

Praktikum

zum Modul

Betriebssysteme 2

Prof. Dr.-Ing. Fritz Mehner
Prof. Dr.-Ing. Hans-Georg Eßer

Fachhochschule Südwestfalen
Fachbereich Informatik und Naturwissenschaften

Version 1.10.2

Stand: 30. September 2021 (WS 2021/22)



Prof. Dr.-Ing. Fritz Mehner
Prof. Dr.-Ing. Hans-Georg Eßer (esser.hans-georg@fh-swf.de)
Fachhochschule Südwestfalen
Fachbereich Informatik und Naturwissenschaften
Frauenstuhweg 31
58644 Iserlohn

© 2007–2014 Fritz Mehner
© 2016–2021 Hans-Georg Eßer

Inhaltsverzeichnis

Einführung	ii
1 Linux-Grundlagen	1
1.1 Anmelden, Abmelden, KDE-Desktop	1
1.2 Unix-Handbuchseiten und help	5
1.3 Lokalisierung der Shell	6
1.4 Dateibaum erzeugen und löschen	7
2 Unix-Kommandos	8
2.1 Dateiinhalte filtern	8
2.2 Auswertung einer Datendatei	9
2.3 Alle Benutzer des Rechners ermitteln	9
2.4 Liste von Verzeichnissen	9
2.5 Sortierte Liste von Verzeichnissen	10
3 Prozesse, Dateien, Shell	11
3.1 Grundlegende Informationen zu Benutzern	11
3.2 Grundlegende Informationen über Prozesse	11
3.3 Shell	13
3.4 Shell-Programmierung	16
4 Benutzerverwaltung	19
4.1 Mehrere Benutzer mit einem Shell-Skript einrichten	19
4.2 Mehrere Benutzer mit einem Shell-Skript löschen	20
5 Handhabung verschiedener Dateisysteme	21
5.1 Dateirechte	23
5.2 Symbolische Links	23
5.3 Gerätedateien	24
5.4 Das proc-Dateisystem	24
5.5 Festplatten-Partitionen als Dateien	24
5.6 Dateien und Verzeichnisse suchen	28
6 Loop-back-Dateisysteme und Secure Shell	30
6.1 CD-Images via fuse-Dateisystem einhängen	30
7 Synchronisieren und Sichern von Dateibäumen	32
7.1 Teile eines Dateisystems mit rsync synchronisieren	32
7.2 Inkrementelle Sicherung	32
Literaturverzeichnis	36

Einführung

Das vorliegende Dokument enthält die Praktikumsaufgaben zum Modul „Betriebssysteme 2“ des Bachelor-Studienganges *Informatik*. Die Durchführung der Praktikumsaufgaben gibt Gelegenheit, das in der Vorlesung Gehörte anzuwenden und zu vertiefen. Zusätzlich muss der Stoff aus geeigneten Büchern oder Tutorien zur Unix-Linux-Systemverwaltung erarbeitet werden. Dazu sollte auch auf dem eigenen Rechner eine vergleichbar eingerichtete Linux-Installation zur Verfügung stehen.

In diesem Semester setzen wir die webfähige Praktikums Umgebung (WPU) der FH ein, die Sie über <https://wpu.fh-swf.de/> erreichen.

Es ist zwingend erforderlich, auch außerhalb der Lehrveranstaltungen das in der Vorlesung erworbene Wissen praktisch nachzuvollziehen, anzuwenden und zu vertiefen.

Der zur Lösung der jeweiligen Aufgaben benötigte Stoff entspricht dem Stand, der in der Vorlesung bis zur Bearbeitungszeit erreicht wurde. Das heißt jedoch nicht, dass es für einzelne Aufgabenteile nicht elegantere Lösungen gibt, die aber Kenntnisse erfordern, die im Augenblick noch nicht zur Verfügung stehen.

Erforderliche Unterlagen

Die folgenden Dokumente sind erforderlich und sollten auf Ihrem dem Rechner vorhanden sein. Zur besseren Handhabung sollten Icons auf dem Desktop oder Links im Browser angelegt werden.

- Terminplan zum Praktikum: Kursseite <http://swf.hgesser.de/bs-b2/ws2021/>
- Bash Reference Manual [FSF19], <http://www.gnu.org/software/bash/manual/>
- Bash Style Guide und Kodierungsrichtlinie [Meh14], <https://lug.fh-swf.de/vim/vim-bash/StyleGuideShell.de.pdf>

Die Literaturliste zum Modul „Betriebssysteme 2“ enthält weitere Angaben.

Durchführung des Praktikums

Teilnahme und Testat Die selbständige Bearbeitung der Praktikumsaufgaben ist Pflicht. Die *Bearbeitungspflicht* gilt als erfüllt, wenn für mindestens 80 Prozent der Praktikumsaufgaben die selbständige und erfolgreiche Bearbeitung durch den Betreuer bescheinigt wurde. Daraufhin wird der Erwerb der Vorleistung bescheinigt. Nur diese Vorleistung berechtigt zur Teilnahme an der Klausur. Die Abgabetermine regelt der Terminplan (siehe Tabelle 1 auf der folgenden Seite).

Eine Pflicht zur regelmäßigen Anwesenheit im Praktikum gibt es nicht. Die Praktikumsstermine dienen dem Testieren der Abgaben und sollen zur persönlichen Beratung genutzt werden.

Terminplan Für die Testierung der Aufgaben gibt es einen Terminplan, der auf der Webseite zu der jeweiligen Veranstaltung eingesehen werden kann. Dieser Terminplan ist *verbindlich*. Eine

Praktikum Betriebssysteme 2

Aufgabe	Teil 1	Teil 2	Teil 3	Teil 4	Teil 5	Teil 6	Summe
1	3	3	2	2	-	-	10
2	2	2	2	2	2	-	10
3	2	2	3	3	-	-	10
4	6	4	-	-	-	-	10
5	1	1	1	1	3	3	10
6	10	-	-	-	-	-	10
7	3	7	-	-	-	-	10
							Σ 70

Tabelle 1: Auflistung der Teilpunkte

Nachfrist für eine Aufgabe wird nur dann gewährt, wenn zum festgesetzten Termin die Lösung im wesentlichen vorlag und deshalb nur Schwächen oder Fehler zu beheben sind.

Der Praktikumsbetreuer kann für einen vorher festgelegten Teil einer Praktikumsgruppe einen um eine Woche späteren Termin festlegen, um die Abgaben zu entzerren und damit mehr Zeit für die Beratung aufwenden zu können.

Die zu testierenden Aufgaben müssen zu *Beginn des jeweiligen Praktikumstermins* vorliegen, um den Betreuern die Durchsicht der Aufgaben aller Teilnehmer zu ermöglichen.

Vorbereitung und Durchführung Die erfolgreiche Durchführung des Praktikums setzt voraus, dass der zugrundeliegende *Stoff im wesentlichen bekannt* ist und dass die Aufgabenstellung *vollständig gelesen und verstanden* wurde.

Die Zeitdauer von zwei Wochenstunden, die formal für das Praktikum angesetzt ist, wird in der Regel nicht zur vollständigen und richtigen Bearbeitung der geforderten Aufgaben ausreichen, so dass wesentliche Teile der Lösung außerhalb des Praktikums und vor dem Abgabetermin erarbeitet werden müssen.

Die Praktikumstermine dienen unter anderem zur Klärung der Aufgabenstellung und bieten Gelegenheit, Einzelfragen zum Stoff der Vorlesung mit dem Betreuer zu besprechen. Weiterhin ist die Diskussion der gewählten Lösungswege und die Festlegung von Verbesserungen und Berichtigungen ein wesentlicher Zweck der Veranstaltung.

Programmdokumentation Für die zu erstellenden Programme gilt ein *Mindeststandard* für die Programmdokumentation, der im Dokument „Bash Style Guide und Kodierungsrichtlinie“ [Meh14] erläutert ist. Nicht oder mangelhaft dokumentierte Programme werden nicht anerkannt.

Programmtests Grundsätzlich ist *jedes vorzulegende Programm* vorher vom Autor zu testen. Das geschieht in der Regel durch die sorgfältige Überprüfung der Lösung. Wenn Testfälle vorgegeben sind, müssen diese zur Vereinfachung der Kontrolle in der angegebenen Form verwendet werden.

Tests sind selbstverständlich auch dann durchzuführen, wenn Sie nicht ausdrücklich in der Aufgabenstellung gefordert werden.

Bewertung der Aufgaben

Ziel der Bearbeitung einer Aufgabe ist die vollständige Bearbeitung und Lösung der Aufgabenstellung. Bei der Testierung können jedoch Teilaufgaben anerkannt werden, so dass ein mangelhafter Anteil nicht das gesamte Testat in Frage stellt. Tabelle 1 listet die Punkte auf, die bei den einzelnen

Teilaufgaben erzielt werden können. Die Nummerierung der Teilaufgaben stimmt mit den Abschnittsnummern überein (Aufgabe 3, Teil 2 findet sich in Abschnitt 3.2). Einige wenige Abschnitte sind nicht aufgeführt, weil diese Erläuterungen zu den davorstehenden Aufgabenteilen enthalten und somit keine eigene Wertung besitzen.

Gesamtpunktzahl Insgesamt können 70 Punkte erreicht werden. Mit **56 Punkten** (80 Prozent) ist die Bearbeitungspflicht erfüllt. *Es empfiehlt sich aber, auch nach Erreichen der nötigen Mindestpunktzahl die verbleibenden Aufgaben weiter zu bearbeiten, da die Inhalte des Praktikums auch klausurrelevant sind.*

Zur Darstellung

Programmcode, Programmausgaben, Programm- und Dateinamen, *Bash*-Schlüsselwörter und Menüeinträge erscheinen in Schreibmaschinenschrift mit fester Zeichenbreite.

In Codebeispielen und Listings werden für Code und Kommentar zur Verbesserung der Lesbarkeit unterschiedliche Schriftstile einer nichtproportionalen Schriftart verwendet: Schlüsselwörter sind halbfett und blau gesetzt (zum Beispiel **while** im Vergleich zu **varname**). In der gewählten Schriftart sind Null (0) und das große O (O) sowie die Eins (1) und das kleine L (l) gut unterscheidbar.

In den abgedruckten *Bash*-Skripten sind die Kopfkomentare aus Platzgründen meist weggelassen. In den zu erstellenden Lösungen müssen sie selbstverständlich vorhanden sein.

Dieses Dokument wurde in L^AT_EX 2_ε unter Linux und macOS erstellt.

1 Linux-Grundlagen

1.1 Anmelden, Abmelden, Gnome-Desktop

3 Punkte

1.1.1 Anmeldung

Linux ist ein Mehrbenutzersystem. Die Anmeldung ist auf dem Ubuntu-System, das Sie über die WPU nutzen, zunächst nur als nichtprivilegierter Benutzer möglich. Verwenden Sie dazu die Zugangsdaten, die Sie von der WPU-Administration erhalten haben; diese geben Sie in der Web-Oberfläche der WPU ein. Der Start des Desktops kann eine halbe Minute oder länger dauern; Sie müssen ggf. bestätigen, die heruntergeladene `.ica`-Datei im Citrix Viewer bzw. Citrix Workspace zu öffnen.

Der Citrix Viewer zeigt dann den Gnome-Desktop an; eine gesonderte Anmeldung „direkt am Linux-System“ ist nicht nötig (und auch nicht möglich). Beachten Sie, dass der Ablauf bei der Anmeldung anders ist, wenn Sie Linux auf einem eigenen Rechner oder in einer virtuellen Maschine (etwa mit VMware oder VirtualBox) installieren.

Die meisten Arbeiten werden Sie mit Hilfe von Befehlen in einem Terminalfenster erledigen. Öffnen Sie ein solches Fenster, indem Sie das weiß markierte Icon in Abbildung 1.1 anklicken. Benötigen Sie mehrere Terminalfenster, können Sie z. B.

- mit einem Rechtsklick auf das Icon ein Kontextmenü öffnen und darin auf *Neues Terminal* klicken;
- in ein geöffnetes Terminalfenster klicken und dann die Tastenkombination `Strg` + `↑` + `N` verwenden.

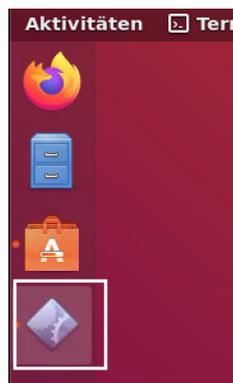


Abbildung 1.1: Über das weiß markierte Icon öffnen Sie ein Terminal-Fenster.

Statt mit `Strg` + `↑` + `N` ein neues Fenster zu öffnen, können Sie auch mit `Strg` + `↑` + `T` einen zusätzlichen Reiter (Tab) im selben Fenster erzeugen.

Eine Anmeldung als Administrator (mit Benutzererkennung `root`) direkt in der WPU ist nicht möglich. Um mit Administratorrechten zu arbeiten, geben Sie das Kommando

```
sudo su
```

und dann auf Aufforderung Ihr Benutzerkennwort (das Sie auch zur Anmeldung verwenden) ein.¹ Sie sehen am veränderten Shell-Prompt (`root@BS2-P`), dass Sie jetzt als Administrator arbeiten.

Da der Systemverwalter alle Rechte besitzt, verwenden Sie diesen Zugang nur dann, wenn es ausdrücklich gefordert ist.

1.1.2 Abmeldung

In der rechten oberen Ecke des Desktops sehen Sie ein Ausschalt-Icon (🔌). Klicken Sie es an, öffnet sich ein Menü, in dem Sie Ihren Namen finden. Klicken Sie diesen an, erscheinen weitere Untermenüpunkte (Abbildung 1.2). Klicken Sie hier auf *Abmelden*, um den Abmeldevorgang einzuleiten. Sie können den Vorgang beschleunigen, indem Sie im angezeigten Dialog erneut auf *Abmelden* klicken; anderenfalls wartet das System noch eine Minute.

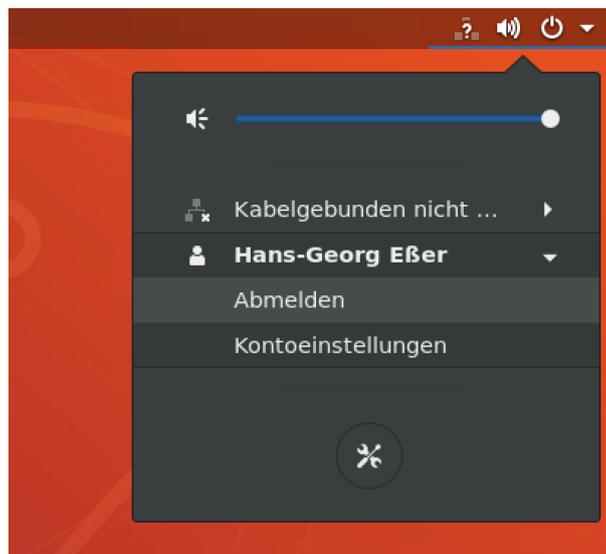


Abbildung 1.2: Über dieses Menü können Sie sich abmelden.

1.1.3 Gnome-Desktop

Machen Sie sich zunächst mit einigen wichtigen Einrichtungen des Gnome-Desktops vertraut.

Anwendung starten Einige Anwendungen sind über das Anwendungsmenü (Icon 🗄️ in der linken unteren Ecke) erreichbar. Weitere Programme können Sie nachinstallieren, indem Sie den Eintrag *Ubuntu-Software* auswählen.

Grafischer Dateimanager Dolphin Das Programm *Dolphin* (Icon 📁) ist der Dateimanager des KDE-Desktops; er ist auf dem WPU-System installiert, weil der Standard-Dateimanager von Gnome (früher *Nautilus*, heute nur noch *Dateien* genannt) nur sehr eingeschränkte Funktionalität bietet.

Textmodus-Dateimanager Midnight Commander (mc) Siehe Abschnitt 1.1.4.

¹Wenn geplante Anpassungen des Systems funktioniert haben, ist keine Passworteingabe nötig.

1.1.4 Dateimanager *Midnight Commander* (mc)

Der Dateimanager *Midnight Commander* mc (Abbildung 1.3) ist eine halbgrafische Konsolenanwendung, die ein sehr schnelles Arbeiten mit Verzeichnissen und Dateien erlaubt. mc steht funktionsgleich auch auf den nichtgrafischen Konsolen² zur Verfügung.

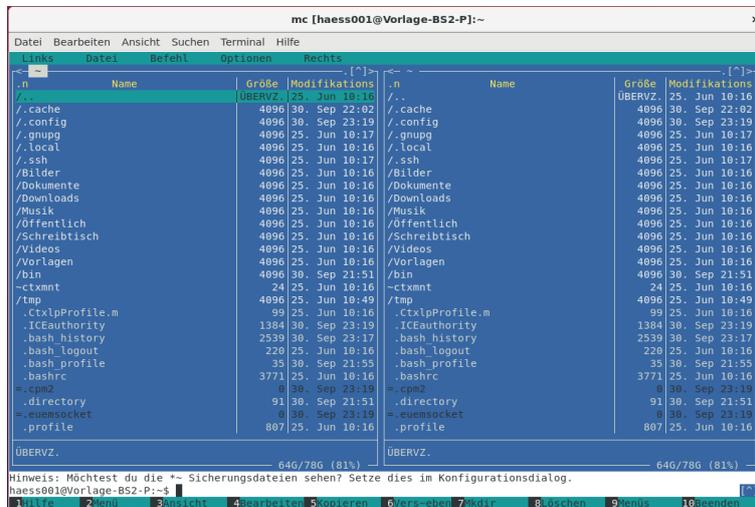


Abbildung 1.3: Der Midnight Commander.

Der Bildschirm des *Midnight Commander* ist in vier Teile gegliedert. Der meiste Platz des Fensters wird durch zwei Verzeichnisse eingenommen. Standardmäßig ist die zweite Zeile am unteren Rand des Fensters die Shell-Befehlszeile. Die untere Zeile zeigt die Belegung der Funktionstasten. Die oberste Zeile ist eine Menüleiste und wird über **F9** erreicht. Die Funktionstasten sind in der folgenden Tabelle aufgelistet.

Taste(n)	Wirkung	
F1	Hilfe	
F2	Menü	fortgeschrittene Einstellungsmöglichkeiten
F3	Anzeige	Inhalt einer Datei anzeigen
F4	Bearbeiten	Datei editieren
F5	Kopieren	Dateien oder Verzeichnisse in das jeweils andere Fenster/Verzeichnis kopieren
F6	Umbenennen	Dateien oder Verzeichnisse umbenennen
F7	Mkdir	ein neues Verzeichnis anlegen
F8	Löschen	Dateien oder Verzeichnisse löschen
F9	Menüs	die Menüs in der Kopfzeile anspringen
Esc , 0	Beenden	(eigentlich F10 – aber das Terminalfenster fängt die Taste ab)

Tabelle 1.1: Die Funktionstasten des *Midnight Commanders*

²Wenn Sie Linux in einer eigenen virtuellen Maschine oder als richtige Installation auf einem eigenen Rechner nutzen, können Sie mit Strg+Alt+F1 bis Strg+Alt+F6 verschiedene Textmodus-Konsolen aktivieren. Sie arbeiten dort in einem Textmodus mit 80x25 Zeichen, ähnlich wie unter MS-DOS.

Sollten Funktionstasten nicht funktionieren, drücken Sie stattdessen  und dann eine Zifferntaste, z. B. ,  statt .

Eine hohe Bediengeschwindigkeit ist nur durch Menübenutzung nicht möglich. Ein wesentliches Bedienkonzept ist deshalb bei vielen Anwendungen die Verwendung von Tastenkombinationen. Die wichtigsten Tastenkombinationen sind in der folgenden Tabelle wiedergegeben.

Tastenkombination	Wirkung
	zum anderen Verzeichnis wechseln
	Datei oder Verzeichnis markieren
 + 	Darstellung wechseln: kurz, lang, benutzerdefiniert
 + 	Verzeichnis-hotlist zeigen und verwenden
	Dateigruppe auswählen
	Dateigruppe abwählen
 + 	Verzeichnisanzeige ein-/ausschalten
 + 	„History“ der Kommandozeile des aktuellen Fensters
 +  + 	Verzeichnis-Chronik aufrufen (zur Navigation)

Tabelle 1.2: Einige Tastenkombinationen des *Midnight Commanders*

Machen Sie sich mit dem Desktop, dem *Dolphin* und insbesondere mit dem *Midnight Commander* so weit vertraut, dass Sie sicher mit Dateien und Verzeichnissen umgehen können.

1.1.5 Grundlegende Bedienmöglichkeiten der *Bash*-Shell

In den folgenden Abschnitten wird das Arbeiten mit der *Bash*-Shell wiederholt bzw. eingeübt. Tabelle 1.3 zeigt eine Auswahl der wichtigsten Bedienmöglichkeiten.

Bedienmöglichkeit (Auswahl)	Befehl / Tastenkombination
Blättern	 +  ,  + 
Kommando- und Dateinamenergänzung	
alle Kommando- und Dateinamenergänzungen	 
Shell beenden	 + 
Programm abbrechen	 + 
Blättern in der <i>History</i> (Kommandowiederholung)	 , 
Suchen in der <i>History</i>	 + 
Eingabeumleitung aus einer Datei	< <i>Dateiname</i>
Ausgabeumleitung in eine Datei	> <i>Dateiname</i>
Pipe (Befehlsverkettung)	<i>Befehl</i> <i>Befehl</i>

Tabelle 1.3: Wichtige Bedienmöglichkeiten der *Bash*-Shell

1.2 Unix-Handbuchseiten und help

3 Punkte

1.2.1 Hilfe zu Shell-Befehlen – manual pages zu eigenständigen Programmen

Online-Hilfe steht für Kommandos und viele Anwendungen durch die Handbuchseiten zur Verfügung. Die Handbuchseiten können aus der Shell heraus über das Kommando `man` eingesehen werden. Tabelle 1.4 zeigt die wichtigsten Hotkeys für den von `man` gestarteten Dateibetrachter `less`. Rufen Sie die Handbuchseite für das Kommando `ls` (`ls` - list directory contents) auf:

```
man ls
```

Stellen Sie für das Kommando `ls` die Bedeutung des Schalters `-l` fest und geben Sie eine entsprechende Verzeichnisliste in einem Shell-Fenster aus.

Bedienmöglichkeit	Befehl
Suchen	<i>/Suchbegriff</i>
nächstes Vorkommen des Suchbegriffs	n (next)
vorheriges Vorkommen des Suchbegriffs	N (Next)
Verlassen	q (quit)

Tabelle 1.4: Die wichtigsten Kommandos im Dateibetrachter `less`

Für das Programm `man` existiert auch ein Handbuch. Es kann mit dem Befehl

```
man man
```

eingesehen werden.

Die Handbuchseiten sind in mehrere Kataloge gegliedert. Ermitteln Sie mit Hilfe von `man man` die Themen der Kataloge mit den folgenden Nummern:

man 1 : _____

man 2 : _____

man 3 : _____

man 5 : _____

man 8 : _____

Die Handbuchseiten bieten Hilfe für Befehle, die als eigenständige, ausführbare Programme oder Skripte vorhanden sind (zum Beispiel `ls` oder `mc`).

1.2.2 Hilfe zu Shell-Builtins

Eine Reihe von Befehlen ist direkt in der Shell eingebaut (und damit ein *Shell-Builtin*). Für diese Befehle steht der Shell-Befehl `help` zur Verfügung. Beispiel: `cd` (change directory) erlaubt den Wechsel des Verzeichnisses. Der Versuch, mit `man cd` eine Handbuchseite aufzurufen, schlägt fehl, während

`help cd`

Auskunft gibt.

Einige Shell-Befehle gibt es als Builtin *und* als externes Programm, z. B. `printf`. Lesen Sie die Hilfe zum Builtin-Befehl `type` durch und erfragen Sie dann mit `type`, welche Versionen von `printf` (alle!) auf Ihrem Rechner vorhanden sind und von welcher Art diese sind.

1.3 Lokalisierung der Shell

2 Punkte

1. Ermitteln Sie den Zweck des Befehls `printf` aus dem Handbuch 1 und den Zweck der gleichnamigen Funktion `printf` aus Handbuch 3 :

`printf` (man 1) _____

`printf` (man 3) _____

Geben Sie in einem Shell-Fenster mit Hilfe des Shell-Befehls `printf` und mit geeigneten Format-Strings nacheinander aus:

- Eine Zeichenkette (zum Beispiel `Hallo, Welt!`),
- die vierstellige ganze Zahl 4711 mit einer Breite von 8 Zeichen,
- die reelle Zahl 47,11 mit einer Breite von 8 Zeichen und 2 Nachkommastellen
- die reelle Zahl 47.11 mit einer Breite von 8 Zeichen und 2 Nachkommastellen.

Dabei sollen jeweils die Ausgaben mit einem Zeilenumbruch (`\n`) abgeschlossen werden.

2. Um den Einfluss der Lokalisierungseinstellungen zu testen, öffnen Sie ein neues Terminalfenster und geben dort zunächst den Befehl

```
export LC_ALL=C
```

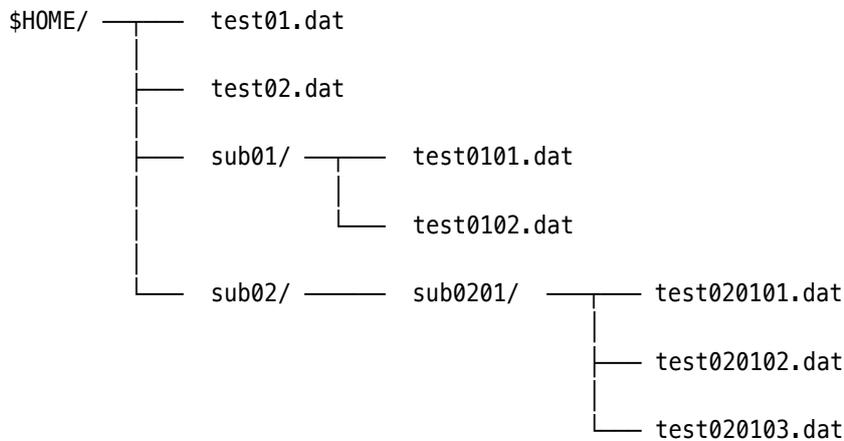
ein: Damit schalten Sie (für dieses Terminalfenster) auf Englisch als Standardsprache um.

Wiederholen Sie im „englischen Fenster“ die letzten beiden Versuche (mit 47,11 und 47.11) Was stellen Sie fest? Vergleichen Sie außerdem die Fehlermeldungen, die erscheinen, wenn Sie in den beiden Fenstern den Befehl `locate` ohne benötigte Parameter eingeben.

1.4 Dateibaum erzeugen und löschen

2 Punkte

Öffnen Sie ein Shell-Fenster und erzeugen Sie mit Hilfe der Befehle `mkdir` und `cd` den folgenden Dateibaum:



(Dabei ist `$HOME` Ihr Home-Verzeichnis `/home/WPU/username/.`)

Die Dateien `test*.dat` werden als leere Dateien durch den Befehl `touch` erzeugt, zum Beispiel
`touch test0101.dat test0102.dat`

Vorgehensweise:

- Kontrollieren Sie nach jedem Schritt mittels `ls`.
- Sehen Sie sich danach das Endergebnis mit einem Dateimanager an (`mc`, `konqueror`).
- Beseitigen Sie danach die Dateien einzeln mittels `rm`.
- Beseitigen Sie dann die Verzeichnisse einzeln mittels `rmdir`.

Führen Sie die Schritte dieses Abschnittes einzeln aus und notieren Sie sich die korrekten Befehle und deren Schalter als **Ergebnisprotokoll zur Vorlage bei der Testierung dieses Aufgabenblattes**.

2 Unix-Kommandos

Erstellen Sie für jede der folgenden Teilaufgaben ein eigenes Skript (`loesung-2-1.sh`, `loesung-2-2.sh`, ...), auch wenn die Lösung nur aus einer Zeile bestehen sollte. Diese Vorgehensweise erleichtert die Abgabe, dokumentiert die Lösung und erspart Tipparbeit.

2.1 Dateiinhalte filtern

2 Punkte

Listing 2.1: Datei `phone.book`

```
Hans!768760!Willy-Brand-Platz 12
Hans-Werner!12780!Ernst-August-Allee 37
Hans-Dieter!88773312!Georg-Friedrich-Haendel-Weg 233
Georg!7772221!Herzog-Wolfgang-Steig 87
Hase!76428765!Wolfsburger Str. 55
Bert!7665654!Haselünner Str. 65
Wolf!8595485!Brunnenweg 77
```

Die Datei `phone.book` (Listing 2.1, Download: <http://swf.hgesser.de/bs-b2/ws2016/prakt/phone.book>) enthält ein winziges Telefonverzeichnis mit den Spalten *Name*, *Telefonnummer* und *Straße*. Die Spalten sind durch das Zeichen `!` getrennt. Die Datei wurde automatisch erzeugt. Das Format kann als gegeben betrachtet werden, das heißt, die Darstellung muss nicht weiter überprüft werden.

Machen Sie sich zum Lösen der folgenden Aufgabenteile insbesondere mit den Werkzeugen `sort`, `cat`, `cut`, `tr`, `head`, `grep` und `column` vertraut. Lesen Sie die zugehörigen Handbuchseiten und verschaffen Sie sich einen Überblick über die vorhandenen Optionen. Lösen Sie folgende Aufgaben:

1. Sortieren der Datei nach den Namen (1. Spalte).
2. Sortieren der Datei nach fallenden Telefonnummern (2. Spalte).
3. Nur die Straßen ausgeben (3. Spalte).
4. Alle Ausrufezeichen in Doppelpunkte umwandeln.
5. Nur die erste Zeile der Datei ausgeben.
6. Nur die Zeile mit den Namen `Hans` und `Wolf` ausgeben. Stellen Sie sicher, dass genau diese beiden Namen in der ersten Spalte gesucht werden.

Hinweis. Bei den ersten drei Aufgaben fällt die Kontrolle schwer, weil keine Formatierung vorhanden ist. Ergänzen Sie deshalb die Sortieranweisungen in der folgenden Art, um eine tabellenartige Ausgabe zu erreichen:

```
sort ... | column -t -s!
```

Was ist die Bedeutung der Schalter `-t` und `-s!`?

2.2 Auswertung einer Datendatei

2 Punkte

Listing 2.2: Beginn der Datei results.csv

```
C7300,ELLIOT,GARY,D,80.1,CA18
C8801,DEARAUJO,KRINESH,C,73,CA18
C9001,JEFFERIES,DIANNE,D,83.4,CA18
C9201,HUAT,KENNETH,D,83.62,CA18
```

Listing 2.2 zeigt den Beginn der Datei results.csv (Download: <http://swf.hgesser.de/bs-b2/ws2016/prakt/results.csv>). Die Einträge innerhalb einer Zeile sind durch Kommas getrennt (csv, comma separated values) und haben folgende Bedeutung:

Matrikelnummer, Nachname, Vorname, Note, Punkte, Abschluss

1. Bestimmen Sie die Anzahl der Studierenden in dieser Datei. Achtung: die Datei kann Leerzeilen enthalten!
2. Erzeugen Sie eine aufsteigend sortierte Liste aller vorkommenden Matrikelnummern.
3. Erzeugen Sie eine nach Namen sortierte Liste aller Studierenden mit der Note HD. Beachten Sie dabei, dass die Zeichenfolge HD Bestandteil eines Namens oder einer Abkürzung sein könnte. Es reicht also keinesfalls, nur nach HD zu suchen. Verwenden Sie zur Kontrolle der Ausgabe column (siehe Aufgabe 2.1).

2.3 Alle Benutzer des Rechners ermitteln

2 Punkte

Informieren Sie sich mittels `man 5 passwd` über das Format der zentralen Passwortdatei `/etc/passwd`. Erstellen Sie aus dieser Datei mit Hilfe von `cut` eine Datei `benutzer`, die die Liste aller aufsteigend sortierten Benutzernamen enthält.

Befehl: _____

(Hinweis: Ihr eigener Benutzername taucht in der Liste nicht auf, weil das Ubuntu-System der WPU einen alternativen Login-Mechanismus verwendet.)

2.4 Liste von Verzeichnissen

2 Punkte

Erstellen Sie mit Hilfe von `du` und `sort` eine nach fallender Größe (in Bytes) geordnete Liste aller Verzeichnisse unterhalb von `/usr`. Die jeweils angezeigte Verzeichnisgröße soll nicht die Größe der Unterverzeichnisse enthalten. Zeigen Sie die Liste mit dem Pager `less` an.

Befehl: _____

2.5 Sortierte Liste von Verzeichnissen

2 Punkte

Erstellen Sie mit Hilfe von `ls` eine nach fallender Größe geordnete Liste aller Dateien im umfangreichsten Verzeichnis unterhalb von `/usr` (wurde in Abschnitt 2.4 ermittelt). Zeigen Sie die Liste mit dem pager `less` an.

Befehl: _____

Testen Sie Ihren Befehl vorher an einem kleinen, selbsterstellten Verzeichnisbaum und überprüfen Sie sorgfältig das Ergebnis!

3 Prozesse, Dateien, Shell

3.1 Grundlegende Informationen zu Benutzern

2 Punkte

Stellen Sie mit Hilfe des Kommandos `groups` fest, zu welchen Gruppen `root` gehört:

Gruppen: _____

Melden Sie sich in einem neuen Shell-Fenster als Benutzer `student` an (`su student`, Passwort `student`). Stellen Sie fest, zu welchen Gruppen der Benutzer `student` gehört:

Gruppen: _____

Stellen Sie mit Hilfe des Kommandos `id` für `root` die numerische Benutzernummer (`uid`), die numerische Gruppennummer (`gid`) fest und die Gruppennummern der Mitgliedsgruppen fest:

uid: _____ **gid:** _____

Gruppen: _____

Nummern: _____

3.2 Grundlegende Informationen über Prozesse

2 Punkte

Schließen Sie alle Terminalfenster und öffnen Sie drei „frische“ Terminalfenster. Ordnen Sie diese so an (und ändern Sie ggfs. die Fenstergrößen), dass alle drei Fenster vollständig zu sehen sind. Sie können dazu jeweils über den Menüpunkt *Terminal / 80x24* eine Standardbreite und -höhe des

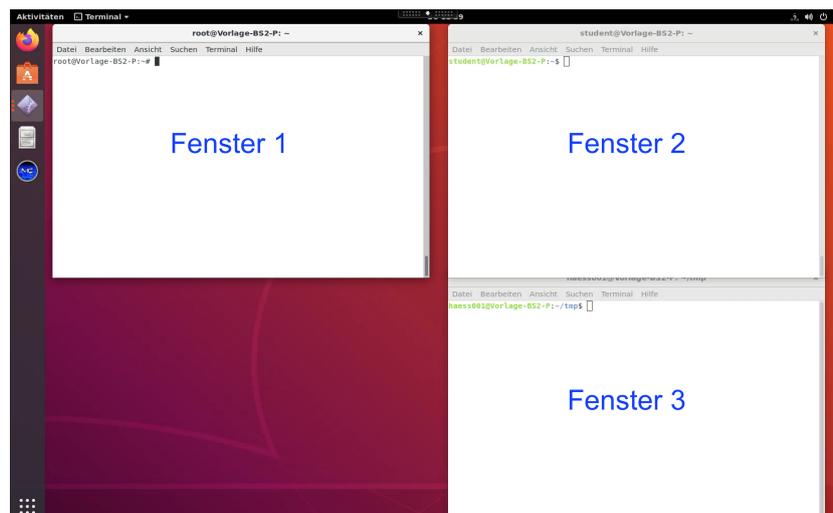


Abbildung 3.1: Drei Terminal-Fenster; in Nummer 1 arbeiten Sie als `root`.

Terminalfensters einstellen; sollten sich dann Fenster überdecken, können Sie mit **Strg** + **[-]** die Schriftgröße verkleinern. (Das müssen Sie in jedem Fenster separat tun, weil die Anzeigeeinstellungen unabhängig sind.)

Legen Sie eine Nummerierung (1, 2, 3) für die drei Fenster fest; in Abbildung 3.1 ist das linke Fenster die Nummer 1, die beiden rechten Fenster sind die Nummern 2 und 3.

1. Werden Sie in **Fenster 1** zum Administrator **root**; dazu geben Sie den Befehl `sudo su -` (mit abschließendem Minuszeichen) ein.
2. Starten Sie in **Fenster 2** den Midnight Commander (`mc`).
3. Rufen Sie in **Fenster 3** die Manpage von `ps` auf.

Erläutern Sie die Wirkung folgender Kommandos, die Sie in **Fenster 1** als **root** oder in **Fenster 3** als normaler Benutzer testen können:

Kommando	Wirkung	Benutzer
<code>ps</code>	_____	root
<code>ps</code>	_____	normaler Benutzer
<code>ps a</code>	_____	root
<code>ps au</code>	_____	root
<code>ps x</code>	_____	root
<code>ps aux</code>	_____	root

Stellen Sie mit Hilfe des Befehls `wc` fest, wieviele Prozesse augenblicklich laufen. Beachten Sie, dass `ps` ohne weitere Maßnahme eine Kopfzeile ausgibt, die nicht mitgezählt werden darf.

Anzahl laufender Prozesse: _____

Starten Sie die Anwendung `top`, um eine laufende Anzeige der Prozessliste zu erhalten. `top` können Sie mit **Q** beenden.

Starten Sie in einem anderen Terminalfenster einen Prozess durch Eingabe von
`yes > /dev/null &`

Beobachten Sie die Ausgabe von `top` und stellen Sie die Prozessnummer von `yes` fest:

PID: _____

Brechen Sie den Prozess mit dem Kommando `kill PID` ab.

Ermitteln Sie den Zweck des Befehls `yes` und erläutern Sie die oben verwendete Befehlszeile.

3.3 Shell

3 Punkte

3.3.1 Prozesse im Vordergrund und Hintergrund

Führen Sie folgende Befehle in einem Shell-Fenster aus:

```
sleep 10
sleep 10 &
```

Wie verhalten sich der Prompt und die Shell?

Listen Sie rekursiv alle Dateien unterhalb von /usr und leiten Sie die Ausgabe in eine Datei um:

```
ls -lR /usr > usr.list &
ps
fg
```

Holen Sie den Prozess (solange er noch läuft – das Kommando braucht nur ca. drei Sekunden) mit dem letzten Kommando in den Vordergrund.

3.3.2 Einfache Shell-Muster zur Beschreibung von Dateinamen (globbing)

Ermitteln Sie zunächst den Hauptzweck des Werkzeuges touch, indem Sie die zugehörige Handbuchseite lesen. Wenden Sie touch auf eine Testdatei an und überprüfen Sie die Wirkung.

Erzeugen Sie in einem Unterverzeichnis mittels touch die folgenden 14 Dateien:

```
feb86    jan12.89
jan19.89 jan26.89
jan5.89  jan85    jan86 jan87
jan88    mar88    memo1 memo10
memo2    memo2.sv
```

Führen Sie die folgenden Befehle aus. Erläutern Sie die Befehle und deren Ergebnisse.

```
echo *
echo *[^0-9]
echo m[a-df-z]*
echo jan*
echo *.*
echo ?????
echo *89
echo jan?? feb?? mar??
echo [fjm][ae][bnr]*
```

3.3.3 Dateien mit besonderen Namen

Erzeugen Sie Dateien mit den folgenden Namen:

```
stars* stars1 stars2
-top
hello my friend
"goodbye"
```

Kommandos zum Anlegen der Dateien:

Löschen Sie nun diese Dateien wieder. Befehle zum Löschen der Dateien:

3.3.4 Sortierreihenfolge und Lokalisierung

Die Bash-Shell für Benutzer-Logins ist *lokalisiert*: Fehler- und Meldungstexte der Shell und vieler Werkzeuge werden in der Landessprache ausgegeben. Die zur Lokalisierung gehörenden Einstellungen werden durch den Befehl `locale` ermittelt. Die erste Zeile der Ausgabe zeigt die Spracheinstellung (`LANG=de_DE.UTF-8`: Deutsch für Deutschland, 8-Bit-Unicode-Codierung), die sechste Zeile die zugrundeliegende Sortiereinstellung (`LC_COLLATE=...`):

```
LANG=de_DE.UTF-8
LANGUAGE=
LC_CTYPE="de_DE.UTF-8"
LC_NUMERIC="de_DE.UTF-8"
LC_TIME="de_DE.UTF-8"
LC_COLLATE="de_DE.UTF-8"
...
```

Ermitteln Sie zunächst die **Spracheinstellung** für die Benutzer `root` und den normalen Nutzer:

```
root _____
normal _____
```

Legen Sie nun mittels `touch` in einem leeren Unterverzeichnis Dateien mit folgenden Namen an:

```
a b c x y z A B C X Y Z ä ö ü Ä Ö Ü
```

Werden Sie mit `sudo su` zum Administrator, führen Sie die folgenden Befehle aus und notieren Sie die Ergebnisse.

Befehl (als root)	Ergebnis
<code>echo *</code>	_____
<code>echo [a-z]*</code>	_____
<code>echo [A-Z]*</code>	_____
<code>find *</code>	_____

Verlassen Sie dann mit `exit` die Root-Shell, führen Sie die gleichen Befehle als normaler Benutzer aus und notieren Sie ebenfalls die Ergebnisse (→ nächste Seite).

Befehl (als normaler Nutzer)	Ergebnis
echo *	_____
echo [a-z]*	_____
echo [A-Z]*	_____
find *	_____

Vergleichen Sie die Ergebnisse und beschreiben Sie die Unterschiede.

3.3.5 Ein-/Ausgabeumleitung

1. Erzeugen Sie eine Liste der Benutzernamen aus dem ersten Feld der Datei `/etc/passwd`. Die Liste soll in alphabetischer Reihenfolge sortiert sein. Verwenden Sie dazu `sort` und `cut`. Das Ergebnis wird durch *Ausgabeumleitung* in die Datei `benutzerliste.txt` geschrieben.
2. Ermitteln Sie die Anzahl der Zeilen in der Datei `/etc/passwd`, die die Zeichenkette `bash` enthalten. Verwenden Sie dazu nur `grep`:
 - Versorgen Sie `grep` mittels *Eingabeumleitung* mit dem Inhalt der Datei `/etc/passwd`.
 - Übergeben Sie den Dateinamen `/etc/passwd` als Kommandozeilenparameter an `grep`.
 - Überprüfen Sie das Ergebnis, indem Sie die Datei `/etc/passwd` im Editor (nur zum Lesen) öffnen und von Hand die Zeilen abzählen, in denen die Zeichenkette vorkommt.

3.4 Shell-Programmierung

3 Punkte

Erstellen Sie das Skript `create_dirs.sh` (Listing 3.1).

Listing 3.1: Skript `create_dirs.sh`

```
1 #!/bin/bash -
2 #=====
3 #
4 #     FILE: create_dirs.sh
5 #
6 #     USAGE: ./create_dirs.sh number_of_dirs
7 #
8 #     DESCRIPTION: Unterverzeichnisse d0 d1 ... erzeugen
9 #
10 #     OPTIONS: ---
11 #     REQUIREMENTS: ---
12 #     BUGS: ---
13 #     NOTES: ---
14 #     AUTHOR: Vorname Nachname, Nachname.Vorname@fh-swf.de
15 #     ORGANIZATION: Fachhochschule Südwestfalen, Iserlohn
16 #     CREATED: 26.08.2013 18:40
17 #     REVISION: ---
18 #=====
19
20 #-----
21 # Aufruf / Anzahl der Aufrufparameter überprüfen
22 #-----
23 if [ $# -lt 1 ]
24 then
25     echo -e "\n\tAufruf: $0 Anzahl_der_Verzeichnisse\n"
26     exit 1
27 fi
28
29 #-----
30 # Verzeichnisse anlegen und zählen
31 #-----
32 anzahl=0
33 erfolg=0
34 while [ $anzahl -lt $1 ]
35 do
36     mkdir d$anzahl && ((erfolg++))
37     ((anzahl++))
38 done
39
40 #-----
41 # Kontrollausgabe
42 #-----
43 echo -e "\n${erfolg}/${anzahl} Verzeichnisse angelegt\n"
```

Machen Sie die Datei ausführbar und erzeugen sie 20 Verzeichnisse mittels

```
chmod +x create_dirs.sh
./create_dirs.sh 20
```

Kontrollieren Sie, ob alle Verzeichnisse vorhanden sind.

Erstellen Sie nun das Skript `mehrfachkopieren.sh` dessen **Anfang** in Listing 3.2 angegeben ist.

Listing 3.2: Anfang des Skripts `mehrfachkopieren.sh` (einige zu erstellende Zeilen ausgeblendet)

```

1  #!/bin/bash -
2  #=====
3  #
4  #      FILE: mehrfachkopieren.sh
5  #
6  #      USAGE: ./mehrfachkopieren.sh number_of_dirs
7  #
8  #      DESCRIPTION: Datei (Kommandozeilenparameter) in alle vorhandenen
9  #                  Unterverzeichnisse kopieren
10 #
11 #      OPTIONS: ---
12 #      REQUIREMENTS: ---
13 #      BUGS: ---
14 #      NOTES: ---
15 #      AUTHOR: Vorname Nachname, Nachname.Vorname@fh-swf.de
16 #      ORGANIZATION: Fachhochschule Südwestfalen, Iserlohn
17 #      CREATED: 26.08.2013 18:40
18 #      REVISION: ---
19 #=====
20
21 #-----
22 # Aufruf / Anzahl der Aufrufparameter überprüfen
23 #-----
24
25 #-----
26 # Prüfen, ob die zu kopierende Datei existiert
27 #-----
28
29 #-----
30 # Unterverzeichnisse ermitteln
31 #-----
32 liste=$(find -type d -name "[^.]*)

```

Ermitteln Sie die Bedeutung des Zusatzes `-name "[^.]*)` im `find`-Befehl in Zeile 32. Das kann der Einfachheit halber außerhalb des Skriptes in einem Shell-Fenster geschehen.

mit `-name` _____

ohne `-name` _____

Bedeutung _____

Weitere Vorgaben:

- Die Datei muss, wie jedes Skript, einen Kopfkomentar enthalten (siehe Listing 3.2).
- Die an das Skript übergebenen Parameter sind am Anfang des Skripts auf Vollständigkeit und Gültigkeit zu überprüfen.
- Dateien und Verzeichnisse, die als Parameter übergeben werden, sind auf Vorhandensein und Lesbarkeit (Rechte) zu überprüfen.
- Die Anzahl der kopierten Dateien wird gezählt und am Ende des Skriptes ausgegeben.

Erstellen Sie nun ein Skript `mehrfachloeschen.sh`, welches eine Datei (der Name ist der erste Aufrufparameter) in allen nicht versteckten Unterverzeichnissen löscht. Der Parameter wird wie oben überprüft. Die Anzahl der gelöschten Dateien wird gezählt und am Ende des Skriptes ausgegeben.

Erstellen Sie weiterhin ein Skript `remove_dirs.sh`, welches alle nicht versteckten Unterverzeichnisse und alle darin enthaltenen Dateien und Verzeichnisse löscht. Die Anzahl der gelöschten Unterverzeichnisse wird gezählt und am Ende des Skriptes ausgegeben.

4 Benutzerverwaltung

4.1 Mehrere Benutzer mit einem Shell-Skript einrichten

6 Punkte

Die Datei `user+num.txt` enthält 50 Namen und Matrikelnummern neuer Student:innen. Mit dem zu entwickelnden Skript `user-add.sh` soll diese Datei eingelesen werden und danach mit Hilfe dieser Informationen automatisch alle Benutzerzugänge und `home`-Verzeichnisse erzeugt werden. Die benötigten Passwörter werden ebenfalls automatisch erzeugt und zur weiteren Verwendung zusammen mit den Login-Namen der neuen Benutzer:innen in eine Ausgabedatei geschrieben. Der Vorgang besteht aus mehreren Schritten, die zunächst einzeln entwickelt und erprobt werden müssen.

Diese Aufgabe muss als Benutzer `root` ausgeführt werden.

4.1.1 Klartextpasswörter erzeugen

Das Handbuch von `pwgen` informiert über den Aufruf und die Kommandozeilenparameter. Erzeugen Sie zunächst zehn Passwörter der Länge 8:

```
pwgen 8 10
```

4.1.2 Entwicklung des Skriptes `user-add.sh`

Das zu entwickelnde Skript `user-add.sh` soll die Datei `user+num.txt` einlesen und jeden Eintrag einzeln verarbeiten. Folgende Vorgaben sind zu beachten:

Benutzer <code>root</code>	Es wird überprüft, ob das Skript vom Benutzer <code>root</code> (UID 0) aufgerufen wird. Wenn das nicht der Fall ist, wird das Skript abgebrochen.
Eingabedatei	Die Lesbarkeit der Eingabedatei ist zu überprüfen.
Klartextpasswort der Länge 8 erzeugen	Das Passwort wird an eine Variable <code>password</code> zugewiesen.
Login-Name erzeugen	Der Nachname jedes Benutzers wird in Kleinbuchstaben umgewandelt und an die Variable <code>loginname</code> als Login-Name zugewiesen.
Benutzer anlegen	<p>Mit Hilfe des Befehls <code>useradd</code> werden ein Benutzer und dessen Home-Verzeichnis angelegt (siehe Handbuch). Der Login-Name steht in der Variablen <code>loginname</code>. Die Matrikelnummer wird als Kommentar in <code>useradd</code> übernommen.</p> <p>Der Schalter <code>-p</code> (beziehungsweise <code>--password</code>) von <code>useradd</code> darf nicht verwendet werden, weil sonst das (verschlüsselte) Passwort für andere Benutzer in der Prozessliste sichtbar würde. Wenn der Benutzer erfolgreich angelegt wurde, wird das Passwort durch</p> <pre>echo \${loginname}:\${password} chpasswd 2>/dev/null</pre> <p>gesetzt. Ermitteln Sie die Bestandteile dieser Zeile und erklären Sie die Wirkungsweise. (<code>chpasswd</code> ist für den Einsatz in Shell-Skripten gedacht.)</p>

Benutzereinrichtung überprüfen	Die fehlerfreie Ausführung von <code>useradd</code> wird kontrolliert und gegebenenfalls ein Erfolgszähler erhöht.
Ausgabe	Die Angaben Nachname, Vorname, Matrikelnummer, login-Name und Klartextpasswort jedes Benutzers werden in die Datei <code>user-added-JJJJMMTT-HHMMSS.txt</code> geschrieben. Der Namensanteil <code>JJJJMMTT-HHMMSS</code> ist der Zeitpunkt, zu dem diese Datei angelegt wurde (Zeitstempel).
Statistik	Am Skriptende wird die Anzahl der versuchten Benutzereinrichtungen (Schleifendurchläufe) und die Anzahl der erfolgreichen Benutzereinrichtungen ausgegeben.

4.2 Mehrere Benutzer mit einem Shell-Skript löschen

4 Punkte

Entwickeln Sie ein Skript `user-delete.sh`, welches die von `user-add.sh` erzeugte Datei `user-added-JJJJMMTT-HHMMSS.txt` (siehe oben) einliest und alle darin enthaltenen Benutzer und deren Home-Verzeichnisse löscht. Hierzu muss das Kommando `userdel` verwendet werden.

Das Skript erzeugt eine Protokolldatei `user-removed-JJJJMMTT-HHMMSS.txt`, in der Nachname, Vorname, Matrikelnummer und Login-Name der erfolgreich gelöschten Benutzer eingetragen werden.

Hinweise

- Überprüfen Sie kritische Befehle sorgfältig, indem Sie diese in Ihrem Skript *nicht* sofort in der ausführbaren Form, wie zum Beispiel

```
userdel -r $loginname
```

angeben, sondern diese mit `echo` zunächst nur als Text ausgeben lassen:

```
echo "userdel -r $loginname"
```

- Testen Sie Ihre Skripte zunächst mit einer Eingabedatei, die nur drei Benutzerangaben enthält. Schreiben Sie sich die Namen und die erzeugten drei Passwörter auf und prüfen Sie, ob eine Anmeldung am System mit diesen Zugangsdaten möglich ist – dazu geben Sie den Befehl

```
su - loginname
```

ein und verwenden dann das zugehörige Passwort. Es sollte nach der Anmeldung ein reduzierter Shell-Prompt (nur `$`) erscheinen; testen Sie dann mit `whoami`, dass die Anmeldung erfolgreich war. Die Abbildung zeigt einen erfolgreichen Test für den Loginnamen `test01`.



```

haess001@Vorlage-BS2-P: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
haess001@Vorlage-BS2-P:~$ su - test01
Passwort:
$ whoami
test01
$ █

```

5 Handhabung verschiedener Dateisysteme

Einführung

a) Besitzer, Gruppe, Zugriffsrechte

Hinweis zu Unterschieden WPU / VirtualBox: Die WPU stellt über Citrix ein virtuelles Linux-System bereit, das bei der Benutzer- und Gruppenverwaltung von den auf regulären Linux-Systemen üblichen Standards abweicht. Ihre VM enthält darum auch einen Benutzer-Account `student` (mit Passwort `student`), für den „üblichere“ Einstellungen gelten. Sie können sich aber an der WPU nicht direkt als `student` anmelden, sondern müssen in einem Terminalfenster mit dem Befehl

```
su - student
```

und Eingabe des Passworts (auch `student`) den Benutzer wechseln. Die Anmeldung ist dann nur in dem Terminalfenster wirksam, in dem Sie den Befehl eingegeben haben (vergleichbar mit dem `su`-Aufruf, den Sie verwenden, um als `root` zu arbeiten).

(Wenn Sie nicht die WPU, sondern die VirtualBox-VM verwenden, können Sie sich direkt als Benutzer `student` anmelden.)

Jede Datei bzw. jeder Ordner hat unter Linux einen Besitzer (owner, user) und eine Besitzergruppe (group) – Dateien im Home-Verzeichnis des Benutzers `student` (`/home/student`) gehören z. B. dem Benutzer `student` und der Gruppe `users`.

Auf Dateien können Sie lesend (read, `r`), schreibend (write, `w`) oder ausführend (execute, `x`) zugreifen, wenn passende Zugriffsrechte gesetzt sind. Die Rechte sowie die Eigentümerangaben sehen Sie in der Ausgabe von `ls -l` am linken Rand und in den Spalten 3 und 4. Zum Beispiel steht in

```
-rwxr--r-- 1 student users 1121 Nov 15 11:14 test.sh
```

der Block `rwX` dafür, dass der Dateibesitzer volle Zugriffsrechte (`r+w+x` = lesen, schreiben und ausführen) hat, während Mitglieder der Gruppe `users` nur lesen dürfen (`r--`). Der letzte Block (hier auch `r--`) räumt Leserechte zudem für Benutzer ein, die weder der Eigentümer sind noch zur Gruppe `users` gehören.

Mit dem Kommando `chmod` (change mode) können Sie die Zugriffsrechte ändern. Um das Kommando auf eine Datei oder einen Ordner anwenden zu dürfen, müssen Sie entweder der Besitzer oder der Administrator `root` sein. Das Programm erlaubt verschiedene Schreibweisen, um die neu zugewiesenen Rechte anzugeben:

- Für den Anfang reicht es aus zu wissen, dass Sie ein Recht mit `+` vergeben und mit `-` entziehen können.
- Auf welche Benutzer sich das bezieht, geben Sie vor dem `+/-` an (`u` = user, Besitzer; `g` = group, Gruppe; `o` = others, sonstige).
- Um welches Recht es geht, schreiben Sie hinter das `+/-` (`r` = read, `w` = write, `x` = execute).

Die Befehle `chown` (change owner) und `chgrp` (change group) ändern Besitzer und Gruppenzugehörigkeit und können nur von `root` ausgeführt werden. (Außerdem kann der Dateibesitzer – mit Einschränkungen – über `chgrp` auch die Gruppenzugehörigkeit eigener Dateien ändern.) Die Syntax ist jeweils

```
chown username datei/verzeichnis
chgrp gruppenname datei/verzeichnis
```

Darüber hinaus kann `chown` auch beide Eigenschaften in einem Rutsch ändern, wenn Sie Username und Gruppenname durch einen Doppelpunkt getrennt angeben:

```
chown username:gruppenname datei/verzeichnis
```

Die Tools `chmod`, `chown` und `chgrp` arbeiten beide auf Wunsch auch rekursiv (für alle Unterordner und enthaltenen Dateien), wenn Sie diese mit einer geeigneten Option aufrufen; Details verraten die Manpages.

b) SUID/SGID

Für Dateien kann ein SUID- (Set User ID) und/oder SGID-Bit (Set Group ID) gesetzt sein. Das führt, wenn es sich um ausführbare Programmdateien handelt, dazu, dass das Programm immer mit den Rechten des Dateibesitzers bzw. mit den Rechten der Besitzergruppe läuft. Ein Beispiel dafür ist das Programm `passwd`, welches das Passwort eines Benutzers ändert: Es kann von normalen Benutzern aufgerufen werden, um deren eigene Passwörter zu ändern, muss dafür aber die Datei `/etc/shadow` verändern, für die nur `root` Lese- und Schreibrechte hat. Darum ist bei `/usr/bin/passwd` das `suid`-Recht gesetzt, und das Programm gehört `root`. Sie erkennen gesetzte Bits daran, dass in der Besitzer- bzw. Gruppenspalte der Ausgabe von `ls -l` das `x` durch ein `s` ersetzt ist:

```
student@linux:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59640 Mar 22 2019 /usr/bin/passwd
```

c) Symbolische Links

Ein *symbolischer Link* erlaubt es, über einen anderen Namen auf eine Datei oder ein Verzeichnis zuzugreifen. Technisch ist das nur eine kleine Spezialdatei, die den Pfad zur Datei / zum Verzeichnis speichert. Sie erkennen symbolische Links in der Ausgabe mit `ls -l` daran, dass in der Rechtespalte ganz am Anfang ein kleines `L` („l“) steht (wo normal ein „-“ für Dateien oder ein „d“ für Verzeichnisse steht). Sie erzeugen solche symbolischen Links mit `ln -s` (siehe Aufgabe 5.2); wenn Sie die Option `-s` (symbolisch) vergessen, erzeugt das Tool stattdessen einen *Hard link* – die Unterschiede werden wir in der Vorlesung besprechen.

d) Ausgabeumleitung

Die Umleitung der *Standardausgabe* eines Programms mit `>` haben wir bereits in der Vorlesung gesehen. Daneben lässt sich auf die *Standardfehlerausgabe* mit `2>` umleiten: Darüber landen Fehlermeldungen in einer Datei. Verwenden Sie `>` und `2>` gleichzeitig, erzeugt ein Programm gar keine Ausgabe im Terminal mehr, und die angegebenen zwei Dateien enthalten getrennt die regulären Ausgaben und die Fehlermeldungen. Um *beide* Kanäle in eine Datei umzuleiten, schreiben Sie kommando `>datei 2>&1`.

5.1 Dateirechte

1 Punkte

Kopieren Sie als root das Verzeichnis `/usr/share/man` rekursiv in das Home-Verzeichnis `/home/student` des Benutzers `student`. Für keine der neuen Dateien hat `student` Schreibrechte.

- Setzen Sie für alle Dateien und Verzeichnisse des Unterbaumes Schreibrechte für alle (Benutzerklassen `user`, `group` und `other`) – finden Sie über die Manpage heraus, wie Sie das mit `chmod` mit einem einzigen Kommando erledigen.
- Setzen Sie für alle Dateien und Verzeichnisse des Unterbaumes als Benutzer und Gruppe `student / users`.

5.2 Symbolische Links

1 Punkte

Erzeugen Sie eine leere Datei `myfile1` und einen symbolischen Link `myfile2`; geben Sie dazu die folgenden Zeilen exakt wie dargestellt ein. `^C` steht für die Tastenkombination `Strg` + `C`.

```

1 touch myfile1
2 ln -s myfile1 myfile2
3 ls -l
4 cat > myfile1
5 xxxxxxxx
6 yyyyyyyy
7 zzzzzzzz
8 ^C
9 ls -l my*
10 cat myfile1
11 cat myfile2
12 cat >> myfile2
13 1111111
14 2222222
15 33333333
16 ^C
17 ls -l my*
18 rm myfile1
19 ls -l my*
```

- Erläutern Sie zunächst die Wirkung der unterschiedlichen `cat`-Befehle.

- Wie sind Links in der Verzeichnisliste (`ls -l`) gekennzeichnet: _____
- Erläutern Sie die Wirkung von Zeile 18 (und damit *alle* Unterschiede zwischen den Ausgaben zu Zeile 17 und zu Zeile 19):

5.3 Gerätedateien

1 Punkte

Die Gerätedateien sind im Verzeichnis `/dev` enthalten. Stellen Sie fest, wieviele Einträge der folgenden Typen vorhanden sind (`find, wc`):

block device	regular file	link	socket
character device	pipe	directory	

Stellen Sie fest, mit welchen Rechten die dritte Festplatte (`/dev/sdc`) ausgestattet ist.

Rechte: _____

Setzen Sie Schreib-/Leserechte für alle Benutzer.

5.4 Das proc-Dateisystem

1 Punkte

Das `proc`-Dateisystem ist eine Schnittstelle zum Betriebssystemkern, die als Dateisystem ausgebildet ist. Die Dateien können mit Hilfe eines Browsers (zum Beispiel `mc`) ausgelesen werden. Beschaffen Sie folgende Informationen (die Angaben gelten für die im Praktikum verwendete virtuelle Maschine):

Datei	gesucht	Ergebnis
<code>cpuinfo</code>	CPU-Modell Frequenz Cache-Größe	
<code>filesystems</code>	die ersten 5 unterstützten Dateisysteme, bei denen <i>nicht</i> <code>nodev</code> in der ersten Spalte steht	
<code>meminfo</code>	Größe des Hauptspeichers	
<code>modules</code>	die ersten drei geladenen Kernel-Module	
<code>partitions</code>	Blockanzahl der Root-Partition (die zugehörige Gerätedatei ermitteln Sie mit <code>df /</code>)	
<code>version</code>	Versionsnummer des BS-Kerns	

5.5 Festplatten-Partitionen als Dateien

3 Punkte

Die meisten Geräte werden in Unix-artigen Betriebssystemen im Verzeichnisbaum unter `/dev` als Dateien virtualisiert. Tatsächlich kann der Inhalt von Partitionen als Image-Datei von einer Partition auf einem Datenträger (Festplatte, USB-Stick, und so weiter) kopiert, als Image-Datei ins Dateisystem eingehängt, sowie der Inhalt geändert und auf den Datenträger zurück kopiert werden.

5.5.1 Festplatten-Partition erstellen

Die Ubuntu-VM (sowohl die WPU-basierte Version als auch die für VirtualBox) verwendet die Festplatte `/dev/sda` (kurz: `sda`), die nur eine Linux-Datenpartition (`/dev/sda1`) enthält. Neben `sda` sind noch die beiden Platten

- `sdb` (wird speziell für den WPU-Betrieb benötigt; *nicht anfassen!*)
- `sdc` (unpartitioniert, für Testzwecke)

vorhanden. In dieser Aufgabe arbeiten Sie mit der dritten Platte `sdc`: Diese „verhält“ sich wie eine neu gekaufte und frisch eingebaute (Hardware-) Platte und enthält noch keine Partitionstabelle.

1. Installieren Sie das Partitionierungstool `gparted` und starten Sie es mit Parameter `/dev/sdc`:

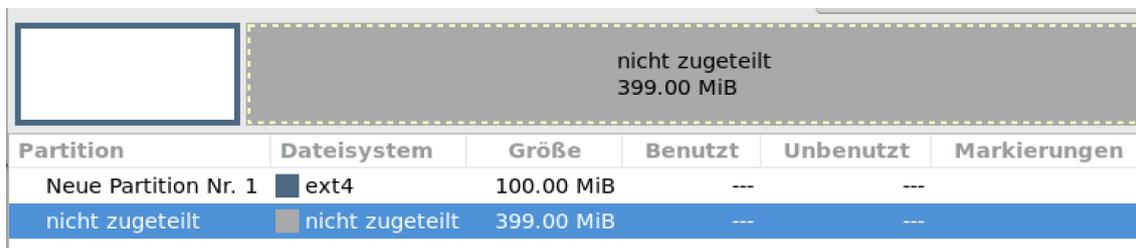
```
sudo apt install gparted
sudo gparted /dev/sdc
```

2. Sie sollten nun die (nicht partitionierte) Platte `sdc` sehen; in der Liste der Partitionen findet sich nur ein Eintrag *nicht zugeteilt* mit ca. 500 MByte Größe (siehe Abbildung).



3. Erzeugen Sie eine (leere) Partitionstabelle, indem Sie *Geräte / Partitionstabelle erstellen* aufrufen, im Dialogfenster die Vorgabe `msdos` für den Tabellentyp übernehmen und auf *Anwenden* klicken. **Achtung: Diese Operation zerstört sofort die bestehende Tabelle. Achten Sie also darauf, dass Sie `sdc` und nicht eine der anderen beiden Platten bearbeiten.**
4. Erstellen Sie eine 100 MByte große Partition. Dazu klicken Sie mit der rechten Maustaste auf den Eintrag *Nicht zugeteilt* in der Tabelle und wählen im Kontextmenü *Neu*. Es erscheint ein Dialog mit Einstellungen für die neue Partition; fast alle können Sie übernehmen – ändern Sie nur im Feld *Neue Größe (MiB)* die Vorgabe auf `100`. Klicken Sie dann auf *Hinzufügen*.

Die neue Partition wird nicht sofort erzeugt; Gparted zeigt nur an, was geplant wird, und bezeichnet die vorgesehene neue Partition als *Neue Partition Nr. 1*:



5. Klicken Sie das grüne Häkchen in der Symbolleiste an und bestätigen Sie die Sicherheitsabfrage per Klick auf *Anwenden*. Nach Abschluss der Repartitionierung erscheint ein weiterer Dialog, den Sie mit einem Klick auf *Schließen* schließen.

In der Partitionsübersicht sollte die neue Partition nun mit dem neuen Gerätenamen (`/dev/sdc1`) erscheinen.

5.5.2 Festplatten-Partition verwenden

Im Folgenden werden Sie

- einige Dateien auf die gerade erstellte Festplatten-Partition schreiben,
- den Inhalt der Partition in eine Image-Datei schreiben,
- die Image-Datei wie eine Festplatten-Partition einhängen und den Inhalt ändern,
- den Inhalt der Festplatten-Partition mit dem Inhalt der Image-Datei überschreiben und
- überprüfen, dass der Inhalt der Festplatten-Partition und der der Image-Datei übereinstimmen.

5.5.3 Mountpoints erstellen

Um den Inhalt der Festplatten-Partition und der Image-Datei unterscheiden zu können, legen Sie zunächst für beide jeweils ein Verzeichnis als mount point an. Werden Sie mit

```
sudo su -
```

zum Administrator und erzeugen Sie mit `mkdir` die Verzeichnisse `/mnt/image` und `/mnt/partition`.

5.5.4 Dateien auf die Festplatten-Partition schreiben

Hängen Sie zunächst die neue Festplatten-Partition ein. Führen Sie dazu auf der Konsole aus:

```
mount /dev/sdc1 /mnt/partition
```

Überprüfen Sie, ob `/dev/sdc1` unter `/mnt/partition` eingehängt ist.

Kopieren Sie einige Dateien in das Verzeichnis `/mnt/partition`. Entsprechend der Größe der eingehängten Partitionen passen etwa 100 MByte in das Verzeichnis.

Hängen Sie die Festplatten-Partition `/dev/sdc1` aus. Es darf keine Programme geben (zum Beispiel Browser, Konsole), die noch auf Dateien im Verzeichnis `/mnt/partition` zugreifen. Führen Sie zum Aushängen folgenden Befehl aus:

```
umount /mnt/partition
```

Überprüfen Sie, dass die Partition tatsächlich ausgehängt ist.

5.5.5 Inhalt der Partition `/dev/sdc1` in eine Image-Datei kopieren

Kopieren Sie jetzt den Inhalt der Festplatten-Partition in eine Image-Datei. Führen Sie dazu folgenden Befehl aus:

```
dd if=/dev/sdc1 of=/tmp/partition_sdc1.img
```

5.5.6 Inhalt des Dateisystems in der Image-Datei ändern

Hängen Sie die nun die neue Image-Datei in das Dateisystem ein:

```
mount -t ext4 -o loop /tmp/partition_sdc1.img /mnt/image/
```

(Die Optionen `-t ext4 -o loop` sind bei aktuellen Distributionen, wie auch beim eingesetzten Ubuntu 18.04, nicht nötig: Der `mount`-Befehl erkennt selbständig, welches Dateisystem auf der Partition angelegt wurde.)

Nehmen Sie nun einige Änderungen vor. Löschen Sie dazu eine Datei aus `/mnt/image` und kopieren Sie eine andere Datei in das Verzeichnis `/mnt/image`. (Sie ändern dadurch den Inhalt der Image-Datei `/tmp/partition_sdc1.img`, die unter `/mnt/image` eingehängt ist.)

Hängen Sie die Partition mittels

```
umount /mnt/image
```

anschließend aus.

5.5.7 Inhalt der Partition `/dev/sdc1` mit dem Inhalt der Image-Datei überschreiben

Führen Sie den folgenden Befehl aus und **achten Sie auf die exakte Schreibweise (`/dev/sdc1`)**:

```
dd if=/tmp/partition_sdc1.img of=/dev/sdc1
```

5.5.8 Inhalte auf der Partition `/dev/sdc1` und in der Image-Datei vergleichen

Hängen sie die Image-Datei und die Festplatten-Partition mit

```
mount /tmp/partition_sdc1.img /mnt/image  
mount /dev/sdc1 /mnt/partition
```

in den Dateibaum ein.

Vergleichen Sie den Inhalt von `/mnt/image` und `/mnt/partition` mittels `diff`

```
diff -r /mnt/image /mnt/partition
```

Die unter `/mnt/partition` eingehängte Festplatten-Partition muss jetzt den gleichen Inhalt haben wie die unter `/mnt/image` eingehängte Image-Datei, weil Sie die Partition mit dem Inhalt der Image-Datei überschrieben haben – darum sollte `diff` hier *nichts* ausgeben.

Nehmen Sie deshalb in beiden Verzeichnissen ein bis zwei Änderungen vor und wiederholen Sie den Vergleich, um zu prüfen, dass `diff` Unterschiede findet.

Hängen Sie die Festplatten-Partition und die Image-Datei anschließend wieder aus.

```
umount /mnt/partition  
umount /mnt/image
```

5.6 Dateien/Verzeichnisse mit bestimmten Merkmalen suchen 3 Punkte

Erarbeiten Sie die Lösungen für alle nachfolgenden Teilaufgaben zunächst in einem Shell-Fenster. Tragen Sie dann die Lösungen für alle Teilaufgaben in ein Skript ein, das die entsprechenden Ergebnisse nacheinander in eine Berichtdatei `report.txt` schreibt. Die Abschnitte sind durch Zwischenüberschriften zu trennen, so dass eine einfache Navigation in einem Editor möglich ist.

5.6.1 Verzeichnisse der Größe nach auflisten

Erstellen Sie mit `du` eine nach fallender Größe sortierte Liste aller Verzeichnisse unterhalb des Wurzelverzeichnisses. Die Liste soll auf ein Dateisystem beschränkt sein: Es ist der Abstieg in andere Dateisysteme (zum Beispiel `/proc`, `/dev` usw.) zu unterbinden.

5.6.2 Dateien der Größe nach auflisten

Erstellen Sie mit `ls` eine nach fallender Größe sortierte Liste aller Dateien im Verzeichnis `/usr/bin`.

5.6.3 Dateien ohne Besitzer auf diesem Rechner

Erstellen Sie mit `find` eine sortierte Liste aller Dateien unterhalb der Wurzel `/` des Dateibaumes, deren Besitzer oder Gruppe auf diesem Rechner nicht existiert. Dabei ist der Abstieg in andere Dateisysteme (zum Beispiel `/proc`, `/dev` usw.; vgl. Aufgabe 5.6.1) zu unterbinden.

Hinweis: Diese und die folgende Teilaufgabe werden Sie nur lösen können, indem Sie sich ausführlich mit der Manpage zu `find` beschäftigen.

5.6.4 Dateien mit SUID- oder SGID-Bit

Erstellen Sie mit `find` eine sortierte Liste aller Dateien unterhalb der Wurzel `/` des Dateibaumes, bei denen das SUID- oder SGID-Recht (Set User ID bzw. Set Group ID) gesetzt ist (evtl. Sicherheitslücke). Dabei ist der Abstieg in andere Dateisysteme (zum Beispiel `/proc`, `/dev` usw.; vgl. Aufgabe 5.6.1) zu unterbinden.

Hinweis: Unix-Zugriffsrechte lassen sich in einer numerischen (oktalen) Notation und in einer symbolischen Notation darstellen. Betrachten Sie z.B. die Ausgaben von `ls -l /usr/bin/passwd` und `stat -c "%a %A" /usr/bin/passwd` für das Passwortprogramm:

```
# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
# stat -c "%a %A" /usr/bin/passwd
4755 -rwsr-xr-x
```

Die am häufigsten verwendeten (und in der Einleitung beschriebenen) Rechte `r` (read), `w` (write), `x` (execute) werden durch die Zahlen 4, 2, 1 und Summenbildung dargestellt, so gilt z.B. `rwX` = $4 + 2 + 1 = 7$, `r-x` = $4 + 1 = 5$ und `---` = 0. Kombiniert man die Rechte des Dateibesitzers, die der Gruppe und die der sonstigen Anwender, entsteht eine dreistellige Oktalziffer (im Beispiel: `rwXr-xr-x` = 755). Zusatzattribute wie SUID und GUID werden über eine noch weiter links stehende Oktalziffer dargestellt; in der symbolischen Darstellung wird an passender Position das `x` durch ein `s` ersetzt (vgl. Einleitung auf Seite 22).

Die Möglichkeit, sich gezielt Dateiattribute mit `stat -c ...` und einem Format-String ausgeben zu lassen, kennen Sie bereits aus der Vorlesung. Über `%a` und `%A` erhalten Sie die oktale bzw. symbolische Beschreibung der Zugriffsrechte.

Eine weitere Hilfestellung kann ein Blick in die Manpage der C-Funktion `open` mit `man 2 open` sein; suchen Sie dort nach `SUID` und `SGID`.

Prüfen Sie, dass Ihr Befehl funktioniert, indem Sie drei Dateien erzeugen, die nur das `SUID`-Bit, nur das `SGID`-Bit bzw. beide Bits gesetzt haben:

```
$ touch /tmp/file{1,2,3}; chmod u+xs /tmp/file{1,3}; chmod g+xs /tmp/file{2,3}; ls -l /tmp/file?  
-rwsr--r-- 1 root root 0 Dec  4 20:03 /tmp/file1  
-rw-r-sr-- 1 root root 0 Dec  4 20:03 /tmp/file2  
-rwsr-sr-- 1 root root 0 Dec  4 20:03 /tmp/file3
```

6 Loop-back-Dateisysteme und Secure Shell

6.1 CD-Images via fuse-Dateisystem einhängen

10 Punkte

Normalerweise sind die Treiber für die unterstützten Dateisysteme (`ext3`, `vfat`, `iso9660` usw.) Kerntreiber, für deren Verwendung Administrator-Rechte benötigt werden. Mit `fuse` (Filesystem in Userspace) steht eine Möglichkeit zur Verfügung, auch nicht-privilegierten Usern die Einbindung verschiedener Dateisysteme in ihrem Home-Verzeichnis zu erlauben. Mögliche Anwendungen sind die Bereitstellung mehrerer CDs/DVDs oder die zeitweise Einbindung verschlüsselter Dateisysteme. Für die von `fuse` unterstützten Dateisysteme stehen Pakete zur Verfügung, die mit `apt` installiert werden können.

6.1.1 CD-Images einhängen

Sie werden in dieser Übung bemerken, dass alles unter Unix-ähnlichen Systemen eine Datei ist (oder besser: durch den Kernel für den Gebrauch transparent als Datei abgebildet wird) – selbst der Inhalt ganzer CDs, Festplatten-Partitionen und Festplatten. In dieser Übung soll das auf CDs eingesetzte Dateisystem `iso9660` verwendet werden. Angesichts der Kapazität aktueller Festplatten kann man auf diese Weise den Inhalt häufig gebrauchter CDs und DVDs ständig zur Verfügung stellen, ohne „Diskjockey“ spielen zu müssen.

Installieren¹ Sie das Paket `fuseiso` über das Shell-Kommando

```
sudo apt install fuseiso
```

Erstellen Sie (ohne Root-Rechte, also als normaler Nutzer) in Ihrem Home-Verzeichnis ein Unterverzeichnis `CDs` (`$HOME/CDs`). Laden Sie zwei kleine CD-Image-Dateien (mit Dateiendung `.iso`) aus dem Netz herunter und platzieren Sie sie in diesem Ordner – wenn Sie nichts anderes finden, können Sie die Dateien `Core-current.iso` und `TinyCore-current.iso` verwenden, die Sie unter <http://distro.ibiblio.org/tinycorelinux/5.x/x86/release/> finden:

```
mkdir ~/CDs; cd ~/CDs
wget http://distro.ibiblio.org/tinycorelinux/5.x/x86/release/Core-current.iso
wget http://distro.ibiblio.org/tinycorelinux/5.x/x86/release/TinyCore-current.iso
```

Benennen Sie die Dateien so um, dass sie die Endung `.image` statt `.iso` besitzen.

Wenn Sie keine Images herunterladen wollen, können Sie mit dem Befehl

```
sudo mkisofs -J -r -o $HOME/CDs/2.image /etc
```

selbst eine Image-Datei mit einem CD-ISO-Dateisystem erstellen, in dem eine Kopie des Ordners `/etc` (oder analog von anderen Verzeichnissen) landet.

¹ In der VirtualBox-VM ist `fuseiso` vorinstalliert.

Schreiben Sie ein Shell-Skript `cd-mount.sh`, das automatisch alle CD-Images in einem bestimmten Verzeichnis in den Dateibaum einbindet.

- Das Skript kann einen Verzeichnisnamen als optionalen Kommandozeilenparameter übernehmen. Wird kein Parameter übergeben, dann wird das Verzeichnis `$HOME/CDs` als Ersatzwert verwendet.
- Die Dateiendung `image` der CD-Images und der Name des Standard-Basisverzeichnisses werden in Variablen festgehalten. (Das erlaubt eine spätere Anpassung, z. B. an die Dateiendung `iso`.)
- In einer Schleife werden alle Dateinamen mit der Endung `image` eingelesen. Aus dem jeweiligen Namen wird die Endung `.image` entfernt und ein gleichnamiges Unterverzeichnis erstellt, falls es nicht bereits vorhanden ist.
- Durch Untersuchung der Information in `/proc/mounts` kann nun festgestellt werden, ob das Verzeichnis bereits als Mountpoint verwendet wird. Ist das nicht der Fall, wird das CD-Image im zugehörigen Verzeichnis eingehängt:

```
fuseiso "$image" "$mountdir"
```

(Für das Beispiel `Core-current.image` soll also die Datei `~/CDs/Core-current.image` in den Ordner `~/CDs/Core-current/` gemountet werden.)

- Zum Abschluss wird die Anzahl der erfolgreich eingehängten CD-Images in einer kurzen Meldung ausgegeben.

Die richtig eingehängten Images können mit dem Befehl

```
cat /proc/mounts
```

angezeigt werden. Die Inhalte der eingehängten CD-Images werden nun zum Beispiel von einem Dateimanager wie Unterverzeichnisbäume behandelt.

6.1.2 CD-Images aushängen

Schreiben Sie ein weiteres Skript `cd-umount.sh`, welches alle eingehängten CD-Images aus dem Dateibaum aushängt.

- Das Skript kann einen Verzeichnisnamen als optionalen Kommandozeilenparameter übernehmen. Wird kein Parameter übergeben, dann wird das Verzeichnis `$HOME/CDs` als Ersatzwert verwendet.
- Die Dateiendung `image` der CD-Images und der Name des Basisverzeichnisses werden in Variablen festgehalten.
- In einer Schleife werden alle Dateinamen mit der Endung `image` eingelesen. Aus dem jeweiligen Namen wird die Endung `.image` entfernt.
- Durch Untersuchung der Information in `/proc/mounts` kann nun festgestellt werden, ob das Verzeichnis gerade als mount point für ein `fuseiso`-Dateisystem verwendet wird. Ist das der Fall, wird das CD-Image ausgehängt:

```
fusermount -u "$mountdir"
```

- Anschließend werden alle leeren Unterverzeichnisse unterhalb des Basisverzeichnisses gelöscht.
- Zum Abschluss wird die Anzahl der erfolgreich ausgehängten CD-Images in einer kurzen Meldung ausgegeben.

7 Synchronisieren und Sichern von Dateibäumen

7.1 Teile eines Dateisystems mit rsync synchronisieren

3 Punkte

Es gibt eine Reihe von Werkzeugen und Vorgehensweisen, um zwei Verzeichnisbäume auf dem gleichen Stand zu halten (sie zu synchronisieren). Die einfachste und zugleich aufwendigste Möglichkeit ist die Erstellung einer vollständigen Kopie bei jedem Abgleich.

Eine weitere Möglichkeit ist die Verwendung von `rsync`, einem Programm zur Dateiübertragung und zum Verzeichnisabgleich zwischen Rechnern. Dieses Programm besitzt umfangreiche Einstellmöglichkeiten und erlaubt auch eine verschlüsselte Übertragung über Netzwerke. (`rsync` verwendet dazu üblicherweise die Secure Shell `ssh`.) Ein wesentlicher Vorteil von `rsync` ist, dass vor einer Übertragung die Verzeichnisbäume zunächst verglichen werden. Beim nachfolgenden Abgleich werden dann nur die Abweichungen übertragen.

1. Das Programm `rsync` ist in der WPU vorinstalliert. Falls Sie ein anderes System verwenden und das Tool dort nicht verfügbar ist (Test: `rsync` ohne Parameter aufrufen), installieren Sie das Paket `rsync` nach.
2. Lesen Sie den Anfang der Handbuchseite von `rsync`.
3. Richten Sie in Ihrem Home-Verzeichnis zwei Ordner `quelle` und `ziel` ein und füllen Sie den Ordner `quelle` mit einigen Unterverzeichnissen und Dateien.
4. Gleichen Sie die beiden Verzeichnisse ab. Die dafür nötige Anweisung ist:


```
rsync -av ~/quelle/ ~/ziel/
```
5. Ändern oder löschen Sie einige Dateien im Verzeichnis `~/quelle` und wiederholen Sie den Abgleich. Überprüfen Sie das Ergebnis.

7.2 Inkrementelle Sicherung

7 Punkte

Für die Sicherung von System- und Benutzerdaten gibt es – abhängig von den jeweiligen Sicherheitsbedürfnissen – viele unterschiedliche Lösungen.

- Für einen privaten Rechner reicht oft das gelegentliche Brennen einer Sicherungs-DVD aus.
- Bei einer größeren Anzahl vernetzter Rechner im professionellen Einsatz wird in der Regel die Verwendung von netzwerkfähigen Sicherungswerkzeugen notwendig sein, und solche Tools erfordern Einarbeitung, die Bereitstellung von Rechenleistung und spezielle Hardware (zum Beispiel Bandstationen).

Die nachfolgende Übung zeigt eine Möglichkeit, wie man Teile der Dateien eines einzelnen Rechners mit Hilfe von Standardwerkzeugen sichern kann.

7.2.1 Die Verwendung von hard links

Zur Vorbereitung der nachfolgenden Sicherungslösung soll die Verwendung von sog. *hard links* untersucht und der Unterschied zu den bereits eingeführten *symbolic links* herausgearbeitet werden.

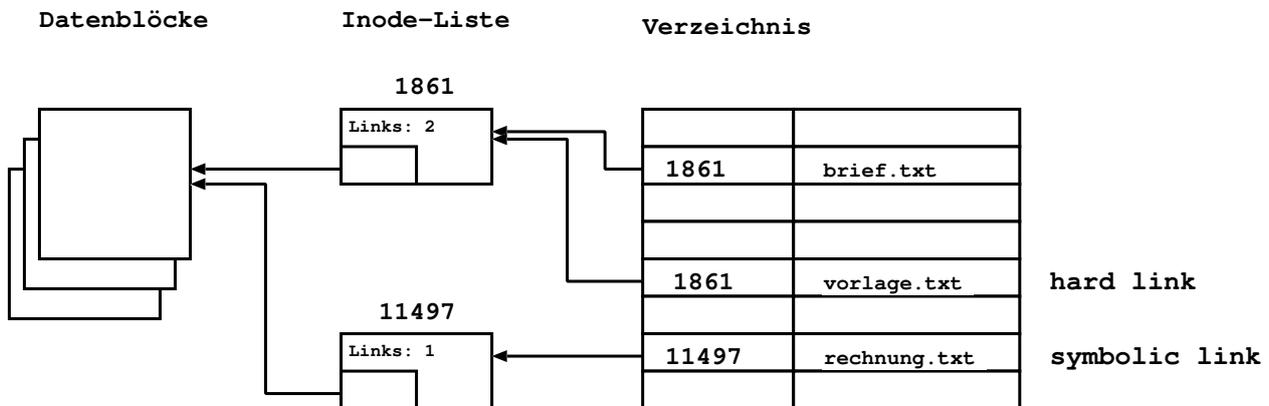


Abbildung 7.1: Datei `brief.txt` mit hard link und symbolic link

Symbolic link In Abbildung 7.1 ist eine reguläre Datei `brief.txt` vorhanden, die den Inode 1861 als Dateibeschreibung besitzt. Der symbolic link `rechnung.txt` ist durch einen eigenen Inode mit der Nummer 11497 dargestellt und zeigt natürlich auf dieselben Datenblöcke.

Wenn die Zieldatei `brief.txt` gelöscht wird, dann zeigt der symbolic link `rechnung.txt` ins Leere.

Hard link In Abbildung 7.1 ist weiterhin ein hard link `vorlage.txt` vorhanden, der ebenfalls auf den Inode 1861 der Zieldatei verweist. Hard links werden im Inode gezählt; der Link-Zähler enthält deshalb den Wert 2.

Wenn die Zieldatei `brief.txt` gelöscht wird, dann werden der zugehörige Inode und die Datenblöcke solange *nicht gelöscht*, wie der Link-Zähler ungleich 0 ist. Über den hard link `vorlage.txt` ist der Inhalt der Datei `brief.txt` nach wie vor erreichbar, obwohl der Verzeichniseintrag nicht mehr vorhanden ist. Hard links sind nur innerhalb eines Dateisystems möglich, können also nicht dateisystemübergreifend sein.

Legen Sie die in Abbildung 7.1 genannten Dateien an (siehe Handbuchseiten zum Befehl `ln`) und überprüfen Sie die Verhältnisse mit Hilfe des Befehls `stat`, insbesondere auch nach dem Löschen der Datei `brief.txt`.

7.2.2 Inkrementelle Sicherung – Überprüfung der Vorgehensweise

Erstellen Sie zunächst zu Testzwecken ein Verzeichnis `original`, in welches Sie einen Dateibaum kopieren (zum Beispiel `/usr/lib/ruby`). Erstellen Sie außerdem die vier Verzeichnisse `backup.0`, `backup.1`, `backup.2` und `backup.3`.

Erstellen Sie nun ein Shell-Skript `backup-test.sh` mit folgendem Inhalt und machen Sie sich mit der Bedeutung der Kommandozeilenschalter vertraut:

```
1 rm --recursive --force backup.3
2 mv backup.2 backup.3
3 mv backup.1 backup.2
4 cp --archive --link backup.0 backup.1
5 rsync --archive --delete original/ backup.0/
```

(Download: `wget http://swf.hgesser.de/bs-b2/ws2020/prakt/backup-test.txt` ; dann Datei in `*.sh` umbenennen.) Zur Funktion des Skripts:

- Zeile 4 enthält den wesentlichen Kunstgriff: der Befehl `cp` legt im Verzeichnis `backup.1` statt echter Kopien nur hard links an. Das Verzeichnis `backup.1` benötigt deshalb nahezu keinen Platz auf der Platte!
- Die Zeilen 1 – 3 sorgen dafür, dass die älteste Sicherung gelöscht wird, und die verbleibenden um eine Stufe „gealtert“ werden. Die Verzeichnisse `backup.1`, `backup.2` und `backup.3` bestehen somit weitestgehend aus hard links.
- Zeile 5 synchronisiert den Sicherungsbestand `backup.0` mit den Originaldateien in `original`.

Aufgabe:

1. Wenden Sie das Skript mehrfach an und löschen Sie dazwischen einige Dateien im Originalbestand.
2. Überprüfen Sie, ob die gelöschten Dateien über die hard links in den Sicherungsverzeichnissen noch vorhanden sind.
3. Überprüfen Sie die Gesamtgröße aller vier Sicherungsverzeichnisse und vergleichen Sie diese mit der Größe des Originalbestandes.

7.2.3 Automatisierte inkrementelle Sicherung

Schreiben Sie ein Skript `backup-home.sh`, das sieben Sicherungsverzeichnisse `backup.0` bis `backup.6` nach der Vorgehensweise des letzten Abschnittes verwaltet. Bei täglich einmaliger Anwendung, zum Beispiel automatisiert durch einen `cron`-Job (zu `cron` vgl. Vorlesung, Folien B-23 und folgende), kann somit ein Schnappschuss pro Tag als Sicherung aufbewahrt werden. Das Skript soll folgende Anforderungen erfüllen:

- Ausführung nur als Benutzer `root`.
- Es wird für jedes Verzeichnis vor der Verwendung überprüft, ob es vorhanden ist.
- Nach abgeschlossener Sicherung wird eine Meldung mit Zeitstempel in eine Logdatei eingetragen (zum Beispiel in `/var/log/BackupHome.log`).
- `rsync` berücksichtigt eine Ausschlussdatei (zum Beispiel `/etc/BackupHome.excludes`), in der Unterverzeichnisse aufgelistet sind, die nicht mitgesichert werden sollen. In dieser Datei stehen zeilenweise die entsprechenden relativen Pfade mit einem vorausgehenden Minuszeichen:

```
- .cache/mozilla/firefox/
- Downloads/
...
```

Damit können zum Beispiel umfangreiche Cache-Verzeichnisse von Browsern oder leicht wieder zu beschaffende Dateien von der Sicherung ausgeschlossen werden.

1. Überprüfen Sie durch Stichproben, ob Änderungen, die zwischen den Sicherungen durchgeführt wurden, berücksichtigt wurden.
2. Überprüfen Sie, ob die Verzeichnisse aus der Ausschlussdatei in der Sicherung fehlen.
3. Legen Sie in `/etc/crontab` einen Cron-Job (zu `cron` vgl. Vorlesung, Folien B-23 und folgende) an, der das Skript einmal pro Tag ausführen würde.
4. Passen Sie das Skript so an, dass es eine (in der Skript-Datei gesetzte) Variable `NUM_BACKUPS` verwendet, über die sich die Anzahl der Backups (bisher: sieben) setzen lässt. Statt `backup.0` bis `backup.6` wird dann mit `backup.0` bis `backup.n` gearbeitet, wobei $n = \text{NUM_BACKUPS} - 1$ ist.

Literaturverzeichnis

- [FSF19] FSF. Bash Reference Manual. <http://www.gnu.org/software/bash/manual/>, Free Software Foundation, 2019. Bash shell, version 5.0. The official manual.
- [Meh14] Fritz Mehner. Bash Style Guide und Kodierungsrichtlinie. <https://lug.fh-swf.de/vim/vim-bash/StyleGuideShell.de.pdf>, 2014.