

## Übung 1: Wichtige Kommandos unter Linux

### 1.1 Einige Shell-Kommandos unter Linux

#### Ziel der Übung:

Diese Übung dient dazu, einfache Shell-Kommandos unter Linux kennenzulernen.

#### Anmelden am System:

Beim Booten wird ein Fenster angezeigt, in dem die zu bootende Partition gewählt werden kann. Wählen Sie die Linux-Partition und drücken Sie die Enter-Taste. Nach einer gewissen Zeit erscheint der Login-Screen. Melden Sie sich mit Ihrer Ibw-Nummer („ifwXXXXX“) als Benutzername und Ihrem AFS-Passwort an (z.B. die ersten acht Ziffern der Matrikelnummer).

(Hinweis: Gleicher Username und gleiches Passwort wie unter dem Windows-System.)

#### Vorbereitende Arbeiten:

Nach dem erfolgreichen Start des Systems benötigen Sie noch ein paar Dateien, an denen Sie die Übungen durchführen. Man kann dazu z.B. die Dateien „Freundliche Helfer\_Linux.doc“, „ps.doc“ und „Setuid.doc“, die ebenfalls im Web im Verzeichnis zur Übung 1 abgelegt sind, herunterladen.

(Hinweis: Die Übung kann mit beliebigen Dateien durchgeführt werden. Um nachzusehen, ob sich Dateien bzw. Ordner in Ihrem Homeverzeichnis befinden, klicken Sie den Home- bzw. „Persönlicher Ordner“-Button auf Ihrem Desktop doppelt an.)

#### Starten der Shell:

Unter einer Shell versteht man die klassische Schnittstelle zwischen dem Benutzer und dem Betriebssystem. Die Shell übernimmt die Interpretation der in der Kommandozeile eingegebenen Befehle und ruft zur Ausführung das Betriebssystem oder die entsprechenden Programme auf. Die Shell ist somit ein Kommandozeileninterpreter.

In unserer Übung verwenden wir die Bash (*Bourne Again Shell*), welche sich als De-Facto-Standard bei Linux etabliert hat. Bash ist eine Weiterentwicklung der älteren Bourne Shell.

Zum Starten der Shell klicken Sie z.B. auf *Applications / System / Terminal* (bzw. nutzen Sie das entsprechende Icon in der Schnellstartleiste). Nun wird ein Terminalfenster angezeigt. Tippen Sie nun „bash“ ein und bestätigen Sie Ihre Eingabe mit der Enter-Taste.

#### Aufgabenstellung:

Im folgenden Abschnitt finden Sie zu den wichtigsten Linux-Kommandos eine kurze Erklärung, die zugehörige Syntaxbeschreibung und jeweils eine kleine Übungsaufgabe.

Versuchen Sie sich bei der Ausführung der Übungen auch ein Bild der Dateisystem-Struktur von Linux zu machen und skizzieren Sie Ihre Erkenntnisse in einer Baumstruktur. Das Datei-

system ist hierarchisch aufgebaut und beginnt mit dem sog. Root-Verzeichnis, dem das Symbol „/“ zugeordnet ist.

### Hinweise zum Heimatverzeichnis (Home-Verzeichnis):

Jedem Benutzer wird ein spezielles Verzeichnis im Dateisystem von Linux zugeordnet. Der Verzeichnisname stimmt in der Regel mit dem Benutzernamen überein. Dieses Verzeichnis wird als Heimatverzeichnis oder Home-Verzeichnis bezeichnet. In diesem Verzeichnis wird man nach dem Login-Vorgang positioniert.

Im Home-Verzeichnis kann man seine eigenen Dateien und Verzeichnisse verwaltet (anlegen, ändern, löschen, anschauen) und hat dazu alle Rechte. Möchte man Dateien in anderen Verzeichnissen (z.B. in Home-Verzeichnissen anderer Benutzer) bearbeiten, so benötigt man meist spezielle Rechte. Das Rechtekonzept von Linux werden wir weiter unten noch genauer betrachten. Ein spezieller Benutzer, der sog. Superuser, darf alle Dateien im gesamten Dateisystem bearbeiten.

#### ▪ **man**

<b>man</b>	
<b>Beschreibung</b>	Zeigt die Manual-Page, ein Kurzmanual zum spezifizierten Befehl an.
<b>Syntax / Beispiele</b>	man <Kommando> man mkdir

*Übung:* Lassen Sie sich die Man-Page für einen beliebigen Befehl ausgeben. Welche Angaben werden angezeigt?

#### ▪ **whoami**

<b>whoami</b>	
<b>Beschreibung</b>	Welcher Benutzer ist am System angemeldet?
<b>Syntax / Beispiele</b>	whoami

*Übung:* Mit welchem Usernamen sind Sie angemeldet?

#### ▪ **pwd**

<b>pwd</b>	
<b>Beschreibung</b>	Gibt das aktuelle Arbeitsverzeichnis aus.
<b>Syntax / Beispiele</b>	pwd

*Übung:* In welchem Verzeichnis befinden Sie sich gerade?

#### ▪ **ls**

<b>ls (list)</b>	
<b>Beschreibung</b>	Zeigt den Inhalt des aktuellen oder des angegebenen Verzeichnisses an.
<b>Optionen</b>	-l: Zeigt ausführlichere Informationen an. -a: Auch versteckte Dateien werden angezeigt.

<b>Syntax / Beispiele</b>	ls <Optionen> <Verzeichnis> ls: Zeigt den Inhalt des aktuellen Verzeichnisses an. ls <Verzeichnis>: Zeigt den Inhalt des angegebenen Verzeichnisses an.
---------------------------	---

*Übung:* Lassen Sie sich nun den Inhalt des aktuellen Verzeichnisses anzeigen.

#### ▪ mkdir

<b>mkdir</b>	
<b>Beschreibung</b>	Verzeichnis anlegen.
<b>Syntax / Beispiele</b>	mkdir <Verzeichnisname> mkdir temp

*Übung:* Legen Sie im Homeverzeichnis ein Verzeichnis mit dem Namen „uebung\_winf“ an. Nachdem Sie den Befehl mkdir ausgeführt haben, lassen Sie sich mit ls den Inhalt des aktuellen Verzeichnisses anzeigen. Der Ordner „uebung\_winf“ müsste nun angezeigt werden.

#### ▪ cd

<b>cd (change directory)</b>	
<b>Beschreibung</b>	Verzeichniswechsel
<b>Syntax / Beispiele</b>	cd <Verzeichnis> cd, cd ~, cd \$HOME : Wechselt in das eigene Heimatverzeichnis. cd <Verzeichnis> : Wechselt in das angegebene Verzeichnis. cd .. : Wechselt in das übergeordnete Verzeichnis.

*Übung:* Wechseln Sie in das von Ihnen erstellte Verzeichnis „uebung\_winf“. Lassen Sie sich mit ls den Inhalt anzeigen.

#### ▪ cp

<b>cp (copy)</b>	
<b>Beschreibung</b>	Kopieren von Dateien / Verzeichnissen
<b>Optionen</b>	
<b>Syntax / Beispiele</b>	cp <AlteDatei> <NeueDatei> cp <Datei> <Zielverzeichnis> cp <Datei_1> <Datei_2> ... <Datei_n> <Zielverzeichnis>

*Übung:* Erzeugen Sie mit Hilfe des Befehls `echo > test.txt` eine neue Datei (detaillierte Erläuterung des Befehls folgt in Übung 5). Wechseln Sie zurück ins Homeverzeichnis. Erstellen Sie mit cp ein paar Kopien der erstellten Datei. Geben Sie diesen Dateien die Namen `test_1.txt`, `test_2.txt`, `test_3.txt`, etc.

Reguläre Ausdrücke: Bei der Verwendung von bestimmten Linux-Befehlen bietet es sich an, auf reguläre Ausdrücke zurückzugreifen. Damit muss beispielweise nicht mehr jede Datei einzeln angesprochen werden, sondern es können Ausdrücke erstellt werden, die für eine

Vielzahl von Dateien gelten. Ein kurzer Auszug aus der Liste der Möglichkeiten bietet folgende Tabelle.

### Dateinamenmuster

Ausdruck	Bedeutung
*	Beliebig viele Zeichen.
?	Genau für ein Zeichen.
[a-z]	Genau ein Zeichen von a bis z.
[!Bb]	Weder ein ‚B‘ noch ein ‚b‘ erlaubt.
{info, hinweis, hilfe}.txt	Eine der drei Dateien info.txt, hinweis.txt oder hilfe.txt.
b* info*	Alle Dateien, die mit ‚b‘ oder ‚info‘ beginnen.

Hinweis: Weiterführenden Informationen zu „Regular Expressions“ finden sich beispielsweise auf der in der Fussnote angegebenen Webseite<sup>1</sup>)

Beispiel	Beschreibung
cp *.txt test	Kopiert sämtliche Dateien mit der Endung txt in den Ordner test ( <u>Voraussetzung:</u> Der Ordner „test“ ist vorhanden und Dateien mit dieser Endung existieren).
mv ??april* april	Verschiebt alle Dateien, in deren Dateinamen das Wort april vorkommt in ein Verzeichnis namens „april“. ( <u>Voraussetzung:</u> Der Ordner „april“ ist vorhanden und Beispieldateien mit Namen wie 01april2009.txt oder 09april2007.txt sind ebenfalls vorhanden)

*Übung:* Wechseln Sie zurück ins Homeverzeichnis. Versuchen Sie nun alle Dateien, die mit „test“ beginnen und als Dateiergung „.txt“ besitzen, in ein neues Verzeichnis zu verschieben.

#### ▪ mv

mv (move)	
Beschreibung	Verschieben/Umbenennen von Dateien oder Verzeichnissen
Syntax / Beispiele	mv <Datei_1> <Datei_2> : Benennt Datei_1 in Datei_2 um mv <Datei> <Verzeichnis>: Verschiebt die Datei in das angegebene Verzeichnis.

*Übung:* Verschieben Sie mit mv alle Dateien mit der Endung „.txt“ in das Verzeichnis mit dem Namen „uebung\_winf“.

Wechseln Sie nun in das Verzeichnis uebung\_winf und benennen Sie die Datei „test.txt“ in „test0.txt“ um.

<sup>1</sup> <http://www.regular-expressions.info/> [letzter Zugriff am 27.02.2011]

▪ **rm**

<b>rm (remove)</b>	
<b>Beschreibung</b>	Löscht die angegebene(n) Datei(en) endgültig.
<b>Optionen</b>	-r: Den Inhalt der Unterverzeichnisse mit einbeziehen (recurse). -f: Das Löschen schreibgeschützter Dateien erzwingen (force).
<b>Syntax / Beispiele</b>	rm <Optionen> <Dateiname>

*Übung:* Wechseln Sie zurück ins Homeverzeichnis und löschen Sie alle Dateien aus Ihrem Übungsverzeichnis. (Versuchen Sie dies mit einem einzigen Befehl zu realisieren.)

▪ **rmdir**

<b>rmdir (remove directory)</b>	
<b>Beschreibung</b>	Löscht das/die angegebene(n) Verzeichnis(se) endgültig. Das angegebene Verzeichnis muss bereits leer sein (rm).
<b>Syntax / Beispiele</b>	rmdir <Verzeichnisname>

*Übung:* Löschen Sie das Übungsverzeichnis.

▪ **cat**

<b>cat</b>	
<b>Beschreibung</b>	Zeigt eine Textdatei in der Kommandozeile an.
<b>Syntax / Beispiele</b>	cat <Dateiname>

*Übung:* Lassen Sie sich die Datei ps.doc mit Hilfe des cat-Befehls anzeigen. Was passiert und haben Sie eine Erklärung dafür?

Lassen Sie sich nun mit Hilfe des cat-Befehls die Datei cat\_example.txt anzeigen. Was ist der Unterschied zu vorheriger Übung?

▪ **ps**

<b>ps</b>	
<b>Beschreibung</b>	Listet die aktuell laufenden Prozesse auf.
<b>Optionen</b>	Siehe z.B. Beschreibung in Datei ps.doc oder alternativ durch Eingabe von „man ps“
<b>Syntax / Beispiele</b>	ps x

*Übung:* Lassen Sie sich die aktuell laufenden Prozesse anzeigen. Welche Angaben finden Sie in der Anzeige?

## 1.2 Berechtigungskonzept unter Linux

### Ziel der Übung:


Diese Übung dient dazu, erste Einblicke in das Berechtigungskonzept von Linux zu erhalten. Außerdem wird erläutert wie der Zugriff auf Dateien oder Verzeichnisse eingeschränkt werden kann. Befehle, um dies zu erreichen, werden vorgestellt und geübt.

### Berechtigungskonzept<sup>2</sup>:

In Sachen Berechtigungskonzept unterscheidet sich das Linux- vom Windows-System erheblich. So wird zu jeder Datei explizit der Eigentümer (Benutzer oder Gruppe) gespeichert. Außerdem wird spezifiziert, wer die Datei lesen, ändern oder ausführen darf. Die Information lassen sich über den ls-Befehl mit der Option `-l` anzeigen.

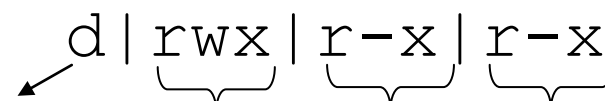
Eine Ausgabe könnte dann in etwa so aussehen:

```
-rwxr-xr-x 2 ifwXX users 43 Nov 17 15:13 bash1.sh
drwxr-xr-x 4 root root 4096 Nov 17 15:13 copy
```



Berechtigung    Eigentümer    Gruppe    Größe    Weitere Informationen

Besonders interessant sind hier die ersten 10 Zeichen - die Berechtigungen. Darum werden diese hier nochmals gesondert erläutert (r= read; w = write; x = execute)



Verzeichnis (d) oder file (-)    Eigentümer    Gruppe    Rest

Die Berechtigungen können auch über die Angabe einer Dezimal-Zahl modifiziert werden. Hierfür werden die Rechte in drei Dreier-Bitgruppen aufgeteilt.

Buchstaben	Binär	Dezimal
<code>rwxr-xr-x</code>	111 101 101	755
<code>rwxrwxrwx</code>	111 111 111	777
<code>rw-r-----</code>	110 100 000	640

*Übung:* Füllen Sie in der folgenden Tabelle die fehlenden Lücken aus!

Befehl	Berechtigung
<code>chmod 440 &lt;dateiname&gt;</code>	

<sup>2</sup> Weiterführende Informationen bzgl. des Linux-Berechtigungskonzepts lassen sich bei Google mit dem Suchbegriff „Linux Berechtigungskonzept“ finden.

	r-xrwx--x
chmod 027 <dateiname>	
	-w--w----

Für die Dateiverwaltung werden Befehle benötigt, um die Attribute von Dateien bzw. Verzeichnissen verändern zu können. Diese werden im Folgenden kurz erläutert.

#### ▪ chmod

chmod	
<b>Beschreibung</b>	Ändert Dateizugriffsrechte
<b>Optionen</b>	-R: Änderungen wirken sich auf den kompletten Unterordner aus.
<b>Syntax / Beispiele</b>	chmod <Modus> <Dateiname> chmod +x bash.sh (Datei wird als ausführbar markiert) chmod 777 test (Lese-,Schreib- und Ausführungsrecht für alle)

*Übung:* Kopieren Sie eine txt-Datei in Ihr Homeverzeichnis und lassen Sie sich anschließend mittels ls-Befehl die Berechtigungsinformationen dazu anzeigen.

Ändern sie nun die Berechtigung für diese Datei auf r--r--r--. Versuchen Sie anschließend den Inhalt der Datei zu ändern (am einfachsten über einen Editor). Was passiert und warum?

#### ▪ chown bzw. chgrp

chown bzw. chgrp	
<b>Beschreibung</b>	Setzt oder verändert den Besitzer bzw. Gruppenzugehörigkeit.
<b>Optionen</b>	-R: Änderungen wirken sich auf kompletten Unterordner aus.
<b>Syntax / Beispiele</b>	chown <User bzw. Gruppe> <Dateiliste>

#### Zusatzübung: Anpassen der Bash

Das Verhalten der Bash lässt sich anpassen und nach individuellen Bedürfnissen gestalten. Im Folgenden werden zwei Varianten aufgezeigt, wie dies realisiert werden kann.

**Aliases:** Mithilfe von Aliases lassen sich Synonyme für Shell-Befehle definieren. Will der Anwender beispielsweise statt dem Befehl `ls` lieber die ausgeschriebene Form `list` verwenden, so muss ein Alias definiert werden.

#### ▪ alias

alias	
<b>Beschreibung</b>	Definiert ein Synonym für einen Befehl
<b>Syntax / Beispiele</b>	alias <aliasname> "<dateiname>" alias list="ls"

	alias: Zeigt alle benutzten Aliase an. unalias <aliasname> : Entfernt den Alias aus der Konfiguration.
--	---

### Übung:

Definieren Sie einen Alias-Befehl „copy“ für die Shell-Anweisung `cp`. Testen Sie anschließend den Alias-Befehl, indem Sie unter Verwendung von `copy` alle `txt`-Dateien in ein beliebiges Verzeichnis verschieben.

### ▪ Verändern des Startvorgangs

**.bashrc:** Bei der Datei `.bashrc` handelt es sich um ein Skript, das beim Aufruf von „bash“ ausgeführt wird. Will man beispielsweise einen Begrüßungstext bei jedem Start der Bash ausführen, so muss in diesem Skript z.B. die Zeile `echo „Hallo Linux-User“` angefügt werden. Schließt man nun die Bash (z.B. mit dem Kommando „exit“) und startet sie erneut, erscheint in der ersten Zeile der definierte Text.

Diese Datei liegt im Home-Verzeichnis jedes Users. Demzufolge kann man an dieser Stelle die Bash User-spezifisch anpassen. Will man systemweit Änderungen beim Starten des Terminals realisieren muss die Datei `/etc/bash.bashrc` entsprechend modifizieren werden. Dazu benötigt man aber entsprechende Rechte.

**.profile:** Eine andere Möglichkeit den Startvorgang zu individualisieren, ist, im Home-Verzeichnis eine `.profile`-Datei erstellen und in dieser Startanweisungen festlegen.

Im Unterschied zur `.bashrc` - Datei wird die `.profile` - Datei nur ausgeführt wenn eine Login-Shell geöffnet wird. Neben der User-spezifischen existiert ebenfalls eine globale `profile`-Datei unter `/etc/.profile`, für deren Änderung ebenfalls entsprechende Rechte erforderlich sind.

Weiterführende Informationen bzgl. **bashrc** und **profile** finden sich unter folgenden Links:

- <http://stackoverflow.com/questions/415403/whats-the-difference-between-bashrc-bash-profile-and-environment> (abgerufen am 15.03.2014, 12:40 Uhr)
- [http://www.joshstaiger.org/archives/2005/07/bash\\_profile\\_vs.html](http://www.joshstaiger.org/archives/2005/07/bash_profile_vs.html) (abgerufen am 15.03.2014, 12:45 Uhr)

### Übung:

Lassen sie sich dazu weiterführende Informationen anzeigen, in dem über die Kommandozeile die Man-Page der bash aufgerufen wird: `man bash` (unter dem Punkt „Invocation“).

### Übung:

Ändern Sie nun die Datei `.bashrc` in Ihrem Home-Verzeichnis so ab, dass bei jedem Start eines Terminals ein Begrüßungstext und der Inhalt Ihres Home-Verzeichnisses ausgegeben wird.

### Zusätzlicher Hinweis zur Labor-Konfiguration:

In unserer Laborumgebung wird beim Login in Linux ein sog. LDAP-Request an einen LDAP-Server abgesetzt. „LDAP“ steht für *Lightweigh Directory Access Protocol*. In diesem Directory (Verzeichnis) verwaltet der LDAP-Server alle Benutzerinformationen der Fakultät.



Als Antwort auf den LDAP-Request übergibt der Server dem Laborrechner alle Informationen zu dem Benutzer, der sich gerade anmeldet, u.a. auch das Home-Verzeichnis und die Login-Shell. Letztere ist bei uns standardmäßig die tcsh, weshalb beim Start eines Terminals auch zunächst mit dieser Shell gearbeitet wird. Das ist auch der Grund, warum wir bei unseren Übungen zuerst in die bash wechseln müssen.