




# LINUX GRUNDLAGEN





<b>1</b>	<b><u>ZEITPLAN:</u></b>	<b>4</b>
<b>2</b>	<b><u>GESCHICHTE:</u></b>	<b>5</b>
2.1	<b>DIE GEBURT VON LINUX:</b>	<b>5</b>
2.2	<b>UNIX:</b>	<b>5</b>
<b>3</b>	<b><u>DISTRIBUTIONEN:</u></b>	<b>7</b>
<b>4</b>	<b><u>INSTALLATION:</u></b>	<b>8</b>
4.1	<b>STANDARDINSTALLATION:</b>	<b>8</b>
4.2	<b>INSTALLATION VIA NFS:</b>	<b>9</b>
4.3	<b>TIPPS ZUR INSTALLATION</b>	<b>9</b>
<b>5</b>	<b><u>GRUNDBEGRIFFE (TCP/IP):</u></b>	<b>10</b>
5.1	<b>IP-ADRESSE: (VERS. 4)</b>	<b>10</b>
5.2	<b>NETZWERKMASKE:</b>	<b>10</b>
5.3	<b>ADRESSBEREICHE:</b>	<b>10</b>
5.4	<b>GATEWAY:</b>	<b>10</b>
5.5	<b>DNS-SERVER:</b>	<b>10</b>
<b>6</b>	<b><u>LINUX ALS WORKSTATION:</u></b>	<b>11</b>
6.1	<b>ARBEITEN MIT GNOME ODER KDE:</b>	<b>11</b>
6.2	<b>DRUCKER UNTER LINUX</b>	<b>11</b>
6.2.1	LPD:	12
6.2.2	CUPS:	12
6.3	<b>DER BOOTMANAGER:</b>	<b>13</b>
6.3.1	LILO – DER LINUX-LOADER:	13
6.3.2	GRUB	13
6.4	<b>WÄHLVERBINDUNGEN UNTER LINUX:</b>	<b>14</b>
<b>7</b>	<b><u>SYSTEMGRUNDLAGEN:</u></b>	<b>14</b>
7.1	<b>TERMINALS:</b>	<b>14</b>
7.2	<b>RUNLEVEL:</b>	<b>14</b>
7.3	<b>BENUTZER / GRUPPEN:</b>	<b>15</b>
7.4	<b>DAS DATEISYSTEM:</b>	<b>15</b>
7.5	<b>DIE ZUGRIFFSRECHTE:</b>	<b>18</b>
7.6	<b>STARTEN VON PROGRAMMEN:</b>	<b>18</b>
7.7	<b>X-WINDOWS:</b>	<b>19</b>



<b>7.8</b>	<b>HILFESYSTEM:</b>	20
<b>7.9</b>	<b>WICHTIGE TOOLS / BETRIEBSSYSTEMBEFEHLE:</b>	21
<b>7.10</b>	<b>WICHTIGE VERZEICHNISSE:</b>	23
<b>7.11</b>	<b>WICHTIGE DATEIEN IM VERZEICHNIS / ETC UND DEREN BEDEUTUNG</b>	24
<b>7.12</b>	<b>DER STARTVORGANG IM DETAIL:</b>	26
<b>8</b>	<b><u>ZUGRIFF AUF ANDERE DATEISYSTEME:</u></b>	<b>26</b>
<b>8.1</b>	<b>LINUX-DATEIZUGRIFF AUF NOVELLSERVER</b>	26
<b>8.2</b>	<b>WINDOWSDRUCKER VON LINUX AUS ANSPRECHEN</b>	27
<b>8.3</b>	<b>ZUGRIFF AUF FREIGELEGTE ORDNER VON WINDOWSRECHNERN</b>	27
<b>9</b>	<b><u>KONFIGURATION VON NETZWERKDIENTEN:</u></b>	<b>27</b>
<b>9.1</b>	<b>FERNWARTUNG:</b>	27
9.1.1	TELNET:	27
9.1.2	SSH (SECURE SHELL):	28
9.1.3	FISH:	28
9.1.4	WARTUNG MIT WEBMIN:	28
<b>9.2</b>	<b>FTP-SERVER:</b>	<b>28</b>
<b>9.3</b>	<b>WEB-SERVER:</b>	<b>30</b>
9.3.1	VERWENDUNG VON PHP:	31
9.3.2	INTRANET	32
9.3.3	WEBSEITEN FÜR BENUTZER:	32
9.3.4	ZUGRIFF AUF BESTIMMTE WEBSEITEN MITTELS AUTHENTIFIZIERUNG:	33
9.3.5	VIRTUELLE SERVER:	33
9.3.6	HTTPS – VERSCHLÜSSELTE VERBINDUNGEN:	35
<b>9.4</b>	<b>BENUTZERPLATZBESCHRÄNKUNG:</b>	<b>35</b>
<b>9.5</b>	<b>WARTUNG VON BENUTZERDATEN</b>	<b>36</b>
9.5.1	YAST	36
9.5.2	KOMMANDOZEILE	36
9.5.3	WEBMIN-USERINTERFACE	36
<b>10</b>	<b><u>PROZESSMANAGEMENT</u></b>	<b>37</b>
<b>10.1</b>	<b>PROZESSE ANZEIGEN</b>	<b>39</b>
<b>10.2</b>	<b>PROZESSE ABBRECHEN</b>	<b>40</b>
<b>10.3</b>	<b>GRAPHISCHE PROZESS-TOOLS</b>	<b>41</b>
<b>11</b>	<b><u>LITERATURVERWEISE (INTERNET):</u></b>	<b>42</b>



# 1 ZEITPLAN:

Montag, 02.06.	
09.30 – 10.15	Begrüßung, Organisation, Aufbau und Anschluss der Rechner
10.30 – 12.00	Installation via NFS, IP Grundlagen, Dateisystem-Grundlagen
13.45 – 15.15	Grundkonfiguration (X-Windows, Netzwerk, Sound), Einrichten und Anpassen der grafischen Oberfläche, Tools, Browser, Software nachinstallieren, Vergleich KDE - GNOME
15.30 – 17.00	Drucker installieren, Updates einspielen, Email, Office, Zugriff auf lokale Dateisysteme
Dienstag, 03.06.	
08.45 – 10.15	Konsolen, Hilfesystem, Manpages, Benutzerstruktur, Zugriffsrechte, Prozessmanagement
10.30 – 12.00	Benutzerstruktur, Zugriffsrechte, Netzzugriff auf Novell- und Windowssysteme, Freigaben für Windowsrechner (SAMBA)
13.30 – 15.00	Fernwartungsmöglichkeiten (telnet, ssh, Webmin, VNC)
15.45 – 16.45	Arbeiten im Textmodus Verzeichnisstruktur, Konfigurationsfiles, Sichern von Verzeichnissen
Mittwoch, 04.06.	
08.45 – 10.15	Einführung in das Erstellen von Shellskripten / Reguläre Ausdrücke Der Startvorgang im Detail
10.30 – 12.00	Webserver – Grundkonfiguration FTP-Zugriff einrichten
13.45 – 16.00	MYSQL, dynamische Seiten, PHP ev. CMS (Typo3, Joomla), ELearning (Moodle)

Die Hinweise in diesem Skriptum beziehen sich auf die Installation und Konfiguration der SuSE Distribution 10.2



## 2 Geschichte:

Die Informationen aus diesem Kapitel stammen überwiegend aus SELFLINUX (<http://www.selflinux.org>)

### 2.1 Die Geburt von Linux:

#### 1991

Der 21jährige finnische Student *Linus Benedict Torvalds* beginnt damit, ein auf Minix angelehntes Betriebssystem für AT-386-Computer zu schreiben.

#### 1992

*Linus Torvalds* verteilte die Version 0.12 per anonymous FTP im Internet, was zu einem sprunghaften Anstieg der Testerzahl führte. Da dieser Anstieg so groß wurde, dass die nötige Kommunikation nicht mehr per Email zu bewältigen war, wurde in den Usenet News die Gruppe [alt.os.linux](#) ins Leben gerufen. Dies hatte zur Folge, dass eine explosionsartige Weiterentwicklung des Systems in ganzen Internet stattfand und von *Linus Torvalds* koordiniert wurde.

### 2.2 UNIX:

Linux wurde wie oben bereits erwähnt aus einer UNIX Version entwickelt. Aus diesem Grund ist das nächste Kapitel einer Übersicht über die Entwicklung von Unix gewidmet.

Die Geschichte von UNIX liest sich stellenweise wie ein Heldenepos oder ein Kriminalroman, manchmal auch wie eine Grotteske. Schon die Geburt von UNIX vollzog sich unter eigenartigen Umständen, denn es entsprang einem gescheiterten Projekt:

Anfang 1969 gab es ein Gemeinschaftsprojekt des *MIT*, *General Electric* und den *Bell Labs* von *AT&T*, das Ideen für eine neue Generation von Betriebssystemen gesammelt hatte und daran ging, unter dem Namen *Multics* diese Ideen umzusetzen. Da weder Zeitplan noch Budget eingehalten werden konnten, zog sich *Bell Labs* sehr schnell aus dem Projekt zurück. *Ken Thompson* und *Dennis Ritchie*, zwei Mitarbeiter von *Bell Labs*, die an *Multics* mitgearbeitet hatten, waren von den Einfällen und Erfahrungen, die sie mit *Multics* gesammelt hatten, so beeindruckt, dass sie kurzerhand eine abgespeckte Version des ursprünglichen *Multics* selbst schrieben und unter dem Namen **Unics**, später **UNIX**, in die Welt setzten. (hauptsächlich um das Spiel „Space Travel“ auf eine PDP-7 zu portieren).

*AT&T* verkaufte die Lizenzen zu Unix zu nominellen Gebühren an Universitäten (u.a. auch an die Universität von Berkley die eine eigene Distribution unter der Bezeichnung BSD (Berkley System Distribution) hervorbrachten). Zur Verwendung von BSD waren jedoch auch Lizenzen von *AT&T* notwendig. Auch kommerzielle Anbieter übernahmen UNIX und entwickelten daraus ihre eigenen Systeme. Nachdem im Jahr 1984 *AT&T* die Lizenzgebühren deutlich an hob, entwickelte jeder Hersteller seine eigene Version weiter und die Systeme wie HP-UP, SunOS, AIX (IBM), ULTRIX (Digital), SINIX (Siemens) und XENIX (MS) wurden immer inkompatibler. Verschiedene Standardisierungsbestrebungen wie System V (*AT&T*) und POSIX scheiterten.



Erfolg hatte dagegen ein Projekt, gestartet Anfang der 80er-Jahre am MIT von *Richard Matthew Stallman*, dem "letzten Hacker der Altvorderzeit", das GNU-Projekt: GNU's Not UNIX. Sein Ziel war es, von Grund auf ein neues, UNIX-ähnliches Betriebssystem zu schreiben, das frei verfügbar sein sollte. *Stallman* wollte damit einen Gegenpol zur zunehmenden Proprietarisierung der Softwarewelt schaffen. Durch seinen intensiven Einsatz und Beiträge anderer Programmierer entstand bis Ende der 90er eine beachtliche und leistungsstarke Sammlung an UNIX-Tools. Auch wenn das System bislang nicht vollständig ist, konnten sich die GNU-Tools dennoch auf vielen UNIXen etablieren, unter anderem auch deshalb, da einzelne UNIX-Anbieter ihre Einnahmequellen noch etwas auszubauen gedachten. Mit dem Grundpaket wurde z. B. kein C-Compiler mitgeliefert, worauf viele Systembetreuer, um Geld zu sparen, auf *Stallmans* GNU C-Compiler zurückgriffen, der ohnehin qualitativ besser war. So wurden die GNU-Tools ein systemübergreifender Quasi-Standard. Die freie Entwicklungsmethode hatte erreicht, woran die proprietären Standardisierungsversuche bislang gescheitert waren.

Auch von akademischer Seite wurde der immer zugeknöpfteren Haltung der UNIX-Vertreiber begegnet: zu Anfang wurde der Quellcode von AT&T den Universitäten offen zur Verfügung gestellt und so vielerorts als Tutorial für die Arbeitsweise eines Betriebssystems verwendet. Als AT&T den Quellcode unter Verschluss brachte, fiel diese Möglichkeit weg. Andrew Tanenbaum, Informatik-Professor an der Freien Universität Amsterdam, entschloss sich daher, für seine Studenten eine eigene Version von UNIX zu schreiben, die nichts mit dem urheberrechtlich geschützten Code von AT&T zu tun hatte. Nach zwei Jahren harter Arbeit brachte er sein System unter dem Namen Minix heraus. Es war weniger für die praktische Arbeit, sondern in erster Linie als Lehrobjekt gedacht. Dennoch wurde es von sehr vielen Studenten auch praktisch auf dem heimischen PC eingesetzt, da es im Gegensatz zu den kommerziellen UNIXen für einen moderaten Preis zu haben war. Allerdings stieß Minix in diesem Einsatzgebiet sehr schnell an seine Grenzen. Viele seiner Anwender machten Tanenbaum Vorschläge und schickten Patches für Erweiterungen und Verbesserungen. Tanenbaum allerdings war damit sehr zurückhaltend, da er Minix in erster Linie als Tutorial sah, kam es ihm mehr auf eine knappe und klare Struktur als auf eine möglichst umfassende Funktionalität an.

Ein Minix-Anwender mit Namen Linus Torvalds gab sich damit nicht zufrieden. Das GNU-System war bis auf den Kernel vollständig, aber das Release des GNU-Kernels mit Namen HURD schien noch auf sich zu warten. Um die zeitliche Lücke bis dahin zu füllen, begann er selbst einen Kernel zu schreiben, der sehr rasch unter dem Namen Linux Verbreitung fand und eine große Entwickler- und Benutzergemeinde zusammenbrachte. Da die meisten Entwickler auf UNIXen arbeiteten, auf denen die GNU-Tools liefen, lag es nahe, den Linux-Kernel so einzurichten, dass er zusammen mit den GNU-Tools verwendet werden konnte: GNU/Linux. Der Kernel HURD ist über "akademische" Anfänge bislang nicht hinausgekommen, so dass das anfänglich als "Provisorium" gedachte Linux sich an seiner Stelle etabliert hat.

Zur gleichen Zeit löste sich BSD aus seiner ursprünglichen Abhängigkeit von AT&T: eine Gruppe von BSD-Leuten ersetzte alle Anweisungen im Quellcode, die noch von AT&T beigesteuert waren, durch neue und erstritt in einem



langwierigen Gerichtsverfahren für BSD die Freiheit. Daraus gingen die Projekte FreeBSD, NetBSD und OpenBSD hervor, die auch eine beachtliche Verbreitung gefunden haben und manchmal als Linux-Vettern bezeichnet werden (und so manche Linux-Distribution enthält das ein oder andere "Schmankerl" aus einem der drei Projekte).

### **3 Distributionen:**

(aus <http://www.learninglinux.de/tutorials/distributionen.php> )

Linux selbst ist nur der Kernel (Betriebssystemkern) und frei im Internet verfügbar. Dazu gibt es eine große Anzahl von ebenfalls freier Software wie beispielsweise tausende GNU-Tools.

Einige Distributionen haben es sich zur Aufgabe gemacht aus dem Kernel und freier Software (GNU-Tools, XFree, einer Shell, verschiedener Anwendungsprogramme und vieles mehr) lauffähige Systeme zu erstellen. Das Ergebnis bezeichnet man als Linux oder GNU/Linux als Anerkennung für das GNU-Projekt.

Hier nun die bekanntesten Distributionen im Überblick:

#### **SuSE**

SuSE ist wahrscheinlich eine der am leichtesten und komfortabelsten zu installierenden Distributionen auf dem Markt. Das ist wohl auch einer der Gründe, warum sich viele Anfänger für diese Distribution entscheiden. Mit (meist) wenigen Handgriffen kommt man zu einem funktionierenden Linux-System für den Desktop.

#### **Red Hat**

Red Hat hat generell einen sehr guten Ruf und wird vor allem in professionellen Bereichen häufig verwendet. Ohne Vorkenntnisse könnte man als Linux-Neuling mit dieser Distribution jedoch etwas überfordert sein, insbesondere bei der Integration neuer Hard- und Software.

#### **Fedora Core**

Fedora Core ist ein Community-Projekt und aus Red Hat entstanden. Die Distribution wird oft mit SuSE auf eine Stufe gestellt, da ein sehr komfortabler Installer Linux-Anfängern den Weg zu einem funktionierenden System vereinfacht.

#### **Mandrake**

Mandrake setzt zunehmend auf eine einfachere Bedienung und eine grafische Benutzerführung bei der Installation und Konfiguration. Dies hat wohl auch entscheidend für die zunehmende Beliebtheit dieser Distribution in der letzten Zeit beigetragen.

#### **Knoppix**

Knoppix eignet sich ideal für Linux-Interessierte, die keine Daten auf die Festplatte installieren wollen, da es komplett von CD startet. Deswegen ist das von Dipl.Ing. Klaus Knopper entwickelte Knoppix auch besonders als Test- und Rettungssystem geeignet.

#### **Debian**

Debian unterscheidet sich von anderen Distributionen dadurch, da keine kommerzielle Firma dahintersteckt, sondern nur freiwillige Entwickler. Debian gilt als schlankes und stabiles System, ist im Vergleich zu anderen Distributionen jedoch schwer zu konfigurieren und daher eher für fortgeschrittenere Anwender zu empfehlen.

#### **Slackware**



Slackware ist die Distribution die Unix am nächsten steht und auf der auch Distributionen wie SuSE und Red Hat basieren. Slackware ist eher eine Distribution für "Linux-Experten", da es keine grafischen Werkzeuge wie beispielsweise YaST bei SuSE zum Einrichten des Systems gibt. Da alle Einstellungen in den Konfigurations-Dateien editiert werden können ist es wahrscheinlich auch deswegen bei erfahrenen Linux-Usern sehr beliebt.

### **Ubuntu**

Ubuntu Linux ist eine relativ neue Distribution, die auf Debian basiert und daher sehr stabil läuft. Ihr Hauptaugenmerk liegt auf aktuellen Softwarepaketen und guter Hardwareerkennung und Unterstützung. Neue Versionen erscheinen halbjährlich. Ubuntu hat sich innerhalb weniger Monate zu einer der beliebtesten Distributionen entwickelt und ist sehr benutzerfreundlich.

### **Gentoo**

Gentoo hat seinen Namen von dem kleinen und schnellen Eselspinguin und ist eine Distribution für Benutzer, die ein sehr individuelles System wollen. Für die Installation sollte man schon weitreichendere Linux-Kenntnisse besitzen, da dabei aus den Quellen der Pakete kompiliert wird.

### **Arch Linux**

Arch Linux unterscheidet nicht wie andere Distributionen zwischen Entwicklungs- und stabilen Versionen, da die Programme immer auf dem neuesten Stand gehalten werden. Es ist eine sehr schlanke Distribution und kann fast ausschließlich über eine einzige Datei konfiguriert werden.

## **4 Installation:**

### **4.1 Standardinstallation:**

Standardmäßig wird Linux via CD/DVD Laufwerk installiert. Dazu sind die ersten beiden CD's als Boot-CD's ausgeführt. Falls man über einen Rechner mit älterem BIOS (der nicht von CD's booten kann) verfügt, kann man auch eine Boot-Diskette erstellen, von der aus der CD-Zugriff erfolgen kann (unter DOS/Windows das Setup von der CD starten).

Seit 6.3 hat man die Wahl zwischen einer grafischen Oberfläche während der Installation (yast2, mind. 64MB Ram erforderlich!; Booten von CD1) und der bisherigen Installationsoberfläche yast1 (Textmodus, Booten von CD2)

Ab Linux 7.1 wird die Distribution auch auf einer DVD ausgeliefert. Falls man über eine DVD -Laufwerk verfügt ist diese Variante sicherlich zu bevorzugen, da kein Wechsel des Installationsmediums notwendig ist.

Da für das Seminar nicht so viele CD-Sätze wie notwendig vorhanden sind, wird hier ein anderer Weg gewählt. Für die Installation wird ein NFS (Network File System) – Server eingerichtet, auf dessen Festplatte der Inhalt der Installations – DVD's (in das Verzeichnis /nfsserver/10.2) kopiert wurde.

Die Arbeitsplatzrechner werden über die CD1 gebootet (Alle CD/DVD Images stammen von <http://www.opensuse.de>)



## 4.2 Installation via NFS:

Voraussetzung: funktionierende Netzwerkverbindung zum NFS-Server (Hardwareseitig und richtig konfiguriertes TCP/IP-Protokoll; Infos zu TCP/IP z.B.(bei installiertem Linuxrechner) [http://localhost/sdb/de/html/cep\\_ip\\_base.html](http://localhost/sdb/de/html/cep_ip_base.html) ; diese Seite ist auch am SuSE Webserver zu finden; weitere Infos auch am PI-Server: <http://www.pinoe-hl.ac.at> ). Die zur Installation notwendigen Disketten (Bootdisk, Moldules1 und Modules3) befinden sich als Images auf der ersten CD im Verzeichnis /boot. Falls der Rechner ein SCSI System, besitzt ist auch die Modules2 Diskette erforderlich. Zur Erzeugung der Disketten kann das ebenfalls auf der CD enthaltene Programm Raw-Write (im Ordner /dosutils) verwendet werden.

Bei der aktuellen Version werden die Rechner mit CD1 gebootet. Beim Auswahlschirm sind (damit eine Installation über das Netzwerk abläuft ein paar Parameter zu ändern:

F2 → Installationsprache auswählen

F3 → Auflösung wählen (Vorschlag: 1024X768)

F4 → Installationsmedium (NFS: Server: 10.0.1.1 Verzeichnis: /nfsserver/10.2

)

TIPP: Für eine Remote-Installation kann weiters noch der Parameter vnc=1 (+ ev. vnpasswoord=

.....) gesetzt werden. Weitere Informationen siehe Novell/SuSE – Dokumentation: <http://www.novell.com/de-de/documentation/opensuse102/>

## 4.3 Tipps zur Installation

Bei der Installation sind einige Entscheidungen notwendig. Eine wichtige Entscheidung ist die Partitionierung der Festplatte. Eine Standardinstallation verwendet i.a. mindestens 3 Partitionen:

/boot kleine Partition für die Startdateien des Systems (sollten vor dem 1024. Zylinder liegen)

swap Partition für Auslagerungsdaten (hier verwendet Linux ein eigenes Dateisystem)

/ Die Root- (Wurzel) Partition des Dateisystems

Für die Datenpartitionen stehen 3 unterschiedliche Dateisysteme zur Verfügung:

Ext2: Das Extended-2- Filesystem (bisher das Standarddateisystem unter Linux). Es unterstützt Benutzerplatzbeschränkung (Quotas). Ein Nachteil ist, dass bei einem Systemabsturz bei einem neuerlichen Systemstart erst ein e2fsck (FileSystemCheck) durchgeführt wird, der bei größeren Platten einige Minuten dauern kann.

ReiserFS: Ein Journaling-Filesystem. Erlaubt bei einem Systemabsturz einen schnellen Systemstart. Nachteil: Quotas sind nicht möglich.

Ext3: Vereinigt die Vorzüge beider Systeme.

Es ist auch möglich (z.B im Serverbereich) Software-Raid einzurichten und Partitionen in ein solches Raid-System zu legen.



## **5 Grundbegriffe (TCP/IP):**

(Siehe auch PI-Weberserver → Grundlagen von IP und Netzwerken)

### **5.1 IP-Adresse: (Vers. 4)**

Vergleichbar mit Telefonnummer (weltweit eindeutig!!), besteht aus 4 Byte (durch Punkte getrennt)

### **5.2 Netzwerkmaske:**

Bitmaske, die mit der IP-Adresse mit „UND“ verknüpft wird, um Netzwerkanteil (~Vorwahl) und Hostanteil (~lokale Nummer) zu ermitteln.

255.255.255.0 C-Klasse  
255.255.0.0 B-Klasse  
255.0.0.0 A-Klasse

### **5.3 Adressbereiche:**

- Klasse A:
  - 1. Bit = 0 → 0 – 126 .x.x.x
- Klasse B:
  - 1. Bit = 1, 2. Bit = 0 → 128 – 191.x.x.x
- Klasse C:
  - 1.Bit=1, 2.Bit=1, 3.Bit=0 → 192 – 223.x.x.x

#### **private Adressbereiche:**

- 1 Class A
  - 10.0.0.0 – 10.255.255.255
- 16 Class B
  - 172.16.0.0 – 172.31.255.255
- 256 Class C
  - 192.168.0.0 – 192.168.255.255

#### **Loopback:**

- 1 Class A
  - 127.0.0.0 – 127.255.255.255

### **5.4 Gateway:**

Adresse des nächsten Routers am Weg ins Internet (wohin sollen alle Pakete geschickt werden, die nicht ins lokale Netz gehören)

### **5.5 DNS-Server:**

Adresse des (der) Domain-Name-Servers (liefert die Übersetzung der Internetnamen in IP-Adressen)



## **6 LINUX als WORKSTATION:**

Nach einer einfachen Installation von SuSE Linux (z.B. Standard mit Office) kann man aus dem Linuxrechner relativ einfach einen vollwertigen Arbeitsplatz machen. Auf der grafischen Oberfläche findet man Tools zur Konfiguration von Netzwerkdiensten (Anschluss über Netzwerkkarte, Einwahl via PPP) und zur Konfiguration von Druckern. Zu den Druckern wäre zu erwähnen, dass Unix-Systeme grundsätzlich PostScript-Drucker erwarten. Falls man einen Drucker verwendet, der kein PostScript versteht, muss über einen passenden Filter (aps-Filter) der Druckbefehl aufbereitet werden. Für die gängigsten Druckertypen findet man passende Druckerfilter als Teil der Distribution.

Das Paket OpenOffice wurde bei der Standardinstallation bereits installiert. Es ist pro Benutzer nur noch das Abarbeiten eines kleinen Installationskripts notwendig. Für alle anderen Tätigkeiten findet man in der Distribution, oder im Internet passende Anwendungen zur Erledigung der Aufgaben.

### **6.1 Arbeiten mit GNOME oder KDE:**

Die beiden derzeit mächtigsten Windowmanager für das Linux X-Window System sind GNOME und KDE. Für beide stellt SuSE umfangreiche Dokumentationen zur Verfügung:

<http://www.novell.com/de-de/documentation/opensuse102/>

### **6.2 Drucker unter Linux**

Wie bereits erwähnt erwartet Linux einen postscriptfähigen Drucker. Um auch nicht Postscriptdrucker verwenden zu können verwendet man einen geeigneten Filter (vgl. Druckertreiber), der die Postscriptbefehle in ein für den Drucker verständliches Format umwandelt. Falls der Drucker lokal am Rechner angeschlossen ist kann Linux gängige Drucker automatisch erkennen. In der Schulumgebung sind die Drucker an Printserver angeschlossen, die man unter Linux direkt ansprechen kann. Dafür gibt es in der Druckerkonfiguration auch bereits passende Eingabemasken, um etwa die IP - Adresse des Servers und die Bezeichnung des Druckers am Server angeben zu können. Wie Drucker anderer Systeme (Windows) angesprochen werden können folgt etwas später im Skriptum.

Im Prinzip hat man nun die Auswahl zwischen 2 verschiedenen Drucksystemen: LPD und CUPS. Bei einer Standardinstallation wird nun CUPS installiert. Eine Ausnahmesituation, in der es notwendig ist auf das ältere LPD-System umzusteigen ist das Ansprechen einer Novell-PRINTQUEUE aus LINUX. (bei modernen Novell-Installation wird allerdings das neue IPRINT System verwendet, wofür es auch einen Linuxclient gibt.)



### 6.2.1 LPD:

Das LPD (LinePrinterDämon) System erzeugt im allgemeinen Filterwarteschlangen, in denen die Postscriptbefehle des Systems in für den jeweiligen Drucker verständliche Befehle übersetzt werden. Wenn nun die Druckerwarteschlange (z.B lp ) am Rechner mit der IP-Adresse 10.0.2.10 freigegeben ist, können alle anderen Rechner des Netzwerkes in diese Warteschlange drucken. Dabei ist zu achten, das auf den Clientrechnern keine weitere Aufbereitung (Umwandlung) der Druckdaten erfolgen darf. (RAW-Warteschlange). Bei einer Umleitung in eine Novellwarteschlange ist zu beachten, dass von yast aus die Datei /etc/lpfilter/xxxxx/redirect (xxxxx steht für den Namen der lokalen Druckerwarteschlange) möglicherweise fehlerhaft erzeugt wird (wenn z.B. der Novellbenutzer kein Passwort besitzt) und händisch korrigiert werden muss.

### 6.2.2 CUPS:

Bei Cups unterscheidet man zwischen Client und Serverkonfigurationen. Der Rechner, an den der Drucker direkt angeschlossen ist dient als Druckserver und muss daher als CUPS-Server konfiguriert werden. (dies ist die Standardinstallation!!). Die Konfiguration dieses Druckers erfolgt über das entsprechende YAST Modul. (Auch die Freigabe dieses Druckers für das Netzwerk).

Auf allen anderen Rechnern des Netzwerkes, die diesen Drucker verwenden wollen wird über YAST eingestellt, dass sie CUPS-Clients sind. In die entsprechende Maske trägt man nur noch die Adresse des CUPS-Servers ein. Damit können sie alle am CUPS Server definierten Drucker verwenden.





Bei Verwendung des CUPS Drucksystems kann die Druckerverwaltung auch über WEB (Port 631) erfolgen. Damit dies vollständig funktioniert ist unter SuSE 9.2 die Zuweisung eines Ip-Passwortes für eine Benutzerkennung (die die Administration durchführen soll) notwendig. Dies kann z.B. mit dem Befehl „lppasswd -g sys -a <USERNAME>“ erfolgen.

## **6.3 DER BOOTMANAGER:**

### **6.3.1 LILO – Der Linux-Loader:**

Lilo eignet sich auch als Bootmanager des Rechners bei verschiedenen Betriebssystemen. Er wird über die Datei /etc/lilo.conf konfiguriert. Die erste Konfiguration wird als Standardbetriebssystem verwendet. Nach einer Änderung in lilo.conf muss noch der Befehl lilo ausgeführt werden.

### **6.3.2 GRUB**

Als Alternative zu LILO wird in den neueren Distributionen GRUB (GRand Unified Bootloader) installiert.

GRUB verwendet als Konfigurationverzeichnis den Ordner /boot/grub und wird in den MBR installiert. Grub verfügt über ein Kommandozeileninterface sowie eine Menüauswahl. Die Konfiguration kann über einen Editor oder über das Konfigurationstool YAST --> Konfiguration des Bootloaders erfolgen. Eine detaillierte Information erhält man über "info grub" in einem Shellfenster.

Beispieldatei /boot/grub/menu.1st

```
# Modified by YaST2. Last modification on Sun Jun  8 18:34:36 2003
```

```
color white/blue black/light-gray
default 1
gfxmenu (hd0,2)/boot/message
timeout 8
```

```
title linux
  kernel (hd0,2)/boot/vmlinuz root=/dev/hda3 vga=0x314 splash=silent showopts
  initrd (hd0,2)/boot/initrd
```

```
title windows
  root (hd0,0)
  chainloader +1
```

```
title floppy
  root (fd0)
  chainloader +1
```

```
title failsafe
```



```
kernel (hd0,2)/boot/vmlinuz.shipped root=/dev/hda3 showopts ide=nodma
apm=off acpi=off vga=normal nosmp noapic maxcpus=0 3
initrd (hd0,2)/boot/initrd.shipped
```

Zu beachten ist, dass in der Grub-Konfiguration die Partitionen der Festplatte anders als üblich angesprochen werden: (hd0,0) bedeutet erste Festplatte (hd0) und darauf die erste Partition.

Mit dem Kernelparameter wird auf eine bestimmte Datei verwiesen, die den Kernel enthalten sollte, der gebootet wird. Optional können hier gewisse Startparameter übergeben werden.

Auch Betriebssysteme wie Windows oder OS/2, die eigentlich von GRUB nicht direkt unterstützt werden können mit diesem Bootsystem gestartet werden. Dazu dient der Parameter „chainloader +1“, der GRUB anweist in der jeweiligen Partition jeweils den nächsten Sektor anzusteuern und zu booten.

**6.4 Wählverbindungen unter Linux:**

Ab SuSE 8.0 empfiehlt sich eine Konfiguration über yast2 (auch Dial-on-Demand) kann hier entsprechend gesetzt werden.

**7 SYSTEMGRUNDLAGEN:**

**7.1 Terminals:**

Der Rechner bietet bei Standardkonfiguration an der lokalen Konsole mindestens 6 Terminals an. Ein Umschalten zwischen den einzelnen Konsolen erfolgt mit der Tastenkombination [Strg]+[Alt]+[Fx], wobei mit [Fx] eine der Funktionstasten gemeint ist.

Konsolen 1-6	Textkonsole
Konsole 7	Grafische Oberfläche
Konsole 10	Systemmeldungen (kein Login-Prompt)

**7.2 Runlevel:**

Das installierte Standardsystem kennt mehrere Betriebsstufen (RUNLEVELS)

RUNLEVEL	AUFGABE
0	„halt“ Anhalten des Systems
1	Betrieb ohne Netzwerkfunktionen
2	Lokaler Betrieb mit remote-Netzwerk
3	Betrieb mit allen Netzwerkfunktionen
5	Betrieb mit allen Netzwerkfunktionen und grafischer Oberfläche
6	„reboot“ Neustarten des Systems



### 7.3 Benutzer / Gruppen:

Systembenutzer werden am einfachsten über yast → Administration des Systems → Benutzerverwaltung angelegt. Jedem Benutzer wird eine eindeutige numerische ID zugewiesen, mit der er im System identifiziert wird. Weiters kann dem Benutzer eine Login-Shell (STANDARD: /bin/bash), eine primäre Gruppenzugehörigkeit (STANDARD: users) und ein Home - Verzeichnis zugewiesen werden. Diese Informationen werden in die Datei „/etc/passwd“ eingetragen. Das Passwort wird aus Sicherheitsgründen verschlüsselt in der Datei „/etc/shadow“ gespeichert.

Analog kann man auch über den Menüpunkt Gruppenverwaltung eine Benutzergruppe anlegen. (Diese Definition wird in die Datei „/etc/group“ eingetragen).

Eine Sonderstellung hat der Benutzer „root“. Er ist vergleichbar mit dem Administrator unter NT bzw. dem Supervisor / Admin unter Novell. Für diesen Benutzer (übrigens mit der ID 0) gelten keine Einschränkungen. Daher muss diese Kennung besonders abgesichert werden (Passwort; darf sich nur von lokalen Konsolen einloggen (Ausnahme ssh); ftp-Zugriff ist nicht erlaubt).

Für die Verwaltung der Benutzer und Gruppen steht auf der KDE-Oberfläche auch der Usermanager zur Verfügung.

### 7.4 Das Dateisystem:

Unter Linux werden die verschiedensten Dateisysteme (auch der Speicher) in eine gemeinsame Wurzel („/“ root) hineingemountet. Als Trennzeichen zwischen den Verzeichnissen dient der Slash „/“. (Achtung: Linux unterscheidet zwischen Groß- und Kleinbuchstaben). Welche Dateisysteme beim Start unter welcher Bezeichnung („Mountpoint“) in das Dateisystem eingegliedert werden, steht in der Datei „/etc/fstab“.

Device	Bedeutung
/dev/hda	1. IDE -Platte (Master auf 1. Contr.)
/dev/hda 1	1. Partition auf 1. Platte
/dev/hda 2	2. Partition auf 1. Platte
.....	
/dev/hdb	Slave auf 1. IDE-Contr.
/dev/hdc	Master auf 2. IDE-Contr.
/dev/sda	1.SCSI-Platte
/dev/sda 1	1. Partition auf 1. SCSI-Platte
/dev/sdb	2. SCSI-Platte



Bei einer Standardinstallation werden 3 Partitionen auf der Platte angelegt:

Mount-Point	Funktion
/boot	Enthält die Boot-Dateien des Systems (müssen vor dem 1024. Zylinder liegen) (wird bei modernen Rechnern nicht mehr extra angelegt!!)
	Swap-Partition
/	Root-Partition (enthält die Wurzel des Dateisystems)

Händisch den Zugriff auf Dateisysteme ermöglichen:

Diskette a: `mount /dev/fd0 /media/floppy`  Diskette ist unter `/media/floppy` im System erreichbar

CD: `mount /dev/hdc /media/cdrom`  CD-Rom ist unter `/media/cdrom` im System erreichbar

(Wenn das CD-RomLaufwerk als Master am 2. Controller angeschlossen ist)

**ACHTUNG: Veränderungen werden zunächst nur im Arbeitsspeicher abgelegt (schnellerer Zugriff) und erst später mit dem Inhalt des Datenträgers synchronisiert. Diese Synchronisation erfolgt spätestens beim Deaktivieren (umount) des Datenträgers. Dies ist vor allem für das Kopieren auf Disketten wichtig.**

**Hinweis zur aktuellen Version (SuSE9.1):**

**Mit der neuen Kernelversion (2.6.x) ist da händische Mouten von CD´s, DVD´s, .. nicht mehr notwendig.**

**Weitere Informationen:**

Mountpoint: Ein (leerer) Verzeichniseintrag, über den via Mount-Befehl die Verbindung zu einem Dateisystem hergestellt.

Wer darf Mouten? Das Einhängen (mounten) von Dateisystemen ist üblicherweise dem root-Benutzer vorbehalten. Eine Ausnahme bilden jene Devices, die in der Datei `/etc/fstab` definiert werden und den Parameter `user` enthalten.

Was kann gemountet werden:

Lokale Systeme: `/dev/fd0` (Diskette), `/dev/hdc` (z.B. CD), `/dev/hda2` (Partition)

NFS-Platten: `10.0.1.1:/suse` (freigebener Ordner `/suse` auf `10.0.1.1`)

Unter Linux werden alle Geräte über Devices angesprochen. Devices sind definierte Schnittstellen zu der Hardware, wobei der direkte Zugriff auf die Hardware nur dem Kernel vorbehalten ist. Im Grunde sind Devices Dateien mit einer Inode, jedoch ohne Datenteil und sind im Verzeichnis `/dev` gespeichert.

Sie werden durch drei Informationen beschrieben: der Major Nummer, der Minor Nummer und dem Zugriffstyp. Die Major Nummer bestimmt den Linuxkerneltreiber,



der für dieses Device verantwortlich ist. Hiervon gibt es zur Zeit 25 Treiber. Die Minor Nummer spezifiziert die Geräteart, z. B. 720 kB Floppy oder 1,44 MB Floppy, bei Festplatten gibt sie die Partition an, die verwaltet werden soll. Beim Zugriffstyp gibt es zwei Arten, blockorientiert für gepufferte Geräte wie Festplatten und zeichenorientiert für ungepufferte Geräte wie serielle Schnittstellen.

Zum Teil finden Sie im /dev-Verzeichnis auch Links. Diese werden mit einem beschreibenden Namen für die Vereinfachung auf Standardschnittstellen gelegt, z. B. für das Modem oder die Maus. Eine Übersicht über die häufig benötigten Schnittstellen ist in nachfolgender Tabelle aufgeführt.

Devicename	Gerät
*bm	Bus Mäuse
console	aktive Konsole
fd*	Diskettenlaufwerk
ftape*	Floppystreamer (ohne Rückspulfunktion) (Link)
hd*	IDE / ATA Festplatten oder CDROM
lp*	Parallele Schnittstellen
mem	Arbeitsspeicher
modem	Standardschnittstelle für das Modem (Link)
mouse	Standardschnittstelle für die Maus (Link)
null	unendlich großes Loch (``schwarzes Loch'')
psaux	PS/2 Maus
nst*	SCSI Bandlaufwerk ohne Rückspulen
port	E/A-Schnittstellen
ptyp*	Terminalschnittstellen unter X (Master)
rmt	Bandlaufwerk (ohne SCSI)
sd*	SCSI Laufwerk
scd*	SCSI CD-ROM
st*	SCSI Bandlaufwerk mit automatischer Rückspulfunktion
tape*	Standardstreamer (Link)
tty*	Virtuelle Textterminals
ttyp*	Terminalslaves unter X
ttyS*	Serielle Schnittstellen

Die Sternchen in der Tabelle sind Platzhalter für eine nähere Bezeichnung, welches Gerät angesprochen werden soll. Hierfür gibt es zwei Notationen. Bei zeichenorientierten Geräten wird numerisch von ``0" an begonnen, die Geräte durchzuzählen. Z. B. ist die erste serielle Schnittstelle /dev/ttyS0 (unter DOS COM1). Bei blockorientierten Geräten gibt es zwei Methoden. Zum einen wird z. B. beim Diskettenlaufwerk oder SCSI CD-ROM die Spezifizierung wie bei zeichenorientierten Geräten vorgenommen, /dev/fd0 zeigt auf das erste Floppy. Bei Festplatten und CD-ROMs wird die Spezifizierung mittels Buchstaben vorgenommen, /dev/sda zeigt auf



das erste SCSI Laufwerk, /dev/hda auf die Master-Festplatte am ersten IDE-Controller. Die einzelnen Partitionen einer Festplatte werden über eine weitere Zahl beschrieben, hier jedoch wird mit der Nummerierung mit ``1" begonnen! Z. B. kennzeichnet /dev/hdc5 die erste logische Partition der Festplatte, die als Master am zweiten IDE-Port hängt (siehe auch , Seite ).

Der Systemverwalter kann weitere Devices anlegen, wenn die bereits durch die Installation vorhandenen Devices nicht ausreichen. Hierfür gibt es den Befehl mknod. Syntax von mknod:

```
mknod [Optionen] Name Typ [Major Minor]
```

Typ:      b           legt blockorientiertes Device an  
          c, u       legt zeichenorientiertes Device an  
          p           legt eine FIFO an

### 7.5 Die Zugriffsrechte:

RECHT	Bezeichnung	Numerischer Wert
Lesen	r	4
Schreiben	w	2
Ausführen	x	1

Diese drei Zugriffsrechte werden nun 3mal vergeben:

Für den Besitzer der Datei:           owner

Für die Besitzergruppe der Datei:   group

Und für alle anderen:                others

Der Befehl zum Ändern der Zugriffsrechte lautet: „chmod“ . Der Befehl chmod 644 /etc/passwd weist der Datei /etc/passwd den Modus 644 zu. Dies bedeutet, dass der Besitzer die Datei lesen und schreiben (verändern) darf, die Gruppe und auch alle anderen dürfen diese Datei nur lesen. Eine ausführbare Datei wird durch das Setzen des Execute-Bits gekennzeichnet.

Auf der grafischen Oberfläche können die Zugriffsrechte wie unter Windows üblich über das Kontextmenü gesetzt werden.

Für eine feinere Steuerung von Zugriffsrechten kann mit so genannten Access Control Lists (ACL) durchgeführt werden. Diese werden vom neuen Kernel unterstützt (vorzugsweise auf EXT3 Dateisystemen) und bieten vor allem im Zusammenhang mit der Verwendung von Samba den Vorteil die Rechtesteuerung von Windows auf Linux zu übertragen. Unter Linux selbst fehlen bei vielen Tools noch die Unterstützungen für ACLs (z.B bei Konqueror).

### 7.6 Starten von Programmen:

Wenn eine Datei als „Ausführbar“ gekennzeichnet ist, kann sie aus einer Shell heraus gestartet werden. Entweder starten sie das Programm mit vollständigem Pfad, oder, wenn das Programmverzeichnis im Suchpfad enthalten ist, genügt ein Aufruf ohne Pfadangabe (z.B.: yast). Selbst wenn man mittels „cd Verzeichnis“ ins Programmverzeichnis gewechselt hat muss der Programmaufruf mit ./PROG\_NAME erfolgen. Wenn sie mehrere Kommandos nacheinander ausführen wollen können sie sie in einen Aufruf (durch ; getrennt) schreiben.



Sonderformen:

Ping 131.130.1.11&	Startet ein Ping zum angegebenen Host als Hintergrundprozess
Ping 131.130.1.11& >/dev/tty10	Startet den Hintergrundprozess und leitet die Ausgabe auf die Konsole 10 um
Nohup ping 131.130.1.11&	Dieser Prozess wird auch beim Ausloggen des Benutzers nicht beendet

### 7.7 X-WINDOWS:

**HINWEIS:** Ab SuSE 9.1 ist der XDMCP Dienst (starten von X-Windows Programmen auf fremden Rechnern) als Standard deaktiviert. Mit dem Sysconfig-Editor kann man 2 notwendige Variable (XDMCP starten und Port 6000 (TCP) aktivieren in der Gruppe System -> Displaymanager aktivieren. Danach wird ein Neustart der grafischen Oberfläche benötigt.

Bei einem laufenden X-Server (grafischer Oberfläche) ist es auch möglich anderen Rechnern zu gestatten X-Windows-Programme auf dem (fremden) Rechner zu starten. Dies kann z.B. verwendet werden um mittels telnet oder ssh einen fremden Rechner zu übernehmen, ein Programm für die grafische Oberfläche zu starten und die Ausgabe dieses Programmes (wie kwrite oder kfmclient (KONQUERER)) auf den eigenen Rechner umzuleiten.

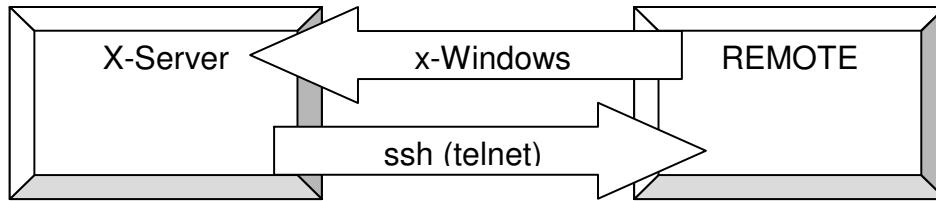
**HINWEIS:** Der Novellserver bietet die Möglichkeit (konfigurierbar über inetcfg an der Serverkonsole) einen Remotezugriff einzurichten. Neben der RCONSOLE, die nur das IPX-Protokoll verwendet kann auch eine X-Konsole eingerichtet werden. Damit ist von außerhalb eine telnet-Verbindung zum Server möglich, die Rückverbindung vom Server zum eigenen Rechner startet dann ein Fenster auf der X-Windows-Oberfläche, von dem aus die Serverkonsole dann administrierbar wird. Da die Novellserver im allgemeinen jedoch nicht direkt im Internet sichtbar sind. Wäre für die Einrichtung einer solchen Verbindung noch eine spezielle Anpassung der Firewallregeln am Linuxrechner (wird im Teil 3 behandelt) notwendig.

**ERGÄNZUNG:** Eine einfachere Fernwartung des Novellservers erhält man jedoch, wenn man ein vt100 kompatibles Terminal verwendet. Dazu kann man unter Linux mit dem Befehl „export TERM=vt100“ in diesen Modus umsteigen. (Kontrolle mit „echo \$TERM“). Danach kann man mit einer Telnet-Verbindung zum Novellserver die Serverkonsole auf einen Linuxrechner umleiten (z.B.: Linux mit ssh übernehmen und dann vom Linuxrechner mit telnet die Serverkonsole übernehmen). Wenn ein direkter Zugriff auf den Novellserver möglich ist kann man z.B. auch mit dem Telnet von Windows 2000 die Konsole übernehmen:

- telnet (ohne Zielangabe)           startet die telnet-Sitzung
- set TERM vt100                    schaltet in den VT100 Modus
- open 10.x.y.z                     verbindet zum Server
- quit                                schließt Telnet



**Siehe Grafik:**



**BEFEHLE:**

Starten eines Programmes mit Ausgabe auf einem anderen Rechner:

```
kwriteln -display 10.0.1.1:0.0 &
```

wenn vorher der Zugriff auf den Rechner 10.0.1.1 mit xhost MY\_IP erlaubt wurde.

- Xhost + schaltet die Zugriffskontrolle aus (alle dürfen)
- xhost - schaltet die Zugriffskontrolle wieder ein

Oder export DISPLAY=10.0.1.1:0.0

```
kwriteln
```

.....

Xn.hosts anlegen (steuert den Zugriff auf den Server n)

xlsfonts zeigt die inst. Fonts

xfontsel zeigt einen Probetext in der gew. Schriftart

```
xterm -geometry 70x35 -fn lucidasanstypewriter-bold-24 &
```

```
xterm -geometry +100+50 70x35 -fn lucidasanstypewriter-bold-24 // BS
```

Position und Größe

.Xresources im jeweiligen Home Verzeichnis des Users. Dort können Sie Zeilen wie die folgenden eintragen:

```
XTerm*font: -misc-fixed-bold-r-normal--13-100-100-100-c-70-iso8859-1
```

```
XTerm*Background: bisque2
```

```
XTerm*Foreground: blue
```

```
XTerm*geometry: 90x40
```

**Starten einer weiteren Oberfläche**

```
/usr/X11R6/bin/X :1
```

```
BEFEHL -display localhost:1.0
```

bzw.: einen Fenstermanager auch auf der neuen Oberfläche starten:

```
export DISPLAY=localhost:1.0
```

```
/usr/X11R6/bin/windowmaker
```

Für die Verbindung zu einem X-Server wird der Port 6000+n verwendet, wobei n die Nummer des X-Servers ist. D.h. die erste X-Windows Oberfläche ist über Port 6001 erreichbar, die 2. über 6002 u.s.w.

**7.8 Hilfesystem:**

Am Linuxrechner gibt es mehrere Möglichkeiten Informationen zu installierten Diensten und Tools zu erhalten. Bei installierter grafischer Oberfläche (und gestartetem Webserver) kann man die Hilfedateien von Suse mit einem Webbrowser



betrachten. (Diese Informationen kann man natürlich auch auf der SuSE-Homepage [www.suse.de](http://www.suse.de) betrachten.) Auf der Textoberfläche kann man mit dem Befehl „man DIENST“ die MAN-Page (manual) zu einem bestimmten Dienst ansehen. (z.B. man chmod). Falls man in diesen beiden Hilfesystemen nicht die passenden Informationen findet und man z.B. zum Dienst wuftp (ein FTP-Server unter Linux) Konfigurationsmöglichkeiten nachlesen will bietet sich (bei installiertem Dienst) noch ein Nachsehen im Verzeichnis /usr/share/doc/packages/wuftp an. In dieses Verzeichnis (/usr/share/doc/packages) werden zu jedem installierten Paket einige Informationsdateien abgelegt. Falls auch damit nicht die gewünschte Information zur Verfügung steht bleibt nur noch der Weg ins Internet.

### 7.9 Wichtige Tools / Betriebssystembefehle:

Befehlssyntax	Zweck
echo "some text"	schreibt some text auf den Bildschirm
ls	listet Dateien auf
wc -l file wc -w file wc -c file	zählt die Zeilen in einer Datei oder zählt die Wörter in einer Datei oder zählt die Anzahl der Buchstaben
cp sourcefile destfile	kopiert sourcefile nach destfile
mv oldname newname	benennt eine Datei um bzw. verschiebt sie
rm file	löscht eine Datei
grep 'pattern' file	sucht nach Zeichenketten in einer Datei Beispiel: grep 'searchstring' file.txt
cut -b colnum file	holt Daten aus Textspalten mit fester Breite Beispiel: hol die Zeichenpositionen 5 bis 9 heraus cut -b5-9 file.txt Verwechsle diesen Befehl nicht mit "cat", der etwas ganz anderes macht
cat file.txt	schreibt file.txt nach stdout (deinen Bildschirm)
file somefile	beschreibt, von welchem Typ die Datei somefile ist
read var	fordert den Benutzer zur Eingabe aus und schreibt sie in eine Variable (var)
sort file.txt	sortiert Zeilen in file.txt
uniq	löscht doppelte Zeilen, wird in Kombination mit sort benutzt, da uniq nur aufeinander folgende doppelte Zeilen entfernt Beispiel: sort file.txt   uniq
expr	rechnen mit der Shell Beispiel: addiere 2 und 3 expr 2 "+" 3
find	sucht nach Dateinamen Beispiel: suche nach Namen: find . -name filename -print



	Dieser Befehl hat viele verschiedene Möglichkeiten und Optionen. Es ist leider zu viel, um sie alle in diesem Artikel zu erklären.
tee	schreibt Daten nach stdout (deinen Bildschirm) und in eine Datei Normalerweise wie folgt benutzt: somecommand   tee outfile Es schreibt die Ausgabe von somecommand auf den Bildschirm und in die Datei outfile
basename file	gibt nur den Dateinamen eines gegebenen Namens zurück und schneidet den Verzeichnispfad ab Beispiel: basename /bin/tux gibt nur tux zurück
dirname file	gibt nur das Verzeichnis eines gegebenen Namens zurück und schneidet den tatsächlichen Dateinamen ab Beispiel: dirname /bin/tux gibt nur /bin zurück
head file	druckt einige Zeilen vom Dateianfang
tail file	druckt einige Zeilen vom Dateiende
sed	sed ist grundsätzlich ein finde und ersetze Programm. Es liest Text von der Standardeingabe (z.B. von einer pipe) und schreibt das Ergebnis in stdout (normalerweise der Bildschirm). Das Suchmuster ist ein regulärer Ausdruck (siehe Referenzen). Diese Suchmuster dürfen nicht mit der Shellwildcardsyntax verwechselt werden. Um die Zeichenkette linuxfocus mit LinuxFocus in einer Textdatei zu ersetzen, benutze: cat text.file   sed 's/linuxfocus/LinuxFocus/' > newtext.file Dies ersetzt das erste Auftreten der Zeichenkette linuxfocus in jeder Zeile mit LinuxFocus. Wenn es Zeilen gibt, in denen linuxfocus mehrmals vorkommt und man alle ersetzen will, schreibt man: cat text.file   sed 's/linuxfocus/LinuxFocus/g' > newtext.file
awk	Meistens wird awk benutzt, um Felder aus einer Textzeile herauszuziehen. Der Standard-Feldtrenner ist ein Leerzeichen. Um einen davon verschiedenen zu spezifizieren, benutzt man die Option -F. cat file.txt   awk -F, '{print \$1 "," \$3 }'  Hier benutzen wir ein Komma (,) als Feldtrenner und drucken die erste und dritte (\$1 \$3) Spalte. Wenn file.txt Zeilen hat, wie die folgenden: Adam Bor, 34, India Kerry Miller, 22, USA  dann ergibt dies: Adam Bor, India Kerry Miller, USA



Es gibt sehr viel mehr, was man mit awk tun kann, aber dies ist eine sehr oft vorkommende Anwendung.

### 7.10 Wichtige Verzeichnisse:

Verzeichnisse und Dateien, die mit einem „.“ beginnen, sind versteckte Dateien.

VERZEICHNIS	BEDEUTUNG
/boot	Bootdateien, Kernelimage
/media/cdrom	Vorbereitet für das Mounten von CD's
/etc	Konfigurationsdateien
/etc/httpd	Konfigurationen für Apache
/etc/mail	Konfigurationen für Sendmail
/etc/sysconfig	Weitere Konfigurationsdateien (Firewall,...)
/media/floppy	Vorbereitet für das Mounten von Disketten
/home	Arbeitsverzeichnisse der Benutzer
/lib/modules/2.xxxx	Ladbare Module, wobei xxxx für die Kernelversion steht.
/root	Arbeitsverzeichnis des Root-Benutzers
/etc/init.d	Startskripte; Darunter die Links für die einzelnen Runlevels. Vor 7.1 findet man diese Dateien unter /sbin/init.d
/usr/bin	Betriebssystembefehle (für alle Benutzer)
/usr/share/doc/packages	Dokumentationen zu installierten Paketen
/srv/ftp	ftp-Server (anonymous-ftp)
/srv/www	WWW-Server
/usr/local/nwe	Root-Verzeichnis für Novellserver Emulation
/usr/src/linux	Quellen des Kernels
/var/lib/mysql	Datenbanken des SQL-Servers
/var/lib/named	Konfigurationsdateien des Domain Name Servers
/var/spool/mail	Mailverzeichnis (Eingehend)
/var/spool/mqueue	Mailverzeichnis (Ausgehend)
/var/squid/cache	Cache-Verzeichnisse des Proxyserver



**7.11 Wichtige Dateien im Verzeichnis /etc und deren Bedeutung**

DATEI	BEDEUTUNG
modules.conf	Optionseinstellung für das Laden von Modulen: z.B: Netzwerkkarten.
crontab	' cron.... Konfigurationsdateien für automatisches Starten von Diensten Zugeordnete Verzeichnisse: Cron.daily Cron.weekly Cron.monthly
default	Verzeichnis für Default Einstellungen: z.B Anlegen neuer Benutzer
dhcpd.conf	Konfigurationsdatei für den DHCP-Server
exports	Dateisysteme die exportiert werden (via NFS - UNIX Netzwerk)
ftphosts	Einschränkungen (Hostseitig) von wo aus FTP-Zugriff erfolgen kann.
fstab	Wie werden die Dateisysteme gemountet.....
ftpaccess	Einstellungen für den FTP - Server
ftpusers	Diese Benutzer dürfen nicht mittels FTP auf den Server zugreifen
group	Benutzergruppen
host.conf	Reihenfolge der Dienste für Namensauflösung (hosts, bind)
HOSTNAME	Definiert den primären Namen des Rechners
hosts	Liste mit einer Übersetzung zwischen Name und IP - Adr.
hosts.allow	hosts... Dateien in denen der Zugriff von außen auf diesen Rechner
hosts.deny	eingeschränkt werden kann
hosts.equiv	
hosts.lpd	
httpd	in diesem Verzeichnis befinden sich die Konfigurationsdateien für den Web - Server
inetd.conf	Steuert einige wichtige Dienste: z.B ftp, telnet, Pop3 (Mail),
inittab	steuert z.B. den RunLevel des Systems
isapnp.conf	Konfiguration von ISA - PNP Karten (wird z.B mit dem Befehl: pnpdump >/etc/isapnp.conf erzeugt)
issue	Begrüßungstext für Login
issue.net	Begrüßungstext für Login aus dem Netz (Telnet, )
lilo.conf	Konfigurationsdatei für LiLo (LinuxLoader) (Bearbeiten über YAST)
login.defs	Login - Einstellungen



<b>mail/aliases</b>	Mailaliases (Textdatei)
mail/aliases.db	Mailaliases (Datenbank wird mit »newaliases« aus der Textdatei erzeugt )
motd	Tagestext (Anzeige nach LOGIN)
mtab	gemountete Filesysteme
named.conf	Konfigurationsdatei für Nameserver
networks	Netzwerkdefinitionen (Subnetze)
passwd	Definifion der Benutzer des Rechners (Textdatei)
permissions	Sicherheitseinstellungen für den Rechner
permissions.easy	(kann über Yast konfiguriert werden)
permissions.local	
permissions.paranoi d	
permissions.secure	
printcap	Druckerdefinitionen (über Yast erstellen)
profile	Systemweite Einstellungsdatei (Suchpfade, Zeitzone,..), kann durch zusätzliche Benutzereinstellungsdateien (\$home/.bashrc) ergänzt werden
rc.config	Globale Einstellungen des Rechners (kann über yast od. einen Texteditor bearbeitet werden)
rc.config.d	Verzeichnis mit weiteren Ergänzungen zur Systemdatei rc.config (z.B.: firewall.rc.config, sendmail.rc.config)
route.conf	Einstellungen für Routing (Weiterleiten von Paketen)
sendmail.cf	Konfigurationsdatei des Mailservers (Sendmail) Auswahl der passenden Konfiguration über Yast
services	Übersicht Portnummer – Protokoll
shadow	Speichert die Paßwörter der einzelnen Benutzer (verschlüsselt)
shells	Auflistung der gültigen Kommandoshells
skel	Verzeichnis mit Dateien, die beim Anlegen des Benutzers automatisch in dessen Homeverzeichnis kopiert werden.
smb.conf	Konfigurationsdatei für Samba
smbpasswd	Speichert die PW der Samba – Benutzer (Windows – Clients)
squid/squid.conf	Konfigurationsdatei für SQUID (Proxy – Server)
ssh	Verzeichnis mit Konfigurationsdateien für die SecureShell

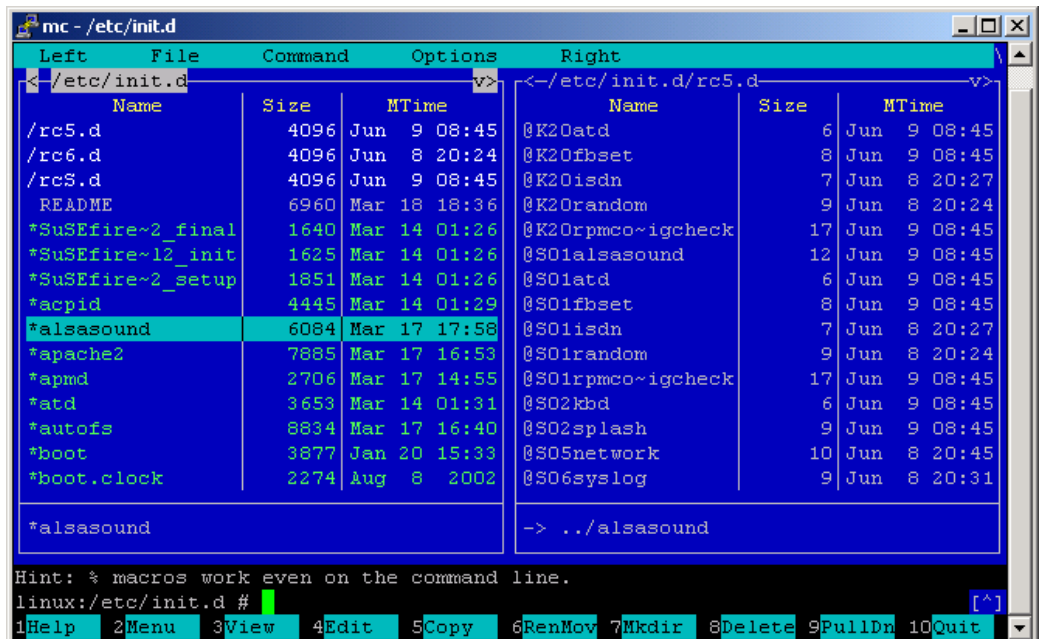


## 7.12 Der Startvorgang im Detail:

Wie bereits erwähnt unterscheidet Linux mehrere Betriebsstufen. Welche Dienste in welchem Level gestartet werden, ist relativ einfach nachzuvollziehen:

Im Verzeichnis /etc/init.d befinden sich die Startskripte für die unterschiedlichsten Dienste am Rechner. In den meisten Fällen wird am Beginn des Skriptes eine Systemvariable auf ihren derzeitigen Wert abgefragt. (z.B: START\_HTTPD).

Wenn diese Variable z.B. auf „no“ gesetzt wurde wird das Skript beendet. Im



anderen Fall wird der entsprechende Dienst gestartet.

Für die einzelnen Runlevel befinden sich nun Unterverzeichnisse in /etc/init.d. Wenn nun z.B. im Ordner rc3.d ein Link auf das Skript „apache“ gelegt wird, wird der Dienst gestartet sonst nicht. Im Allgemeinen findet man jeden Link jedoch 2-mal. Einmal mit S.... und einmal mit K.... . Einer dieser Links wird ausgeführt beim Wechsel in diesen Runlevel (Start....) und der andere beim Verlassen des Runlevels (Kill....)

Im Unterschied zu den Vorversionen, wo die Links grundsätzlich vorhanden sind und der Wert einer Startvariable (definiert über rc.config) überprüft wird, werden die Links für die notwendigen Runlevel unter SuSE 8.0 angelegt bzw. entfernt. Dazu sollte man das Modul Runlevel-Editor aus yast2 verwenden.

## 8 ZUGRIFF AUF ANDERE DATEISYSTEME:

### 8.1 LINUX-DATEIZUGRIFF auf NOVELLSERVER

Sie können auch vorhandene Novell - Filesysteme in ihren Linux - Verzeichnisbaum einbinden. Dies erfolgt mit einem ncpmount-Befehl (enthalten im Pakete ncpfs)

ncpmount -S server -U user -P password -V volume MOUNTPOINT

z.B:

ncpmount -S fs411 -U guest -V vol1 /novell

Bevor sie auf ein Novell-Dateisystem zugreifen können müssen sie noch die IPX-Unterstützung im Kernel aktivieren. Mit dem Befehl

**ipx\_configure - -auto\_interface=on - -auto\_primary=on**

wird das ipx-Routing richtig eingestellt. Danach sollten sie eine Verbindung zum Novellserver herstellen können.



## **8.2 WINDOWSDRUCKER von LINUX aus ansprechen**

Für diese Option finden sie in YAST unter „Administration“ einen Punkt mit dem Namen Drucker im Windows-Netzwerk ansprechen. In der darauf folgenden Eingabemaske können sie nun die Druckerverbindung definieren. Sie erhalten beim Beenden der Maske einige Informationen, welche Drucker das System für sie angelegt hat.

Von Linux aus können sie mit dem Befehl „lpr“ Druckbefehle auslösen und so die Funktion des soeben definierten Netzwerkdruckers testen.

## **8.3 ZUGRIFF auf freigegebene Ordner von Windowsrechnern**

Sie können mittels SAMBA nicht nur Ordner für Windowsrechner freigeben, sondern auch in umgekehrter Richtung auf (von Workstation freigegebene) Ordner zugreifen. Wenn sie z.B. in ihrer Arbeitsgruppe einen Rechner mit Netbiosnamen „PC“ und einem freigegebenen Verzeichnis mit der Bezeichnung „C\_PLATTE“ besitzen, können sie mit einem smbmount-Befehl diesen Ordner in ihren Verzeichnisbaum mounten:

```
smbmount //PC/C_PLATTE /windowsrechner
```

(Der Mountpoint /windowsrechner muss vorher bereits erzeugt worden sein)

Wenn sie das Einbinden vereinfachen wollen können sie die passenden Veränderungen in der Datei /etc/fstab vornehmen: z.B :

```
//PC/C_PLATTE /windowsrechner smbfs noauto,user 0 0
```

(Der Parameter noauto verhindert, dass der Ordner automatisch (beim Systemstart) aktiviert wird)

## **9 KONFIGURATION VON NETZWERKDIENTSTEN:**

### **9.1 Fernwartung:**

Der Linuxrechner erlaubt ein Login auch von nicht-lokalen Konsolen. Stellvertretend seien hier 4 Möglichkeiten dafür genannt:

#### **9.1.1 TELNET:**

Wenn der Telnet-Dämon (Teil des inetd; inetd.conf bzw. neuerdings xinetd) gestartet ist, wartet er am Port 23 (nachzulesen in der Datei /etc/services) auf Anmeldungen via Netzwerk. Aus Sicherheitsgründen ist es dem Root-Benutzer nicht erlaubt direkt eine Telnetverbindung herzustellen. Um trotzdem Wartungsarbeiten durchführen zu können kann man folgende Lösung praktizieren:

Mit telnet a.b.c.d öffnet man eine Telnet-Verbindung (z.B. von einer Windows-WS aus) und loggt sich unter einer gültigen Benutzerkennung ein. Mit dem Befehl „su root“ (SwitchUser) wechselt man in die Root-Identität (Root-PW erforderlich!!) und kann nun alle Wartungsarbeiten am System durchführen. Der Nachteil der Wartung mittels Telnet ist, dass die Daten (auch die Passwörter!!) unverschlüsselt übers Netz (Internet) verschickt werden und somit einem eventuellen Mitlauscher direkt zur Verfügung stehen. Günstiger ist es daher, die Wartung über eine SecureShell (ssh) durchzuführen.



### 9.1.2 SSH (Secure Shell):

Hier werden die Daten mittels RSA-Algorithmus verschlüsselt. Der einzig unsichere Moment ist das Übertragen des Schlüssels (wenn dies über Internet geschieht). Wenn man das nicht will, kann man mittels Diskette, Abschreiben,.. den Public-Key des Fernzuwartenden Rechners in z.B: die Datei /root/.ssh/known\_hosts eintragen. (Wenn die Wartungsarbeit am anderen Rechner auch als Root-Benutzer durchgeführt wird). Die Konfiguration des SecureShellDämons (sshd) erfolgt über die Datei /etc/ssh/sshd\_config. Es empfiehlt sich (aus Sicherheitsgründen) über den Parameter Protocol nur die Version 2 des ssh-Protokolls zuzulassen.

Als Client-Programm kann man von einem Linuxrechner aus die Kommandozeilenuitility ssh verwenden. Von Windowsrechnern aus bietet sich das Programm putty an, das man z.B von [www.tucows.at](http://www.tucows.at) downloaden kann.

**ACHTUNG: Bei SuSE 9.2 ist SSH bei Standardinstallation nicht aktiv!!**

### 9.1.3 FISH:

Von einem Linux-Desktop mittels Web\_Browser Konqueror: in die Adresszeile folgende URL eingeben: fish://ip\_server Damit erfolgt eine SSH (bzw. SCP) Zugriff auf den Rechner ip\_server. Über den Browser kann man nun auf Dateien des entfernten Rechners zugreifen und diese direkt editieren und verändern.

### 9.1.4 Wartung mit WEBMIN:

Das Tool Webmin (<http://www.webmin.com>) ermöglicht eine weitestgehende Wartung des Rechners über einen Webbrowser. Dazu kann man das Paket als \*.rpm – Paket herunterladen und via yast installieren. Über dieses Tool kann das System und nahezu alle Serverdienste gesteuert werden. Es ermöglicht außerdem das Anlegen einer Benutzerstruktur für die Webmin – Oberfläche.

## 9.2 FTP-Server:

Standard FTP-Server ist der Dienst vsftpd, der über xinetd gestartet wird. Die Konfiguration dieses Dienstes erfolgt (etwas einfacher) über die Konfigurationsdatei /etc/vsftpd.conf. Eine genauere Beschreibung der Parameter erhält man mit „man vsftpd.conf“

```
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are very paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
#
# Allow anonymous FTP?
#anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
```



```
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
# Make sure PORT transfer connections originate from port 20 (ftp-data).
connect_from_port_20=YES
#
# If you want, you can arrange for uploaded anonymous files to be owned by
# a different user. Note! Using "root" for uploaded files is not
# recommended!
#chown_uploads=YES
#chown_username=whoever
#
# You may override where the log file goes if you like. The default is shown
# below.
#xferlog_file=/var/log/vsftpd.log
#
# If you want, you can have your log file in standard ftpd xferlog format
#xferlog_std_format=YES
#
# You may change the default value for timing out an idle session.
#idle_session_timeout=600
#
# You may change the default value for timing out a data connection.
#data_connection_timeout=120
#
# It is recommended that you define on your system a unique user which the
# ftp server can use as a totally isolated and unprivileged user.
#nopriv_user=ftpsecure
#
# Enable this and the server will recognise asynchronous ABOR requests. Not
# recommended for security (the code is non-trivial). Not enabling it,
# however, may confuse older FTP clients.
#async_abor_enable=YES
#
# By default the server will pretend to allow ASCII mode but in fact ignore
# the request. Turn on the below options to have the server actually do ASCII
# mangling on files when in ASCII mode.
# Beware that turning on ascii_download_enable enables malicious remote parties
```



```
# to consume your I/O resources, by issuing the command "SIZE /big/file" in
# ASCII mode.
# These ASCII options are split into upload and download because you may wish
# to enable ASCII uploads (to prevent uploaded scripts etc. from breaking),
# without the DoS risk of SIZE and ASCII downloads. ASCII mangling should be
# on the client anyway..
#ascii_upload_enable=YES
#ascii_download_enable=YES
#
# You may fully customise the login banner string:
#ftpd_banner=Welcome to blah FTP service.
#
# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd.banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd.chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES

pam_service_name=vsftpd
chroot_local_user=YES
```

### 9.3 Web-Server:

Bei einer Standardinstallation wird der Webserver APACHE automatisch mitinstalliert und auch bereits gestartet (Einstellung über START\_HTTPD – Variable in rc.config). Bei konfigurierterem Netzwerk können sie von einer Workstation bereits mit einem Webbrowser unter <http://a.b.c.d> auf den Webserver des Linuxrechners zugreifen, wenn a.b.c.d stellvertretend für eine der Adressen des Linuxrechners steht. Wenn für diese Adresse bereits ein Rechner/Domainname vergeben ist und auch im entsprechenden Nameserver eingetragen wurde, kann der Zugriff auch über den Namen erfolgen.

Die Konfigurationsdatei des Servers ist /etc/httpd/httpd.conf.

**Das Root-Verzeichnis des Webservers ist nun standardmäßig auf /srv/www** (zuvor: /usr/local/httpd) eingestellt. Hier findet man u.a. 2 wichtige Verzeichnisse:

#### cgi-bin:

Enthält ausführbare Dateien (\*.cgi). In der Konfigurationsdatei wird dieses Verzeichnis als Script-Verzeichnis definiert, wodurch ein Ausführen von



Programmen via WEB in diesem Verzeichnis erlaubt wird. Wenn man in einem anderen Verzeichnis auch das Ausführen von Scripts erlauben will, muss die Option ExecCGI für diesen Ordner eingeschaltet werden.

**htdocs:**

Das Wurzelverzeichnis für HTML-Dokumente. Beim Zugriff auf den Webserver wird zunächst nach der Datei index.html gesucht. Wenn Sie nun ihre eigene Homepage gestalten wollen, überspielen sie ihre Homepage (mit allen zugehörigen Dateien und Verzeichnissen) ins Verzeichnis /srv/www/htdocs, wobei die Startseite ihrer Homepage index.html heißen sollte. Benennen Sie vorher die bereits vorhandene Datei index.html z.B. in suse.html um. Damit können sie mit <http://a.b.c.d/suse.html> jederzeit auf die Suse-Dokumentation zugreifen.

Beachten Sie, dass alle Dateien ihrer Homepage mit dem Recht 644 (lesen für „others“) versehen werden und alle Verzeichnisse das Recht 755 (Ausführen/lesen für „others“) besitzen.

Es empfiehlt sich für die Wartung der Homepage einen eigenen Benutzer anzulegen, ihn als Besitzer aller Dateien unter /srv/www festzulegen. Mittels ftp-Zugriff kann dieser Benutzer dann die Homepage von anderen Rechnern aus warten.

Weitere Tipps und Tricks zu den Wartungsaufgaben finden sie am PI-Webserver (<http://www.pinoe-hl.ac.at> ) unter AG-Informatik  Anbindung eines Netzes ans Internet  WWW-Server Apache.

**9.3.1 Verwendung von PHP:**

Das folgende Beispiel demonstriert die Verwendung von PHP in HTML Seiten. Damit die Seite richtig interpretiert wird muss die Datei die Endung .php besitzen. Damit diese Datei (index.php) vor einer eventuell vorhandenen Datei index.html verwendet wird, muss der Parameter DirectoryIndex in der Datei /etc/httpd/httpd.conf angepasst werden. Für die Funktion der nachfolgenden Datei, die einen Zähler realisiert, muss im selben Verzeichnis eine Datei counter.txt existieren, die für die Gruppe others auch Schreibrechte hat.

```
<html>
<head>
  <title>Homepage von USER</title>
  <meta content="">
  <style></style>
</head>
<body>
Dies ist eine Testseite für Benutzerhomepages <br>
<?PHP
  $fp = fopen("counter.txt","r");
  $zahl =(int) fgets($fp,10);
  fclose($fp);
  $zahl++;
```



```

echo "Sie sind der ".$zahl.". Besucher"; // Strings werden mit
zusammengefügt
$fp = fopen("counter.txt","w");
fputs($fp,$zahl);
fclose($fp);
?>
</body>
</html>

```

### 9.3.2 INTRANET

Das folgende Beispiel zeigt ein Verzeichnis, das nur aus dem internen Netz betrachtet werden, oder von außerhalb, wenn man gültige Benutzerkennungen hat. Wenn der Parameter „Satisfy any“ auf „Satisfy all“ ändert ist ein Betrachten nur dann möglich, wenn beide Bedingungen erfüllt werden.

```

<Directory /srv/www/htdocs/intra>
AuthType Basic
AuthName intranet
AuthUserFile /etc/httpd/users
AuthGroupFile /etc/httpd/groups
Require group customers
Order allow,deny
Allow from 10.0
Satisfy any // Standardeinstellung ist Satisfy all
</Directory>

```

### 9.3.3 Webseiten für Benutzer:

Sie können es ihren Benutzern am Linuxrechner gestatten sich eigenständig Webpages zu gestalten. Der Zugriff auf diese Seiten erfolgt mit <http://a.b.c.d/~USER>, wenn USER der Name des Benutzers ist. Dazu müssen sie in der Konfigurationsdatei den entsprechenden Teil für das Verzeichnis /home/\*/public\_html aktivieren, wobei sie keine so komplizierte Definition wie im Beispiel benötigen.

**Unter SuSE 9.2 sind die Benutzerhomepages automatisch aktiviert (das Modul userdir wird automatisch geladen) .**

(Die entsprechende Datei heißt nun mod\_userdir.conf im Verzeichnis /etc/apache2)

Es genügt z.B.:

```

<directory /home/*/public_html>
Options Indexes
Order allow,deny
Allow from all
</directory>

```

### **Aufgaben für den jeweiligen Benutzer:**



Er muß der Gruppe others das Execute-Recht für sein Homeverzeichnis einräumen (Read ist nicht notwendig) und dies ebenfalls für das anzulegende Verzeichnis public\_html tun. Für die Zugriffsrechte auf die Dateien unter public\_html gilt das Gleiche wie oben für die Homepage erwähnt.

(Dieses Verzeichnis wird in den neueren Versionen automatisch beim Anlegen eines Benutzers in sein Homeverzeichnis kopiert)

### 9.3.4 Zugriff auf bestimmte Webseiten mittels Authentifizierung:

Man kann einzelne Verzeichnisse am Webserver sperren, sodass ein Zugriff nur mit gültiger Benutzerauthentifizierung möglich ist. (Siehe Dokumentation PI-Server  Web-Zugriff auf Verzeichnisse sperren)

Dazu muss eine eigene Passwortdatei angelegt werden:

z.B.: mit dem Befehl: "htpasswd -c /usr/local/httpd/allowed.users lehrer" eine Passwortdatei z. B. für den Benutzer **lehrer** anlegen. Danach erfolgt eine zweimalige Abfrage eines Passwortes. Dieses Passwort wird dann verschlüsselt in der Datei **allowed.users** abgespeichert.

In dem Verzeichnis, das gesperrt werden soll (z. B. **/usr/local/httpd/htdocs/geheim**) wird mit einem Editor eine Datei mit dem Namen **.htaccess** angelegt. In diese Datei wird folgendes geschrieben:

```
AuthName "Verzeichnis geheim"
AuthType Basic
AuthUserFile /usr/local/httpd/allowed.users
require user lehrer
```

(oder require valid-user  Zugriff kann mit jeder gültigen Benutzererkennung erfolgen)

Bei den Standardeinstellungen des Apache-Servers werden die Einstellungen der Datei **.htaccess** ignoriert. Es muss daher für das Verzeichnis **/usr/local/httpd/htdocs/geheim** diese Möglichkeit aktiviert werden. Dazu editiert man die Datei **/etc/httpd/httpd.conf** und fügt für dieses Verzeichnis eine eigene Definition ein:

```
<Directory "/usr/local/httpd/htdocs/geheim">
AllowOverride All
</Directory>
```

Wenn man weitere gültige Benutzerkennungen zur Passwortdatei hinzufügen will, so kann dies mit dem Befehl

"htpasswd /usr/local/httpd/allowed.users NocheinBenutzer" (d.h. ohne -c) erfolgen.

### 9.3.5 Virtuelle Server:

Es ist auch möglich auf einem Rechner mehrere Webserver zu betreiben. Dazu ist es notwendig das DNS-Service entsprechend einzurichten (NickNames für den Rechner



eintragen) und es Apache mitzuteilen wo die Dokumente für diesen virtuellen Server liegen. Für das angeführte Beispiel war es z.B. notwendig im DNS-Server für den Novellserver (IP 10.0.1.2) die Synonyme student.brg-wrn.ac.at, sta.brg-wrn.ac.at, al.brg-wrn.ac.at,..... einzutragen. Die Konfigurationsdatei für APACHE unter Novell befindet sich (bei installiertem Webserver) im Verzeichnis sys:\apache\conf und hat die gleiche Syntax wie unter Linux (bis auf die Verzeichnisangaben).

**Ergänzung:** Unter Novell wird der Server mit load sys:\apache\apache2 gestartet.

HINWEIS: auch für den bereits laufenden (Standard-) Server ist eine virt. Hostdefinition erforderlich. Sie sollte an erster Stelle stehen, damit dieser Server zum Default Server wird. (d.h. alle Anfragen die nicht exakt einem der nachfolgenden virt. Hostnamen entsprechen werden auf den ersten Server geleitet.

NameVirtualHost 10.0.1.2 //Unter welcher IP laufen die virt. Server  
 // mit der Anweisung NameVirtualHost 10.0.1.2:5045 wäre es z.B. möglich die virt.  
 // Server auf einem anderen Port laufen zu lassen.

```
<VirtualHost 10.0.1.2>
  ServerAdmin sta@brg-wrn.ac.at
  DocumentRoot sys:/apache/student
  ServerName student.brg-wrn.ac.at
  <Directory home:/schueler/*/public.www>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
      Order allow,deny
      Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS PROPFIND>
      Order deny,allow
      Deny from all
    </LimitExcept>
  </Directory>
</VirtualHost>
```

```
<VirtualHost 10.0.1.2>
  ServerAdmin sta@brg-wrn.ac.at
  DocumentRoot home:/lehrer/sta/public.www
  ServerName sta.brg-wrn.ac.at
</VirtualHost>
```

```
<VirtualHost 10.0.1.2>
  ServerAdmin sta@brg-wrn.ac.at
  DocumentRoot home:/lehrer/al/public.www
  ServerName al.brg-wrn.ac.at
</VirtualHost>
```

Anstelle der IP-Adresse (10.0.1.2) kann auch ein \* eingegeben werden. Damit akzeptiert der Webserver Anfragen auf allen angeschlossenen Netzwerkkarten. Das obige Beispiel zeigt sog. „NameBasedVirtualHosts“. Alternativ können auch IP-Basierende virtuelle Hosts verwendet werden. Dazu ist es jedoch notwendig für jeden virtuelle Server eine eigene IP-Adresse zu besitzen. (Konfiguration siehe Apache-Doku od. SuSE Handbuch)



### 9.3.6 HTTPS – Verschlüsselte Verbindungen:

Mit dem Apacheserver können auch virtuelle Hosts so konfiguriert werden, dass man nur mittels SSL-Zertifikaten zugreifen kann. Dazu findet man Dokumentationen am System selbst (apache-doc) und im Internet. Im Verzeichnis /etc/apache2/vhosts.d findet man eine Beispielkonfiguration für einen virtuellen Host mit https. Dieses Muster passt man sich bei der ersten Verwendung am besten an die eigenen Bedürfnisse an. Zu Beachten ist:

a.) Erzeugen eines Zertifikates: mit /usr/bin/gensslcert

- C Common name "\$name"
- N comment "\$comment"
- c country (two letters, e.g. DE) \$C
- s state \$ST
- l city \$L
- o organisation "\$O"
- u organisational unit "\$U"
- n fully qualified domain name \$CN (\\$FQHOSTNAME)
- e email address of webmaster webmaster@\$CN
- y days server cert is valid for \$srvdays
- Y days CA cert is valid for \$CAdays
- d run in debug mode
- h show usage

For example:

```
/usr/bin/gensslcert -c US -s TN -l "Oak Ridge" -o Your_organization -e  
your_name@your_isp.com -d -n 192.168.1.10
```

b.) Neustarten des Webservers mit der Option SSL!!

(in /etc/sysconfig/apache2 wird die Variable „APACHE2\_OPTS“ auf -D SSL gesetzt)

### 9.4 Benutzerplatzbeschränkung:

Bei einer größeren Anzahl von Systembenutzern ist es notwendig, dass der, den Benutzern zur Verfügung stehende Plattenplatz beschränkt wird. Dies kann mit den sog. QUOTAS erfolgen.

Wenn sie das Paket QUOTA (Serie ap1) installiert haben, müssen die Beschränkungen noch konfiguriert und aktiviert werden. Dazu legen sie im Wurzelverzeichnis des Dateisystems, auf dem sie Beschränkungen verwenden wollen die Datei "quota.user" an. Für die Standardinstallation in NÖ ist dies eigentlich nur für die Root-Partition sinnvoll und kann z.B. mit dem Befehl "touch /aquota.user" durchgeführt werden (als root-Benutzer). Weisen sie dieser Datei mit "chmod 600 aquota.user" die passenden Filerechte zu.

Editieren sie nun die Datei "/etc/fstab". In dieser Datei finden sie eine Zeile, in der die Optionen für das Mouneten der Root-Partition stehen: (Quotas sind nur in z.B:



**/dev/hda2 / ext2 defaults, usrquota .....**

Fügen sie in dieser Zeile die Option `usrquota` hinzu. Diese Option aktiviert die Quotas auf der entsprechenden Partition.

Speichern sie die Änderungen ab und editieren sie danach die Datei `/etc/rc.config`. Setzen sie in dieser Datei den Parameter `START_QUOTA` auf `yes`. Jetzt müssen die bereits vorhandenen Dateien ihren Besitzern zugeordnet werden. Die erste Initialisierung wird mit dem Befehl `quotacheck -acuvqm` durchgeführt, danach starten sie den Rechner neu.

Nach dem neuerlichen Start des Rechners werden die Quotas aktiviert. Mit dem Befehl `quota Benutzername` können die Quotas eines Benutzers eingesehen werden. Zum Einstellen von Beschränkungen kann der Befehl `edquota Benutzername` verwendet werden. Sie können im Editor (`vi`) nun für den jeweiligen Benutzer ein Softlimit (darf einige Zeit überschritten werden) und ein Hardlimit einstellen. (Im `vi` speichert man mit `":w"` ; Verlassen: `":q"`). Für eine größere Anzahl verwendet man einen Beispielbenutzer und macht die anderen äquivalent zu diesem. (kann mit `edquota` gemacht werden)

Hilfe erhalten sie über die entsprechende Manpage. Quotas können übrigens auch mit den `PSNTOOLS` eingestellt werden.

Eine einfache Überprüfung und Einstellung der Quotas kann auch über das `WEBMIN` – Interface erfolgen.

## 9.5 WARTUNG VON BENUTZERDATEN

### 9.5.1 Yast

Über die Systemverwaltung mittels `YAST` (Textmodus oder Grafische Oberfläche) können über das Modul Sicherheit und Benutzer bestehende Benutzer und Gruppen verändert, oder neue angelegt werden.

### 9.5.2 Kommandozeile

<code>useradd</code>	Neuen Benutzer anlegen
<code>usermod</code>	Existierenden Benutzer ändern
<code>groupadd</code>	Gruppe anlegen
<code>groupmod</code>	Existierende Gruppe ändern

### 9.5.3 WEBMIN-USERINTERFACE

Man kann auch über das `WEBMIN`-Interface Benutzer aus eine Steuerdatei anlegen und löschen. Diese Variante ist durch die unsaubere Lösung der `PSN-Tools` bei neueren Distributionen vorzuziehen.

`create:username:passwd:uid:gid:realname:homedir:shell:min:max:warn:inactive:expire`

`modify:oldusername:username:passwd:uid:gid:realname:homedir:shell:min:max:warn:inactive:expire`



`delete:username`

*In create lines, if the uid field is left empty, Webmin will assign a UID automatically. If the gid field is empty, Webmin will create a new group with the same name as the user. The username, homedir and shell fields must be supplied for every user - all other fields are allowed to be empty. If the passwd field is blank, no password will be assigned for the user. If it contains just the letter x, the account will be locked. Otherwise, the text in the field will be taken as the cleartext password and encrypted. In modify lines, an empty field will be taken to mean that the corresponding user attribute is not to be modified.*

## **10 Prozessmanagement**

Das gesamte Management der Prozesse unterliegt dem Kernel. Er verteilt die Rechenzeit, überwacht und ermöglicht die saubere Kommunikation unter den Prozessen und kontrolliert die Rechte. Linux ist ein Multitasking-System, was bedeutet, dass mehrere Prozesse gleichzeitig abgearbeitet werden können. Dies kann aber nur auf Mehrprozessormaschinen wirklich erfolgen, auf Single-Prozessorsystemen wird ein System benötigt, das dem Anwender scheinbar die parallele Abarbeitung der Prozesse vorspielt. Hier kommt der Prozess-Scheduler zum Einsatz. Er verteilt die Rechenzeit und Systemressourcen auf die einzelnen Prozesse für ein bestimmtes Zeitintervall.

Der Mehrprogrammbetrieb steigert den Informationsdurchfluss der Hardware. Stoppt z.B. Programm n, weil es einen E/A-Kanal aufruft, so verarbeitet der Prozessor das bereitstehende Programm m, bis dieses einen Stop erreicht usw. Ein Zuteilungsprogramm legt zuvor die Prioritäten fest.

Steht jedem im Hauptspeicher befindlichen Programm eine feste Zeitspanne zu, so spricht man von Zeitscheibenverfahren (engl.: time slicing)." [[Breuer](#), Seite 197]

Welcher Prozess als nächster für die Bearbeitung durch die CPU ausgewählt wird, errechnet der Prozess-Scheduler mit Hilfe von Prioritätsparametern, Prozesszustand und Wartezeit. Auf die Priorität von Prozessen kann auch direkt Einfluss genommen werden über den Befehl nice. Normale Anwender können Prozesse herunterstufen, allein dem Systemadministrator root ist das Recht vorbehalten, Prozesse heraufzustufen. Der Aufruf erfolgt über:

```
nice -<nice-Wert> <Kommandofolge>
```

Für den nice-Wert kommt ein Zahlenbereich von -20 bis 20 in Frage, wobei ein niedriger nice-Wert eine höhere Priorität bedeutet. Der in der Syntax angegebene Bindestrich gehört zum Befehl dazu, so dass bei heraufsetzen der Priorität zwei Bindestriche einzugeben sind. Nachträglich, wenn der Prozess schon gestartet ist, kann die Priorität mit dem Befehl renice geändert werden. Die Syntax hierfür lautet:

```
renice -<nice-Wert> <PID>
```

Der PID ist eine eindeutige Identifizierung eines Prozesses (Prozess Identifier). Jedem Prozess wird eine eindeutige Nummer zugewiesen. Wie schon in Kapitel [erwähnt](#), hat der Vater aller Prozesse, der init-Prozess, die PID 1.

Ein Prozess besteht immer aus einem Textsegment (Befehle), einem Datensegment und einem Stack-Bereich. Wird nun ein Befehl oder Kommando zwei oder mehrfach gestartet, so wird das Textsegment nicht noch einmal geladen, sondern im Speicher



nur Bereiche für das Datensegment und den Stack-Bereich reserviert. Somit können die Speicherressourcen geschont werden.

Im Verzeichnis /proc befindet sich ein Abbild aller Prozesse sowie alle Informationen des Kernels. Es ist ein spezielles Dateisystem mit keinen echten Dateien und Verzeichnissen, sondern bildet die Schnittstelle zum Kernel. Hier kann der direkte Zugriff auf Informationen des Kernels realisiert werden. Bei Arbeiten in diesem Verzeichnis ist größte Sorgfalt angebracht, da Änderungen an den Dateien direkt den Kernel betreffen und das System in einen instabilen Zustand versetzen können, bis zum Absturz des Systems. Trotz allem ist dieses Verzeichnis eine der wichtigsten Informationsquellen zum System. Im nachfolgenden Listing ist ein Auszug aus dem Verzeichnis /proc wiedergegeben:

```

dr-xr-xr-x 3 root  root          0 Jul 10 08:51 bus
-r-r-r-  1 root  root          0 Jul 10 08:51 cmdline
-r-r-r-  1 root  root          0 Jul 10 08:51 cpuinfo
-r-r-r-  1 root  root          0 Jul 10 08:51 devices
-r-r-r-  1 root  root          0 Jul 10 08:51 dma
-r-r-r-  1 root  root          0 Jul 10 08:51 filesystems
dr-xr-xr-x 2 root  root          0 Jul 10 08:51 fs
dr-xr-xr-x 4 root  root          0 Jul 10 08:51 ide
-r-r-r-  1 root  root          0 Jul 10 08:51 interrupts
-r-r-r-  1 root  root          0 Jul 10 08:51 ioports
-r----  1 root  root    134221824 Jul 10 08:51 kcore
-r----  1 root  root    134221824 Jul 10 08:51 kcore_elf
-r----  1 root  root          0 Jul 10 08:40 kmsg
-r-r-r-  1 root  root          0 Jul 10 08:51 ksyms
-r-r-r-  1 root  root          0 Jul 10 08:51 loadavg
-r-r-r-  1 root  root          0 Jul 10 08:51 locks
-r-r-r-  1 root  root          0 Jul 10 08:51 meminfo
-r-r-r-  1 root  root          0 Jul 10 08:51 memstat
-r-r-r-  1 root  root          0 Jul 10 08:51 misc
-r-r-r-  1 root  root          0 Jul 10 08:51 modules
-r-r-r-  1 root  root          0 Jul 10 08:51 mounts
dr-xr-xr-x 3 root  root          0 Jul 10 08:40 net
-r-r-r-  1 root  root          0 Jul 10 08:51 partitions
-r-r-r-  1 root  root          0 Jul 10 08:51 pci
dr-xr-xr-x 2 root  root          0 Jul 10 08:51 scsi
-r-r-r-  1 root  root          0 Jul 10 08:51 sound
-r-r-r-  1 root  root          0 Jul 10 08:51 stat
-r-r-r-  1 root  root          0 Jul 10 08:51 swaps
dr-xr-xr-x 9 root  root          0 Jul 10 08:51 sys
dr-xr-xr-x 4 root  root          0 Jul 10 08:51 tty
-r-r-r-  1 root  root          0 Jul 10 08:51 uptime
-r-r-r-  1 root  root          0 Jul 10 08:51 version

```

Zum Beispiel enthält die Datei /proc/cpuinfo Informationen über die CPU, die Datei /proc/modules Informationen, welche Module geladen sind und die Datei /proc/meminfo Informationen über Speicherverbrauch und -ressourcen.



## 10.1 Prozesse anzeigen

Alle laufenden Prozesse können über die Befehle ps, top oder pstree angezeigt werden. Dabei gibt der Aufruf von ps viele wichtige Informationen über das System. Das nachfolgende Listing zeigt eine gekürzte Ausgabe des Befehls ps -aux, die Parameter werden in Kapitel 10 beschrieben.

USER	PID	%CPU	%MEM	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	204 ?	S	12:28	0:04		init [2]
root	2	0.0	0.0	0 ?	SW	12:28	0:00		[kflushd]
root	3	0.0	0.0	0 ?	SW	12:28	0:00		[kupdate]
root	4	0.0	0.0	0 ?	SW	12:28	0:00		[kpiod]
root	5	0.0	0.0	0 ?	SW	12:28	0:00		[kswapd]
bin	94	0.0	0.3	400 ?	S	12:29	0:00		/sbin/portmap
root	101	0.0	0.2	360 ?	S	12:29	0:00		/usr/sbin/scanlogd
root	107	0.0	0.4	632 ?	S	12:29	0:00		/usr/sbin/syslogd
root	185	0.0	0.4	560 ?	S	12:29	0:00		/usr/sbin/inetd
root	200	0.0	0.4	572 ?	S	12:29	0:00		/usr/sbin/lpd
root	231	0.0	0.4	628 ?	S	12:29	0:00		/usr/sbin/cron
nobody	235	0.0	0.5	692 ?	S	12:29	0:00		/usr/sbin/in.identd -e
root	236	0.0	0.5	692 ?	S	12:29	0:00		/usr/sbin/in.identd -e
root	270	0.0	0.7	1016	tty1	S	12:29	0:00	login - root
root	271	0.0	0.7	1016	tty2	S	12:29	0:00	login - tobias
root	272	0.0	0.3	428	tty3	S	12:29	0:00	/sbin/mingetty tty3
root	273	0.0	0.3	428	tty4	S	12:29	0:00	/sbin/mingetty tty4
root	274	0.0	0.3	428	tty5	S	12:29	0:00	/sbin/mingetty tty5
root	275	0.0	0.3	428	tty6	S	12:29	0:00	/sbin/mingetty tty6
root	301	0.0	1.0	1328	tty1	S	12:30	0:00	-bash
tobias	345	0.0	1.0	1328	tty2	S	12:30	0:00	-bash
tobias	2784	0.0	1.5	1545	tty2	R	12:34	0:00	find
root	2785	0.0	0.7	928	pts/1	R	12:40	0:00	ps -aux

Dabei bedeuten die Spalten im einzelnen

[USER:]der aufrufende Benutzer des Prozesses;

[PID:]eindeutige ID eines Prozesses;

[RSS:](Resident Set Size) Größe des Prozesses im Speicher in Kilobyte;

[TTY:]kontrollierender Terminal (ein ? bedeutet, daß diesem Prozess kein Terminal zugeordnet ist);

[Stat:]Laufzeitzustand des Prozesses;

[R:]Prozess läuft im Moment;

[S:]Prozess "schläft", befindet sich also im Wartezustand;

[D:]Prozess "schläft" und kann nicht zwingend unterbrochen werden;

[T:]Prozess ist gestoppt;

[Z:]Prozess ist ein Zombie;

[Time:]Bisher verbrauchte Prozessor-Zeit des Prozesses;

[Command:]Name des Kommandos, mit dem der Prozess erzeugt wurde;

Der Befehl top gibt dieselben Informationen wie der Befehl ps, jedoch werden einige Zusammenfassungen und Gesamtspeicherinformationen zusätzlich angezeigt. pstree



zeigt einen Prozessbaum an, wobei gleiche Kommandos und Befehle zusammengefasst werden und die Anzahl dieser als Multiplikator vorangestellt wird. Das nachfolgende Listing gibt eine gekürzte Ausgabe des Befehls pstree wieder:

```
init-+-atd
  |-cron
  |-httpd--httpd
  |-in.identd--in.identd--5*[in.identd]
  |-inetd
  |-kflushd
  |-klogd
  |-kpiod
  |-kswapd
  |-kupdate
  |-login--bash--pstree
  |-5*[mingetty]
  |-portmap
  |-scanlogd
  ` -syslogd
```

## 10.2 Prozesse abbrechen

Um laufende Prozesse abbrechen gibt es zum einen die Möglichkeit, mit der Tastenkombination STRG-C einen Prozess abbrechen. Leider funktioniert dies nicht immer, besonders, wenn die Prozesse abgestürzt sind oder als Zombies die Systemressourcen verbrauchen. Hier kommen die Kommandos kill und killall zu Hilfe.

### Kill-Signale

1	HUP (hang up)
2	INT (interrupt)
3	QUIT (quit)
6	ABRT (abort)
9	KILL (non-catchable, non-ignorable kill)
14	ALRM (alarm clock)
15	TERM (software termination signal)

Mit kill können Signale an einen Prozess gesendet werden, nicht nur zum Abbrechen sondern z.B. auch um einen Prozess anzuhalten und nachher wieder fortzusetzen. Beim normalen Aufruf von kill über kill <PID> wird dem Prozess das Signal SIGTERM gesendet, welches den Prozess zum Beenden auffordert. Reagiert der Prozess auf dieses Signal nicht, kann die brachiale Methode verwendet werden, indem man über kill -9 <PID> oder kill -SIGKILL <PID> den zwingenden Abbruch des Prozesses fordert. Die Syntax des Befehls lautet:

```
kill -Signal <PID> oder kill -Signal-ID <PID>
```

Eine Auswahl der Signale ist in Tabelle wiedergegeben.

Beim Befehl killall muss nur als Parameter der Name des Kommandos oder Befehls, welcher abgebrochen werden soll, angegeben werden und alle (!) Prozesse mit diesem Kommando- bzw. Befehlsname werden abgebrochen.



### 10.3 Graphische Prozess-Tools

Natürlich gibt es unter Linux mehrere graphische Tools, um die Arbeit und die Übersicht zu erleichtern. Das Programm ktop ist ein graphisches Front-End zum Befehl top. Mit diesem Programm können die Prozesse ausgewählt werden, welche angezeigt werden sollen (alle, System, Nutzer, eigene), ob als Liste oder als Prozess-Baum und es können auch Prozesse abgebrochen werden. Zusätzlich kann man sich noch Informationen über die Systemauslastung anzeigen lassen (siehe Abbildung ).

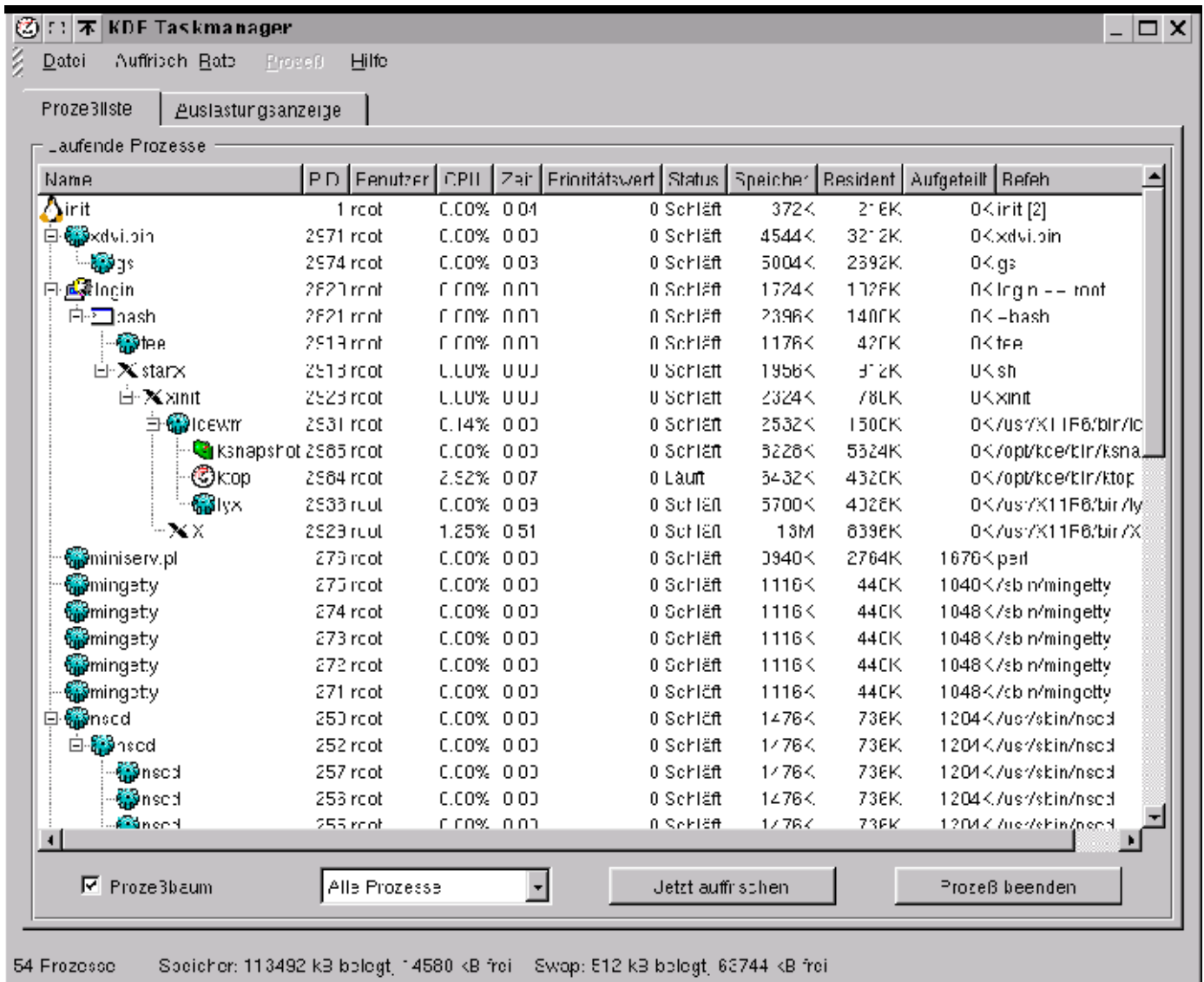
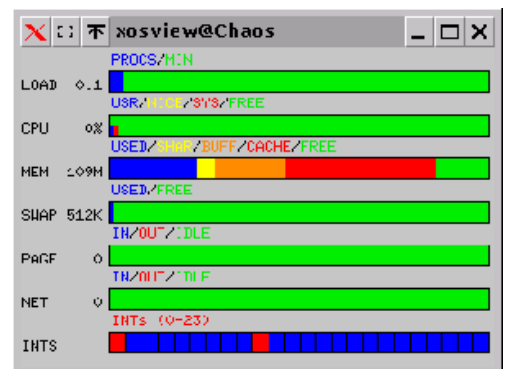


Abbildung: Das Programm ktop

Das Programm xosview zeigt zwar nicht einzelne Prozesse an, bietet aber durch die Anzeige der Systemauslastung und der Arbeitsspeicherbelegung eine gute Übersicht über das System (siehe Abbildung ).





## **11 LITERATURVERWEISE (INTERNET):**

Dokumentationen von Novell/SuSE (allgemein):

<http://www.novell.com/de-de/documentation>

Opensuse 10.2 <http://www.novell.com/de-de/documentation/opensuse102/>

Einführungstutorials (allgemein):

<http://www.learninglinux.de/tutorials/index.html>

<http://www.selflinux.org> (etwas veraltet)

Arbeiten in der Shell:

Shellskripte: <http://www.luetticken.net/linux/shskript.htm>

<http://www.linuxfocus.org/Deutsch/September2001/article216.shtm>

!

reguläre Ausdrücke: <http://www.linuxfocus.org/Deutsch/July1998/article53.html>