

Internettechnik

<http://worgtsone.scienceontheweb.net/worgtsone> – <mailto:worgtsone@hush.com>

13. Oktober 2011

Inhaltsverzeichnis

1	Einleitung	5
1.1	Zweck	5
1.2	Verwendung	5
1.3	Internet im Unternehmen (Intranet)	5
2	HTML	6
2.1	Skelett einer Seite	6
2.2	Regeln für Tags	6
2.3	Verbreitete Tags	6
2.3.1	Anatomie einer URL (Unified Resource Locator)	6
2.4	Sonderzeichen	7
2.5	Übung	8
2.5.1	HTML-Seiten	8
3	Interaktive Webseiten	10
3.1	Metacrawler benutzen	10
3.2	Metacrawler verstehen	10
3.3	Übung: Startseite von <code>www.metacrawler.com</code> verkürzen	10
3.3.1	Lösung:	11
3.3.2	Erläuterung	11
3.4	Übung: Weitere Tags	12
4	File Transfer protocol - FTP	13
4.1	Server und Clients	13
4.1.1	Server: Definition	13
4.1.2	MultiUser	13

4.1.3	Woher bekomme ich Linux?	13
4.1.4	Benutzerrechte	14
4.2	Übung	14
4.2.1	FTP-Server im Eigenbau	14
4.2.2	FTP-Client benutzen	15
5	Mail	16
5.1	Geschichtliches	16
5.2	Authentifizierung	16
5.3	Übung	17
5.3.1	Mail mit Eudora	17
6	Prozesse und Portnummern	18
6.1	Server-Dienste	18
6.2	Prozesse	18
6.3	Portnummern	18
7	Telnet	20
7.1	Was ist Telnet?	20
7.2	Dateisystembefehle	20
7.3	Bash - die shell	21
7.4	Editoren	23
7.4.1	vi	23
7.4.2	pico und joe	23
7.5	Übungen	24
7.5.1	Pascal	24
7.5.2	Dateisystembefehle	24
7.5.3	Alle	24
7.5.4	Ausgaben	24
7.5.5	Hilfsprogramm seq	24
7.5.6	Selbstgeschriebene Bilder	24
7.5.7	Weitere Umgebungsvariablen	25
7.5.8	Zusatzprogramme	25
7.5.9	Skripte	25
8	Netzwerkschichten nach ISO/OSI	26
8.1	Netzwerk-Komponenten	27

8.1.1	Gateway	27
8.1.2	Router	27
8.1.3	Bridge	27
8.1.4	Proxy	27
9	PHP - Private HomePages	28
9.1	Funktionsweise	28
9.2	PHP	28
9.3	Schreiben von PHP-Dateien	29
9.4	Ausgewählte Sprachelemente	29
9.4.1	Ausgabe	29
9.4.2	Variablen	30
9.4.3	Zufallszahlen	30
9.4.4	Verzweigung	30
9.4.5	For-Do-Schleife	31
9.4.6	Variablenübergabe	31
9.5	Übung	33
9.5.1	Gib "Hallo Welt" aus.	33
9.5.2	Gib die Sinusse von 0 bis 9 aus.	33
9.5.3	Iteratives Wurzelziehen	33
9.5.4	Zufallszahl	33
9.5.5	Katze	33
9.5.6	Eingabeüberprüfung mit <code>isset</code>	33
9.5.7	Eingabe verarbeiten	33
9.5.8	Mail	33
9.5.9	Gästebuch	34
10	Virtual networked Computing (VNC)	35
10.1	Windos	35
10.2	Telnet	35
10.3	Die Lösung: <code>vnc</code>	36
10.4	Übung	36
11	Common Internet File System (CIFS oder Samba)	37
11.1	Zweck	37
11.2	Primary Domain Controller	37
11.3	Master Browser	37

11.4 Samba	38
11.5 Übung	38
11.5.1 Samba	38

Disclaimer

Wissen ist zum Teilen da. Ich teile mein Wissen mit Ihnen, lieber Kollege.

Ich bin aber nicht perfekt. Unter worgtsone@hush.com nehme ich dankbar Ihre Verbesserungsvorschläge entgegen.

*

Legal Blurb: Alle Informationen in diesem Dokument sind falsch, unvollständig, irreführend, irrelevant und / oder funktionieren einfach nicht.

Wenn Sie es trotzdem benutzen, und es geht dabei etwas kaputt, ist das Ihr Problem, nicht meins.

*

Bitte teilen Sie meine Web-Adresse nicht Ihren Schülern mit.

1 Einleitung

1.1 Zweck

Das Internet dient dem schnellen, globalen Informationsaustausch durch

- Browsen `http`
- Email `pop`, `smtp`
- `ftp` ("Leech")
- Internettelefonie

Dieser Datenaustausch funktioniert schnell, sicher, günstig und arbeitszeitunabhängig

1.2 Verwendung

- anonymer Kommunikation (Email und Chat)
- Werbung
- Finden neuer Kunden
- E-Commerce
- Kundenpflege (CRM - Customer Relationship Management)
- Online-Kataloge, -Datenbanken, -Support
- Vermeiden von Filialen, wenn's doch Email, Cash-Systeme und Kurierdienste gibt

1.3 Internet im Unternehmen (Intranet)

Gemeinsames Benutzen von

1. Druckern
2. Massenspeichern (zB Festplatten auf einem Rechner, von dem wirklich backups gemacht werden)
3. Mail-Server und Fax-Server

Wirtschaftlich sehr bedeutungsvoll, Bedeutung wächst täglich.

2 HTML

Internet-Seiten, die im Browser (zB MSIE) angezeigt werden, sind in HTML geschrieben.

2.1 Skelett einer Seite

```
<html>
  <head>
    <title>
      Seitentitel by grm
    </title>
  </head>
  <body>
    irgendein text
  </body>
</html>
```

2.2 Regeln für Tags

1. Eine HTML-Seite besteht aus Kopf und Körper.
2. Im Kopf steht der Titel
3. Formatierungen werden durch Schlüsselworte (Tags) gestartet (ohne /) und beendet (mit /).
4. Die Schlüsselworte stehen in spitzen Klammern.
5. Tags darf man ineinanderschachteln (wie begin-end-Konstruktionen in Pascal).

2.3 Verbreitete Tags

Stehen am Anfang des Übungs-Blattes.

2.3.1 Anatomie einer URL (Unified Resource Locator)

Das Beispiel

```
http://www.go.org/joachim/rules/rule06.html
```

zeigt folgende Bausteine:

- `http`: das Protokoll, mit dem die Internet-Seite angefordert wird. Andere verbreitete Protokolle sind `ftp`, `pop`, `smtp`, `telnet`, `gopher`.

- `www.go.org` der Full Qualified Hostname (völlig zureichende Rechnername). `www` ist der Rechnername, er steht organisatorisch in der subdomain `go` in der Domain `org`.

`org` ist reserviert für non-profit organizations. Andere domains sind `gov`, `edu`, `com` für Regierungs- bzw. Ausbildungs-Organisationen (Unis etc.) bzw. gewinnorientierte Firmen (commercial) in den USA, `de`, `uk`, `au` für Deutschland, United Kingdom (Großbritannien), Australien etc. Es gibt eine internationale Liste für 2-buchstabile Länderabkürzungen.

Die meisten `www`-Server heißen `www` oder tun wenigstens so.

- `joachim` ist das Unterverzeichnis des `www`-Servers.
- `rules.html` ist der Dateiname.
Es muß nicht immer eine `html`-Datei sein - auch `.wav`, `.zip`, `.bz2`, `mpg` etc. sind erlaubt.
- `#rule06` ist ein Anker innerhalb der Datei, vermutlich die Go-Regel Nr. 6.

2.4 Sonderzeichen

Sonderzeichen heißen

`& etwasdasmansichmerkenkann ;`

Quelltext	Merkhilfe	Ergebnis
<code>&auml;</code>	a-Umlaut	ä
<code>&ouml;</code>	o-Umlaut	ö
<code>&uuml;</code>	u-Umlaut	ü
<code>&Auml;</code>	A-Umlaut	Ä
<code>&Ouml;</code>	O-Umlaut	Ö
<code>&szlig;</code>	s-z-ligatur	ß
<code>&amp;</code>	ampersand	&
<code>&lt;</code>	lower-than	<
<code>&gt;</code>	greater-than	>

2.5 Übung

tag	Bedeutung
neue zeile br p	macht nix Zeilenumbruch (branch) großer Zeilenumbruch (paragraph)
hr hr width=80p hr width=40%	horizontal rule hr, 80 Pixel breit hr, 40% des Fensters breit
b i u tt H1 .. H6	boldface (fett) italic (kursiv) underline Schreibmaschinenschrift Überschrift
 Listenpunkt 1 Listenpunkt 2 	unnummerierte Liste mit 2 Items
ol	nummerierte Liste (ordered list)
pre	preformatted: Zeilenumbruch gibt neue Zeile
	Definition eines Ankers namens rule06
	zeigt ein Bild an dieser Stelle an
	ein verzerrtes Bild
Hier Klicken	Der Text "Hier Klicken" wird unterstrichen dargestellt und ist ein Hyperlink auf die angegebene Seite.

2.5.1 HTML-Seiten

Schreibe zwei HTML-Seiten. Beide enthalten Deinen Namen und das Datum von heute in der ersten sichtbaren Zeile.

Die erste enthalte zudem

- 6 verschieden große Überschriften,
- fetten, kursiven und unterstrichenen Text, z.T. in Schreibmaschine,
- ein Bild, das gleichzeitig ein Link auf die zweite Seite ist, und
- eine unnummerierte Liste.

Die zweite enthalte zudem

- horizontale Linien über 50% der Breite des Fensters,

- eine numerierte Liste, in die eine weitere numerierte Liste geschachtelt ist,
- zwei Links auf zwei Anker,
- deutsche Sonderzeichen und
- den Quelltext für die Liste in lesbarem HTML.

Beide Seiten sollen zudem Links auf die jeweils andere Seite sowie auf die numerierte Liste und die deutschen Sonderzeichen haben.

3 Interaktive Webseiten

3.1 Metacrawler benutzen

Wenn man nach `www.metacrawler.com` geht, sieht man zunächst eine Seite, die neben viel Werbung ein Eingabeformular enthält. In das Eingabeformular kann man Worte eingeben und den Suchbereich spezifizieren.

Nach Klick auf `GoGetIt!` sieht man eine Seite, die Ergebnisse zu den Suchworten anzeigt. Man braucht sie nur noch anzuklicken und erhält beliebig viele Informationen zum eingegebenen Suchbegriff.

Die URL hat sich ebenfalls geändert:

1. Protokoll ist immer noch `http:`.
2. Der befragte Rechner heißt nun `search.metacrawler.com`.
3. Die aufgerufene Datei heißt nun `taxis/search`. Sowohl aus der Extension als auch aus dem Verhalten darf man schließen, daß sie keine `html`-Datei, sondern ein Programm ist.
4. Dahinter, durch `&` getrennt, stehen ein paar Name-Wert-Paare ("Hashes") (z.B. `method=1`).

Was immer der User eingegeben hat, steht nun, mit einem Namen versehen, mit in der URL.

Man kann die URL direkt ändern. Als Resultat erhält man die entsprechenden Seiten. Das beweist: Die Parameter werden tatsächlich über die URL weitergegeben.

3.2 Metacrawler verstehen

Der normale Benutzer mag sich nun denken: Aha, `metacrawler` holt sich meine Suchworte und sieht in seiner Datenbasis nach, welche Seiten dazu passen. Bei mehrfachen Suchen nach denselben Suchworten stellt man dabei fest: Oh, es sind jedesmal unterschiedlich viele Ergebnisse.

Die Erklärung: Metacrawler sucht nicht in seiner eigenen Datenbasis, sondern in den Datenbasen anderer Suchmaschinen. Verschiedene Suchergebnisse erklären sich durch unterschiedliches Zeitverhalten der befragten Suchmaschinen.

`metacrawler` ist eine Art Suchmaschine der Suchmaschinen, oder mathematischer ausgedrückt: eine Meta-Suchmaschine.

3.3 Übung: Startseite von `www.metacrawler.com` verkürzen

Hole die Startseite von `www.metacrawler.com`, speichere sie auf dem Desktop, sieh den Quelltext an, und wirf alles heraus, was nicht dringend nötig ist.

3.3.1 Lösung:

```
<body> <center>

<form name=querybox
action=http://search.metacrawler.com/texis/search?method=get>

<input type=text name=q value='' size=40> <p>

<input value="Suche jetzt" type=submit>

<input type=radio name=method value=1>any
<input type=radio name=method value=2>Phrase
<input type=radio name=method value=0 checked>all

</form> </body> </html>
```

3.3.2 Erläuterung

Die erste Zeile ist klar.

In der zweiten Zeile wird ein Formular definiert, das in der vorletzten Zeile abgeschlossen wird.

Das Formular bekommt noch ein paar Name-Wert-Paare (engl.: hashes) zugewiesen.

name Eine HTML-Seite kann mehrere Formulare enthalten, die man dann anhand ihrer Namen auseinanderhalten kann.

Jedes Formular-Element hat aus demselben Grund ebenfalls Namen.

action Diese URL wird aufgerufen. Beim Abschicken der Seite wird die Seite `search` angefordert. Diese ist wahrscheinlich gar keine HTML-Seite, sondern ein Programm, das zu den eingegebenen Suchworten passende Ausgaben *dynamisch* generiert.

method Die Benutzer-Eingaben können mittels GET oder POST übergeben werden. Im ersten Fall kann man sie in der Adresszeile des Browsers als Name-Wert-Paare sehen, im zweiten Fall nicht.

Es folgen verschiedene Eingabefelder:

text dient zum Text eingeben. Die *hashes* erklären sich selbst.

submit Durch Klick hierauf wird das ausgefüllte Formular abgeschickt. Die Aufschrift kann man durch ein *hash* `value=` ändern, zB zu `GoGetIt`.

radio Wie an einem Radio springen die anderen Knöpfe heraus, wenn man einen wählt. Gegenteil=checkbox.

value=0 ist voreingestellt checked.

3.4 Übung: Weitere Tags

Finde heraus, wie man Schriftarten, Schriftgrößen und Schriftfarben ändert, und wie man Tabellen erstellt.

Dazu gibt es zwei Möglichkeiten:

- Suche dir eine Webseite, die die gewünschten Elemente enthält, und suche die Tags im Quelltext der Seite.
- Geh zu einer www-Seite, die dir mit HTML auf die Sprünge hilft, und schlag dort im Inhaltsverzeichnis nach.

4 File Transfer protocol - FTP

4.1 Server und Clients

4.1.1 Server: Definition

Ein **Server** ist **hardwaremäßig** ein Rechner wie jeder andere auch: mit Festplatte und Netzwerkkarte.

Software: Es läuft ein kleines Programm darauf, das auf Anfragen von Clients wartet und diese dann ggf. befriedigt.

Man kann jeden netzwerk- und multitaskingfähigen Rechner zum Server aufrüsten, selbst eine Windows95-Kiste. (s. www.google.de : download free w95 ftp server).

**Ein Server ist ein Rechner wie jeder andere auch.
Zum Server wird er erst durch Programme, die kontinuierlich darauf
laufen und hereinkommende Anfragen von Clients beantworten.**

4.1.2 MultiUser

Ein Server dient meistens mehreren Leuten gleichzeitig. Damit diese sich nicht in die Quere kommen (gegenseitig Dateien löschen usw.), muß er bzw. das Server-Programm multiuser-fähig sein.

Es gibt zwei Möglichkeiten, das zu realisieren:

1. Wir schreiben das Server-Programm so, daß es verschiedene Benutzer und ihre Rechte verwalten kann.
Das müssen wir dann für jedes Server-Programm tun.
Oder
2. Wir benutzen ein Betriebssystem, das von Natur aus multiuser-fähig ist. Das verbreitetste, das kostenlos verteilt wird und samt seiner (ebenfalls kostenlosen) Server-Programme als extrem sicher und stabil gilt, heißt Linux.

4.1.3 Woher bekomme ich Linux?

Linux ist eine Idee, ein Prozess, work in progress. Es läuft auf fast jeder Hardware (zB Xbox, Premiere-Decoder, 386er aufwärts).

Linux kauft man in Distributionen. Auf einer Distribution sind die anwesenden Programme zueinander kompatibel - sie finden ihre Bibliotheken etc.

Wenn ein Programm nicht dabei ist, saugt man den Quelltext dazu aus dem Netz und versucht es zu kompilieren, installieren, konfigurieren und starten.

Der `linuxserver` benutzt Redhat-Linux 6.2 auf einem P166, Wechsel sind nicht geplant.

4.1.4 Benutzerrechte

Unter Un*x gibt es drei Gruppen:

- den Dateieigentümer,
- die Gruppe des Dateieigentümers,
- den Rest der Welt.

Jeder der drei kann folgende Rechte an der Datei haben:

- r - readable, lesbar.
- w - writeable, schreibbar.
- x - executable, ausführbar.

```
$ ls -l di*x
-rw-r--r--  1 myname  users      5378 Dez  1 10:41 dint.tex
```

- rw für `myname`, den Besitzer.
- r für `users`, die Gruppe des Besitzers.
- r für den Rest der Welt.
- Dateigröße = 5378 Bytes.
- letzte Änderung am 01.12 diesen Jahres um 10:41.
- Dateiname = `dint.tex`.

4.2 Übung

4.2.1 FTP-Server im Eigenbau

1. Erzeuge `c:\deleteme`.
2. Saug `G6_FTPServer_beta_2.0`, und installiere ihn.
3. Sein Wurzelverzeichnis sei `c:\deleteme`.

4. Lege die benutzer `juan` und `new` an (passwort=username). Sie dürfen alles.
5. Legen den Benutzer `anonymous` an. Er darf nur lesen.
6. Lege 5 Dateien von 1kB, 10kB, 100kB, 2MB und 30MB nach `c:\deleteme`.
7. Starte den Server. Schreibe seine IP an die Tafel.

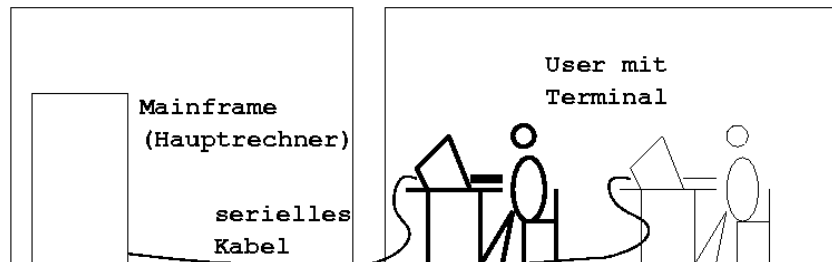
4.2.2 FTP-Client benutzen

Gehe im MSIE nach `ftp://juan@<IPanDerTafel>`, sauge und lade Dateien hoch.

5 Mail

5.1 Geschichtliches

Als Rechner noch teuer waren, saßen nur amerikanische Studenten vor Terminals, die an den Uni-Hauptrechner (Host, Mainframe) angeschlossen waren.



Rechner-Raum (nicht-öffentlich) Benutzer-Raum (öffentlich)

Abbildung 1: Die User sitzen an Terminals - fast wie am Rechner.

Die mails, die sie eintippten, kamen von einem vertrauenswürdigen host. Zur Übertragung auf andere Rechner waren keine Sicherheitsabfragen notwendig. Das Protokoll hieß `smtp` (simple mail transfer protocol).

Als Rechner billig wurden und man sich über Modems zeitweise ins WWW einhängen konnte, brauchte man ein Protokoll, mit dem die User folgendes konnten:

- sich authentifizieren;
- ihre Mail vom Server holen;
- sie anschließend von dort löschen

Das Protokoll heißt `pop` (post office protocol). Mittlerweile gibt es Erweiterungen namens `pop2`, `pop-3`, `imap`.

Zum Senden und Empfangen von Mail werden 2 verschiedene Protokolle benutzt: `pop` zum Empfangen, `smtp` zum Senden.

5.2 Authentifizierung

Üblicherweise werden Sender wie folgt authentifiziert:

1. Zunächst holt der User (z.B. `juan`) über `pop` seine Mail (mit Passwort).

2. Dabei schaltet der Mailserver den Absender `juan@wo.auch.immer` für 5 Minuten frei.
3. Der User darf nun 5 Minuten lang Mail als `juan@wo.auch.immer` versenden. Einen anderen Absender weist der Mailserver zurück.
4. Wenn das in diesen 5 Minuten jemand anders versucht, könnte es gelingen - es sei denn, der Mailserver überwacht auch die IP des sendenden Rechners.
5. Nach Ablauf der 5 Minuten muß `juan` zunächst wieder seine Mail holen, bevor er erneut senden darf.

5.3 Übung

5.3.1 Mail mit Eudora

Saugen Sie sich `eudora.exe`, installieren und konfigurieren Sie es, und senden und empfangen Sie mails. Ihr Benutzername ist `username@linuxserver`.

6 Prozesse und Portnummern

6.1 Server-Dienste

Wir haben bis jetzt verschiedenste Server-Dienste des `linuxserver` in Anspruch genommen:

- `httpd`;
- `ftpd`;
- `popd`;
- `smtpd`.

Das `d` in der obigen Aufzählung kommt übrigens vom Wort Daemon.

6.2 Prozesse

Unix-Systeme sind extrem aufgeräumt. Man kann jederzeit nachschauen, welche Prozesse gerade laufen. Mit den notwendigen Rechten kann man auch bestehende Prozesse schlafen legen oder killen oder neue aufrufen.

Prozesse, die im Hintergrund laufen und auf Dienstanweisungen warten, heißen Dämonen. Offensichtlich MUSS auf dem `linuxserver` ein Programm laufen, das `http`-Anforderungen entgegennimmt und abarbeitet - und möglichst wenig Ressourcen verbraucht, während es nichts zu arbeiten hat.

Es heißt logischerweise `http`-Daemon oder `httpd`.

Prozesse können weitere Prozesse anstoßen. Deshalb sieht man mit

```
$ ps -ax | grep httpd
```

sowohl den ersten `httpd`-Prozeß (kenntlich an der kleinsten Prozeßnummer), als auch seine 10 Child-Prozesse (Kinder).

6.3 Portnummern

Die oben genannten Server-Programme laufen alle auf dem `linuxserver`. Die Anfragen kommen alle über dasselbe Netzwerk-Kabel an denselben Rechner. Der Rechner erkennt an der Portnummer, an welches Server-Programm er die Anfrage weiterleiten muß. `http`-Anfragen werden vom Browser in der Regel an Portnummer 80 geschickt.

Man kann sich das wie folgt vorstellen:

Im Server ist eine TCP-IP-Maschinerie mit vielen Steckern ("Ports"). Hinter einigen Ports sitzt ein Dämon und schläft. Wenn nun eine Anfrage für Port 80 kommt, leitet die

Maschinerie diese zum Port 80. Davon wacht der Dämon auf, beantwortet die Anfrage und schläft anschließend weiter.

Wenn man eine `http`-Anfrage an Port 23 sendet, bekommt man keine `html`-Seite zurück, sondern eine Fehlermeldung vom `telnet`-Dämon.

Die Client-Programme wissen in der Regel, welchen Port sie ansprechen müssen.

7 Telnet

7.1 Was ist Telnet?

Wir werden in den nächsten Stunden den `telnetd` nutzen. Dieser stellt uns eine Telnet-Konsole zur Verfügung.

Eine Konsole ist ein zeichenbasiertes Ein-/Ausgabegerät. In historischer Zeit war jedes Terminal über eine serielle Schnittstelle an den Hauptrechner angeschlossen.

Terminals stellen die vom Hauptrechner empfangenen Zeichen auf dem Bildschirm dar und senden die vom User eingetippten Zeichen dorthin. Das hat zwei wesentliche Vorteile:

1. Der User kann arbeiten, als säße er direkt vor dem Hauptrechner.
2. Terminals sind billig.

Telnet kann dasselbe, nur daß die räumliche Entfernung zwischen dem Telnet-Server und dem Telnet-Client egal ist. Telnet funktioniert quer durch die Schule oder auch über das `www` um die halbe Welt.

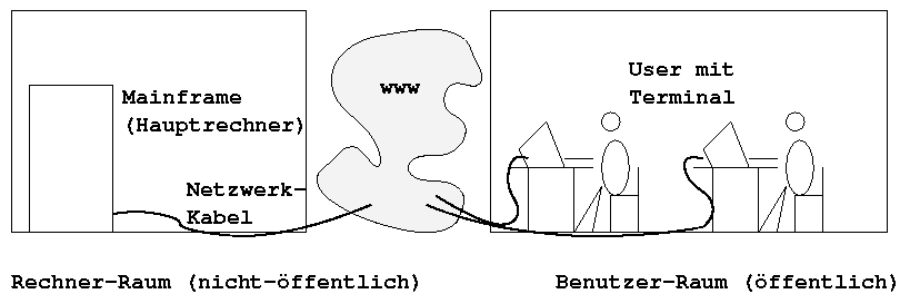


Abbildung 2: Die User sitzen am Terminal - egal wo der Rechner steht.

Da die einzelnen Tastendrücke gesendet werden, können Rechner, über die diese Signale gehen, sie leicht dekodieren und so Passwörter sammeln. Dies ist in der Regel unerwünscht. Im Internet benutzt man daher ein anderes Protokoll (SSH, Secure Sockets Layer), wobei jeder Tastendruck verschlüsselt wird.

Innerhalb des Hauses gibt es keine Rechner, die dazwischenhängen.

7.2 Dateisystembefehle

Alle Programme unter Un*x werden beim Aufruf durch Optionen und Parameter gesteuert.

Vorteile: Man sagt dem Programm in einer Zeile, wie und woran es arbeiten soll.

Nachteile: Man muß die Optionen und Parameter in den Manual-Pages nachschauen.

ls	zeigt Dateien im aktuellen Verzeichnis an
ls -la	dito.
pwd	zeigt aktuelles Verzeichnis
joe filename	editiert Datei
cd ..	wechselt ein Verzeichnis höher
mkdir tex	erzeugt Verzeichnis tex
rmdir tex	löscht Verzeichnis tex, wenn es leer ist
cd tex	wechselt ins Unterverzeichnis tex

Tabelle 1: Die wichtigsten Dateisystembefehle unter Un*x

Die Vorteile überwiegen, denn die meisten Un*x-Server sollen arbeiten, während der Sysop irgendwo in Ruhe Kaffee trinkt.

7.3 Bash - die shell

Telnetd sorgt nur für die Verbindung. Das Programm, das uns Befehle über die Tastatur eingeben läßt und anschließend ausführt, heißt shell.

Wir verwenden bash. bash kann tolle Sachen:

- alien expansion - wenn wir * eingeben, expandiert bash das zu allen Dateien im aktuellen Verzeichnis.
- command line completion - wer nicht jedesmal `latex MeineDiplomarbeit.tex` eintippen will, benutzt `lat<tab> M<tab>.tex`.
- Prozesse schlafen schicken mit `C-z` und wieder hervorholen mit `fg` oder im Hintergrund weiterlaufen lassen mit `bg`.
- Eingebaute shell-Befehle laufen lassen. Der interessanteste ist `echo` oder `echo -n`. Hilfe dazu gibt es mit `help echo`.
- Umgebungsvariablen auslesen, setzen und testen, z.B.:
`r=asdfghjk` - setzt die Umgebungsvariable `r` auf `tasdfghjk`.
`echo "rr $r"` - gibt sie dreimal aus, einmal ohne Leerzeichen, einmal mit Leerzeichen dazwischen.
- for-do-schleifen laufen lassen, zB

```
for i in $(seq 10 3) zwei 1 BUUUUUUUUMMMMM ; do
  sleep 1 ;
  echo $i ;           # zählt runter von 10 bis BUUUUMMMM
done
```

- STDOUT umlenken und pipen (zB nach STDIN), z.B.:

`echo $r > r.my.file.name.txt` - schreibt die Ausgabe des Befehls in die benannte Datei.

`echo $r | ppmtogif > r.my.file.name.txt` - sendet die Ausgabe des Befehls in das Programm `ppmtogif` und schreibt dessen Ausgabe in die benannte Datei.

7.4 Editoren

7.4.1 vi

Der verbreitetste Editor unter Linux ist `vi`. Befehle:

<code>vi</code> <code>vi filename</code>	ruft <code>vi</code> mit einer leeren Datei auf öffnet <code>filename</code> mit <code>vi</code> .
Innerhalb von <code>vi</code> : <code>ESC</code> <code>i</code>	wechselt in den Befehlsmodus. wechselt in den Einfügemodus.
Im Befehlsmodus: <code>:wq</code> <code>dd</code> <code>99dd</code> <code>dw</code> <code>17dw</code> <code>P</code> <code>w</code> bzw. <code>b</code>	speichert die Datei und beendet <code>vi</code> . löscht die aktuelle Zeile. löscht die aktuellen 99 Zeilen. löscht das nächste Wort. löscht die nächsten 17 Worte. fügt die zuletzt gelöschte Zeile wieder ein. springt wortweise vor bzw. zurück.

Tabelle 2: Wichtige `vi`-Befehle

7.4.2 pico und joe

`pico` und `joe` haben ein etwas benutzerfreundlicheres Aussehen.

Befehle in `joe`:

`^KX` beendet `joe` mit Speichern.

`^KH` öffnet oder schließt die Hilfe in `joe`.

7.5 Übungen

7.5.1 Pascal

Lies **Pascal auf dem linuxserver**. Schreibe anschließend 2 Pascal-Programme:

Das erste gibt den Sinus der Zahlen 0 - 10 aus.

Das zweite erzeugt eine Zufallszahl zwischen 0 und 99, zieht 50 ab, gibt das Ergebnis aus und untersucht und gibt aus, ob es größer, kleiner oder gleich 0 ist.

7.5.2 Dateisystembefehle

1. Wie heißt dein Zuhause auf `linuxserver`?
2. erzeuge mit einem der Editoren `joe`, `pico`, `vi` oder dem Editor von `mc` eine Datei `plain.txt`.
3. Lasse dir die Dateien in deinem Zuhause oder in `/usr/bin` in verschiedenen Formaten anzeigen.

7.5.3 Alle

* bedeutet umgangssprachlich **alle**. Zu welchen Zeichen expandiert `bash` NICHT? Warum nicht?

7.5.4 Ausgaben

Geben Sie `Hallo Welt!` aus.

Geben Sie `Hallo Welt! =) (/%%$$Û*' Ä<>Ö` aus.

Setzen Sie die Variable `r` auf `Hallo Welt! =) (/%%$$Û*' Ä<>Ö`, und geben Sie sie aus. Geben Sie sie anschließend 10mal aus.

Setzen Sie die Variable `r` auf `Hallo Welt! =) (/%%$$Û*' Ä<>Ö`, und geben Sie sie in die Datei `deleteme.txt` aus - 10x mit LF am Ende, 10mal ohne.

7.5.5 Hilfsprogramm seq

Lesen Sie `man seq`, und geben Sie die Zahlen von 0 bis 999 auf den Bildschirm aus - einmal nebeneinander, einmal untereinander.

7.5.6 Selbstgeschriebene Bilder

Schreiben Sie ein Pascal-Programm, das folgendes ausgibt:

```
P2 100 100 256
100mal 0
100mal 1
.
100mal 99
```

Pipen Sie das output in die Datei `bild1.ppm`.

Pipen Sie das Output in den STDIN von `ppmtogif`, und pipen Sie dessen output in die Datei `bild1.gif`. Falls Ihnen das Output nicht gefällt, pipen Sie es zunächst durch `ppmdither`.

7.5.7 Weitere Umgebungsvariablen

Gucken Sie sich mit `export` alle Umgebungsvariablen an.

7.5.8 Zusatzprogramme

Lesen Sie `man date`, und geben Sie das aktuelle Datum des Rechners in deutschem Format aus.

Lesen Sie `man cat`. Geben Sie `bild1.ppm` auf den Bildschirm aus. Schreiben Sie `bild1.ppm` dreimal hintereinander nach `3xbild1.ppm`.

Schauen Sie mit `less bild1.ppm` an.

Pipen Sie das Output von `ls -l /usr/bin/*` nach `less`.

Lesen Sie `man rm`, und löschen Sie einige Ihrer Dateien.

Lesen Sie `man cut`, setzen Sie `$r` auf einen langen String, und schneiden Sie den 5. bis 9. Buchstaben heraus.

7.5.9 Skripte

Statt all diese Befehle wieder und wieder einzugeben, schreiben Sie besser Skripte.

8 Netzwerkschichten nach ISO/OSI

Bis jetzt haben wir gesehen, daß die verschiedenen Protokolle (`http`, `ftp`, ...) über ein gemeinsames Netzkabel überall hinrutschen.

Die International Standardization Organization (Internationale Normungs-Behörde) hat vernommen, daß die Protokolle dabei gestapelt werden, und ein 7-schichtiges Protokoll-Stapel-System definiert.

1. Physical Layer : elektrische und mechanische Kennwerte.

In dieser Netzwerkprotokollebene müssen zB die Spannungsschwankungen im Netzkabel definiert werden.

2. Link Layer : Datensicherung. Hier haben wir die Selbstorganisation des Netzwerks, zB was mit vergessenen oder besonders wichtigen Daten geschieht.

Diese beiden Schichten sind unter effektiven Betriebssystemen der TCP/IP-Welt hardwareabhängig, aber völlig transparent optimiert und zu einer zusammengefaßt: Network Layer.

Übliche Protokolle in der TCP/IP-Welt sind zB SLIP, PPP, Ethernet,...

3. Network Layer. Unter TCP/IP = Internet Layer. Protokolle = IP, IPv6.

Die Daten im IP-Protokoll müssen irgendwo ins Ethernet-Protokoll umgewandelt werden (wenn sie über Ethernet reisen sollen), und irgendwann auch wieder zurück. Dies geschieht, hardware-programmiert, auf der NIC (Network Interface Card, Netzkarte).

4. Transport Layer. Unter TCP/IP = Host-to-Host Transport Layer. Protokolle = TCP, UDP.

Die Wandlung von TCP nach IP und rückwärts wird von Software erledigt.

5. Session Layer,

6. Presentation Layer,

7. Application Layer.

Sie haben jede bestimmte Aufgaben, sind aber unter TCP/IP zum Application Layer zusammengefaßt. Die verbreiteten Applikationen (FTP-Clients, Browser) können gleichzeitig präsentieren und FTP bzw. HTTP sprechen.

Unter TCP/IP gibt es also nur 4 baulich getrennte Schichten. Das OSI/ISO-Modell wird von vielen als überladen betrachtet.

ISO/OSI	TCP/IP	Protokolle (Beispiele)
7 application layer	application layer	FTP-, HTTP-, SMTP-client
6 presentation layer		----- ftp, http, smtp
5 session layer	host-to-host- transportation layer	tcp, udp
4 transport layer	internet layer	ip, ipv6
3 network layer	network access layer	slip, ppp, ethernet
2 link layer		
1 physical layer		

8.1 Netzwerk-Komponenten

8.1.1 Gateway

ist eine allgemeine Komponente zur Verbindung von Netzwerken, die auf verschiedenen Layern arbeiten kann.

Netzwerke mit verschiedener Hardware (zB Glasfaser-ThinWire) oder verschiedenem Low-Level-Protokoll (Ethernet vs. TokenRing) müssen über Gateways verbunden werden.

Wenn das Gateway auf Layer 3 arbeitet, heißt es Router.

Wenn das Gateway auf Layer 7 arbeitet, heißt es Proxy oder Application Level Gateway.

8.1.2 Router

Verbindet Netzwerke auf Level 3.

IP und IPX dürfen durch, Broadcasts nicht.

8.1.3 Bridge

Eine Bridge verbindet Netzwerk-Segmente auf Level 2, auf der Basis von MAC-Adressen.

Z.B. Zur Umsetzung Glasfaser-TwistedPair.

8.1.4 Proxy

„Stellvertreter“. Nimmt Anforderungen von Rechner A entgegen und gibt sie, ggf. gefiltert, anonymisiert oder zeitverzögert, an Rechner B weiter.

Manche Proxies cachen.

9 PHP - Private HomePages

9.1 Funktionsweise

Wenn eine HTML-Seite vom Server über Port 80 angefordert wird, sendet er diese. Falls sie nicht vorhanden ist, sendet er eine Fehlermeldung.

Der Server (als eigenständiges Programm) kann aber sehr viel mehr als Nachprüfen oder Nachdenken. Praktisch ist es recht einfach, die HTML-Seite durch einen (oder mehrere) Präprozessoren (Vorverdauern) zu pipen, bevor sie zum Empfänger geleitet wird.

Im Apache-Webserver, der kostenlos bei jeder Linux-Distribution dabei ist und als wesentlich stabiler, schneller, zuverlässiger und sicherer gilt als jedes andere (teure) Produkt, schaltet man den Präprozessor in der Konfigurationsdatei ein.

Er schickt dann Dateien, die auf `.php` enden, zunächst durch den PHP-Präprozessor.

9.2 PHP

PHP ist eine Skriptsprache, die in ihrer Syntax eng an C/C++, perl und bash angelehnt ist. Sie unterstützt die üblichen Programmiersprachenelemente wie Funktionen, Skalare, Arrays, Hashes, Listen, Verzweigungen, Schleifen etc. und stellt darüber hinaus eine unglaubliche Anzahl von Funktionen zur Verfügung, mit denen man mit wenigen Zeilen Code folgendes kann:

- Berechnungen durchführen;
- Dateien öffnen, lesen, schreiben, anhängen, löschen;
- Datenbanken updaten und auslesen;
- GIFs erzeugen;
- Authentifizierung;
- Rechtschreibprüfung;
- (De)Kompression;
- Cookie-Erzeugung und -verwaltung;
- IMAP, LDAP, PDF, ODBC, Perl, SNMP, virtuelle Mailaccounts;
- etc.

Diese Liste ist nicht vollständig.

Wir beschränken uns hier auf die grundsätzlichen Sprachelemente Variablen, echo, Zufallszahl, if, Dateioperationen, fgets, trim, strpos, explode, isset, mail.

9.3 Schreiben von PHP-Dateien

Der http-Server auf `linuxserver` ist PHP-fähig. Bestimmte User dürfen in ihrem Home-Verzeichnis ein Verzeichnis `public.html` anlegen, dessen Inhalt dann unter `http://linuxserver/~username` zur Verfügung steht.

9.4 Ausgewählte Sprachelemente

PHP-Dateien enden auf `php3`. Sie enthalten plain HTML und zusätzlich PHP-Code wie folgt:

```
<html> <body> <center>

<?php
    echo "Hallo Welt!";
?>

</center> </body> </html>
```

Beliebig viele PHP-Blöcke sind erlaubt, HTML-Tags sind optional:

```
<?php
    echo "<html> <body> <center>";
?>
<?php
    echo "Hallo Welt!";
    echo "</center> </body> </html>";
?>
```

PHP-Anweisungen müssen mit einem Semikolon abgeschlossen werden. Die gezeigte Einrückung ist nicht vorgeschrieben, aber gewährleistet Lesbarkeit.

Variablennamen werden zu Variablenwerten expandiert.

9.4.1 Ausgabe

<pre>print gibt einen String aus. echo gibt eine Liste von Strings aus.</pre>

Beide ersetzen in den auszugebenden Strings erst die aufgefundenen Variablennamen durch die Werte der Variablen.

Beide ignorieren Funktionsaufrufe. Wenn das Ergebnis einer Funktion ausgegeben werden soll, muß man das zweistufig tun.

9.4.2 Variablen

Variablennamen beginnen immer mit einem \$.

Sie dürfen Buchstaben, Ziffern und _ enthalten.

Der Datentyp ist nicht klar definiert.

Variablen werden nicht initialisiert, außer man benutzt sie. D.h. sobald man eine Variable benutzt, gibt es sie auch.

Einer Variablen weist man durch = einen Wert zu. Das kann auch das Ergebnis einer Dateioperation (ein `filehandle`) oder einer Datenbank-Operation (zB eine Verbindungsnummer oder ein `select`-Resultat) sein.

Stringoperationen Stringoperationen sind immer erlaubt.

Strings werden durch . aneinandergereiht.

Zahlen und Rechnen Sofern PHP die Strings als Zahlen interpretieren kann, kann man auch damit rechnen, sonst gibt es eine Fehlermeldung.

PHP unterstützt (* / - +).

Beispiel Im folgenden Beispiel werden 5 Strings zusammengehängt, der fünfte String ist dabei das Ergebnis einer Berechnung.

```
$r=1.234;
$s=2.345;
echo $r."*".$s."=".( $r*$s );
```

9.4.3 Zufallszahlen

```
                                // dies ist ein Kommentar
srand(time());                 // startet Zufallszahlengenerator neu
echo rand(0,9);                 // Zufallszahl zwischen 0 und 9 inclusive
```

9.4.4 Verzweigung

PHP unterstützt

```
== >= <= != > < && ||
```

Die Bedingung steht in Klammern, die Anweisungsblöcke in geschweiften Klammern. `elif`- und `else`- Blöcke sind optional.

```
if ($a=="ein_string" || $b==3.141)
{
    echo "Erfüllt!";
}
elseif ($z==$z) then
{
    echo "Ist immer erfüllt.";
}
else
{
    echo "Unvorhergesehener Fehler!";
}
```

9.4.5 For-Do-Schleife

```
// simuliert durch eine while-do-Schleife
$i=1;
while ($i<10)
{
    $temp=sin($i);
    echo "sin($i) = $temp <br>";
    $i++;
}

// oder in echt
for ($i=0 ; $i<10 ; $i++)
{
    $temp=sin($i);
    echo "sin($i) = $temp <br>";
}
```

9.4.6 Variablenübergabe

Erinnern Sie sich an das Eingabefeld für Metacrawler? Die Parameter wurden als Name-Wert-Paare (engl.: hashes) in der URL transportiert.

HTTP unterstützt 2 Übertragungsmethoden: GET und POST. Bei GET kommen die Parameter auf der URL (wie bei `metacrawler`), bei POST stehen sie ohne ein abschließendes CR oder LF an STDIO.

Variablen zu extrahieren, war in früheren Zeiten sehr mühsam. Aber jetzt gibt es PHP.

Jedes Eingabefeld hat einen Namen. Kaum ist PHP gestartet, sucht es nach neuen Eingabeparametern.

Mit `isset()` kann man überprüfen, ob ein Parameter gesetzt wurde oder nicht.

Falls ja, existiert eine Variable namens `$Name`, und die hat automatisch den richtigen Wert.

Das kann zB eine Zahl sein oder ein String oder ein längerer String, z.B. ein komplettes Pascal-Programm aus einem `<TEXTAREA>`.

```
<?php
    if (isset $temporaer)                // Datei eingabe.php3
    {
        echo "Ihre Eingabe war: <hr> <pre>";
        echo $temporaer;                  // $ Kommentar
        echo "</pre> <hr>";
    }
    else
    {
        echo "<h1> Eingabemaske</h1> <hr>";
        echo "<form method=post action=eingabe.php3>";
        echo "<textarea name=temporaer cols=60 rows=10>";
        echo "program p;\n\n";
        echo "begin\n";
        echo "  writeln ('sin(4) = , sin(4));\n";
        echo "end."
        echo "</textarea> <hr>";
        echo "<input type=submit value=Jetzt_losschicken!> </form>";
    }
?>
```

9.5 Übung

9.5.1 Gib "Hallo Welt" aus.

9.5.2 Gib die Sinusse von 0 bis 9 aus.

9.5.3 Iteratives Wurzelziehen

Weise Z 16 zu. Weise a Z zu. Weise b 1 zu. Wiederhole:

$a \leftarrow (a+b)/2$ $b \leftarrow Z / a$ Gib a und b aus.

Bis A nicht weiter als 0,000001 von b entfernt.

9.5.4 Zufallszahl

Setze eine Variable auf einen Zufallswert zwischen -3 und 3. Prüfe anschließend, ob das Ergebnis größer 0, gleich 0 oder kleiner 0 ist.

9.5.5 Katze

Schreiben Sie `katze.php`. Sie kann schnurren, spielen, schlafen, pirschen und sich putzen.

Benutzen Sie Zufallszahlen, um das Verhalten der Katze zu simulieren. Benutzen Sie `sleep()` und `usleep()`, um zwischen den einzelnen Aktionen Pausen einzulegen.

9.5.6 Eingabeüberprüfung mit `isset`

Erzeugen Sie ein Eingabeformular, wo der User einen Text eingeben kann. Falls er einen eingibt, soll er zwischen zwei horizontalen Linien wieder ausgegeben werden.

9.5.7 Eingabe verarbeiten

Erzeugen Sie ein Eingabeformular mit einem Radio (Optionen: I will be back), einem Check-Balken (ene, mene, muh) und einem Runterfallmenü (Optionen: PHP ist toll, PHP ist spitze, PHP ist das Beste), und zeigen Sie, was der User gewählt hatte.

9.5.8 Mail

Erzeugen Sie ein Formular, wo der User seine eigene Emailadresse, den Empfänger, ein Betreff und einen Text eintragen kann.

Falls Absender gesetzt ist, soll bei "GoSendIt!" die Email verschickt werden, sonst soll "Gib mir eine Rückadresse!!!" ausgegeben werden.

Falls das Senden klappt, soll "Mail erfolgreich abgeschickt!" ausgegeben werden, sonst "Mail failed for delivery."

9.5.9 Gästebuch

Basteln Sie sich `gaestebuch.html`. Wer vorbeikommt, darf Kommentare eintragen, muß aber seine email-adresse angeben. Kommentare mit dem Wort "sex" oder "porn" sollen zurückgewiesen werden.

10 Virtual networked Computing (VNC)

10.1 Windos

Wir kennen bereits Windos. Man sitzt vor einem großen, teuren, schnellen, lauten PC und kann auf das Netzwerk zugreifen.

User sitzen vor dem local Rechner. Der Server ist der remote Rechner.

Wenn man eine große Datei bearbeiten möchte, muß man sie auf den lokalen Rechner holen (das dauert...) und das Anwendungsprogramm dazu (das dauert länger.....). Und dann geht's los.

Vernünftiger wäre es, wenn

- die große Datei auf dem remote Rechner bleiben könnte, wo sie eh schon ist;
- die Anwendung auf dem remote Rechner laufen könnte. Dann könnte man sie (und die Benutzer) dort zentral administrieren;
- der remote Rechner (also einer für alle) große, teure Rechnerleistung mit viel Speicher besitzt;
- der lokale Rechner nur zur Anzeige und zum Eingeben (via Tastatur oder Maus) benutzt würde.

10.2 Telnet

Wir kennen bereits `telnet`: man sitzt vor Rechner A (lokal) und arbeitet auf Rechner B (remote), so als säße man direkt davor.

Das ist sinnvoll, wenn

- Rechner B eine deutlich höhere Verarbeitungsgeschwindigkeit hat;
- auf Rechner B große Dateien liegen, die sonst mühsam zum lokalen Rechner und zurück geschaufelt werden müßten;
- man Programme, Dateien, User und Userrechte zentral administrieren möchte;
- man Backups zentral durchführen möchte;
- man extrem billige, leistungsschwache Arbeitsplatzrechner (zB 25-MHz-386er) für anspruchsvolle Aufgaben (zB Programmentwicklung oder CAD) verwenden möchte.

10.3 Die Lösung: vnc

Leider sind die meisten User heutzutage an bunte, klickbare GUIs gewöhnt, so daß sie `telnet` befremdlich finden.

Zum Glück gibt es `vnc`, der (wie `telnet` auch) ein Server-Client-Konzept enthält:

Auf dem Hauptrechner läuft mindestens 1 `vncserver`. Dieser ist für die X-Anwendungen ein vollwertiger X-Server. Er schickt seine bunten Bilder aber nicht an den lokalen Rechner, sondern an einen remote Rechner, auf dem ein `vncclient` läuft.

Der `vncclient` wirkt für den Benutzer wie ein extrem leistungsfähiges Grafik-Terminal: mit Maus, Start-Knopf, Passwort-Abfrage und Hunderten von Programmen.

Man kann jeden 386er zum X-Terminal aufrüsten. Er braucht nicht einmal eine Festplatte, wenn die Bootdiskette geschickt geschrieben ist. Ohne Festplatte ist er zudem extrem leise.

10.4 Übung

Holen Sie sich `vncviewer` von `linuxserver`, und verbinden Sie sich mit einem der `vncserver` auf `linuxserver`. Die Displaynummern erhalten Sie durch

```
cat /etc/X11/xdm/Xservers
```

oder

```
ps -ax | grep Xvnc
```

11 Common Internet File System (CIFS oder Samba)

11.1 Zweck

Microsoft hat ein eigenes Netzwerk-Management erfunden, das die Nachteile der Unix-Sicherheit (in diesen Rechner kommst du nur mit dem richtigen Passwort) und der NIS-Sicherheit (???) verbindet.

Mehrere Rechner sind dabei zu einer Workgroup zusammengeschlossen. Die Workgroup kann eine Domain sein. Eine Domain wird durch einen Domain Controller kontrolliert. In diesem Fall braucht man sich nur einmal am Netzwerk anmelden und hat dann Zugriff auf die freigegebenen Verzeichnisse und Drucker ("Shares").

Definition:

Eine Share ist etwas, das man mit anderen Benutzern teilen möchte. Das können Verzeichnisse oder Drucker sein. Man kann sie mit oder ohne Passwort einrichten und zum Lesen oder Schreiben freigeben.

11.2 Primary Domain Controller

Der PDC weiß die zugelassenen User und Passwörter und kann so ein Netzwerk zentral verwalten. Das System ist ein schlechter Nachbau des NIS (Network Information Service) (auch bekannt als "Yellow Pages") von Sun, aber langsamer.

Er kann bestimmten Usern ein Benutzer-Profil zuordnen, das bei der Anmeldung automatisch geladen wird. In dem Profil kann man diesem Nutzer zB das Ausführen von DOS-Programmen, das Ändern des Bildschirm-Hintergrundes oder das Suchen nach Dateien verbieten.

Falls der PDC (Primary Domain Controller) abstürzt, kann ein BDC (Backup Domain Controller) einspringen. Das Konzept erscheint in der Praxis etwas unausgereift. Man hat schon ganze Domains durch den Absturz eines einzigen PDC lahmgelegt.

11.3 Master Browser

In einer besonderen Sicht kann man sich alle freigegebenen Shares innerhalb der Workgroup ansehen. Diese Sicht wird vom Master Browser erzeugt, der Anfragen auf Port 139 bearbeitet. Port 139 gehört zu den meistgescannten Ports im Internet.

Als Master Browser wird einer der Rechner im Netzwerk aufgrund bekannter Kriterien gewählt. Die anderen Rechner melden ihm dann, welche Shares aktuell freigegeben sind.

Wenn ein Rechner neu dazukommt, erzwingt das eine erneute Wahl des Master Browsers. Wenn ein Rechner herunterfährt, versucht er sich beim Master Browser abzumel-

den.

Da der Master Browser (unter Windos) gelegentlich abstürzt, wird in festgelegten Zeitintervallen (ca. 15 min) der Master Browser erneut gewählt. Häufig, aber nicht immer, ist der PDC auch der Master Browser.

11.4 Samba

Unter Linux emuliert die Software-Suite Samba einen File- und Druckerserver, der (bei neueren Versionen) einen PDC vollständig ersetzen kann. Er ist bedeutend günstiger in der Anschaffung und zuverlässiger im Betrieb.

11.5 Übung

11.5.1 Samba

Vorsicht! Bei dieser Übung können Sie alle Ihre Dateien versehentlich löschen!!!

Finden Sie den Status von Samba auf `linuxserver` heraus mit

```
/etc/rc.d/init.d smb status
```

Falls er nicht läuft, bitten Sie den Sysop, Samba zu starten.

Falls er läuft, verbinden Sie sich mit `\\linuxserver\deiname`, und sehen Sie sich Ihre Dateien einmal auf diese Weise an.

Geben Sie Verzeichnisse und ggf. Drucker frei, mit Passwort und ohne.