

# UNIX-Grundlagen-Referenz

## **ls [optionen] [namen]:**

**ls** zeigt den Inhalt von Verzeichnissen. Sind keine *namen* angegeben, werden die Dateien im aktuellen Verzeichnis aufgelistet. Sind eine oder mehrere *namen* angegeben, werden entweder die Dateien im Verzeichnis *name* aufgelistet oder alle Dateien, deren Name zu *name* paßt.

**ls -l:** Gibt Daten im erweiterten Format aus. Dies beinhaltet die Angabe von Zugriffsrechten, Besitzer, Länge, Zeitpunkt der letzten Änderung und mehr.

## **cd [verzeichnis]:**

**cd** wechselt das Verzeichnis nach *verzeichnis*. Ohne Angabe eines *verzeichnisses*, wird in das Heimatverzeichnis des Benutzers gewechselt.

## **mkdir verzeichnis:**

**mkdir** legt ein Verzeichnis mit dem Namen *verzeichnis* an. Dies ist allerdings nur möglich, wenn der Benutzer eine Schreibberechtigung im Elternverzeichnis hat.

## **rmdir verzeichnis:**

**rmdir** löscht das angegebene *verzeichnis* (nicht dessen Inhalt). Zum Löschen eines *verzeichnisses* muß dieses leer sein (siehe **rm**-Befehl).

## **rm [option] daten:**

**rm** löscht eine oder mehrere *dateien*. Um eine *datei* löschen zu können, wird eine Schreibberechtigung für das Verzeichnis benötigt, indem die *datei* liegt.

## **rm -r verzeichnis:**

Wenn anstelle einer Datei ein *verzeichnis* auf Kommandozeile angegeben ist, werden das *verzeichnis* und sein gesamter Inhalt mit allen Unterverzeichnissen rekursiv gelöscht. **Achtung:** Dieser Befehl ist gefährlich.

## **mv quelle ziel:**

**mv** verschiebt oder benennt Dateien und Verzeichnisse um.

## **cp [option] datei1 datei2:**

### **cp [option] datei verzeichnis:**

**cp** kopiert *datei1* nach *datei2* oder kopiert eine oder mehrere *dateien* in das *verzeichnis*. Wenn das Ziel eine bestehende Datei ist, wird diese überschrieben.

## **cp -R verzeichnis1 verzeichnis2:**

Das *verzeichnis1* wird mit allen Unterverzeichnissen samt Inhalt in *verzeichnis2* rekursiv kopiert.



**chmod 777 datei -> rwxrwxrwx**

**Achtung:** Dieses volle Zugriffsrecht hat es in sich. Jeder beliebige Benutzer des Systems darf so eine *datei* auslesen, verändern und ausführen. Mit solchen Dateien werden Sicherheitslöcher im System geöffnet. Aus diesem Grund **niemals** einer Datei höhere Rechte zuordnen, als gerade notwendig.

*option -R*

**chmod -R modus verzeichnis:**

Geht rekursiv durch alle Unterverzeichnisse und führt die Modusänderung durch.

**exit:** **exit** beendet die **telnet**-Verbindung oder die aktuelle Shell.

Dies ist nur ein kleiner Überblick über die wichtigsten Befehle unter Unix. Weitere detailliertere Dokumentationen findet man auf den 'manpages' (**man**).

**man befehl:**

**man** zeigt die "Online-Handbücher" an.

**telnet [host]:**

Das Programm **telnet** ist die Benutzerschnittstelle, über die die Kommunikation mit einem Netzwerkrechner abgewickelt wird. Wenn **telnet** ohne Angabe eines *Hosts* (z.B. rfhs0005) aufgerufen wird, startet es in den Befehlsmodus, was durch die Eingabeaufforderung `telnet>` angezeigt wird. Gefolgt von der Eingabe 'open' erscheint die Frage nach dem *Host* (to). Nach Eingabe von z.B. rfhs0005 wird die Verbindung aufgebaut und es erscheint ein Login.

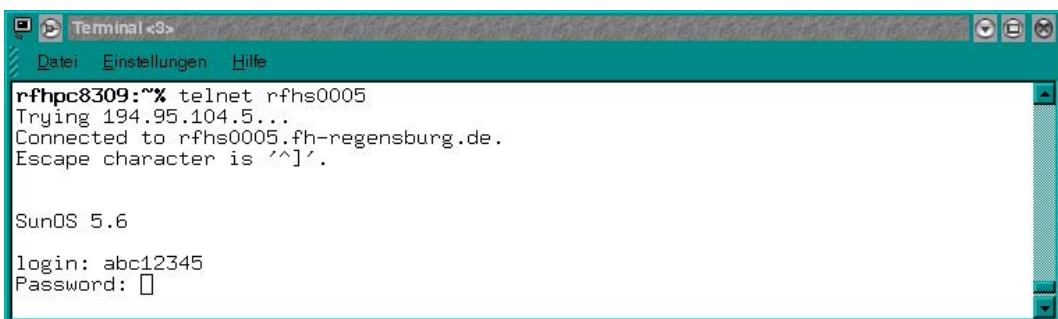


```
r-fhpc8309:~% telnet
telnet> open
(to) rfhs0005
Trying 194.95.104.5...
Connected to rfhs0005.fh-regensburg.de.
Escape character is '^['.

SunOS 5.6

login: abc12345
Password: █
```

Einfacher ist es, den *host* sofort beim Aufruf von **telnet** mit anzugeben.



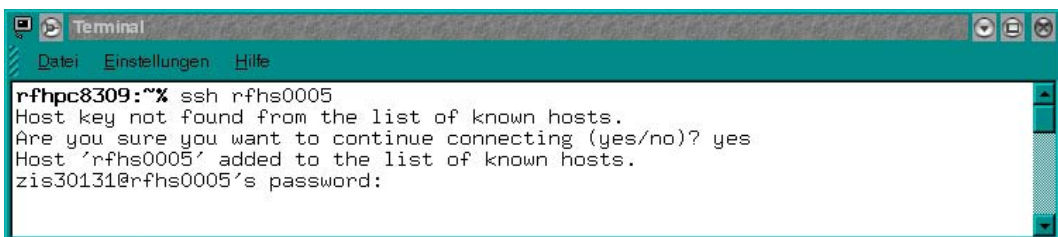
```
r-fhpc8309:~% telnet rfhs0005
Trying 194.95.104.5...
Connected to rfhs0005.fh-regensburg.de.
Escape character is '^['.

SunOS 5.6

login: abc12345
Password: █
```

**ssh [host]:**

**ssh (secure-shell)** funktioniert analog zu **telnet**, allerdings werden bei **ssh** die Passwörter verschlüsselt übertragen, was bei **telnet** nicht der Fall ist.

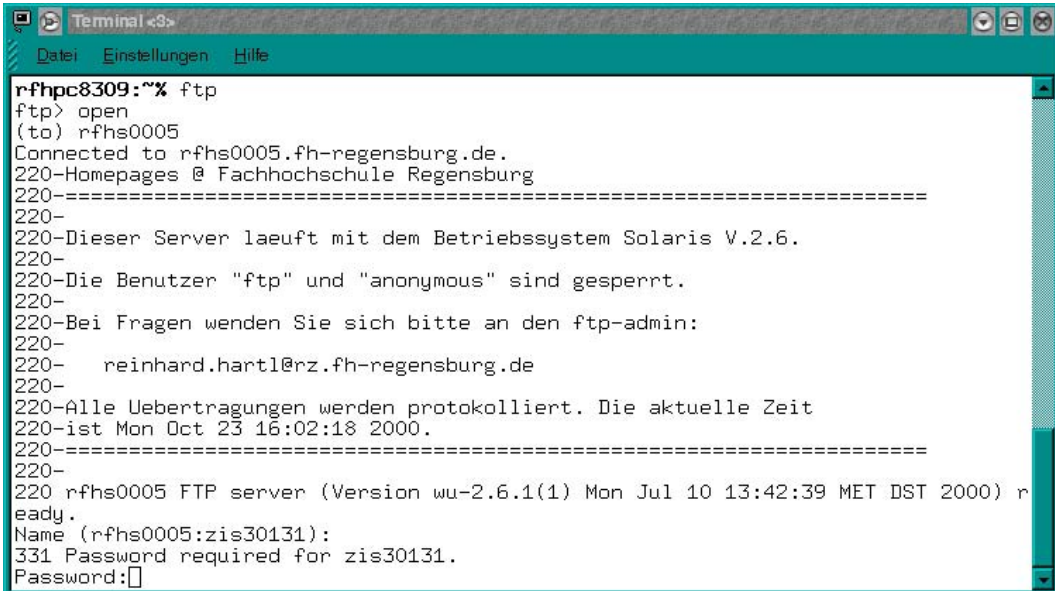


```
r-fhpc8309:~% ssh rfhs0005
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'rfhs0005' added to the list of known hosts.
zis30131@rfhs0005's password:
```

**ftp [host]:**

Das Programm **ftp (file transfer protocol)** überträgt Dateien zwischen dem lokalen Rechner und einem entfernten *Netzwerkrechner*. Das Einwählen erfolgt wie bei telnet.

Ohne Angabe eines *Hosts* (z.B. rfhs0005), startet es in den Befehlsmodus, ftp> erscheint. Gefolgt von der Eingabe 'open' erscheint die Frage nach dem *Host* (to). Nach Angabe von z.B. rfhs0005 wird die Verbindung aufgebaut und es erscheint ein Login.



```

r-fhpc8309:~% ftp
ftp> open
(to) rfhs0005
Connected to rfhs0005.fh-regensburg.de.
220-Homepages @ Fachhochschule Regensburg
220-=====
220-
220-Dieser Server laeuft mit dem Betriebssystem Solaris V.2.6.
220-
220-Die Benutzer "ftp" und "anonymous" sind gesperrt.
220-
220-Bei Fragen wenden Sie sich bitte an den ftp-admin:
220-
220-   reinhard.hartl@rz.fh-regensburg.de
220-
220-Alle Uebertragungen werden protokolliert. Die aktuelle Zeit
220-ist Mon Oct 23 16:02:18 2000.
220-=====
220-
220-rfhs0005 FTP server (Version wu-2.6.1(1) Mon Jul 10 13:42:39 MET DST 2000) r
eady.
Name (rfhs0005:zis30131):
331 Password required for zis30131.
Password:

```

Einfacher ist es, den *host* sofort beim Aufruf von **ftp** mit anzugeben.

**ascii:** Setzt den Transfertyp auf **ASCII**. Textdateien werden hiermit übertragen. (Voreinstellung)

**binary:** Setzt den Tranfertyp auf **binär**. Graphiken oder Programme werden binär übertragen.

**get entfernte\_datei [lokale\_datei]:**

**get** holt die *entfernte\_datei* von der Gegenstelle und speichert sie auf dem lokalen Rechner. Wenn die *lokale\_datei* nicht angegeben ist, wird der Name der *entfernten\_datei* verwendet.

**put lokale\_datei [externe\_datei]:**

**put** speichert eine *lokale\_datei* auf der Gegenstelle. Wenn kein Name für die *externe\_datei* angegeben ist, wird der lokale Dateiname verwendet.

**bye:** **bye** beendet die **ftp**-Verbindung.