01 01 00 06 0E BE 94 3E 00 00 00 0C 61 09 00
00001CE 00 00 81 95 05 0E FE 7D 4A 61 09 00 72 4E 03 00
00001DE 00 CHSStart 83 CHSEnde BC AF C0 00 BD C5 7C 00
00001EE 00 `` 82 `` 79 75 3d 01 00 00 80 00
AK      CHSStart ID CHSEnde LBA Adresse LBA Länge
    
```

Beispiel für eine Partitionstabelle

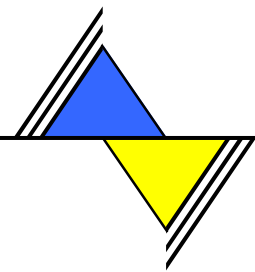


## Bootprozess unter LINUX



### 2.3.1 Das Programm init

Das Programm `/sbin/init` ist dafür zuständig, dass das System korrekt initialisiert wird. Es ist allen anderen Prozessen übergeordnet und wird deswegen auch Vater-Prozess (parent) genannt, d.h. dass alle weiteren Prozesse von `init` oder dessen Kind-Prozessen (child) gestartet werden. `init` hat die höchste Priorität. Demnach wird er bevorzugt vor allen anderen Prozessen ausgeführt. Stirbt `init`, sterben auch alle anderen Prozesse und das System stürzt ab. Zugleich ist `init` das einzige Programm, das direkt vom Kernel gestartet wird und es kann auch mit dem Kommando `kill` nicht beendet werden (weitere Informationen zur Prozessverwaltung sind im gleichnamigen Kapitel zu finden). Über die Datei `/etc/inittab` kann `init` konfiguriert werden. In dieser Datei werden die einzelnen Runlevel, und was in diesen geschehen soll, definiert. Die verschiedenen Skripten, die `init` startet, können durch die Einträge in dieser Konfigurationsdatei festgelegt werden. Diese Skripten sind übersichtlich im Verzeichnis `/sbin/init.d` zusammengefasst.



## 2.3.2 Runlevel

Die Runlevel unter Linux definieren den Zustand des Betriebssystems. Was sich hinter dem einzelnen Runlevel Code verbirgt und in welchem Runlevel das System hochgefahren werden soll (default runlevel), ist in der Datei /etc/inittab dokumentiert. Beachten Sie bitte, dass die Runlevel versions- und distributionspezifisch sind.

### Runlevel 0

Das System wird angehalten.

Es werden alle laufenden Prozesse beendet und zuletzt init selbst abgeschlossen.

Das Abschalten des Rechners ist nach Erreichen dieses Zustandes gefahrlos möglich.

### Runlevel 1

Der Single User Mode ist für Wartungsarbeiten oder Datensicherungen gedacht.

Es läuft ein Linux Kernel, jedoch ist nur die root Partition gemountet.

Nur der Benutzer root darf sich anmelden, alle anderen Benutzer haben keinen Zugang zum System.

Alle Terminals sind getrennt und es ist nur noch die Arbeit an der Konsole möglich.

### Runlevel 2

Im Multi User Mode ohne Netzwerk sind zwar sämtliche Benutzer am System zugelassen, allerdings ist es nicht möglich auf Remote-Systeme zuzugreifen, weil das Netzwerk nicht gestartet wird.

### Runlevel 3

Das Netzwerk wird mitgestartet und das Einloggen erfolgt textbasiert.

### Runlevel 4

Derzeit ist dieser nicht belegt.

### Runlevel 5

Nachdem der X-Server gestartet wurde, dürfen sich alle Benutzer anmelden, auch wenn weiterhin alle Konsolen zur Verfügung stehen.

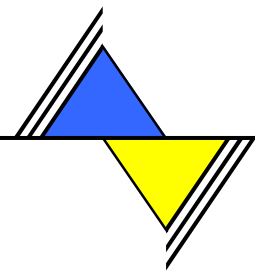
Zudem wird das Netzwerk mitgestartet.

### Runlevel 6

Beim Wechsel in diesen Runlevel wird das System, nachdem alle Dienste beendet und alle Dateisysteme ausgehängt sind, durchgestartet.

Als Systemadministrator (root) ist es jederzeit möglich in einen anderen Runlevel zu wechseln, indem mit dem Kommando init die gewünschte id des Runlevels an das System Übergeben wird:

```
init x
```

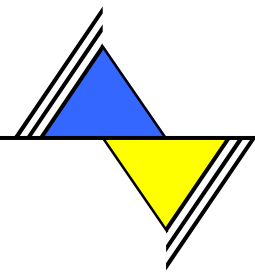


### 2.3.3 Die init-Scripten

Im Verzeichnis `/sbin/init.d` gibt es zwei Arten von Skripten:

- Skripten, die direkt von `init` aufgerufen werden, was beim System-Boot, beim sofortigen Herunterfahren oder beim Neustart geschieht.
- indirekt von `init` gestartete Skripten.

- ⊗ Beim Wechsel von Runleveln wird immer das übergeordnete Skript `/sbin/init.d/rc` ausgeführt, woraufhin die relevanten Skripten in der richtigen Reihenfolge ausgeführt werden.
- ⊗ Alle Skripten sind unter dem Verzeichnis `/sbin/init.d` gespeichert.
- ⊗ Auch die Skripten, die beim Wechseln des Runlevels gestartet werden, sind dort abgelegt, aber sie werden in der Regel als symbolischer Link aus den Unterverzeichnissen `/sbin/init.d/rcx.d` aufgerufen.
- ⊗ Diese Organisation ist sehr übersichtlich und vermeidet ein mehrfaches Vorkommen der Skripten, weil jedes Skript sowohl als Start- als auch als Kill-Skript gestartet werden kann. Der Unterschied liegt in der Parameterübergabe, die `START` oder `STOP` lautet.
- ⊗ Die Kill- und Start-Skripten eines Runlevels sind in entsprechend benannten Verzeichnissen organisiert (`sbin/init.d/rcx.d`) und dort als Link realisiert. Durch ihre Benennung wird ersichtlich, um welchen Typ von Skript es sich handelt.
- ⊗ Die Nummer im Namen der Links gibt an, in welcher Reihenfolge die zahlreichen Skripten abgearbeitet werden. Es kann jedoch auch ein so genannter default runlevel festgelegt werden, in dem das System immer gestartet werden soll.
- ⊗ Dies ist durch Anpassung der Einträge in der Datei `/etc/inittab` möglich. Hier könnte beispielsweise der default runlevel geändert werden oder das Verhalten des Systems beim Drücken der Tastenkombination `Strg + Alt + Entf` festgelegt werden.
- ⊗ Wenn ein Runlevel gewechselt wird, werden die Kill-Skripten des aktuellen Runlevels ausgeführt. Diese beenden verschiedene Programme des Runlevels.
- ⊗ Anschließend werden die Start-Skripten des neuen Runlevels ausgeführt. `init` liest also die Konfigurationsdatei `/etc/inittab` und ruft das Skript `/sbin/init.d/rc` auf, dem es den neuen Runlevel als Parameter übergibt.
- ⊗ Anschließend ruft `rc` alle Kill-Skripten des aktuellen Runlevels auf, die im neuen Runlevel kein entsprechendes Start-Skript aufweisen.



## **3 Systemadministration unter WINDOWS-XP**

### **3.1 Benutzerverwaltung**

Wird später ergänzt

### **3.2 Backup und Recovery**

Wird später ergänzt

### **3.3 Boot-Konzept**

Wird später ergänzt

#####  
#####