

SelfLinux-0.12.2



Unterschiede / Gemeinsamkeiten zwischen Windows und Linux



Autor: Alexander Fischer (selflinux@tbanus.org)
Formatierung: Torsten Hemm (T.Hemm@gmx.de)
Lizenz: GFDL

Inhaltsverzeichnis

1 Vorwort

2 Einzel- und Mehrbenutzersysteme

3 Das Dateisystem

4 Die Zugriffsrechte

5 Das persönliche Verzeichnis - Heimatverzeichnis

6 Der Desktop - grafische Benutzeroberfläche

7 Die Shell

8 Zeilenumbrüche

9 Befehlsunterschiede MS-DOS / Linux

1 Vorwort

In diesem Dokument sollen die wesentlichen Unterschiede zwischen Windows und Linux gezeigt werden. Denn gerade die Personen, die in die Unix-Welt umsteigen wollen, stellen meist die Frage: Was kann Linux, das Windows nicht kann? Oder: Kann mir Linux den Komfort bieten, den mir Windows bietet? Auf all diese und weitere Fragen werden wir eingehen und dabei versuchen, auch für **Laien** verständlich zu sein. Wenn Sie dennoch weitere Fragen haben, zögern Sie nicht, eine E-Mail an den Autor oder an das SelfLinux-Team zu senden.

Viele der hier aufgeführten Unterschiede gelten übrigens für Unix im allgemeinen und nicht nur für Linux. Wir werden daher in diesen Fällen meist auch von Unix sprechen. Linux als Teil der Unix-Familie ist dann immer mitgemeint. Wenn wir von Eigenschaften sprechen, in denen sich Linux womöglich von anderen Unix-Varianten unterscheidet, werden wir auch jeweils Linux als Bezeichnung verwenden.





2 Einzel- und Mehrbenutzersysteme

Unix wurde von Anbeginn als ein Mehrbenutzersystem konzipiert. Daher benötigen Sie, bevor Sie mit dem System arbeiten können, einen **Benutzernamen**, ein zugehöriges Passwort und im allgemeinen auch ein sogenanntes Heimatverzeichnis für Ihre persönlichen Daten. Benutzername und Passwort werden auch als Benutzerkonto bezeichnet. Anders als bei einigen Windows-Varianten kann ein gewöhnlicher Benutzer kein neues Benutzerkonto eröffnen. Diese Aktion ist unter Unix nur dem Administrator (root) erlaubt.

Unter Unix ist es möglich, dass mehrere Benutzer **gleichzeitig** auf ein und demselben Rechner arbeiten. Zu früheren Zeiten wurde dies bei Unix-Systemen tatsächlich ausgenutzt, indem man mehrere Terminals, d.h. Monitore und Tastaturen, an einen Rechner anschloss, die von unterschiedlichen Benutzern verwendet werden konnten. Heute wird diese Funktionalität meist über Netzwerkverbindungen genutzt. Unterschiedliche Benutzer können sich z.B. auf einem besonders leistungsfähigen Rechner von ihrem lokalen und weniger leistungsfähigen Rechner aus über ein Netzwerk anmelden, um bestimmte rechenintensive Programme oder Dienste zu benutzen. Unix verwaltet diese unterschiedlichen Prozesse so, dass die Benutzer völlig unabhängig voneinander arbeiten können.

Windows ist primär ein Einzelplatzsystem. Frühere, auf MS-DOS basierende Windows-Versionen wie Windows 95 / 98 / ME bieten überhaupt keine Mehrbenutzerfähigkeit im oben beschriebenen Sinne. Spätere, auf NT-Technologie basierende Versionen bieten zwar echte Mehrbenutzerfähigkeit auf Prozessebene, aber kaum sinnvolle Mechanismen, um auf einer entfernten Maschine komfortabel arbeiten zu können. Diese Lücke wird von den **Windows Terminal Server**-Varianten geschlossen, für die jedoch zusätzlich die Verwendung spezieller Clients notwendig ist. Insgesamt bietet Windows also ebenfalls Mehrbenutzerfähigkeit mit der Möglichkeit der Nutzung entfernter Rechner, doch sind entweder die Möglichkeiten recht eingeschränkt oder aber die Installation spezieller Windows-Versionen auf Server- und Clientseite erforderlich. Über Gegenwart und Zukunft der .NET Technologie wollen wir an dieser Stelle keine Aussage treffen - zu ungewiss ist der Ausgang dieses gerade angebrochenen Kapitels.

3 Das Dateisystem

Windows und Unix verwenden unterschiedliche Dateisystemtypen. Windows 95 bis ME arbeiten ausschließlich mit den Dateisystemen  **FAT16** und  **FAT32**. Windows NT / 2000 / XP verwenden primär das Dateisystem  **NTFS**. Sicherlich haben Sie diese Begriffe schon einmal gehört, sei es beim Partitionieren der Festplatte oder bei sonstigen Arbeiten. Unter Linux ist das Dateisystem  **EXT2** am weitesten verbreitet, aber es gibt weit mehr als ein Dutzend Alternativen. Am interessantesten sind hierbei protokollierende Dateisysteme

(engl.: journaling filesystems), die eine größere Fehlertoleranz bieten und nach Systemabstürzen keinen umfangreichen Prüfdurchlauf durchführen müssen, was gerade bei größeren Festplatten eine große Zeitersparnis beim Wiederhochfahren des Systems bedeuten kann. Als protokollierende Dateisysteme werden unter Linux vor allem **EXT3**, **ReiserFS**, **XFS** und **JFS** verwendet. NTFS ist ebenfalls ein protokollierendes Dateisystem, das zumindest für einen lesenden Zugriff auch unter Linux unterstützt wird. Generell gilt, dass Sie mit Linux alle unter Windows verwendeten Dateisysteme zumindest lesen können. Umgekehrt können Sie jedoch mit Windows nicht auf die Linux spezifischen Dateisysteme zugreifen.

Durch das Dateisystem wird unter anderem bestimmt, wie groß eine Partition sein darf. Aktuelle NTFS-Versionen sind ebenso wie die meisten unter Linux verwendeten Dateisysteme in der Lage, Partitionen bis zur einer Größe von mehreren Terabyte zu verwalten. Die **Linux-Kernel** der Reihe 2.4.x (oder kleiner) limitierten diese Größe jedoch auf ca. ein Terabyte. Mit der Veröffentlichung von Kernel 2.6 wurde durch die Einführung sogenannter **Large Block Devices** (welche die Verwendung größerer Blöcke auf den Festplatten erlauben) diese Limitierung aufgehoben. Wer extrem hohe Datenmengen innerhalb einer Partition verwalten möchte, sollte sich informieren, ob dies vom jeweiligen Kernel und Dateisystem unterstützt wird. Für normale Benutzer sind Datenmengen im Terabyte-Bereich freilich weit jenseits des Üblichen.

Linux ist aufgrund der verwendeten Dateisysteme schon sehr viel länger in der Lage, lange Dateinamen zu verarbeiten. Dabei braucht man auch keine Rücksicht auf die Kombination 8+3 nehmen. Diese wurde vorrangig von MS-DOS eingesetzt.

Unix unterscheidet zwischen Groß- und Kleinschreibung. In einem Verzeichnis könnten z.B. die Dateien **Text.txt** und **text.txt** liegen. Es handelt sich dann um unterschiedliche Dateien, die natürlich auch potentiell einen unterschiedlichen Inhalt haben.

Unter Unix steht es Ihnen frei, wie Sie Ihre einzelnen Dateien benennen. Unter Windows müssen Sie häufig auf die Dateiendung achten. So sollte z.B. bei einem Textdokument die Endung **.txt** sein. Unter Unix können Sie wählen: **.txt**, **.dokument**, **.privat** oder auch ein Dateiname ohne Endung sind möglich. Der Punkt in dem Dateinamen stellt nicht eine Trennung sondern ein ganz neutrales Zeichen in der Benennung dar. Das System erkennt anhand der dateiinternen Kennzeichnungen um welchen Dateityp es sich handelt. Ergänzend sollte man allerdings festhalten, dass beispielsweise Desktop-Systeme wie **KDE** oder Web-Browser die Möglichkeit bieten, Dateien mit einer bestimmten Endung Applikationen zuzuordnen. Diese Zuordnungen sind vom Benutzer konfigurierbar.

4 Die Zugriffsrechte

Unter Windows 95 / 98 / ME haben Sie das Problem, dass Sie jede Datei auf dem System ansehen, verändern, speichern oder gar löschen können. Egal ob es sich nun um eine Systemdatei oder um einem geheimen Liebesbrief handelt. Ebenso können Sie jede beliebige Software installieren und selbstverständlich auch wieder entfernen. Ihnen steht es auch frei, einfach aus Spaß die gesamte Festplatte zu formatieren. Das muss leider nicht immer aus bösem Willen geschehen, sondern kann auch einem erfahrenen Benutzer aus Unachtsamkeit passieren. Auch die so genannten Anmelde-Profile helfen in diesem Fall leider nicht weiter. Sie speichern lediglich die persönlichen Einstellungen und den eingerichteten Desktop. Es ist trotzdem möglich, dass einzelne Benutzer auf die Daten ihrer Mitbenutzer zugreifen.

Unter Unix müssen Sie stets mit einem Benutzernamen angemeldet sein. Somit weiß das System immer, wer Sie sind. Entsprechend wurden Ihnen vom Systemadministrator gewisse Rechte oder Verbote erteilt. Sie können auf dem System genau so viel arbeiten, einsehen oder löschen, wie es Ihnen vom Administrator erlaubt wurde. Die eigentlichen Systemdateien darf ein normalen Benutzer beispielsweise nicht ändern. So können Sie nicht aus Versehen das System beschädigen, Konfigurationsdateien löschen oder gar die Festplatte formatieren - und aus Absicht ebenso wenig. Sie können auch nicht (außer es wurde Ihnen erlaubt) auf die Dateien von anderen Benutzern zugreifen oder diese einsehen (ob lokal oder über das Netz). Grundsätzlich gilt folgende Regel: Es ist

alles verboten, das nicht ausdrücklich erlaubt ist. Wundern Sie sich also nicht, wenn Sie einmal die Meldung **Permission denied** lesen.

Zusätzlich können unter Unix auch so genannte Benutzergruppen erstellt werden. In diesen Gruppen können Sie mehrere Benutzer zusammenfassen und eine Datei für jede Gruppe individuell freigeben oder verbieten. Somit sparen Sie sich die Arbeit, für jeden einzelnen Benutzer eine Datei mit den Zugriffsrechten zu versehen. Das ist in großen Netzwerken mit vielen Benutzern recht hilfreich und erleichtert die Verwaltung erheblich.

Die auf NT-Technologie basierenden Windows-Systeme (NT / 2000 / XP) bieten ein ausgereifteres Benutzersystem an, welches viele der Möglichkeiten bietet, welche unter Unix bereits vor 30 Jahren üblich waren.

5 Das persönliche Verzeichnis - Heimatverzeichnis

Den Begriff Heimatverzeichnis gibt es unter Windows 95 bis ME eigentlich nicht. Wenn sie den Rechner starten, gehört der Computer Ihnen. In allen Verzeichnissen können Sie tun und lassen, was Sie wollen, und ebenso können Sie überall Ihre Daten speichern.

Nehmen Sie Veränderungen an den Systemeinstellungen vor, dann werden diese System weit übernommen und gelten ab sofort für den gesamten PC. Das so genannte Family-Logon ermöglicht es Ihnen immerhin, Ihren Desktop und das Startmenü individuell anzupassen. Alle anderen Einstellungen haben globale Auswirkungen.

In der Unix-Welt (und selbstverständlich auch unter Linux) existiert für jeden Benutzer ein so genanntes Heimatverzeichnis. Nach der Anmeldung befinden Sie sich automatisch in Ihrem persönlichen Verzeichnis. Dieses Heimatverzeichnis gehört Ihnen, und sie können damit (fast) alles tun und lassen, was sie wollen. Hier können Sie Ihre Daten speichern, Ihre E-Mails archivieren und kleinere Programme installieren. Die individuellen Einstellungen für Ihren Desktop und für viele andere Programme werden hier ebenso gespeichert. Ändern Sie Ihren Desktop, die Einstellungen Ihres E-Mail-Programms oder die Schriftfarbe der Konsole, betrifft das die anderen Benutzer in keiner Weise. Auch das Betriebssystem selbst wird dadurch nicht verändert. Das Heimatverzeichnis ist Ihr privater Bereich, und alle Konfigurationen, die Ihre persönliche Arbeitsumgebung betreffen, werden hier gesammelt gespeichert. Diese Daten werden weder vom Betriebssystem oder einem anderen Benutzer gebraucht. Wenn Sie Ihr Heimatverzeichnis löschen oder zerstören, interessiert das außer Ihnen selbst niemanden.



In größeren Netzwerken oder Firmen wird das Heimatverzeichnis in der Regel für private Dateien und Einstellungen verwendet, und die relevanten Firmendaten liegen gesammelt an einer anderen Stelle. Das soll gewährleisten, dass die privaten und die business-Dateien getrennt gehalten werden. Die Daten in Ihrem Verzeichnis können von keiner anderen Person gelesen, geschrieben oder ausgeführt werden, außer Sie erlauben es Ihr.

6 Der Desktop - grafische Benutzeroberfläche

Bei den heutigen modernen Betriebssystemen fällt es meist schwer, zwischen Kernel und grafischer Benutzeroberfläche zu unterscheiden. In den damaligen Zeiten vom MS-DOS war das ganze noch sehr einfach. Hier bildete MS-DOS das eigentliche Betriebssystem und Windows wurde aus dem Prompt heraus als eine grafische Benutzeroberfläche nachgeladen. (Besonders deutlich wird diese Trennung in Windows 3.11, dem Vorgänger von Windows 95.) Seit Windows NT / 2000 sind die beiden Komponenten zu einer Einheit verschmolzen. Die grafische Oberfläche war und ist ab diesem Zeitpunkt fester Bestandteil des Kernels, und lässt sich weder abschalten noch umgehen. Microsoft Windows ist zum Betriebssystem geworden.

In der Unix-Welt wird nach wie vor eindeutig zwischen Kernel und grafischer Oberfläche unterschieden. Unix lässt sich ohne grafische Oberfläche betreiben, und gerade im Server-Bereich ist dies auch häufig der Normalfall. Über eine Textkonsole lassen sich mittels einer Shell alle wichtigen Konfigurationsaufgaben eines Servers komfortabel erledigen. Das Aussehen erinnert an MS-DOS, und tatsächlich bilden die DOS-Kommandos ja auch nur eine kleine Untermenge der damaligen Unix-Kommandos - lediglich die Namen wurden ein wenig abgeändert und die Optionen anstelle mittels - durch einen Schrägstrich / übergeben. Da eine grafische Oberfläche viele Vorteile hat, ist diese auch unter Unix verfügbar. Der wesentliche Unterschied liegt darin, dass diese Oberfläche (meist das [X Window-System](#)) kein fester Bestandteil des Betriebssystems ist, sondern wie jedes andere Programm gestartet werden muss.

Mit einem X Window-System können Sie an sich noch nicht viel anfangen. Es bietet lediglich eine Reihe grafischer Grundfunktionen an, die von anderen Programmen verwendet werden können. Auf diesen Funktionen baut meist ein so genannter Window-Manager auf. Dieser stellt beispielsweise Titelleisten und andere

Rahmenbedingungen für einzelne Fenster bereit und sorgt für die Verwaltung und Anordnung unterschiedlicher Fenster auf einer Arbeitsfläche. Häufig bildet der Window-Manager seinerseits die Grundlage für eine Desktop-Umgebung wie  [KDE](#) oder  [Gnome](#), die den Funktionsumfang der grafischen Oberfläche noch erweitern. Es gibt allerdings zahlreiche sehr brauchbare Window-Manager, die ohne eine solche Desktop-Umgebung hervorragend zu verwenden sind und bereits einen sehr hohen Komfort bieten. Die Vielfalt der verfügbaren Window-Manager ermöglicht eine hohe Konfigurierbarkeit der grafischen Oberfläche. All dies mag für Sie ein wenig verwirrend klingen, aber Microsoft Windows macht das im Prinzip nicht anders. Nachdem das Betriebssystem geladen wurde, startet die Windows-Umgebung (gewissermaßen X Window und der Window-Manager in einem). Sie haben unter Microsoft Windows allerdings nicht die Wahl, welches Window-System oder welchen Window-Manager Sie starten möchten.

7 Die Shell



Ein wichtiger Unterschied zwischen Unix und Windows besteht in der Bedeutung der Shell-Verwendung. In den Zeiten von MS-DOS haben Sie Ihre Befehle noch am DOS-Prompt eingetippt und ausgeführt. Mit Windows hat sich dies grundlegend geändert. Fast alle Programme werden mittels Mausklick ausgeführt. Es existiert zwar noch ein DOS-Fenster, doch dieses hat fast keinerlei Bedeutung mehr.

Unter Unix stellt die Shell nach wie vor ein sehr wichtiges Arbeitsmittel dar. Alle Programme können von dort gestartet werden (wobei X-Anwendungen natürlich eine X Window-Umgebung für den Start benötigen). Die unter Unix verfügbaren Shells sind weit mächtiger als die Shell des Kommandozeilenfensters unter Windows.

Selbst wenn Sie unter der grafischen Benutzeroberfläche arbeiten, möchten Sie sehr oft nicht auf eine Shell verzichten. Sie können dann eine Konsolen-Anwendung wie beispielsweise `xterm` starten, in der Sie eine Shell verwenden.

Sie sollten sich aus diesen Gründen unbedingt mit der Funktionsweise einer Shell vertraut machen.

8 Zeilenumbrüche

Dateien werden in  [Binärdateien](#) und  [ASCII-Dateien](#) unterteilt. ASCII-Dateien zeichnen sich durch die besondere Bedeutung von Zeilenumbrüchen aus: Am Ende einer Zeile wird durch eine spezielle Zeichenkombination der Zeilenumbruch markiert. Leider gibt es für die Verwendung dieser Zeichenkombination keinen Betriebssystem übergreifenden Standard. Und tatsächlich unterscheiden sich Windows und Unix in der Handhabung von Zeilenumbrüchen. Unter Unix wird ein sogenannter Zeilenvorschub (engl.: line feed, ASCII-Zeichen **LF**) als Markierung eines Zeilenendes verwendet. Macintosh verwendet übrigens das Zeichen für Wagenrücklauf (engl.: carriage return, ASCII-Zeichen **CR**). Und Windows benutzt eine Kombination dieser beiden Zeichen, nämlich **CR-LF**. Das bedauerliche Ergebnis dieser Unterschiede ist, dass Textdateien, die nicht den für das jeweilige System typischen Zeilenumbruch verwenden, häufig unbrauchbar sind. Im einfachsten Fall werden sie in einem Editor falsch dargestellt oder enthalten Schmierzeichen. In schlimmeren Fällen schlägt das Parsen von Textdateien innerhalb einer Anwendung fehl und verursacht damit gravierendere Probleme.

Um derartige Probleme zu beheben, kann man ein Programm zur Konvertierung der betreffenden Dateien einsetzen. Das ist eigentlich nicht weiter kompliziert, da jeweils nur die Zeichenkombination, welche das Zeilenende markiert, an das jeweilige Betriebssystem angepasst werden muss.

Für den Datenaustausch wird meist ein **FTP**-Programm (File Transfer Protocol) verwendet. Aktuelle **FTP**-Programme haben eine Umschaltmöglichkeit zwischen ASCII- und Binärmodus. Binärdateien werden hier unverändert übertragen, während ASCII-Dateien vor der Abspeicherung entsprechend konvertiert werden. Allerdings wissen die **FTP**-Programme leider nicht, welche Ihrer Dateien in welchem Format gespeichert sind. Das heißt, dass Sie bei jedem Transfer angeben müssen, in welchem Format (ASCII oder binär) die Datei vorliegt. Sie können prinzipiell alle Dateien im Binär-Modus übertragen. Bei Textdateien sollten Sie jedoch in den ASCII-Modus umschalten. Damit wird gewährleistet, dass die Dateien konvertiert werden. FTP dient primär dem Datenaustausch. Die Konvertierung der Zeilenenden ist nur eine Nebenfunktion. Für die korrekte Verwendung müssen Sie selbst sorgen.

9 Befehlsunterschiede MS-DOS / Linux

Es folgt eine Liste der wichtigsten Befehle von MS-DOS und ihr Analogon in Linux. Dies soll Ihnen beim Einstieg in die Benutzung der Shell ein wenig helfen.

<code>attrib</code>	<code>chmod</code>	Dateiattribute (wie schreibgeschützt) ändern
<code>chkdsk</code>	<code>fsck</code>	Den Datenträger überprüfen
<code>cd</code>	<code>cd</code>	Verzeichnis wechseln
<code>cd</code>	<code>pwd</code>	Anzeigen der Position im Verzeichnisbaum
<code>copy</code>	<code>cp</code>	Dateien kopieren
<code>date, time</code>	<code>date</code>	Zeit und Datum anzeigen / ändern
<code>del</code>	<code>rm</code>	Dateien löschen
<code>deltree</code>	<code>rm -R</code>	Einen Verzeichnisbaum löschen
<code>diskcopy</code>	<code>dd</code>	Eine Diskette kopieren
<code>dir</code>	<code>ls</code>	Den Inhalt eines Verzeichnisses anzeigen
<code>dir</code>	<code>df</code>	Anzeige des verwendeten Speichers von Dateisystemen
<code>echo</code>	<code>echo</code>	Einen Text auf dem Bildschirm ausgeben lassen
<code>fc</code>	<code>diff</code>	Dateien miteinander vergleichen
<code>fdisk</code>	<code>fdisk</code>	Festplatte partitionieren
<code>find</code>	<code>grep</code>	Dateien nach einem Schlüsselwort durchsuchen
<code>find</code>	<code>find</code>	Eine Datei suchen
<code>format</code>	<code>mkfs</code>	Einen Datenträger formatieren
<code>md, mkdir</code>	<code>mkdir</code>	Ein neues Verzeichnis erstellen
<code>more</code>	<code>more, less</code>	Den Inhalt einer Datei am Screen ausgeben lassen
<code>move</code>	<code>mv</code>	Datei oder Verzeichnis verschieben
<code>rd, rmdir</code>	<code>rmdir</code>	Ein Verzeichnis löschen
<code>rename</code>	<code>mv</code>	Datei oder Verzeichnis umbenennen
<code>sort</code>	<code>sort</code>	Dateien oder Verzeichnisse sortieren
<code>type</code>	<code>cat</code>	Den Inhalt einer Datei auf den Bildschirm ausgeben
<code>xcopy</code>	<code>cp -a</code>	Mehrere Dateien oder ganze Verzeichnisse kopieren