



Zugriff auf Daten auf den Rechnern des FB Informatik der FH Wiesbaden

Zugriff per **SSH/SCP/SFTP**



Inhaltsverzeichnis

Zugriff auf Daten auf den Rechnern des FB Informatik der FH Wiesbaden.....	1
Zugriff per SSH/SCP/SFTP.....	1
1. Allgemeines.....	3
1.1. Symbole und Formatierungen in diesem Dokument.....	3
1.2. Der SSH-Zugang an der FH Wiesbaden.....	4
1.2.1. Die Rechnernamen.....	4
1.2.2. Unterstützte Protokolle.....	4
2. Die Zugriffsmethoden.....	5
2.1. SSH – Secure Shell	5
2.2. SCP – Secure Copy.....	7
2.3. SFTP – Secure File Transfer Protocol.....	9
2.4. SSH-Tunnel.....	11
3. Konfiguration und Nutzung von graphischen SSH-Clients.....	12
3.1. PuTTY (Windows).....	12
3.1.1. Installation.....	12
3.1.2. Einrichten einer SSH-Verbindung.....	12
3.1.3. Einrichten eines SSH-Tunnels.....	15
3.2. WinSCP (Windows)	17
3.2.1. Installation.....	17
3.2.2. Konfiguration.....	17
3.2.3. Bedienung.....	21
3.2.3.1. Beschreibung der Oberfläche.....	21
3.2.3.2. Dateien zwischen lokalem und entferntem Rechner kopieren.....	23
3.2.3.3. Weitere Funktionen von WinSCP.....	26
3.3. Konqueror (Linux, KDE)	27
3.3.1. Installation.....	27
3.3.2. Konfiguration.....	27
3.3.3. Benutzung von SCP mit Konqueror.....	28
4. Anhang.....	32
4.1. Kostenlose SSH-Clients.....	32
4.2. Kostenlose SCP/SFTP-Clients.....	33
4.3. Links.....	34

1. Allgemeines

Jeder Student des Fachbereichs Informatik an der FH Wiesbaden besitzt ein Homeverzeichnis zum Ablegen von eigenen Dateien. Auf diese und weitere freigegebene Dateien, wie Skripte von Professoren, kann man auch von zu Hause aus zugreifen. Diese Dokumentation beschreibt, wie dies funktioniert.

1.1. Symbole und Formatierungen in diesem Dokument



Das Ausrufezeichen weist auf wichtige Anmerkungen hin.



Das Informationszeichen soll Dich auf Tipps aufmerksam machen.

kursiv

Kursiver Text stellt eine Eingabe oder einen anklickenden Link dar, den Du in der Weboberfläche anklicken mußt.

blau

Gibt eine URL/Internet-Adresse oder eine E-Mail-Adresse an.

fett

Wichtige Begriffe werden fett dargestellt.

Die in dieser Dokumentation in den Beispielen und Screenshots verwendeten Namen und E-Mail-Adressen sind frei erfunden und sollen nur der Illustration dienen.

1.2. Der SSH-Zugang an der FH Wiesbaden


Dieses Kapitel soll einen kurzen Überblick über den vom FB Informatik der FH Wiesbaden angebotenen SSH-Zugang geben.

1.2.1. Die Rechnernamen

Zum Zugriff von außen stehen die drei Rechner *login1* und *login2* (Wiesbaden) und *login1* (Rüsselsheim) zur Verfügung:

- `login1.informatik.fh-wiesbaden.de`
- `login2.informatik.fh-wiesbaden.de`
- `login1.mi.fh-wiesbaden.de`

Innerhalb der FH Wiesbaden genügt es, die Namen *login1* oder *login2* anzugeben.

 Die Rechnernamen wie *fizzie* oder *adorno* sollten nicht mehr verwendet werden, da diese sich im Gegensatz zu oben genannten Aliasen (= andere Namen) ändern können.


1.2.2. Unterstützte Protokolle

Aus Sicherheitsgründen sollte der SSH-Zugriff nur mit Protokoll Version 2 (*SSH2*) erfolgen. Alle gängigen SSH-Implementationen und Clients unterstützen dies bzw. verwenden automatisch SSH2.


2. Die Zugriffsmethoden

In diesem Kapitel werden die möglichen Zugriffsmethoden beschrieben und anhand der Kommandozeilen-Implementation von OpenSSH beschrieben.

Detaillierte Erläuterungen einzelner Clients und Programme erfolgen gesondert in Kapitel 4.

 Alle folgenden Beispiele werden anhand des Rechners *login1.informatik.fh-wiesbaden.de* erläutert, funktionieren aber auch mit den beiden anderen oben angegebenen Login-Rechnern des Fachbereichs.

OpenSSH ist die Standardimplementierung des SSH-Protokolls unter Unix/Linux. Eine Portierung auf Windows ohne Cygwin ist ebenfalls erhältlich (siehe 4. Anhang). Ansonsten kann auch auf die Kommandozeilentools von PuTTY zurückgegriffen werden (siehe Anhang).

 Die folgenden Beschreibungen behandeln nur die gebräuchlichsten Kommandos und Parameter. Weitere Informationen und Konfigurationsmöglichkeiten findest Du in den man-Pages unter *man ssh*, *man scp* und *man sftp*.

2.1. SSH – Secure Shell

SSH (Secure Shell) ist ein Protokoll zum sicheren, verschlüsselten Zugriff auf entfernte Rechner über ein Netzwerk, z.B. Internet. Auf dem Zielsystem (hier: die Linux-Rechner in der FH Wiesbaden) lassen sich per Kommandozeile sämtliche Befehle, wie auch lokal nutzen. Mit SSH ist kein Down- bzw. Upload von Dateien möglich.

a) Öffne eine Shell unter Linux bzw. die Kommandozeile unter Windows.

b) Verbinde Dich auf den FHW-Server mit

```
ssh amust999@login1.informatik.fh-wiesbaden.de
```

Statt *amust999* musst Du natürlich Deinen Benutzernamen angeben.

Durch obigen Befehl wird eine SSH-Verbindung mit dem angegebenen Benutzernamen zum angegebenen Server aufgebaut.

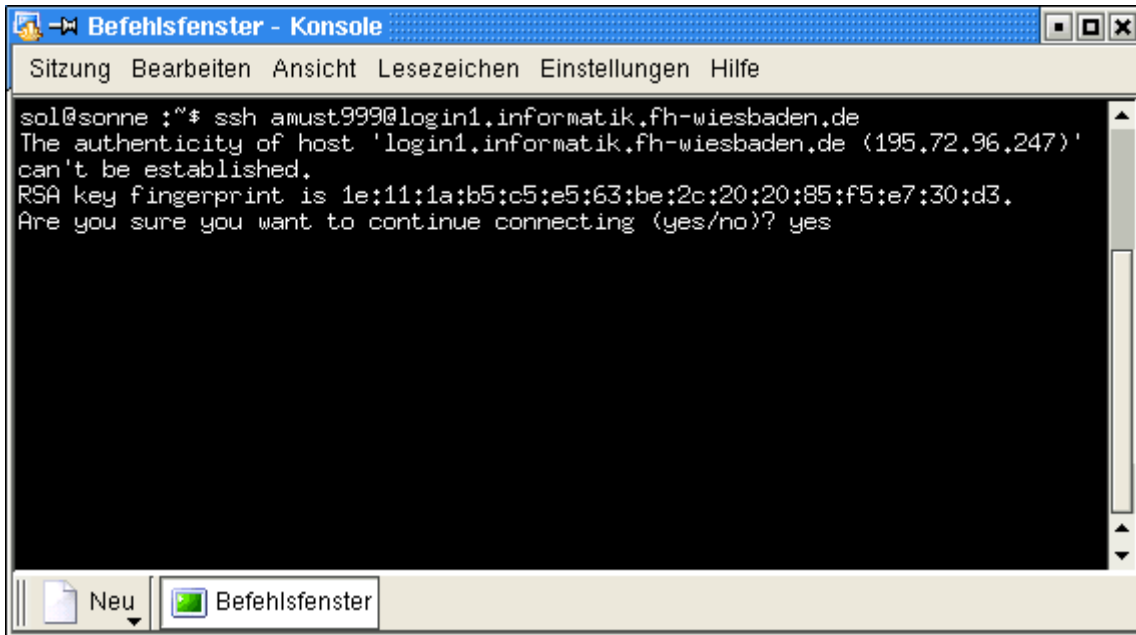
Alternativ kann man auch die Schreibweise

```
ssh -l amust999 login1.informatik.fh-wiesbaden.de
```

 verwenden.

Bei der ersten Verbindung wird gefragt, ob die Verbindung zugelassen werden soll, da der Zielrechner *login1.informatik.fh-wiesbaden.de* noch nicht bekannt ist. Beantworte die Abfrage mit *yes*, falls Du die Verbindung zulassen möchtest und *no*, wenn nicht. Im Falle von *yes* wird zukünftig keine Abfrage mehr erfolgen.

Der angezeigte RSA key fingerprint dient dazu, die Identität des Login-Rechners festzustellen und somit Manipulationsversuche (Man-in-the-Middle, IP-Spoofing etc.) zu erkennen, bei denen sich ein fremder Rechner als derjenige des FB Informatik ausgibt.



```
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
sol@sonne :~$ ssh amust999@login1.informatik.fh-wiesbaden.de
The authenticity of host 'login1.informatik.fh-wiesbaden.de (195.72.96.247)'
can't be established.
RSA key fingerprint is 1e:11:1a:b5:c5:e5:63:be:2c:20:20:85:f5:e7:30:d3.
Are you sure you want to continue connecting (yes/no)? yes
```

Um auf die FHW-Server zugreifen zu können, muss der SSH-Client die **SSH-Version 2** unterstützen. Viele Clients können dies und benutzen beim Zugriff automatisch diese Protokoll-Version (so auch OpenSSH für Windows). Sollte es beim Verbindungsaufbau zu einem Fehler ("*Disconnecting: Corrupted check bytes on input*" o.ä.) kommen, dann musst Du explizit SSH2 aktivieren:

```
ssh -2 amust999@login1.informatik.fh-wiesbaden.de
```

```
bzw. ssh -2 -1 amust999 login1.informatik.fh-wiesbaden.de
```

c) Nach Eingabe Deines Passwortes (das gleiche, mit dem Du Dich auch an den Linux-Rechnern der FH Wiesbaden anmelden kannst) ist der Login erfolgt und Du kannst nun wie lokal auch die Shell nutzen.



```
Sitzung Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
sol@sonne :~$ ssh amust999@login1.informatik.fh-wiesbaden.de
The authenticity of host 'login1.informatik.fh-wiesbaden.de (195.72.96.247)'
can't be established.
RSA key fingerprint is 1e:11:1a:b5:c5:e5:63:be:2c:20:20:85:f5:e7:30:d3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'login1.informatik.fh-wiesbaden.de,195.72.96.247'
(RSA) to the list of known hosts.
amust999@login1.informatik.fh-wiesbaden.de's password:
Last login: Thu Oct 30 01:57:25 2003 from dsl-007-007-007-007.arcor-ip.net

Dies ist einer der beiden Login-Rechner des FB Informatik. Die zur Verfüegung
stehenden Dienste sind: ssh, telnet, ftp.

Bei Problemen bitte email an unix-admin@informatik.fh-wiesbaden.de schreiben.
amust999@fizzie(~)$
```

2.2. SCP – Secure Copy

SCP (Secure Copy) ist ein einfaches Protokoll zum sicheren Kopieren von Dateien zwischen entfernten Rechnern auf Basis von SSH. Über SCP kannst Du also Dateien zwischen Deinem Rechner zu Hause und Deinem Homeverzeichnis in der FH Wiesbaden austauschen.

Es ist jedoch nicht möglich, sich Dateilisten anzeigen zu lassen, so daß beim Kopieren vom FHW-Rechner der Dateiname und der Pfad zur Datei bekannt sein muß.

- a) Öffne eine Shell unter Linux bzw. die Kommandozeile unter Windows.
- b) Kopiere nun nach folgendem Schema entweder eine Datei vom FH-Server bzw. zum FH-Server:

```
scp user@quelle:file user@ziel:file
```

Die Angabe des *user* ist optional und kann jeweils für den lokalen Rechner weggelassen werden.

file ist der Name der zu kopierenden Datei bzw. der neue Name auf dem Zielrechner.

Beispiel 1:

Um die Datei *file.txt* vom lokalen Rechner auf den Server der FH Wiesbaden zu kopieren, ist Folgendes anzugeben:

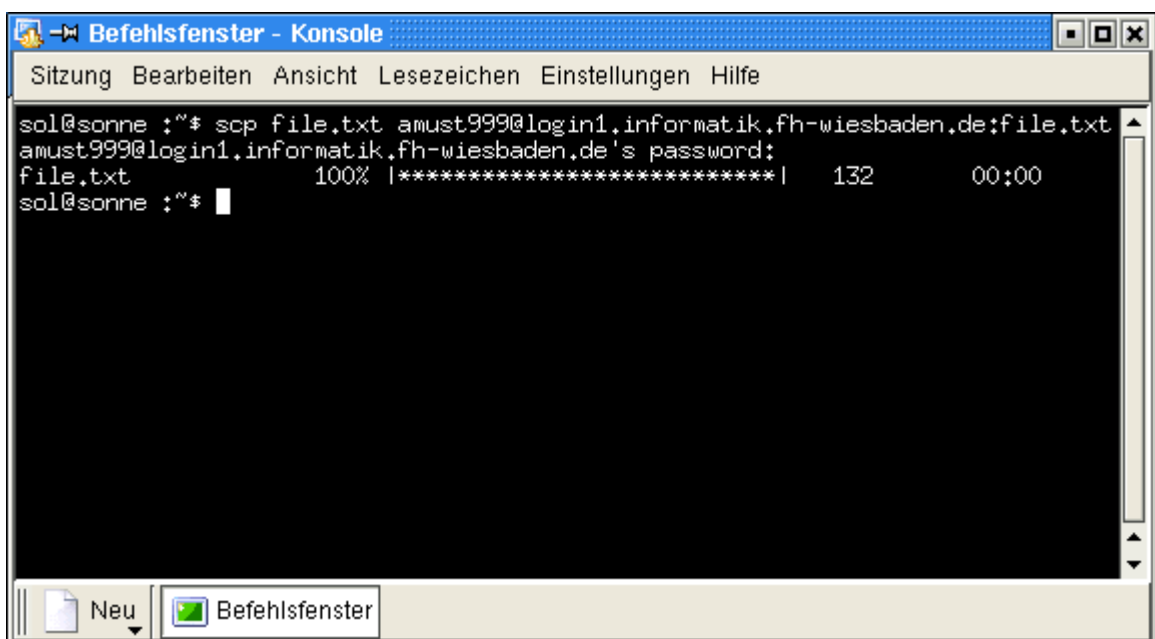
```
scp file.txt amust999@login1.informatik.fh-wiesbaden.de:file.txt
```

Beispiel 2:

Um die Datei *file.txt* vom FHW-Server herunterzukopieren, ist Folgendes anzugeben:

```
scp amust999@login1.informatik.fh-wiesbaden.de:file.txt file.txt
```

Der Fortschrittsbalken zeigt an, wieviel der Datei bereits kopiert wurde.



- ❗ Wenn Du als Quelle oder Ziel einen Datei- oder Ordnernamen mit Leerzeichen oder einem Sonderzeichen das durch die Shell interpretiert werden kann (z.B. *, ?, &) angeben möchtest, dann mußt Du diesen in einfache Anführungszeichen ' setzen.
- ❗ Mit dem Parameter *-r* kannst Du rekursiv den Inhalt ganzer Verzeichnisse und deren Unterverzeichnisse kopieren. So lädt das Kommando

```
scp -r amust999@login1.informatik.fh-wiesbaden.de:~/files /  
~/downloads/
```

das Verzeichnis *~/files/* und alle darunter befindlichen Dateien und Verzeichnisse nach *~/downloads/* herunter.

```
scp -r amust999@login1.informatik.fh-wiesbaden.de:'~/files/*'  
~/downloads/
```

würde hingegen alle Dateien aus *~/files/* nach *~/downloads/* herunterladen, jedoch ohne das Verzeichnis *~/files/* selbst.

2.3. SFTP – Secure File Transfer Protocol

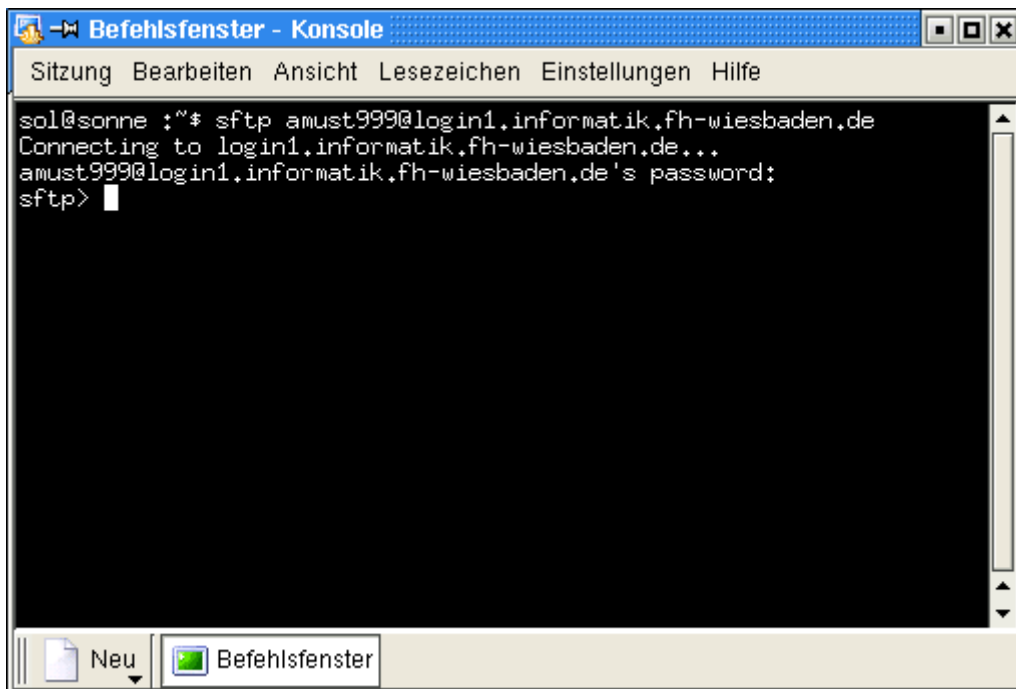
Komfortabler als SCP ist SFTP (Secure File Transfer Protocol) – die sichere Alternative zu FTP.

Nach dem erfolgreichen Einloggen befindest Du Dich in einer interaktiven Shell, in der Du die meisten der vom FTP-Befehl bekannten Kommandos ausführen kannst, u.a. auch Wechseln/Anzeigen des Verzeichnisses und natürlich Up- und Download von Dateien.

- a) Öffne eine Shell unter Linux bzw. die Kommandozeile unter Windows.
- b) Zum Server der FH Wiesbaden kannst Du Dich so verbinden:

```
sftp amust999@login1.informatik.fh-wiesbaden.de
```

Nach Eingabe Deines Passwortes gelangst Du in die interaktive SFTP-Shell.



An dieser Stelle möchte ich die wichtigsten Kommandos kurz ansprechen, die Du in der SFTP-Shell verwenden kannst.

i Wenn Du als Quelle oder Ziel einen Datei- oder Ordernamen mit Leerzeichen oder einem Sonderzeichen, das durch die Shell interpretiert werden kann (z.B. *, ?, &) angeben möchtest, dann mußt Du diesen in einfache Anführungszeichen ' setzen.

• Herunterladen einer Datei

Um eine Datei vom Server auf Deinen lokalen Rechner herunterzuladen (Download) ist das *get*-Kommando zu verwenden:


```
get remote-file [local-file]
```

remote-file gibt den Dateinamen der Datei auf dem Server an, dabei dürfen auch Wildcards verwendet werden. Der optionale Parameter *local-file* gibt den Namen bzw. den Pfad an, unter dem die Datei auf dem lokalen Rechner abgelegt werden soll. Wird nichts angegeben, so wird der ursprüngliche Name beibehalten.

Beispiel:

Du möchtest die Datei *file.txt* aus dem Verzeichnis *files* in Deinem Homeverzeichnis auf Deinen Windowsrechner nach *C:\Downloads* kopieren:

```
get files/file.txt C:\Downloads\
```

 Mit den bekannten Wildcards der Shell kannst Du auch Gruppen von mehreren Dateien spezifizieren, die einem gemeinsamen Namensschema gehorchen. So lassen sich z.B. mit `get *.c C:\Downloads\` alle C-Quellcodefiles herunterladen.

 Wenn Du unter Windows kein Zielverzeichnis angibst, dann wird die Datei direkt unter *C:* abgelegt.

- **Hochladen einer Datei**

Um eine Datei von Deinem lokalen Rechner auf den Server hochzuladen (Upload) ist das *put*-Kommando zu verwenden:

```
put local-file [remote-file]
```

local-file gibt die Datei auf dem lokalen Rechner an, die Du unter dem optionalen Namen *remote-file* auf dem Server ablegen möchtest.

Beispiel:

Du möchtest die Datei *file.txt* aus dem Ordner *C:\Downloads* (Windowsrechner) auf den Server unter dem Verzeichnis *files* in Deinem Homeverzeichnis kopieren:

```
put C:\Downloads\file.txt files/
```

- **Aktuellen Pfad wechseln**

Mit *cd pfad* wechselst Du auf dem Server ins Verzeichnis *pfad* und mit *lcd pfad* kannst Du auf dem lokalen Rechner nach *pfad* wechseln. Dadurch kannst Du Dir z.B. die Angabe von langen Verzeichnisnamen beim Down- oder Upload sparen.

- **Aktuellen Pfad anzeigen**

Mit *pwd* kannst Du Dir den aktuellen Arbeitspfad auf dem Server und mit *lpwd* denjenigen auf dem lokalen Rechner anzeigen lassen.

- **Verzeichnisinhalte anzeigen lassen**

Das Kommando *ls pfad* zeigt Dir den Inhalt des entfernten Verzeichnisses *pfad* an und *lls pfad* den des lokalen Verzeichnisses *pfad*. Ist kein *pfad* angegeben, so wird der Inhalt des aktuellen Arbeitsverzeichnisses ausgegeben.

- **Beenden der SFTP-Verbindung**

bye und *exit* beenden die SFTP-Verbindung.

Weitere Kommandos findest Du in den Hilfeseiten unter *man sftp*.

2.4. SSH-Tunnel

Mit einem sogenannten SSH-Tunnel ("Port Forwarding") kann man neben einer Shell (siehe 2.1. SSH) auch andere TCP-Verbindungen über einen sicheren Kanal von und zur FH Wiesbaden übertragen. Dies hat den Vorteil, daß man so z.B. auch die internen Websites der FH Wiesbaden abfragen kann, die von außerhalb nicht einsehbar sind (siehe Fehlermeldung 403 - "*Permission Failed - You are not allowed to retrieve the document xxx from this server*" o.ä.). Dazu zählen u.a. die interne Dokumentation und Teile der Website des DOPSY-Labors von Prof. Dr. Kröger (siehe Anhang 4.3).

- a) Öffne eine Shell unter Linux bzw. die Kommandozeile unter Windows.
- b) Baue nun mit folgendem Befehl (bitte in einer Zeile eingeben) einen SSH-Tunnel zum internen Dokumentationsserver der FH Wiesbaden auf:

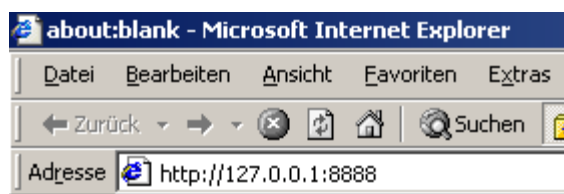
```
ssh amust999@login1.informatik.fh-wiesbaden.de  
-L 8888:www-intern.informatik.fh-wiesbaden.de:80
```

Erläuterung:

Mit der Option *-L* wird *www-intern.informatik.fh-wiesbaden.de* mit der internen IP des FHW-Servers namens "*login1*" vom Port *80* auf den lokalen Port *8888* (dieser kann frei zwischen 1024 und 65535 gewählt werden) auf Deinen Rechner umgeleitet.

amust999 mußt Du natürlich durch Deinen eigenen Benutzernamen ersetzen.

- c) Nach Eingabe Deines Passwortes erscheint wie bei einer gewöhnlichen SSH-Verbindung eine Shell.
- d) Wechsele nun zu Deinem Webbrowser (Internet Explorer, Mozilla, Opera o.ä.). Die zuvor geöffnete Shell darf dabei nicht geschlossen werden! Um die getunnelte Website anschauen zu können, gibst Du einfach <http://127.0.0.1:8888> (oder <http://localhost:8888>) in die Adresszeile des Browsers ein.



Wenn Du beim Aufbau des Tunnels (Abschnitt b) einen anderen lokalen Port als *8888* gewählt hast, dann ist die obige Adresse entsprechend anzupassen.

- e) Um den Tunnel abzubauen, genügt es, die entfernte Shell per *exit* zu schließen.

3. Konfiguration und Nutzung von graphischen SSH-Clients

Dieses Kapitel soll die Installation, Konfiguration und Benutzung von ausgewählten SSH-, SCP- und SFTP-Clients schildern. Diese stellen ein graphisches Frontend für oben erläuterte Kommandos zur Verfügung und bieten dementsprechenden Komfort.

3.1. PuTTY (Windows)

PuTTY ist eine freie Implementation des SSH-Protokolls und bietet eine graphische Oberfläche zum Verwalten und Konfigurieren von verschiedenen Verbindungen.

Neben dem SSH-Client werden auch Kommandozeilen-Tools für SFTP und SCP, sowie Programme zum Erzeugen und Verwalten von Schlüsseln mitgeliefert, auf die aber hier nicht näher eingegangen wird.

Die jeweils aktuelle Version ist stets auf der offiziellen Homepage unter

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
zu finden.


3.1.1. Installation

Einfach die heruntergeladene Datei *putty.exe* in ein beliebiges Verzeichnis kopieren und diese per Doppelklick starten. Es werden keine weiteren Dateien eingerichtet.

3.1.2. Einrichten einer SSH-Verbindung

Im Folgenden werden alle Einstellungen beschrieben, die notwendig sind, um mit PuTTY eine SSH-Verbindung auf den Server *login1* der FH Wiesbaden aufzubauen.

Die genannten Überschriften beziehen sich auf den Namen der *Category* im linken Baum von PuTTY.

 Damit sich PuTTY Deine Einstellungen und Angaben auch nach einem Neustart merkt, mußt Du diese explizit unter *Session* über Schaltfläche *Save* unter einem frei wählbaren Namen speichern! Dies muß vor dem Verbindungsaufbau geschehen, da sich danach das Dialogfenster schließt und die Speicherung nicht mehr möglich ist.

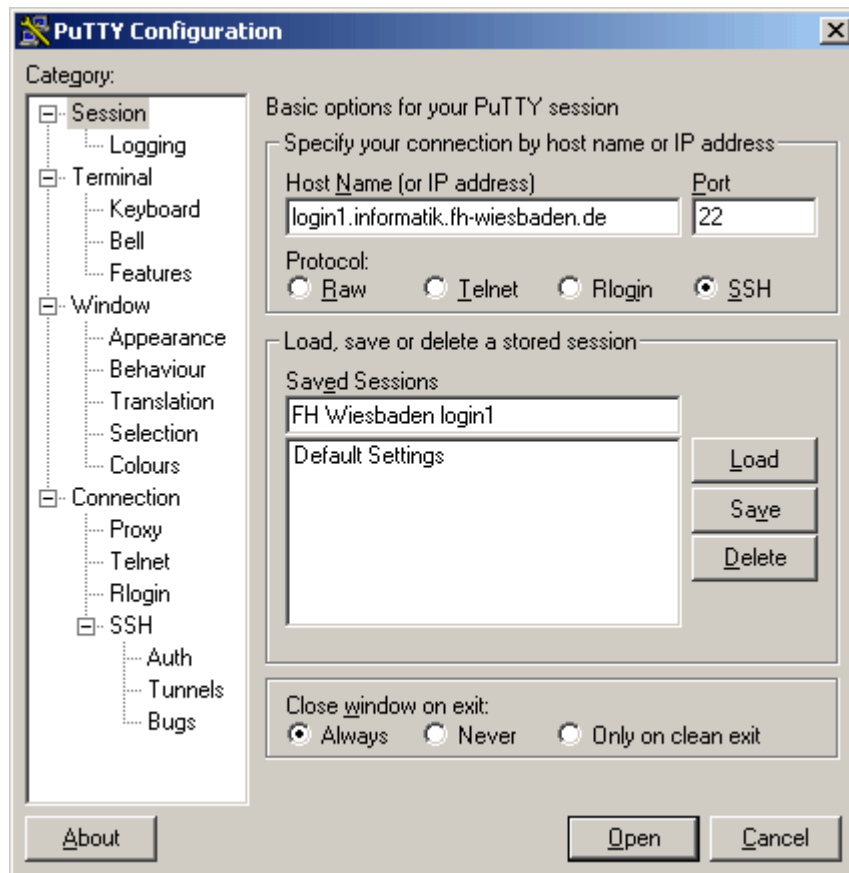
1. Session

Trage bei *Host Name* den Rechnernamen *login1.informatik.fh-wiesbaden.de* ein. Als *Protocol* ist *SSH* auszuwählen. Der *Port* sollte sich dabei selbst auf *22* einstellen, wenn nicht, dann stelle diesen Wert manuell ein.

Einstellungen speichern:

Trage bei *Saved Sessions* einen Namen ein, unter dem die Einstellungen abgelegt werden sollen, z.B. *FH Wiesbaden login1*, und speichere alles per *Save*.

Bei der Option *Close window on exit* kannst Du festlegen, ob das bei einer Verbindung angezeigte Terminalfenster (Shell) bei Beendigung geschlossen werden soll oder nicht.

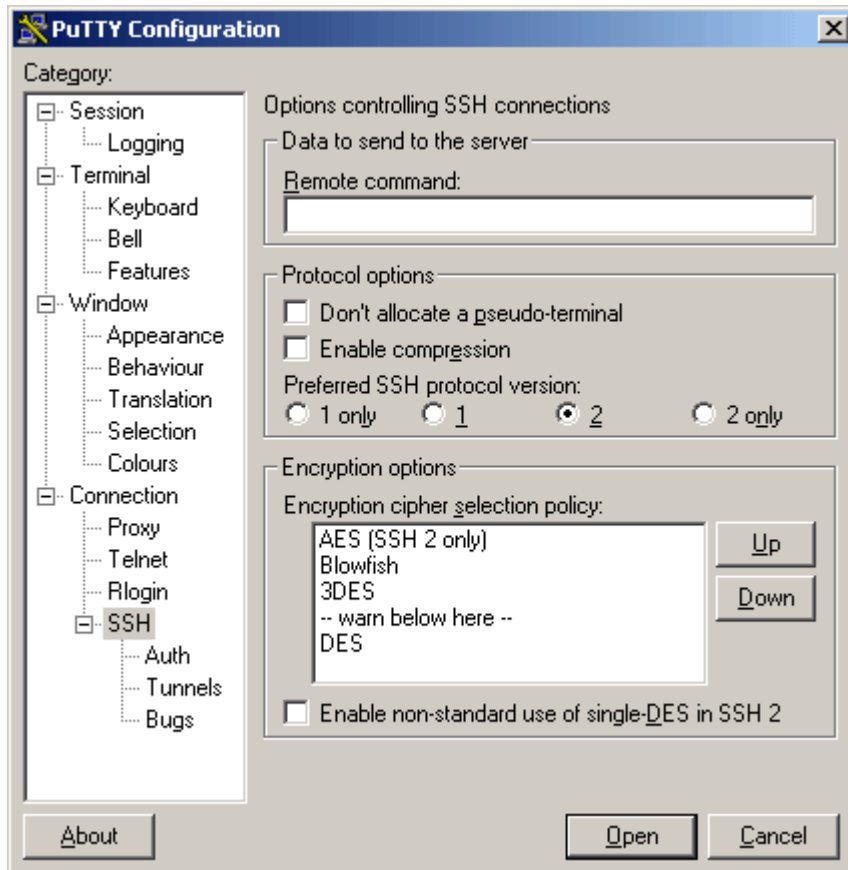


2. Connection → SSH

Setze bei *Preferred SSH protocol version* das Häkchen bei *2* oder gar *2 only*, damit das neuere SSH2 verwendet wird.

In der Auswahlbox *Encryption cipher selection policy* ordnest Du über die Schaltflächen *Up* und *Down* (den gewünschten Eintrag wählen und damit nach oben oder unten verschieben) folgendermaßen an:

- AES (SSH2 only)
- Blowfish
- 3DES
- -- warn below here --
- DES



3. Connection → SSH → Auth

Als *Authentication method* ist als einzige Option *Attempt "keyboard interactive" authentication (SSH2)* auszuwählen. Dies bewirkt, daß Du vor dem Einloggen nach Deinem Usernamen und Passwort gefragt wirst, die Du wie gewohnt per Tastatur eingeben mußt.

Auf die fortgeschrittene Möglichkeit, sich per SSH-Schlüssel (sog. Public Key) zu authentifizieren, wird in dieser Dokumentation nicht eingegangen. An dieser Stelle möchte ich auf die entsprechende offizielle PuTTY-Dokumentation unter

<http://the.earth.li/~sgtatham/putty/0.53b/html/doc/Chapter8.html>

verweisen.

4. Session

Speichere jetzt Deine Einstellungen nochmals über die Schaltfläche *Save*, bevor Du die Verbindung mit *Open* öffnest.

Nach Eingabe Deines Benutzernamens und Passwortes befindest Du Dich in einer Unix-Shell und kannst dort wie gewohnt arbeiten.



```
FH Wiesbaden
login as: amust999
amust999@login1.informatik.fh-wiesbaden.de's password:
Last login: Sat Nov  8 18:28:46 2003 from dsl-213-023-229-181.arcor-ip.net

Dies ist einer der beiden Login-Rechner des FB Informatik. Die zur Verfuegung
stehenden Dienste sind: ssh, telnet, ftp.

Bei Problemen bitte email an unix-admin@informatik.fh-wiesbaden.de schreiben.
amust999@fozzie (~) $ █
```

3.1.3. Einrichten eines SSH-Tunnels

Im Folgenden werden alle Einstellungen beschrieben, die notwendig sind, um mit PuTTY einen SSH-Tunnel zum Server *login1* der FH Wiesbaden aufzubauen, um so die interne Dokumentation des Fachbereichs im Browser anzusehen.

1. Grundeinstellungen

Führe zunächst alle Einstellungen wie im Abschnitt 3.1.2. beschrieben durch und sichere diese unter einem neuen Namen, z.B. *FH Wiesbaden Tunnel*.

2. Connection → SSH → Tunnels

Trage im Feld *X display location* den Wert *127.0.0.1* ein. Dies ist die lokale IP-Adresse Deines eigenen Rechners, auch als *localhost* bekannt.

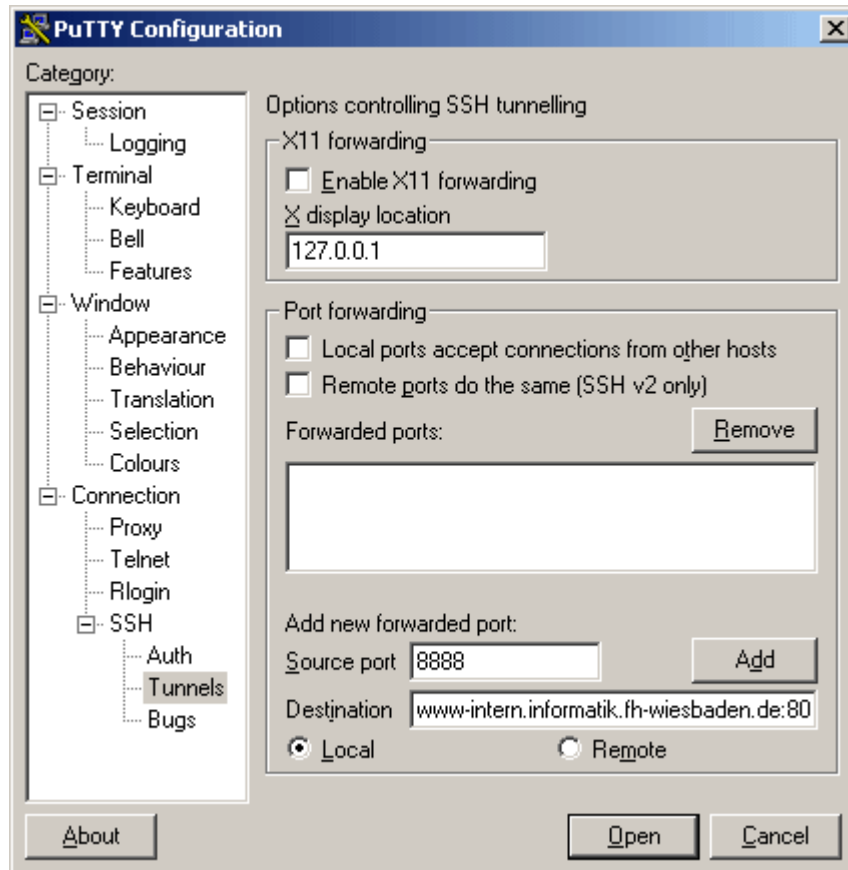
Bei *Add new forwarded port* sind für die interne Dokumentation folgende Angaben zu machen:

Source port: 8888

Destination: www-intern.informatik.fh-wiesbaden.de:80

Das untere Häkchen ist bei *Local* zu setzen. Wenn alle Angaben eingetragen sind, dann kannst Du diese über die Schaltfläche *Add* in die Liste der *Forwarded ports* aufnehmen.

Sichere jetzt Deine Einstellungen unter der Kategorie *Session* nochmals, bevor Du über den Button *Open* den SSH-Tunnel aufbaust (Eingabe von Benutzername und Passwort erforderlich).



3. Aufruf im Webbrowser

Gebe in der Adresszeile Deines Webbrowsers <http://127.0.0.1:8888> ein. Die per PuTTY geöffnete Shell darf dabei nicht geschlossen werden.



Wenn Du bei *Source port* (siehe 2.) einen anderen lokalen Port als 8888 gewählt hast, dann ist die obige Adresse entsprechend anzupassen.

Die interne Dokumentation kann nun wie innerhalb der FH Wiesbaden "durchsurft" werden.


3.2. WinSCP (Windows)

WinSCP ist ein OpenSource-Client für SFTP und SCP unter Windows. Damit lassen sich sichere Dateioperationen und Up- und Downloads zu bzw. von einem entfernten Rechner mit dem Komfort eines gewöhnlichen FTP-Clients oder Dateimanagers durchführen.

Die jeweils aktuelle Version ist unter

<http://winscp.vse.cz/eng/> bzw. <http://winscp.sourceforge.net>

zu finden.

 Die folgenden Beschreibungen beziehen sich auf die Programmversion 4.3.2. Da WinSCP beständig weiterentwickelt wird, ist darauf zu achten, daß Du stets die aktuellste Version installierst.

3.2.1. Installation

Kopiere die heruntergeladene Datei *winscpXXX.exe* (XXX steht dabei für die Programmversion) in ein beliebiges Verzeichnis. Weitere Installationsschritte sind nicht auszuführen. WinSCP kann sofort per Doppelklick gestartet werden.

Alternativ kann auch der ebenfalls auf der Website verfügbare Installer verwendet werden, um die benötigte Datei auf Deinem Rechner einzurichten.

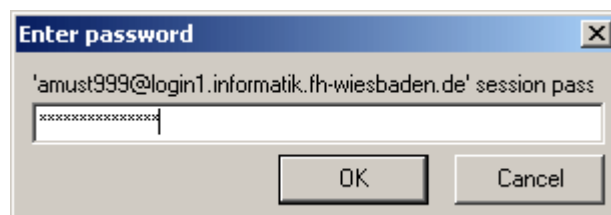
3.2.2. Konfiguration

In diesem Abschnitt wird beschrieben, wie WinSCP für eine Verbindung zur FH Wiesbaden eingerichtet werden muß.

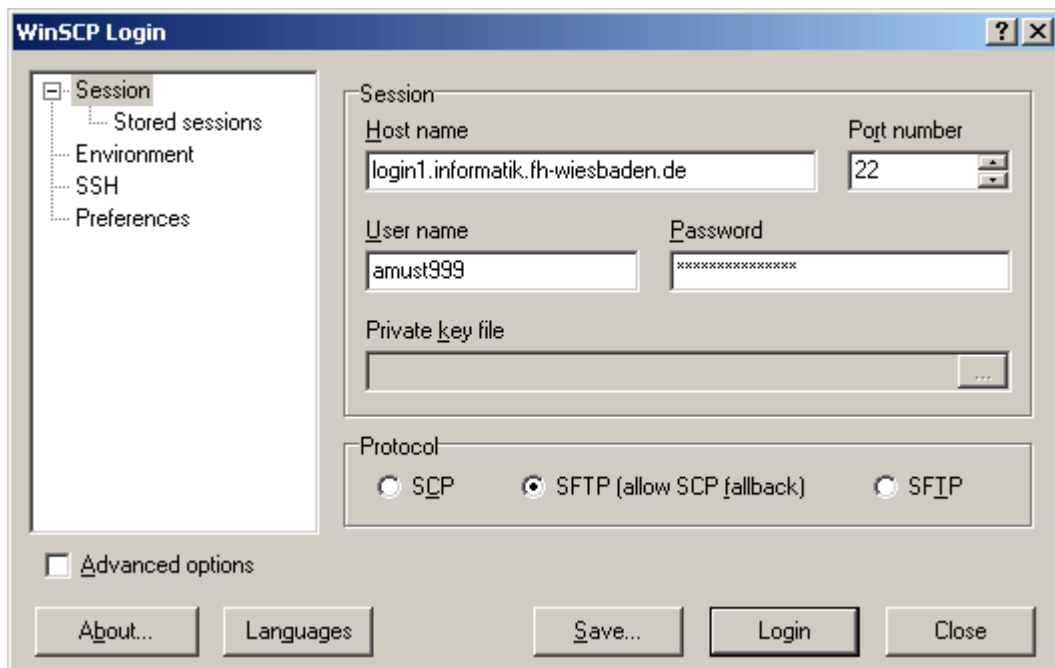
Beim ersten Programmstart erscheint der *WinSCP Login*, über den Du Deine Verbindungsdaten eintragen kannst. Diese Einstellungen werden im Folgenden näher erläutert.

1. Session

Bei *Host Name* trägst Du den Rechner *login1.informatik.fh-wiesbaden.de* (oder einen anderen der in Kapitel 1.2.1 genannten Server) ein. Die *Port number* muß auf dem Standardwert 22 eingestellt bleiben. *User name* und *Password* sind diejenigen, mit denen Du Dich an den Linuxrechnern in der FH Wiesbaden einloggen kannst. Wenn Du das Feld *Password* leer läßt, so wirst Du bei jedem Login danach gefragt:

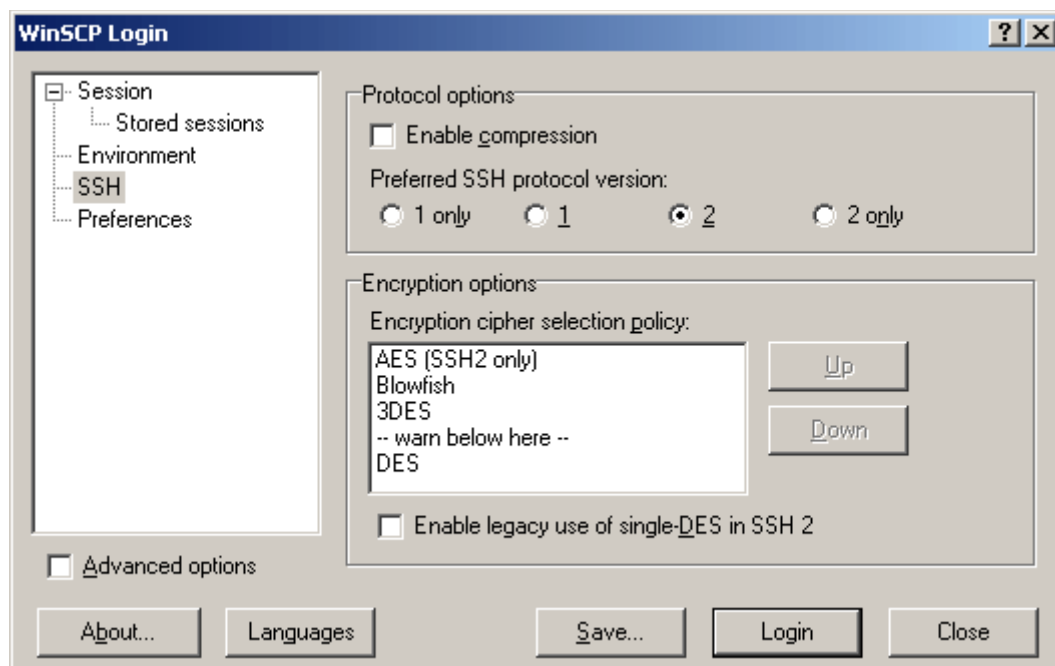


Das *Protocol* kann frei gewählt werden, ich empfehle aber *SFTP (allow SCP fallback)*. Ein Vergleich zwischen SCP und SFTP bei WinSCP befindet sich unter <http://winscp.sourceforge.net/eng/protocols.php>



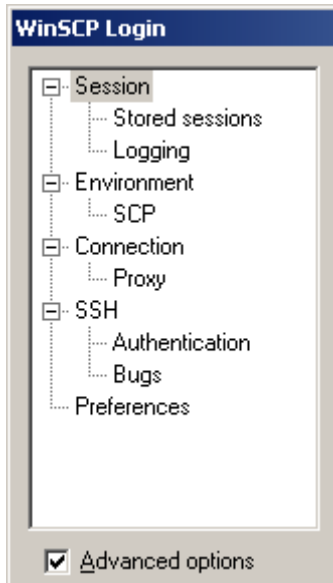
2. SSH

In der Einstellungsseite SSH solltest Du die *Preferred SSH protocol version* auf den Wert 2 oder 2 only setzen und die Einträge bei *Encryption cipher selection policy* gemäß Screenshot über die beiden Schaltflächen *Up* und *Down* anordnen.



3. Weitere Einstellungen

Unter *Environment* und *Preferences* lassen sich Einstellungen bezüglich des Aussehens und Verhaltens der Oberfläche von WinSCP tätigen. Da hier keine Änderungen vorgenommen werden müssen, wird darauf an dieser Stelle nicht näher eingegangen.

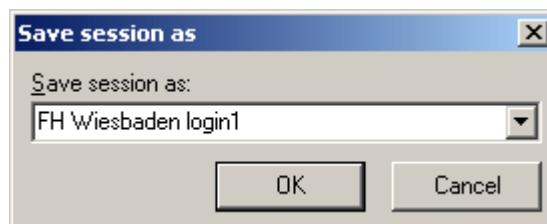


Zudem läßt sich über den Schalter *Advanced Options* der Menübaum erweitern und darüber weitere, fortgeschrittene Konfigurationen vornehmen.

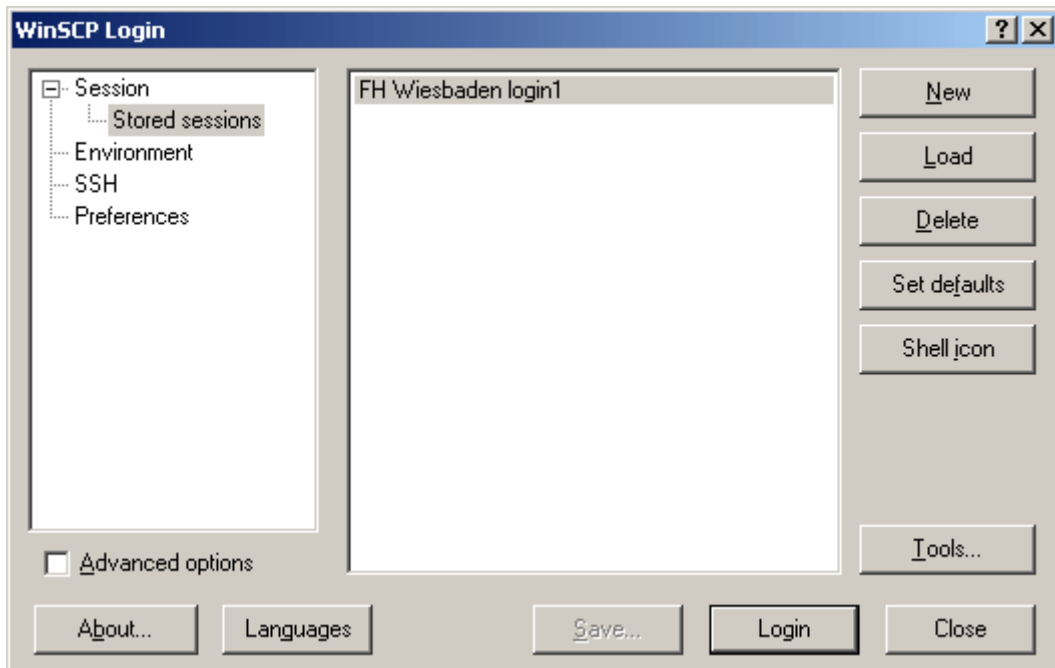
Deren Einstellungsmöglichkeiten werden in dieser Dokumentation jedoch nicht erläutert.

4. Einstellungen sichern

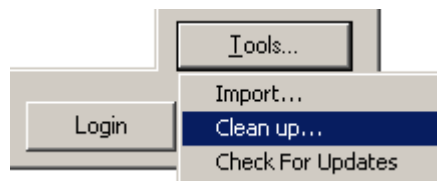
Über die Schaltfläche *Save...* kannst Du nun Deine gemachten Einstellungen sichern. Es erscheint ein Fenster, in das Du einen frei wählbaren Namen für diese Konfiguration (*session*) eingeben kannst.



Deine Einstellungen erscheinen nun unter dem gewählten Namen in der Liste der *Stored sessions* und können von dort geladen werden.



- Da WinSCP seine Einstellungen in der Registry von Windows ablegt, bleiben diese auch nach Löschen der *winscpXXX.exe* erhalten. Über *Tools... → Clean up...* und anschließender Auswahl der zu löschenden Werte können diese Einträge restlos entfernt werden.

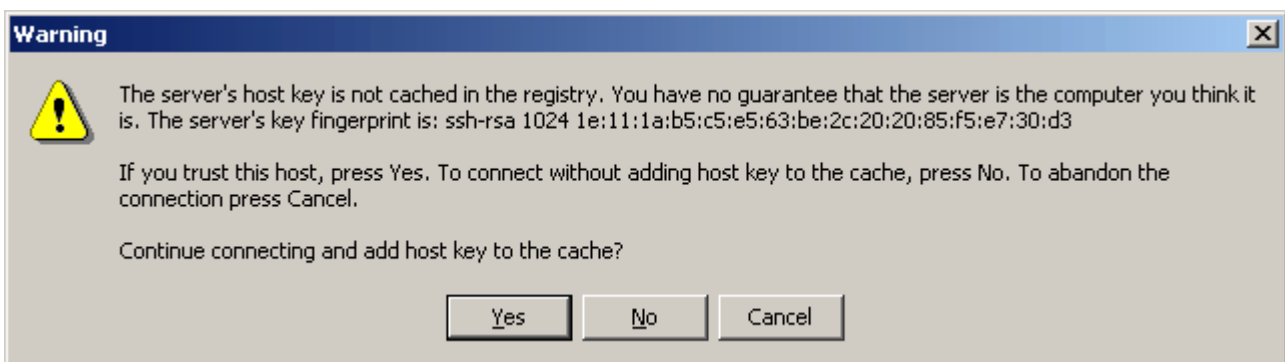


5. Verbindung aufbauen

Über die Schaltfläche *Login* wird nun die Verbindung zum Rechner der FH Wiesbaden aufgebaut.

Bei der ersten Einwahl per WinSCP wird nachgefragt, ob Du die Verbindung zu dem bislang unbekanntem Rechner zulassen möchtest. Über den angezeigten Fingerprint könntest Du die Identität des Rechners überprüfen...

Mit Klick auf *Yes* wird eine Verbindung zu diesem Rechner dauerhaft akzeptiert, mit *No* nur dieses eine Mal. Mit *Cancel* wird der Einwahlversuch abgebrochen.

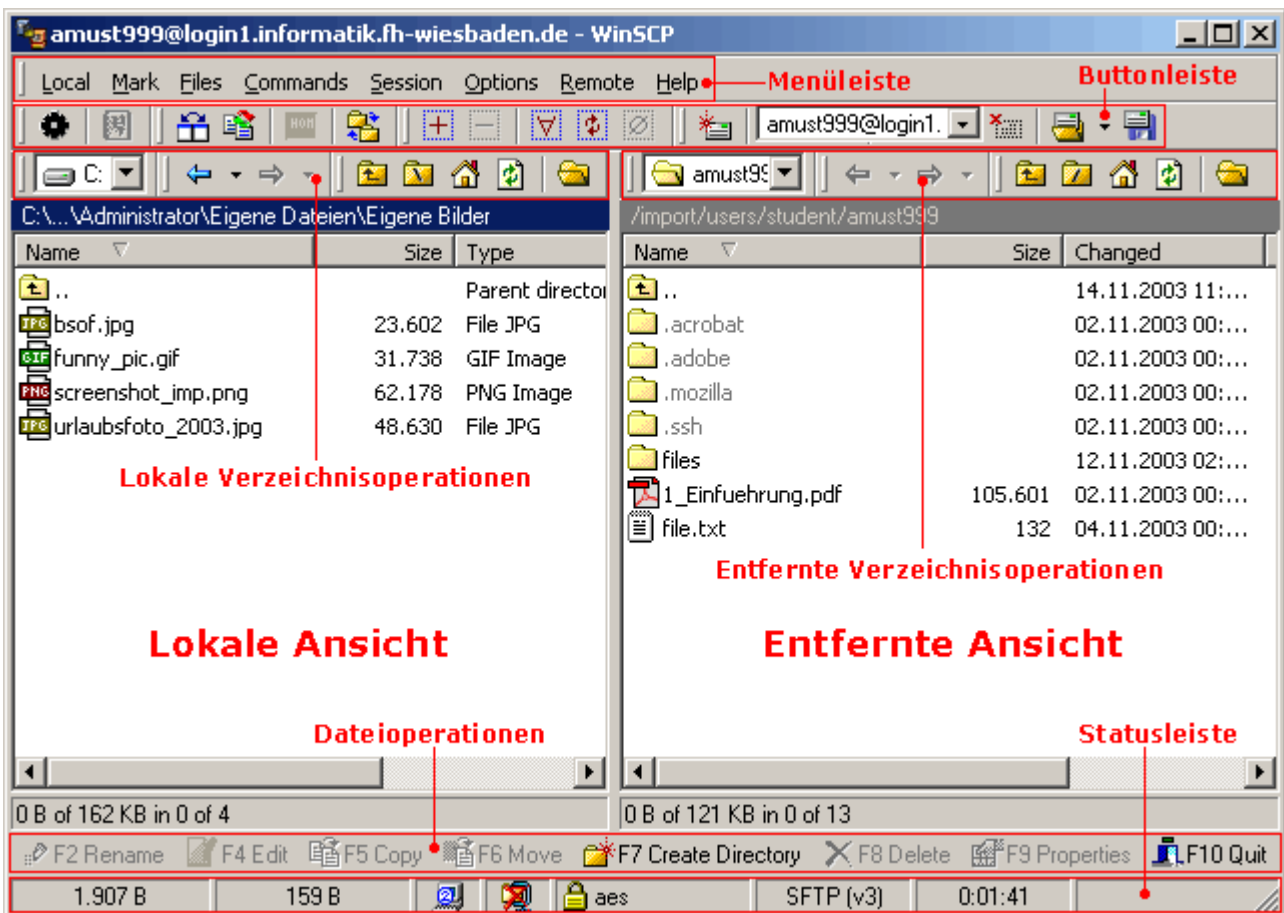


3.2.3. Bedienung

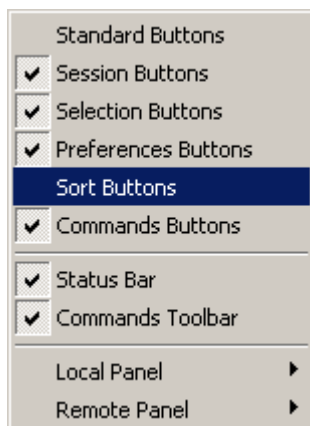
In diesem Kapitel werden die wichtigsten Bedienelemente von WinSCP erläutert, so daß danach die gängigsten Dateioperationen durchgeführt werden können.

3.2.3.1. Beschreibung der Oberfläche

Nach dem erfolgreichen Einloggen wird Dir eine FTP-Client-ähnliche Oberfläche geboten:



Die *Menüleiste* bietet Zugriff auf alle Funktionen von WinSCP. Schnelleres Arbeiten wird durch die einzelnen Schaltflächen in der *Buttonleiste* ermöglicht. Wenn Du kurz mit dem Mauszeiger über einer Taste verweilst, wird angezeigt, welche Funktion diese hat (sog. *Tooltip*).



Anpassen an die persönlichen Bedürfnisse kann man diese über den Menüpunkt *Options* → *Toolbars* bzw. per Klick mit der rechten Maustaste auf die *Buttonleiste*.

In der Auswahl einfach durch Klick auf den Namen der entsprechenden Buttonleiste klicken, um diese ein- bzw. auszublenden.

Die Dateiansicht ist in zwei Hälften geteilt: Links die *lokale Ansicht* Deines Rechners und rechts die Ansicht des *entfernten* Rechners. Die Verzeichnisse wechseln kannst Du wie aus dem Windows-Explorer gewohnt per Doppelclick auf ein Verzeichnis und jeweils über die Schaltflächen für *lokale und entfernte Verzeichnisoperationen*.



Wechselt vor und zurück in der Liste der zuletzt besuchten Verzeichnisse (History).



Wechselt eine Verzeichnisebene nach oben.



Wechselt in das Rootverzeichnis. Unter Linux ist dies / und unter Windows das jeweilige Laufwerk, z.B. C:\



Wechselt in Dein Homeverzeichnis. Unter Linux in ~ und unter Windows nach "*Eigene Dateien*" des jeweils angemeldeten Benutzers.



Aktualisiert die Verzeichnisansicht.



Öffnet ein Fenster, in dem Du Lesezeichen von häufig benötigten Verzeichnissen anlegen, verwalten und öffnen kannst.



Zeigt unter Linux alle übergeordneten Verzeichnisse bis zum Rootverzeichnis an und unter Windows alle existierenden Laufwerksbuchstaben.

Über die Schaltflächen *Dateioperationen* kannst Du die gängigen Dateioperationen wie Bearbeiten, Löschen sowie Kopieren und Verschieben (lokal ↔ entfernt) mit der bzw. den gerade ausgewählten Datei(en) auf der aktiven Seite ausführen.





Welche Seite – die lokale oder entfernte Ansicht – gerade aktiv ist, kannst Du an der blau (statt grau) hinterlegten Pfadangabe über der Dateiliste erkennen. Im Screenshot auf der vorherigen Seite ist die lokale Ansicht links aktiv.

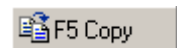
Die *Statusleiste* zeigt Dir diverse Informationen zur aktuellen Verbindung an bzw. beschreibt kurz zusätzlich zum Tooltip, was eine Schaltfläche bewirkt, wenn Du mit dem Mauszeiger darüber stehen bleibst.


3.2.3.2. Dateien zwischen lokalem und entferntem Rechner kopieren

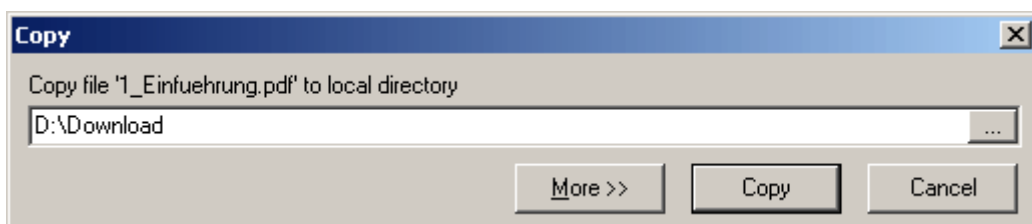
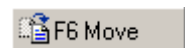
Um eine Datei vom bzw. zum entfernten Rechner zu kopieren, mußt Du diese in der Ansicht des Quellrechners markieren und mit gedrückter linker Maustaste (*Drag'n'Drop*) in die Ansicht des Zielrechners ziehen. Beim Loslassen der Maustaste erscheint ein kleines Fenster, in dem Du das Zielverzeichnis nochmals kontrollieren und gegebenenfalls ändern kannst.


 Selbstverständlich können statt einzelnen Dateien auch mehrere gleichzeitig oder gar ganze Verzeichnisse kopiert werden.

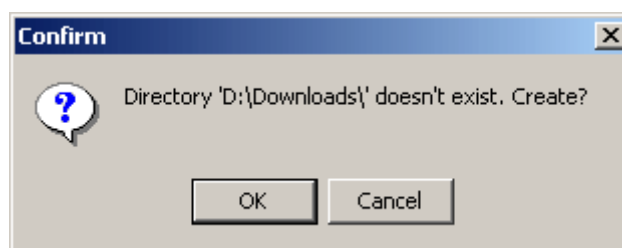
 Der Kopiervorgang kann auch über die Schaltfläche *F5 Copy* in der Leiste *Dateioperationen* oder die Taste *F5* mit der ausgewählten Datei angestossen werden.



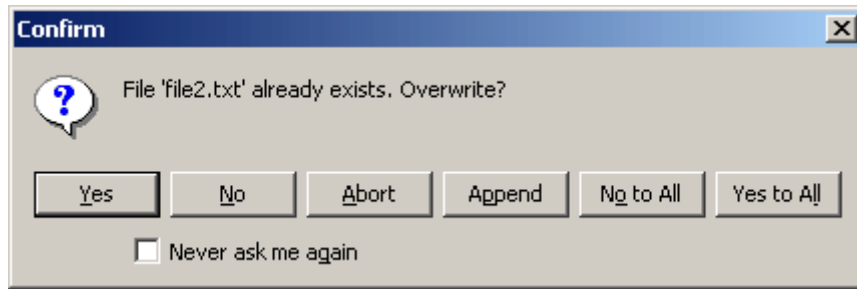
 Über die Schaltfläche *F6 Move* bzw. der Taste *F6* wird die gewählte Datei nicht kopiert, sondern **verschoben**, d.h. nach dem Übertragen der Datei wird diese auf dem Quellrechner gelöscht.



Änderungen am Verzeichnis kannst Du über die Schaltfläche  und der daraufhin erscheinenden Auswahl oder direkt im Textfeld vornehmen. Wenn Du ein nicht existierendes Verzeichnis angibst, dann wirst Du gefragt, ob dieses angelegt werden soll. Mit *Cancel* gelangst Du zurück zur Eingabemaske für das Verzeichnis, ohne daß dieses erstellt wird.

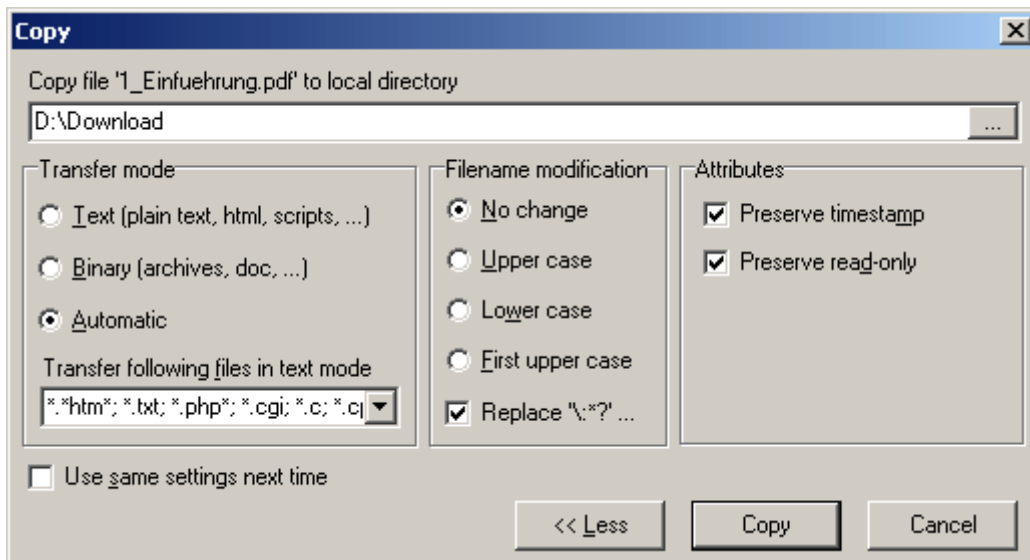


Wenn im Zielverzeichnis bereits eine Datei gleichen Namens existiert, dann wirst Du in einem Dialog nach dem weiteren Vorgehen gefragt. Mit *Yes* wird die bestehende Datei überschrieben, mit *No* brichst Du den Kopiervorgang für diese eine Datei und mit *Abort* auch für alle weiteren Dateien ab. *Append* hängt die zu kopierende Datei ans Ende der bereits bestehenden Datei an. Dies ist nur für Textdateien sinnvoll! *No to All* bzw. *Yes to All* verbietet bzw. erlaubt das Überschreiben für alle weiteren Dateien, insofern Du mehrere Dateien zum Kopieren ausgewählt hast.



⚠️ Sobald Du ein Häkchen bei *Never ask me again* gesetzt hast, wird die nun ausgewählte Option zukünftig für alle Dateien angewandt!

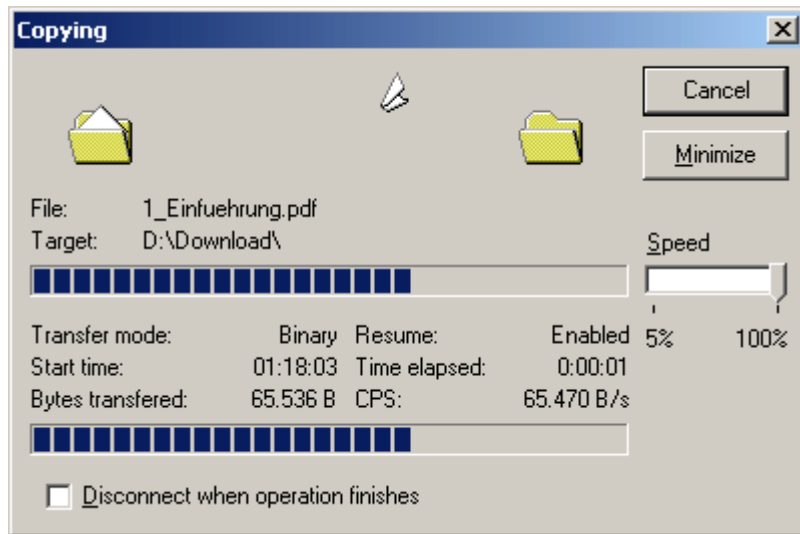
Die Einstellung kannst Du jedoch unter dem Menüpunkt *Options* → *Preferences* → *Environment* → *Overwriting of files* jederzeit wieder rückgängig machen.



Über die Schaltfläche *More >>* kannst Du Dir die fortgeschrittenen Optionen anzeigen lassen. In der Regel sind dort keine Änderungen vorzunehmen.

 Ein Klick auf die Schaltfläche *Copy* startet den Kopiervorgang mit den gewählten Dateien und Einstellungen.

Während des Kopiervorgangs wird (bei kleinen Dateien nur sehr kurz) das unten gezeigte Fenster eingeblendet, das über den Fortschritt informieren soll.



Zudem läßt sich hier über den Schieberegler *Speed* die Down- bzw. Upload-Geschwindigkeit regeln, um so eventuell Bandbreite für andere Netzwerkaktivitäten freizugeben. Der Standardwert liegt bei *100%*. Wird die Option *Disconnect when operation finishes* markiert, dann beendet WinSCP die Verbindung zum entfernten Rechner automatisch, sobald die Datei übertragen wurde.

3.2.3.3. Weitere Funktionen von WinSCP

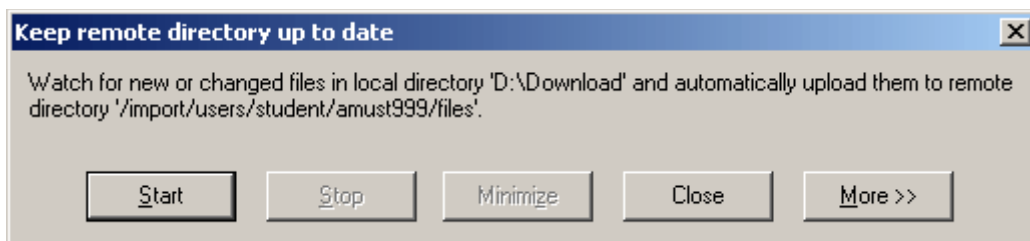
Im Folgenden möchte ich noch kurz einige nützliche Funktionen von WinSCP erläutern. Alle genannten Funktionen sind jeweils über die gezeigten Schaltflächen, die entsprechenden Menüpunkte und über Tastatur-Shortcuts erreichbar.

- **Verzeichnisse vergleichen** 

Die Funktion *Verzeichnisse vergleichen* markiert in beiden Ansichten alle Dateien, die auf der Gegenseite nicht vorhanden sind oder eine andere Dateigröße aufweisen. Die markierten Dateien können anschließend leicht über die in Kapitel 3.2.3.2. beschriebenen Methoden übertragen werden, so daß beide Verzeichnisse inhaltlich synchronisiert sind.

- **Entferntes Verzeichnis "up-to-date" halten** 

Wenn diese Funktion aktiviert wird und über den Button *Start* gestartet wird, dann werden alle Änderungen (neue oder veränderte Dateien) im gewählten lokalen Verzeichnis registriert und die betroffenen Dateien automatisch ins gewählte entfernte Verzeichnis kopiert. Mit *Stop* kann die Überwachung abgeschaltet werden.



- **Dateiansichten synchronisieren** 

Wenn diese Funktion aktiviert ist, dann versucht WinSCP sowohl auf der lokalen als auch auf der entfernten Seite in das gleiche Verzeichnis zu wechseln, wenn dies in einer Ansicht geändert wird. Wenn dies z.B. wegen einem nicht existierenden Verzeichnis fehlschlägt, dann wird eine Fehlermeldung angezeigt und die Funktion deaktiviert.

Sinnvoll ist *Dateiansichten synchronisieren* nur dann, wenn sich auf beiden Rechnern der gleiche Verzeichnisbaum befindet.

- **In PuTTY öffnen** 

Öffnet die aktuelle Verbindung in PuTTY, sofern dieses installiert ist. Der Pfad zum Programm kann unter *Options* → *Preferences* → *Environment* → *Integration* → *PuTTY path* angegeben werden.


3.3. Konqueror (Linux, KDE)

Konqueror ist der in das KDE-Projekt (K Desktop Environment) integrierte Dateimanager und Webbrowser. Über das SSH-Modul *kio-slave fish* kann Konqueror auch mit SCP-Verbindungen umgehen.

Konqueror ist integraler Bestandteil von KDE und ist somit unter

<http://www.kde.org> bzw. <http://www.konqueror.org>

zu finden.

 Die nachfolgenden Erläuterungen und Screenshots beziehen sich auf Konqueror 3.1.3.deutsch, der mit KDE 3.1.3 mitgeliefert wird. Bei anderen Versionen können manche Details etwas abweichen.

3.3.1. Installation

Konqueror selbst wird stets zusammen mit KDE installiert (Paket *kdebase*). Dessen Installation sollte problemlos über den distributionseigenen Paketmanager möglich sein. Weitere Informationen zur Installation befinden sich unter

<http://www.kde.org/documentation/userguide/installation.html>

Da das Modul *kio-slave fish* erst ab KDE 3.1 fest integriert ist, muß bei älteren Versionen das Paket *kio-fish* nachinstalliert werden. Es ist jedoch möglich, daß *kio-fish* in Deiner Distribution trotz älterer KDE-Version bereits vorhanden ist. Die Versionsnummer Deines KDEs erfährst Du auf der ersten Seite des *Kontrollzentrums* (siehe "Startmenü"). Zur Installation des Paketes solltest Du die Dokumentation Deiner Linux-Distribution zu Rate ziehen.


Auf den Rechnern der Linuxpools in der FH Wiesbaden sind alle notwendigen Pakete installiert und eingerichtet.

3.3.2. Konfiguration

Nach der erfolgreichen Installation von KDE, Konqueror bzw. *kio-fish* ist keine weitere Konfiguration notwendig.

Konqueror läßt sich jedoch vielfältig den eigenen Wünschen und Gewohnheiten entsprechend anpassen:

Menüpunkt *Einstellungen* → *Konqueror einrichten...*

 Durch die individuelle Konfiguration von Konqueror kann es zu größeren Abweichungen im Aussehen der Bediener-Oberfläche im Vergleich zu den hier gezeigten Screenshots kommen.

3.3.3. Benutzung von SCP mit Konqueror

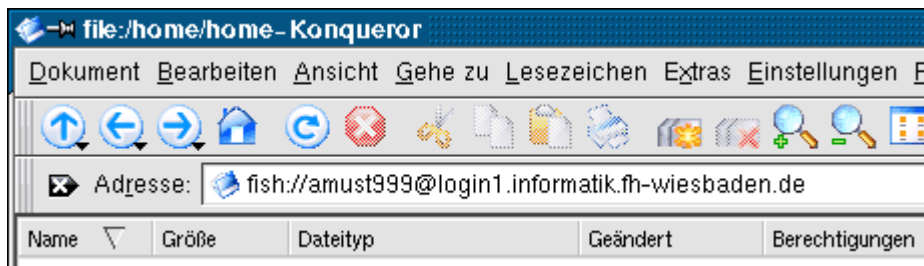
Die Verbindung zum entfernten Rechner kann im Konqueror durch Eingabe einer URL der Form

`fish://user@host`

in der Adresszeile aufgebaut werden. Die Eingabe wird mit *Enter* oder Klick auf die Schaltfläche *Gehe zu* bestätigt.

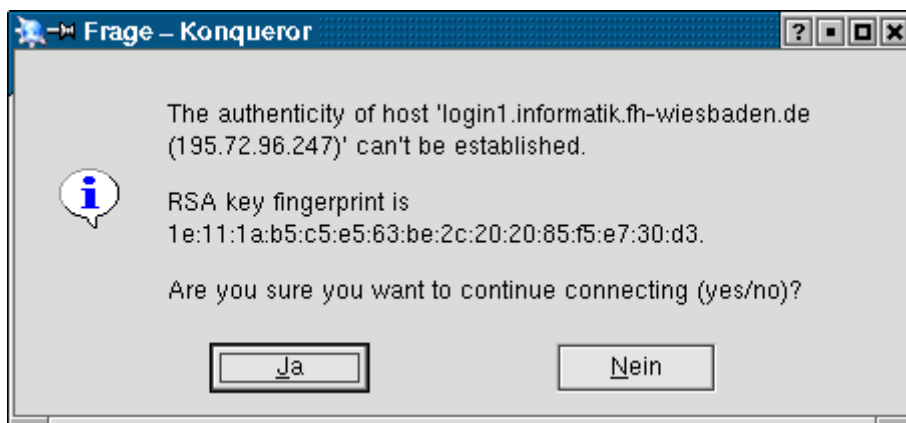


Für die FH Wiesbaden gilt, daß *user* Dein Benutzername und *host* einer der in Kapitel 1.2.1 genannten Loginrechner des Fachbereichs Informatik ist.

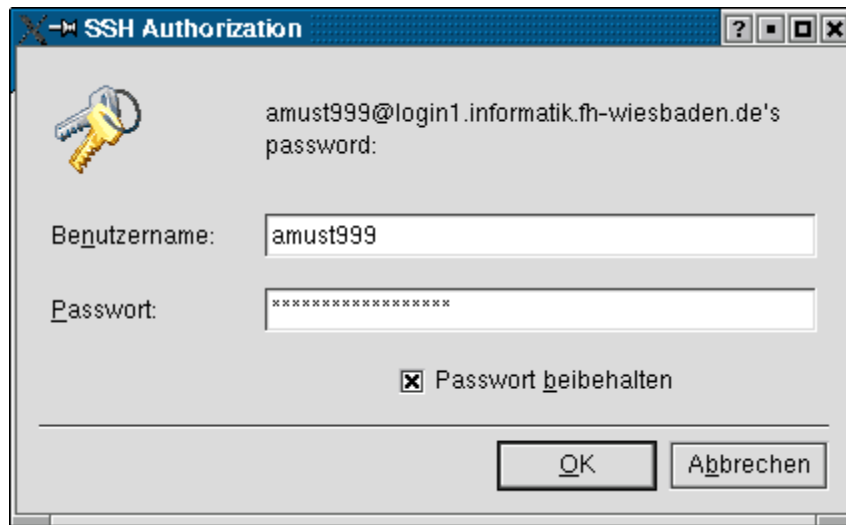


Bei der ersten Verbindungsaufnahme wird, wie bei einer "normalen" SSH-Verbindung auch (siehe Kapitel 2.1) der RSA-Fingerprint des entfernten Rechners überprüft und vom Benutzer eine Bestätigung verlangt. Mit Klick auf *Ja* wird zukünftig keine weitere Abfrage erfolgen, insofern sich der Fingerprint nicht verändert. *Nein* unterbindet die Verbindung mit der Fehlermeldung:

Beim Laden von fish://amust999@login1.informatik.fh-wiesbaden.de ist folgender Fehler aufgetreten: Keine Verbindung zu Rechner Host Key verification failed.

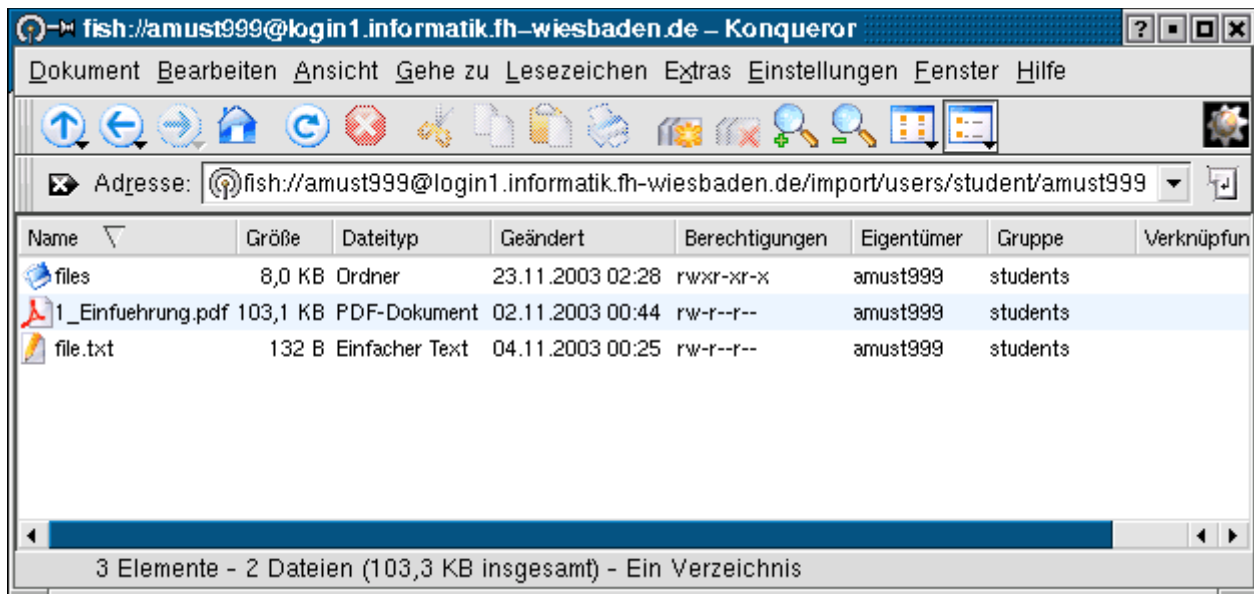


Im nächsten Schritt wirst Du nach Deinem *Password* gefragt:



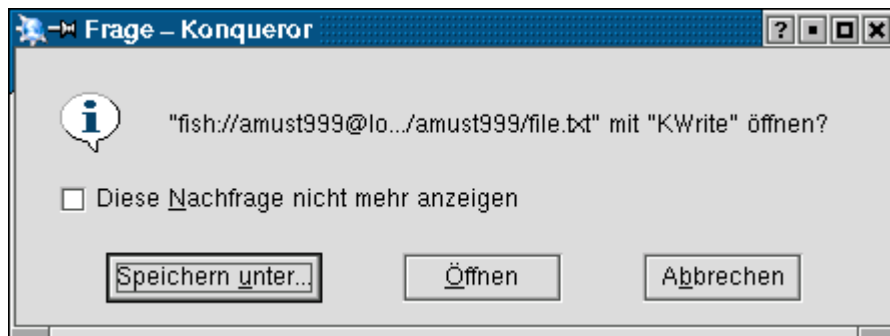
In obiger Eingabemaske kannst Du das *Password* zum angegebenen *Benutzernamen* eintragen. Dieser könnte an dieser Stelle auch noch geändert werden, falls Du Dich bei der Adresse *fish://user@host* verschrieben hast. Wenn die Option *Password beibehalten* aktiviert wird, erfolgt während der Sitzung keine weitere Passwortabfrage.

Wurde dies korrekt eingegeben, dann bist Du auf dem entfernten Rechner eingeloggt und kannst dort, wie lokal auch, Dateioperationen durchführen.



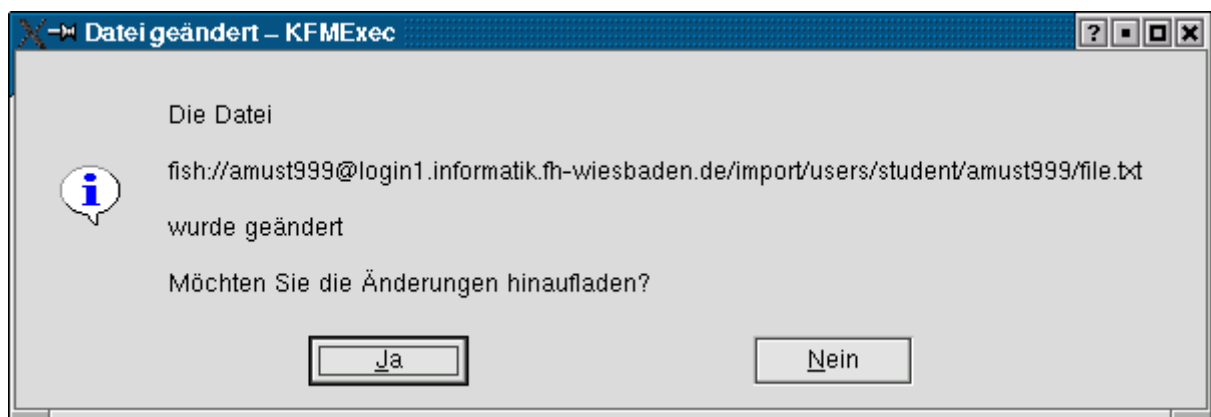
☞ An diesem Icon kannst Du immer erkennen, daß dies die Ansicht eines entfernten Rechners ist.

Bei einem Klick (bzw. Doppelklick, je nach Konfiguration von KDE) auf eine Datei wirst Du gefragt, ob Du die Datei herunterladen oder, abhängig vom Dateityp, mit einem geeigneten Programm öffnen möchtest.



Speichern unter... sichert die Datei auf der lokalen Festplatte – ein Zielverzeichnis kann im erscheinenden Auswahldialog festgelegt werden. *Öffnen* lädt die Datei mit einem Standardprogramm, hier *KWrite*. Mit *Abbrechen* wird keine Aktion ausgeführt.

Wenn eine Datei mit einem Programm geöffnet wird, kann diese dort direkt bearbeitet werden und wird nach dem Beenden des Programms automatisch wieder auf dem Server abgelegt. Manchmal erfolgt dabei eine Abfrage, ob der Upload erfolgen soll oder nicht.

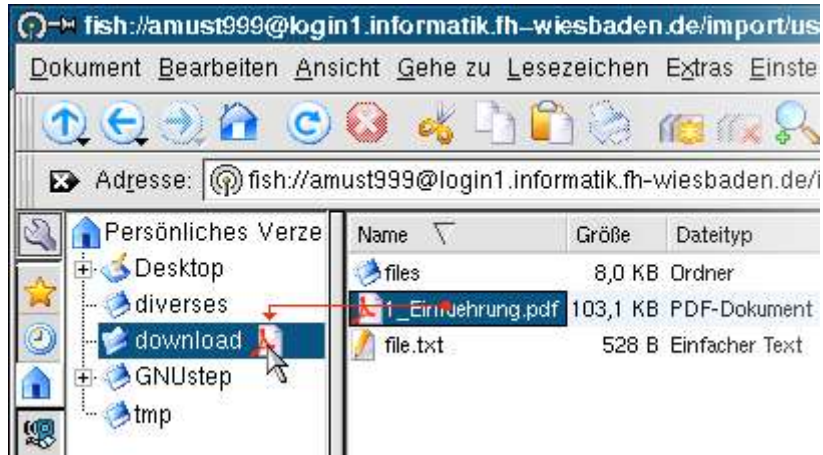


Um nun eine Datei (oder ein Verzeichnis) einfach zwischen lokalem und entferntem Rechner hin- und herzukopieren, ist es sinnvoll, die Darstellung von Konqueror zu ändern, um effizienteres Arbeiten zu ermöglichen. Dies kann nach eigenen Wünschen über den Menüpunkt *Fenster* geschehen:

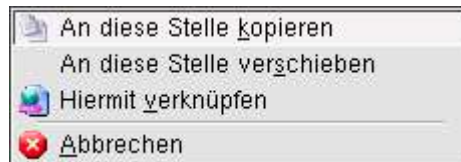
- *Ansicht in rechte und linke Hälfte teilen*
- *Ansicht in obere und untere Hälfte teilen*
- *Navigationsbereich anzeigen*

Bei der Teilung der Ansicht kann in einer Hälfte die lokale und in der anderen die entfernte Ansicht dargestellt werden. *Navigationsbereich anzeigen* blendet links den Verzeichnisbaum der lokalen Ansicht ein.

In beiden Fällen können Dateien einfach per *Drag'n'Drop* umkopiert werden:



Markiere die gewünschte Datei, halte die linke Maustaste gedrückt und ziehe das Icon auf das Zielverzeichnis. Wenn Du dort die Maustaste losläßt, erscheint der folgende Dialog, in dem Du auswählen kannst, ob die Datei *kopiert* oder *verschoben* werden soll. *Hiermit verknüpfen* legt eine Link auf die Datei an und *Abbrechen* stoppt den Vorgang.



Während der Übertragung der Datei wird ein Fenster mit dem Fortschritt des Up- bzw. Downloads eingeblendet:






4. Anhang


4.1. Kostenlose SSH-Clients


 **PuTTY** <http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Kleiner und leistungsfähiger SSH-Client mit GUI, eigene SSH-Implementation, keine Installation notwendig, OpenSource 


 **SSHTerm** <http://3sp.com/products/sshtools/sshterm/sshterm.php>

Java-SSH2-Client, OpenSource, auch als Appletversion installierbar   


 **SSH Secure Shell** <http://www.ssh.com/support/downloads/secureshellwks/non-commercial.html>


Eingeschränkte Version, nur für nicht-kommerziellen Gebrauch kostenlos 

 **SSH Tunnel** <http://www.delight.ch>

Nur zum Aufbau von SSH-Tunneln, nur für private Nutzung kostenlos 

 **OpenSSH** <http://www.openssh.org>

Freie SSH-Implementierung für *nix, Kommandozeile 

 **OpenSSH for Windows** <http://lexa.mckenna.edu/sshwindows>

Windows-Portierung von OpenSSH, Kommandozeile 

 **MacSSH** http://pro.wanadoo.fr/chombier/MacSSH/SSH_info.html

Implementierung des SSH2-Protokolls für MacOS, Kommandozeile 

 **MindTerm** <http://www.mindbright.se/mindterm/>


In Java geschriebener SSH-Client, nur für privaten Gebrauch kostenlos   


 **SecPanel** <http://www.pingx.net/secpanel/>


Graphische Oberfläche für SSH und SCP, benötigt SSH oder OpenSSH 


4.2. Kostenlose SCP/SFTP-Clients


 **PuTTY** <http://www.chiark.greenend.org.uk/~sgtatham/putty/>


SCP- und SFTP-Kommandozeilen-Tools von PuTTY, keine Installation, auch ohne PuTTY nutzbar, OpenSource 


 **WinSCP** <http://winscp.vse.cz/eng/>
<http://winscp.sourceforge.net/eng/>

Explorer- bzw. FTP-Client-ähnliche Oberfläche für SCP und SFTP, OpenSource, viele Funktionen, komfortabel, keine Installation notwendig 


 **Fugu** <http://rsug.itd.umich.edu/software/fugu/>

Graphische Oberfläche für SFTP und SCP 

 **FileZilla** <http://filezilla.sourceforge.net>

Graphischer FTP-Client mit Unterstützung für SFTP 

 **Konqueror** <http://www.konqueror.org>

Der Dateimanager und Webbrowser von KDE kann auch mit SCP umgehen. 

4.3. Links

<http://wwwvs.informatik.fh-wiesbaden.de>

Website des DOPSY-Labors von Prof. Dr. Kröger

<http://www-intern.informatik.fh-wiesbaden.de>

Interne Dokumentationen des FB Informatik der FH Wiesbaden

<http://the.earth.li/~sgtatham/putty/0.53b/html/doc/Contents.html>

Offizielle Dokumentation von PuTTY.