

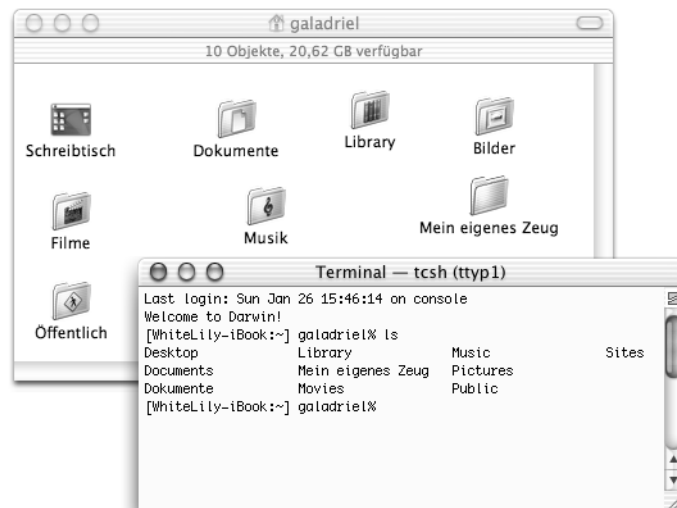
Terminal: Das Tor zu Unix

Ihnen ist sicherlich inzwischen aufgefallen, daß die Ähnlichkeit von Mac OS X mit dem traditionellen Betriebssystem des Mac rein oberflächlich ist. Unter der schönen Haut steckt ein völlig anderes Tier. Hier haben wir Unix, eines der ältesten und bestangesehenen Betriebssysteme der heutigen Zeit.

Wenn Sie es zum ersten Mal sehen, würden Sie schwören, daß Unix ungefähr soviel mit dem Mac OS zu tun hat wie ein Jeep mit einer Melone (siehe Abbildung 15-1).

Abbildung 15-1:

Oben: Woran die meisten Leute bei »Macintosh« denken, ist eine grafische Oberfläche – mit der Maus über Menüs und Symbole für Dateien und Befehle anzusprechen. Unten: Das Terminal bietet einen anderen Weg der Steuerung von Mac OS X: Eine Kommandozeilenschnittstelle – Sie übernehmen die Steuerung durch das Eintippen von Programmcode.



In der Abbildung unten in Abbildung 15-1 sehen wir natürlich eine *Kommandozeilenschnittstelle*: Eine Stelle, an der Sie Anweisungen an den Computer eintippen können. Eine Welt ohne Symbole, Menüs und Dialogfenster; sogar die Maus ist so gut wie nutzlos.

Sie folgen sicher der Ironie dieser Geschichte: Der Original-Mac von 1984 war so brillant, weil er die Kommandozeilenschnittstelle *unnötig machte*, die zu dieser Zeit auf allen Computern noch die Szene beherrschte (wie auf dem Apple II und auf DOS-Rechnern). Die meisten Nicht-Computerfreaks atmeten erleichtert auf und freuten sich darüber, nie wieder Befehle auswendiglernen zu müssen. Und nun haben wir hier Mac OS X, das angeblich so ultramoderne Betriebssystem von Apple, und es hat eine Kommandozeile. Was ist hier los?

In Wirklichkeit war die Kommandozeile nie weg. An den Universitäten und in Unternehmen weltweit waren die Computer-Profis stets fleißig dabei, Dinge am kleinen C:- oder %-Prompt einzuhacken, und sie schätzten die Effizienz und den großen Einfluß, den sie so auf ihre Computer hatten.

Es sei Ihnen vergeben, wenn Ihre erste Reaktion auf das Erlernen von Unix ist: »Ja, zum Kuckuck noch mal – kann ich nicht wenigstens erst einmal mein neues Betriebssystem beherrschen lernen, bevor ich mir noch ein anderes aneigne?«

Aber natürlich. Sie *müssen* die Kommandozeile von Mac OS X nicht benutzen. Apple hat sie sogar ganz weit unter den Teppich gekehrt, in der offensichtlichen Annahme, daß die meisten Leute die schönen Symbole und Menüs des normalen Schreibtischs bevorzugen. In den frühen Versionen von Mac OS X gab es ein paar Funktionen, die *ausschließlich* über die Kommandozeile erreichbar waren; aber von diesen gibt es in 10.2 noch weniger.

Für Mac-Fans mit einigem oder großem Mac-Wissen und etwas Zeit und Neugierde bietet die Kommandozeile eine ganze Welt von Möglichkeiten. Sie können damit in Bereiche von Mac OS X vordringen, auf die Sie vom normalen Schreibtisch aus keinen Zugriff haben. Verschiedene Aufgaben lassen sich weitaus effizienter und schneller durchführen als durch Anklicken und Ziehen von Symbolen. Und Sie erhalten einen faszinierenden Einblick in die Gemütswelt von Leuten, deren Leben im Computer-Bereich stattfindet.

Wenn Sie schon einmal mit ResEdit herumgemacht, mit AppleScript experimentiert oder ein Mac-Netzwerk eingerichtet haben, dann kennen Sie bereits den Schwierigkeitsgrad des Materials, das jetzt kommt. Mag sein, daß die Unix-Kommandozeile *nicht vertraut* ist, aber sie muß nicht besonders technisch sein, besonders dann nicht, wenn Sie wie in diesem und im nächsten Kapitel bestimmten »Rezepten« folgen können.

Hinweis: Unix ist natürlich ein ganz eigenes Betriebssystem. In diesem Kapitel geht es darum, Ihnen einen Eindruck zu verschaffen, damit Sie entscheiden können, ob Sie Unix mögen oder nicht. Falls Sie sich dafür begeistern, finden Sie im Anhang E einige weitere Unix-Quellen.

IM HÖCHSTEN GANG

Die Unix-Wurzeln von Mac OS X

Anno 1969 hatte Bell Labs-Programmierer Ken Thompson etwas Zeit übrig, als sein Hauptprojekt, ein Betriebssystem namens Multics, abgebrochen wurde. Bell Labs hatte sich aus dem teuren Projekt zurückgezogen und war nach vier Jahren harter Arbeit enttäuscht über die Ergebnisse.

Thompson war jedoch immer noch der Ansicht, das Projekt – ein Betriebssystem, das als kooperative Software-Entwicklungs-umgebung funktionierte – sei eine gute Idee. Irgendwann entstand aus einer Zusammenarbeit mit seinem Kollegen Dennis Ritchie ein Betriebssystem, das bald Unix genannt wurde (ein Seitenhieb auf Multics). Bell Labs erkannte den Wert von Unix, unterstützte die weitere Entwicklung und setzte als erstes Unternehmen dieses Betriebssystem ein.

Als Thompson und Ritchie mit der Arbeit an Unix begannen, schrieben die meisten Programmierer noch Code, der nur auf einem speziellen Computer (oder sogar nur auf einem bestimmten Computer-Modell) funktionierte. Unix hingegen war eines der ersten *portablen* Betriebssysteme: seine Programme liefen ohne Änderungen auf verschiedenen Computern. Das lag daran, daß Thompson und Ritchie für Unix eine neue, eigene Programmiersprache erfunden hatten, und zwar C.

Der Programmierer muß seinen Code nur einmal in einer Sprache wie C schreiben. Danach kann ein Software-Helferlein namens *Compiler* die frischgeschlüpfte Software in eine Form bringen, die ein bestimmtes Computer-Modell versteht.

Unix fand bald seinen Weg in die Labors und dank der kostengünstigen Universitätslizenzen von AT&T auch in die Universitäten weltweit. Programmierer auf der ganzen Welt fügten neuen Quellcode hinzu, beseitigten Fehler und gaben danach die

Änderungen an die Gemeinde weiter.

Mitte der 70er Jahre wurde die Universität von Kalifornien in Berkeley zum Schauplatz weitreichender Unix-Entwicklungen. Die Studenten und Dozenten dort verbesserten den Unix-Kernel (den zentralen Hauptteil des Betriebssystems), fügten dem Betriebssystem neue Funktionen hinzu und schrieben neue Unix-Programme. 1977 war soviel Software zusammengekommen, daß man eine eigene Unix-Version veröffentlichte, die erste von mehreren sogenannten *Berkeley Software Distribution* (BSD)-Versionen.

Zu dieser Zeit suchte die amerikanische Defense Advanced Research Projects Agency (DARPA) zufällig ein einheitliches, portables Betriebssystem für ihr wachsendes Weitverkehrsnetzwerk, das ursprünglich ARPAnet hieß (und nun als Internet bekannt ist).

Der DARPA gefiel Unix, und so wurden weitere Forschungen in Berkeley finanziert. Im Januar 1983 änderte die DARPA das Netzwerkprotokoll des ARPAnet auf TCP/IP – und das Internet war geboren und lief hauptsächlich auf Unix-Rechnern.

1985 verließ Steve Jobs Apple und gründete NeXT Computer, dessen Betriebssystem NextStep auf BSD Unix aufgebaut war. Als Apple NeXT 1996 aufkaufte, kamen Herr Jobs, NextStep (dies wurde irgendwann OpenStep getauft) und sein Terminal-Programm mit. Das Unix, das im Herzen von Mac OS X schlägt, ist einfach nur der neue Landeplatz des Betriebssystems, das vom Jobs-Team bei NeXT entwickelt wurde.

Wenn Sie also zum nächsten Mal von Apple etwas über das »neue« Betriebssystem hören, dann wissen Sie, daß die Technik, auf der es basiert, in Wahrheit über 30 Jahre alt ist.

Terminal

Das Schlüsselloch zur Unix-Innenwelt von Mac OS X ist ein Programm namens Terminal, das Sie im Ordner Programme→Dienstprogramme finden (siehe Abbildung 15-2). Terminal hat seinen Namen von den Terminals übernommen (Compu-

tern, die nur aus Monitor und Tastatur bestehen), die an Universitäten und in großen Firmen an einen Großrechner angeschlossen sind. Genauso ist Terminal nur ein Fenster, über das Nachrichten mit dem Hirn des Mac ausgetauscht werden können.

Wenn Sie das Programm Terminal zum ersten Mal öffnen, werden Sie bemerken, daß es in seinem Fenster außer dem Datum und der Uhrzeit Ihres letzten Logins, einem Willkommensgruß und dem *Kommandozeilenprompt* (Abbildung 15-2) nicht viel zu sehen gibt.

Bei den Benutzerfreundlichkeitsfetischisten hat das Terminal keinen guten Stand – technischer als mit dem Prompt geht's bei Computern nicht mehr. Dies sind seine Bestandteile:

- **office-mac:** ist der Name Ihres Mac (das denkt zumindest Unix). Normalerweise wird hier der Rendezvous-Name des Mac verwendet (so wie er unter Sharing in den Systemeinstellungen erscheint), manchmal ist es aber auch der Name Ihres Mac im Internet.
- **~.** Der nächste Teil des Prompts gibt an, in welchem Ordner Sie sich befinden (siehe Abbildung 15-2). Dies ist Ihr *Arbeitsverzeichnis* – also der aktuelle Ordner (vergessen Sie nicht, es gibt unter Unix keine Symbole). Im Grunde gibt dieser Abschnitt an, wo Sie beim Navigieren durch Ihren Rechner gerade gelandet sind.

Wenn Sie Terminal zum ersten Mal ausprobieren, ist das Arbeitsverzeichnis auf das Symbol `~` eingestellt. Wie schon erwähnt, ist das Tilde-Symbol (`~`) eine Abkürzung für »Ihr eigener Privat-Ordner«. Diesen sehen Sie, wenn Sie Terminal zum ersten Mal aufrufen, aber bald werden Sie hier die Namen anderer Ordner sehen – [*office-mac: /Users*] oder [*office-mac: /System/Library*] zum Beispiel (näheres zu dieser Schrägstrichschreibweise auf Seite 52).

Hinweis: Bevor Apple den benutzerfreundlichen Begriff *Ordner* für einen elektronischen Aufbewahrungsort für Dateien aus der Taufe hob, hießen Ordner *Verzeichnisse* (ja, ist das gleiche). Aber in diesem und auch im nächsten Kapitel werden Sie fast ausschließlich auf den Begriff *Verzeichnis* stoßen. In einer Unix-Besprechung ist »Verzeichnis« einfach der korrekte Begriff.

Außerdem riskieren Sie den Tod am Strang, wenn Sie in Gegenwart von Unix-Experten Begriffe wie »Arbeitsordner« fallen lassen.

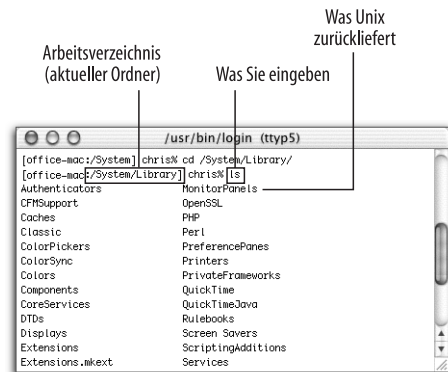
-
- **chris%** fängt mit Ihrem Benutzerkurznamen an. Hier sieht man, wer sich an der *Shell* angemeldet hat (siehe Kasten auf Seite 501), und das ist üblicherweise der Benutzer, der gerade am *Mac* angemeldet ist. Nun zum %-Zeichen: Stellen Sie es sich als Doppelpunkt vor. Eigentlich sagt Unix mit dem ganzen Prompt, der in Abbildung 15-2 zu sehen ist, einfach nur: »Ok, Chris. Ich bin soweit. Was darf's sein?«

Wenn Sie nicht mit den Einstellungen von Terminal gespielt haben, sieht die Einfügemarke wie ein hochkantiges Rechteck am Ende der Kommandozeile aus. Es wandert nach rechts, wenn Sie tippen.

Abbildung 15-2:

Im Web gehört das Terminal von Mac OS X zu den meistdiskutierten Elementen von Mac OS X. Dutzende Tutorials für das schrittweise Ausführen bestimmter Aufgaben sind online im Umlauf, und meistens findet sich in ihnen nicht viel Information darüber, warum man das tippt, was man tippt.

In diesem Kapitel dürfen Sie nicht vergessen, daß beim Terminal die Groß- und Kleinschreibung eine Rolle spielt, obwohl das im Finder nicht so ist. Was Unix betrifft, sind Hallo und hallo zwei ganz verschiedene Sachen.



SPRECHSTUNDE FÜR POWER USER

Von Terminal und Shells

Ein Unix-Programm läuft automatisch, wenn Sie ein Terminal-Fenster öffnen: *tcsh* (wird »Ti-Si-Shell« ausgesprochen).

Technisch gesehen ist eine *Shell* ein Unix-Programm, das die von Ihnen eingegebenen Befehle interpretiert, sie an den *Kernel* (das Gehirn des Betriebssystems) weitergibt und Ihnen dann die Antwort des Kernels anzeigt.

Aber praktisch gesehen ist die Shell der Unix-Finder. In diesem Programm können Sie durch Ihre Festplatte navigieren, den Inhalt bestimmter Ordner einsehen, Programme und Dokumente aufrufen und so weiter.

Es gibt unter Unix sogar verschiedene Shells, von denen jede eine leicht andere Befehlssyntax hat. Es werden mit Mac OS X noch ein paar andere Shells ausgeliefert – zum Beispiel *sh*, die Bourne-Shell, *bash*, die Bourne Again-Shell (hey, das ist gut!), *zsh* und *csh*. Aber das Terminal ist auf *tcsh* eingestellt.

Sie können weitere Terminal-Fenster öffnen (bis zu 36), indem Sie ABLAGE → NEUES SHELL-FENSTER wählen. Jedes Fenster läuft unabhängig von den anderen Fenstern. Und falls Sie einen Beweis brauchen, öffnen Sie einmal mehrere Fenster, und lassen Sie in allen den Befehl *cal* ausführen.

Unix-Programme

Für Unix gibt es eine Unzahl von Programmen. Und dank Tausender von Open Source-Entwicklern – Programmierern auf der ganzen Welt, die zusammenarbeiten und ihre Arbeit dann für die nächste Modifizierungsrunde freigeben – ist ein Großteil dieser Software frei und kostenlos erhältlich – auch für Mac OS X-Benutzer.

Jeder Unix-Befehl ruft im Normalfall eine bestimmte Operation (oder einen Prozeß, wie Experten sagen würden) auf, die startet, eine Aufgabe ausführt und sich danach wieder beendet. Viele der bekanntesten Programme dieser Art sind bei Mac OS X dabei.

Hier ein lustiges Beispiel: geben Sie einfach *uptime* ein, und drücken Sie Enter oder Return (so führen Sie ein Unix-Programm aus: einfach seinen Namen eingeben und Enter drücken). In der nächsten Zeile zeigt Ihnen Terminal, seit wann Ihr Mac ununterbrochen läuft. Dies könnte zum Beispiel so aussehen: »6:00PM up 8 days, 15:04, 1 user, load averages: 1.24, 1.37, 1.45« – was bedeutet, daß Ihr Mac seit 8 Tagen und 15 Stunden nonstop läuft.

Das Programm *uptime* ist beendet. Der %-Prompt erscheint wieder und zeigt an, daß Terminal für den nächsten Ball bereit ist, den Sie ihm zuwerfen.

Versuchen Sie es damit: Tippen Sie am Prompt *cal* ein, und drücken Sie Enter. Unix spuckt schnurstracks einen Kalender des aktuellen Monats aus.

```
[office-mac:~] chris% cal
      September 2002
  S  M Tu  W Th  F  S
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30
[office-mac:~] chris%
```

Versuchen Sie es nun mit *cal 11 2003*, *cal -y* oder *cal -yj*. Über diese drei Befehle erzeugt Unix einen Kalender von November 2003, einen Kalender des aktuellen Jahres und einen *Julianischen* Kalender des aktuellen Jahres.

IM HÖCHSTEN GANG

1x1 der Pfadnamen

In vielerlei Hinsicht funktioniert das Durchstreifen des Festplatteninhalts mit dem Terminal genauso wie mit dem Finder. Sie fangen bei einem Ordner an und gehen dann in seine Unterordner oder in die übergeordneten Ordner hinein.

In diesem und im nächsten Kapitel werden Sie dazu aufgefordert, eine bestimmte Datei oder einen bestimmten Ordner in dieser Ordnerhierarchie anzugeben. Aber Sie können von der Kommandozeile aus ihre Symbole nicht sehen. Wie sollen Sie also die Datei oder den Ordner identifizieren?

Indem Sie den entsprechenden Pfadnamen eingeben. Der Pfadname besteht aus einer Reihe von Ordnernamen, die Sie

wie eine Landkarte von der *Root-Ebene* in den nächsten verschachtelten Ordner, dann in den nächsten und so weiter bringen (für uns Einsteiger entspricht die Root-Ebene dem Hauptfestplattenfenster).

Unter Unix wird dieses durch einen einfachen Schrägstrich gekennzeichnet. Der Ausdruck */Users* bedeutet also »Der Ordner Users in meinem Festplattenfenster« oder, anders ausgedrückt, »das Users-Verzeichnis auf Root-Ebene«.

Möchten Sie auf den Dokumente-Ordner in Ihrem eigenen Privat-Ordner verweisen, dann könnten Sie zum Beispiel */Users/chris/Documents* tippen (natürlich nur, wenn Ihr Name Chris ist).

Tip: Die Maus ist in der Kommandozeile nicht sehr hilfreich. Sie bewegen den Cursor im Allgemeinen nur mit den Tasten Pfeil-nach-links und Pfeil-nach-rechts (die Löschtaste funktioniert genauso wie immer).

Sie *können* aber mit der Maus Text irgendwo im Fenster (oder in einem anderen Programm) auswählen und am Prompt einfügen. Sie können mit der Maus auch ein Symbol vom Schreibtisch ins Terminal-Fenster ziehen, wie in Abbildung 15-3 zu sehen ist. Und im Dialogfenster `TERMINAL` → `FENSTEREINSTELLUNGEN` → `EMULATION` finden Sie eine Option, über die Sie per Option-Klick in eine Kommandozeile die Einfügemarke dorthin setzen können.

Unter Unix navigieren

Wenn Sie keine Symbole für Ordner und Dateien sehen, wie sollen Sie dann mit ihnen arbeiten?

Sie können Unix nur bitten, Ihnen mitzuteilen, welchen Ordner Sie sich gerade ansehen (über den Befehl `pwd`), was darin enthalten ist (über den Befehl `ls`), und ihm dann sagen, zu welchem Ordner Sie gehen möchten (über den Befehl `cd`), wie auf den folgenden Seiten zu sehen.

`pwd` (print working directory oder auch »Wo bin ich?«)

Hier einer der wichtigsten Navigationsbefehle: `pwd`, Abkürzung von *print working directory*, Arbeitsverzeichnis ausgeben. Der Befehl `pwd` druckt nichts auf den Drucker; statt dessen gibt er auf dem Bildschirm den *Pfad* aus, in dem Sie sich laut Unix befinden (das Arbeitsverzeichnis).

Tippen Sie versuchsweise `pwd`, und drücken Sie Enter. Auf der nächsten Zeile zeigt Ihnen Terminal zum Beispiel folgendes:

```
/Users/chris/Movies
```

Terminal gibt den Pfad des Arbeitsverzeichnisses aus – eine Liste mit jeweils durch Schrägstrich voneinander getrennten Ordnern innerhalb von Ordnern, die den Aufenthaltsort eines bestimmten Ordners auf Ihrer Festplatte angibt. `/Users/chris/Movies` führt uns zum Filme-Ordner `Movies` im Privat-Ordner von Chris (der sich wie alle Privat-Ordner im Users-Verzeichnis befindet).

Tip: Die Unterscheidung zwischen Groß- und Kleinschreibung ist wichtig unter Unix. Befehlsnamen werden fast immer klein geschrieben (wie `cal` oder `pwd`). Bei der Eingabe von *Ordernamen* müssen Sie auf die richtige Schreibweise achten.

`ls` (list oder »Was ist hier drin?«)

Über den Befehl `ls` (Abkürzung von *list*) gibt Terminal die Namen aller Dateien und Ordner innerhalb des Ordners an, in dem Sie sich befinden (also in Ihrem Arbeitsverzeichnis). Sie können das gleich einmal ausprobieren: Tippen Sie `ls`, und drücken Sie Enter. Terminal zeigt Ihnen die Namen der Dateien und Ordner in einer waagerechten Liste an:

```
Desktop Documents Library Movies Music Pictures Public Sites
```

Mit anderen Worten sehen Sie eine Liste der Symbole, die im Finder in Ihrem Privat-Ordner erscheinen würden.

Hinweis: Terminal respektiert die verschiedenen Zugriffsrechte der Mac OS X-Accounts (Kapitel 11). Das heißt, der Inhaber eines normalen oder Administrator-Accounts kann keinen Blick in den Privat-Ordner eines anderen Benutzers werfen. Versuchen Sie dies, so sagt Ihnen Terminal »Permission denied«.

Terminal kann Ihnen auch den Inhalt eines beliebigen anderen Verzeichnisses auflisten (das *nicht* das Arbeitsverzeichnis ist), wenn Sie den Pfadnamen als *Argument* hinzufügen. Ein Argument teilt dem Befehl mit, womit er arbeiten soll. (Erinnern Sie sich noch an das Kalenderbeispiel? Als Sie den Kalender für November 2003 benötigten, haben Sie *cal 11 2003* eingegeben. »11 2003« war hier das Argument.)

Alle Dateien in Ihrem Documents-Verzeichnis könnten Sie also über *ls /Users/chris/Documents* abfragen. Und weil das *~*-Symbol eine Kurzform für »mein Privat-Verzeichnis« ist, könnten Sie sich mit *ls ~/Documents* ein wenig Tipparbeit sparen. Der Pfadname »~/Documents« ist ein Argument für den Befehl *ls*.

Über Optionen (»Flags«)

Zwischen einen Befehl und seine Argumente kann man manchmal auch sogenannte *Optionen* (auch *Flags* genannt) einfügen – kurze Ausdrücke, die die Wirkungsweise des Befehls verändern. Im Kalenderbeispiel könnten Sie mit *cal -y* den Kalender für ein ganzes Jahr sehen. *-y* ist eine Option.

Optionen wird fast immer ein Bindestrich vorangestellt (-), allerdings können Sie normalerweise mehrere Optionen hinter einen Bindestrich stellen, zum Beispiel so: *ls -ai* (auf diese Weise wird sowohl *-a* als auch *-i* gesetzt).

Hier ein paar nützliche Beispiele für den *ls*-Befehl:

- **-a.** Damit stellt der *ls*-Befehl auch die Namen von *verborgenen* Dateien und Ordnern dar – zumindest von denen, die im Finder so definiert wurden. Die Unix-Shell hat ihr eigenes System zum Verbergen von Dateien und Ordnern und ignoriert das System des Finders. Das heißt aber nicht, daß Sie alles sehen; die Dateien, die laut Unix-Definition unsichtbar sind, bleiben trotzdem verborgen.

Mit einer der Optionen für *ls* können Sie aber sogar die für Unix verborgenen Dateien hervorzaubern. Fügen Sie einfach die Option *-a* hinzu, und tippen Sie also dies: *ls -a*. Wenn Sie nun Enter drücken, dann sehen Sie ungefähr so etwas:

```
.      Desktop          Movies          Public
..     Documents        Music           Sites
.      CFUserTextEncoding Library         Pictures
```

- **-F.** Wie Sie sehen, beginnen die Namen der verborgenen Unix-Dateien jeweils mit einem Punkt (Unix-Leute nennen sie »Dot-Files« oder auch »Punkt-Dateien«). Aber handelt es sich hier um Dateien oder um Ordner? Dies können

Sie herausfinden, indem Sie `ls` mit der Option `-F` (auf Großschreibung achten) benutzen, und zwar so: `ls -aF Movies`. Es erscheint etwa dies:

```
./ Picnic Movie 1 Yosemite/      Reviews.doc
../ Picnic Movie 2 Tahoe/
```

Die Namen der Einträge selbst haben sich nicht geändert, aber die Option `-F` lässt Schrägstriche an den Verzeichnis-(Ordner-)namen erscheinen. In diesem Beispiel sehen wir vier Verzeichnisse und eine Datei in `~/Movies` (und das ist der Ordner `Movies` in Ihrem Privat-Verzeichnis).

- **-R.** Die Option `-R` erzeugt eine *rekursive* Liste – hier sehen Sie die Verzeichnisse *innerhalb* der Verzeichnisse in der Liste. Geben Sie zum Beispiel `ls -R` ein, dann erhalten Sie ungefähr folgendes:

```
Bad Reviews.doc  Old Tahoe Footage 2 Picnic Movie 2  Reviews.doc

./Old Tahoe Footage 2:
Tahoe 1.mov      Tahoe 3.mov      Tahoe Project File
Tahoe 2.mov      Tahoe 4.mov

./Picnic Movie 2:
Icon?           Media           Picnic Movie 2 Project

./Picnic Movie 2/Media:
Picnic Movie 1 Picnic Movie 3 Picnic Movie 5
Picnic Movie 2 Picnic Movie 4 Picnic Movie 6
```

Tip: Wie Sie schon an den Beispielen mit `cd` und `ls` gesehen haben sind Unix-Befehle sehr kurz. Sie bestehen oft nur aus zwei Buchstaben, und für eine eindrucksvolle Anzahl von ihnen werden *beide Hände* (`ls`, `cp`, `rm` und so weiter) verwendet.

Der Grund hierfür liegt zum Teil im begrenzten Speicher der alten Computer und zum Teil in der Effizienz: Die meisten Programmierer möchten möglichst wenig tippen. Das ist zwar nicht benutzerfreundlich, aber wenn Sie diese Befehle ein paar Monate lang täglich tippen, dann sind Sie irgendwann einmal dankbar für die gesparten Tastendrucke.

`cd` (change directory oder »Zeig mir einen anderen Ordner«).

Jetzt wissen Sie schon, wie Sie herausfinden, in welchem Verzeichnis Sie sich befinden, und wie Sie sehen können, was sich darin befindet, und das alles ohne einen einzigen Doppelklick auf ein Symbol. Das ist schon sehr gut, dient aber nur der Information. Wie aber *make* ich nun wirklich etwas im Kommandozeilen-Finder – zum Beispiel in ein anderes Verzeichnis wechseln?

Sie wechseln Ihr Arbeitsverzeichnis, indem Sie den Befehl `cd` und danach den Pfad des Verzeichnisses eintippen, in das Sie wechseln möchten. Sie wollen sehen, was sich im `Movies`-Verzeichnis Ihres Privat-Verzeichnisses befindet? Geben Sie

`cd /Users/chris/Movies` ein, und drücken Sie Enter. Der `%`-Prompt zeigt Ihnen das Verzeichnis, in dem Sie sich jetzt befinden (das neue Arbeitsverzeichnis). Wenn Sie nun den Befehl `ls` ausführen, dann zeigt Ihnen Terminal den Inhalt des Movies-Verzeichnisses.

Das ist natürlich ganz schön viel Tipparbeit. Anstatt diesen ganzen Pfad (den sogenannten *absoluten* Pfad) einzutippen, können Sie glücklicherweise auch einfach angeben, welches Verzeichnis Sie *relativ* zu dem Verzeichnis sehen möchten, in dem Sie sich gerade befinden.

Ist Ihr Privat-Ordner zum Beispiel gerade das Arbeitsverzeichnis, dann lautet der relative Pfadname für das Verzeichnis Trailers im Verzeichnis Movies `Movies/Trailers`. Das ist viel kürzer als der ganze absolute Pfadname (`/Users/chris/Movies/Trailers`).

Wenn Ihr Hirn nicht schon vor Stress zu kochen angefangen hat, hier eine Zusammenfassung der drei verschiedenen Möglichkeiten, von `~/`(Ihr Privat-Verzeichnis) zu `~/Movies` zu wechseln:

- **`cd /Users/chris/Movies`.** Die lange Methode – der absolute Pfadname. Funktioniert unabhängig vom gerade aktuellen Arbeitsverzeichnis.
- **`cd ~/Movies`.** Auch dies ist ein absoluter Pfadname, den Sie von beliebiger Stelle aus eingeben können. Er macht sich die Abkürzung `~` zunutze (die immer »mein Privat-Verzeichnis« bedeutet).
- **`cd Movies`.** Dieser abgespeckte *relative* Pfad macht sich die Tatsache zunutze, daß Sie bereits in Ihrem Privat-Verzeichnis sind.

Tip: Es gibt eigentlich sogar noch eine vierte Methode, bei der Sie gar nichts tippen müssen: *Ziehen Sie das Symbol* des Verzeichnisses, das Sie angeben möchten, direkt in das Terminal-Fenster. Dies wird in Abbildung 15-3 deutlicher.

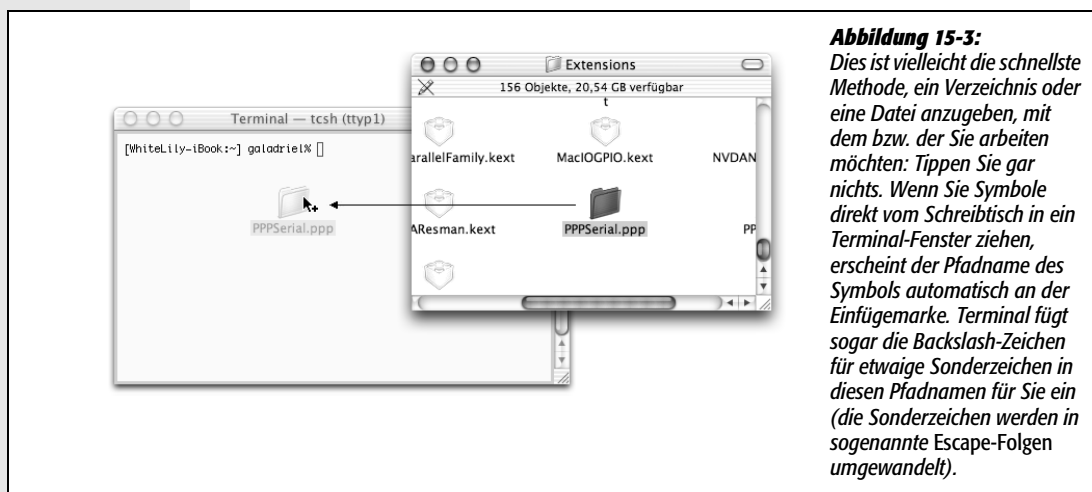


Abbildung 15-3: Dies ist vielleicht die schnellste Methode, ein Verzeichnis oder eine Datei anzugeben, mit dem bzw. der Sie arbeiten möchten: Tippen Sie gar nichts. Wenn Sie Symbole direkt vom Schreibtisch in ein Terminal-Fenster ziehen, erscheint der Pfadname des Symbols automatisch an der Einfügemarke. Terminal fügt sogar die Backslash-Zeichen für etwaige Sonderzeichen in diesen Pfadnamen für Sie ein (die Sonderzeichen werden in sogenannte Escape-Folgen umgewandelt).

. und .. (Punkt und Punkt-Punkt oder »Ich will raus hier«)

Nun haben Sie sich also in Ihr Movies-Verzeichnis vergraben. Wie kommen Sie da wieder raus?

Freilich, Sie könnten jetzt wieder den vollen Pfadnamen des Verzeichnisses eintippen, das Movies enthält – wenn Sie heute nachmittag noch nichts vorhaben. Aber es gibt auch eine Abkürzung: Sie können einen Punkt (.) oder zwei Punkte (..) in einem Pfadnamen tippen. Diese Abkürzungen stehen für das *aktuelle Verzeichnis* und sein *Parent-Verzeichnis* (das Verzeichnis, das das aktuelle Verzeichnis enthält).

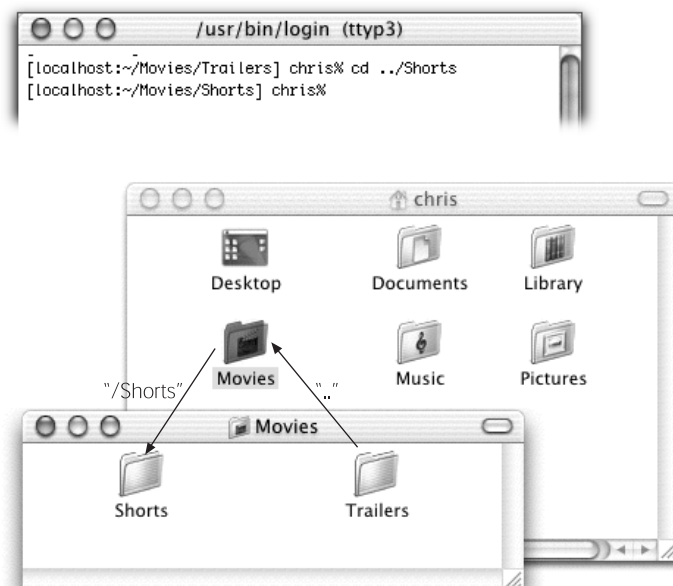
Sie gelangen zum Beispiel von Ihrem Privat-Verzeichnis zu `/Users`, indem Sie einfach `cd ..` tippen (also `cd` gefolgt von einem Leerschritt und zwei Punkten).

Sie können über *wiederholte* `..`-Eingabe viele Verzeichnisse auf einmal hinaufsteigen, und zwar so: `cd ../../`, was bedeuten würde »wechsle das Arbeitsverzeichnis auf das Verzeichnis zwei Ebenen höher«. Befinden Sie sich im Movies-Verzeichnis, würde `../../` das Verzeichnis `Users` als Arbeitsverzeichnis definieren.

Noch ein Trick: Sie können die `..`-Abkürzung mit richtigen Verzeichnisnamen mischen. Nehmen wir zum Beispiel an, Ihr Movies-Verzeichnis enthält zwei Verzeichnisse, `Trailers` und `Shorts`. `Trailers` ist das aktuelle Verzeichnis, aber Sie möchten in das Verzeichnis `Shorts` wechseln. Sie brauchen nur `cd ../Shorts` eingeben, wie in Abbildung 15-4 zu sehen ist.

Abbildung 15-4:

Zwei Punkte weisen Unix an, sein Augenmerk auf das Movies-Verzeichnis zu lenken (und im Verzeichnisbaum nach oben zu wandern), der Rest läßt es vom Movies-Verzeichnis ins Verzeichnis `Shorts` hinunterwandern. Beachten Sie, daß der Prompt stets das aktuelle Arbeitsverzeichnis angibt.



Tastenschläge sparen

Jetzt glauben Sie vielleicht, daß das Anklicken von Symbolen immer noch schneller als all das Getippe ist. An dieser Stelle kommen die Tippkürzel der *tcs*h-Shell zum Zuge.

Tabulatorergänzung

Sie wissen doch, daß Sie im Finder ein Fenster markieren können, indem Sie die ersten paar Buchstaben seines Namens eintippen? Die Tabulatorergänzung funktioniert ganz ähnlich. Im Lauf der Zeit können Sie sich Kilometer von Fingerbewegungen sparen.

Sie schaltet sich ein, wenn Sie gerade mit dem Tippen eines Pfadnamens anfangen. Beginnen Sie mit den ersten ein, zwei Buchstaben des gewünschten Pfads und drücken Sie dann die Tabulatortaste; Terminal ergänzt sofort den Rest des Verzeichnisnamens. Wie in Abbildung 15-5 zu sehen ist, können Sie diesen Vorgang wiederholen und so den nächsten Abschnitt des Verzeichnispfades angeben.

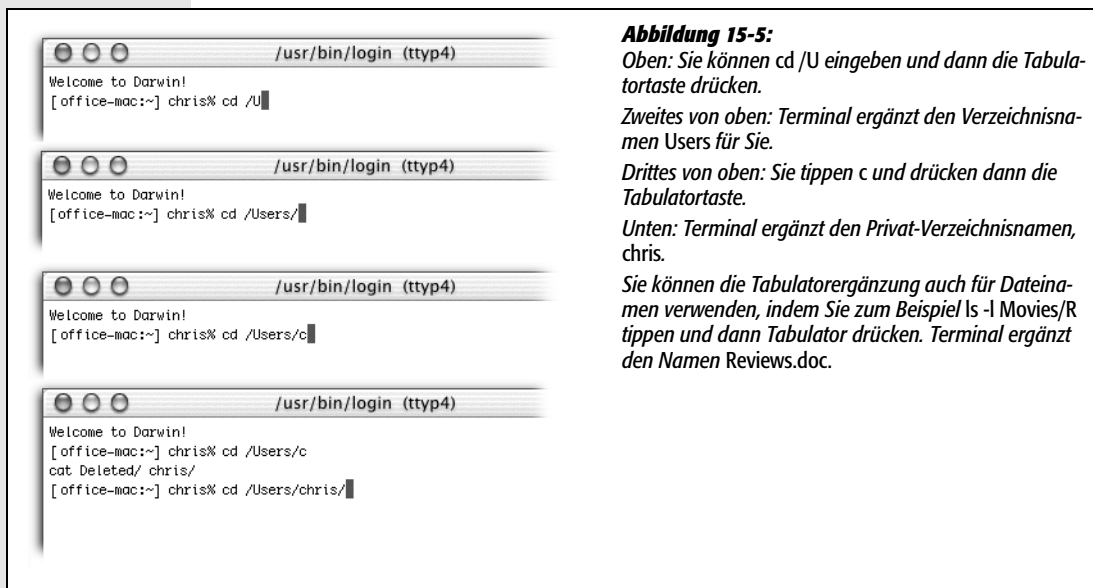


Abbildung 15-5:

Oben: Sie können cd /U eingeben und dann die Tabulatortaste drücken.

Zweites von oben: Terminal ergänzt den Verzeichnisnamen Users für Sie.

Drittes von oben: Sie tippen c und drücken dann die Tabulatortaste.

Unten: Terminal ergänzt den Privat-Verzeichnisnamen, chris.

Sie können die Tabulatorergänzung auch für Dateinamen verwenden, indem Sie zum Beispiel ls -l Movies/R tippen und dann Tabulator drücken. Terminal ergänzt den Namen Reviews.doc.

Einige Tips zur Tabulatorergänzung:

- Die Groß- und Kleinschreibung ist von Bedeutung.
- Terminal fügt automatisch Backslash-Zeichen hinzu, wenn Ihr Verzeichnisname Leerzeichen oder Sonderzeichen enthält. Aber Sie müssen die Backslash-Zeichen selber eingeben, wenn Sie die ersten paar Buchstaben, die die Tabulatorergänzung auslösen.
- Findet Terminal kein Verzeichnis, das mit dem Getippten übereinstimmt, so ertönt ein Warnton.

Findet es *mehrere* Dateien oder Verzeichnisse, die mit dem Getippten übereinstimmen, so piept Terminal und zeigt eine Liste an. Tippen Sie einfach die nächsten ein, zwei Buchstaben des gewünschten Eintrags, und drücken Sie erneut die Tabulatortaste.

Befehlsspeicher verwenden

Irgendwann müssen Sie vielleicht einen bereits erteilten Befehl erneut eingeben, aber der Gedanke an die ganze erneute Tipparbeit ist Ihnen ein Greuel. Es ist jedoch nie nötig, einen Befehl noch einmal zu schreiben. Terminal (oder besser gesagt die Shell, die darauf läuft) erinnert sich an die letzten 150 eingegebenen Befehle. Drücken Sie an einem beliebigen Prompt die Pfeil-nach-oben- oder die Pfeil-nach-unten-Taste, und navigieren Sie so durch die verschiedenen Befehle im Shell-Speicher. Sie erscheinen nacheinander am %-Prompt – gleich in der richtigen Zeile.

Joker (Wildcards)

Joker sind spezielle Zeichen, die andere Zeichen repräsentieren – und sie sparen enorm viel Zeit.

Das beliebteste Jokerzeichen ist das Sternchen (*), und es bedeutet »hier kann beliebiger Text stehen«. Möchten Sie zum Beispiel eine Liste aller Dateien in Ihrem Arbeitsverzeichnis sehen, die mit den Buchstaben *te* enden, könnten Sie *ls *te* eingeben. Terminal zeigt Ihnen die Dateien namens Yosemite, BudLite, Brigitte und so weiter – und alle anderen Dateien der Liste sind verborgen.

Möchten Sie entsprechend alle Dateien und Verzeichnisse einsehen, die mit »Old« beginnen, könnten Sie *ls Old** eingeben und Enter drücken. Sie sehen nur die Namen der entsprechenden Symbole im Verzeichnis, Old Yeller, Old Tahoe Footage, Olduvai Software und so weiter.

Fügen Sie vor *und* hinter dem Suchbegriff ein Sternchen ein, so finden Sie die Einträge, die diesen Ausdruck *irgendwo* in ihrem Namen aufweisen. Tippen Sie *ls *jo**, so erscheinen die Dateinamen Johnson, Mojo, Major Disaster und so weiter.

Tip: Nur * alleine bedeutet »zeig mir alles«. Möchten Sie eine Liste mit dem, was in einem Verzeichnis *und in all seinen Unterverzeichnissen* enthalten ist (als wenn Sie alle Ordner in der Listendarstellung im Finder markieren und dann ⌘-Pfeil-nach-rechts drücken), tippen Sie einfach *ls **.

Verzeichnis wechseln

Ein Bindestrich (-) nach dem Befehl *cd* bedeutet »bring mich zum letzten Arbeitsverzeichnis zurück«. Sind Sie zum Beispiel vom Verzeichnis *~/Movies/Movie 1* nach *~/Documents/Letters* gewechselt, geben Sie einfach *cd -* ein, wenn Sie wieder nach *~/Movies/Movie 1* zurückgelangen möchten. Über ein erneutes *cd -* kehren Sie wieder nach *~/Documents/Letters* zurück. (Leerschritt zwischen *cd* und dem Bindestrich nicht vergessen.)

Tip: Schalten Sie andererseits viel zwischen zwei Verzeichnissen hin und her, so sollten Sie wahrscheinlich lieber zwei Terminal-Fenster mit verschiedenen Arbeitsverzeichnissen öffnen und dann einfach zwischen diesen beiden Fenstern wechseln.

GEHEIME SCHLEICHWEGE

Leerzeichen verboten

Terminal betrachtet ein Leerzeichen nicht als Leerzeichen. Es hält ein Leerzeichen zum Beispiel dafür: »Ich habe gerade einen Befehl eingegeben, und jetzt folgt ein Argument«. Möchten Sie sehen, was in Ihrem Short Films-Verzeichnis enthalten ist, so brauchen Sie es also gar nicht erst mit `ls ~/Movies/Short Films` zu probieren. Sie erhalten nur die Fehlermeldung »No such file or directory« (sehen Sie das Leerzeichen im Namen des Short Films-Verzeichnisses?)

Auch `$`-Zeichen haben unter Unix eine spezielle Bedeutung. Versuchen Sie, eines mit einem Pfadnamen einzutippen (weil ein Verzeichnisname zum Beispiel das `$`-Zeichen enthält), dann gibt es Ärger.

Glücklicherweise können Sie dieses Problem umgehen, indem Sie ein anderes reserviertes Sonderzeichen verwenden: den Backslash (`\`). Er bedeutet: »Ignoriere die Sonderbedeutung des nächsten Zeichens – eines Leerzeichens, zum Beispiel. Es hat jetzt gerade keine besondere Unix-Bedeutung. Ich verwende das nächste Leerzeichen als, naja, als Leerzeichen (und wenn Sie in einem Meeting der Unix-Usergruppe sind, dann könnten Sie auch sagen »Ich möchte das Leerzeichen *escapen*«).

Richtig rufen Sie den Inhalt des Verzeichnisses Short Films also mit `ls ~/Movies/Short\ Films` auf (sehen Sie, wie der Backslash direkt vor dem Leerzeichen bedeutet »Dies ist nur ein Leerzeichen – schön weitermachen, Leute«?).

Wenn Sie natürlich eine Menge Text mit Leerzeichen eingeben müssen, wäre es ziemlich nervig, vor jedem einen Backslash einzufügen. Glücklicherweise können Sie, statt Backslash-Zeichen zu verwenden, den ganzen Kladderadatsch in einfach Anführungszeichen einschließen. Das heißt also, Sie tippen anstatt

```
cd /Users/chris/My Documents/Letters\ to\ finish\Letter\ to\ Craig.doc
```

...einfach nur das:

```
cd '/Users/chris/My Documents/Letters to finish/Letter to Craig.doc'
```

Es kann sogar noch komplizierter werden. Was ist zum Beispiel, wenn der Pfad ein einfaches Anführungszeichen *enthält*? (Antwort: *So eines* schützen Sie dann mit doppelten Anführungszeichen). Das lernst Du schon noch, kleiner Bär.

Das ~-Kürzel

Sie kennen das Tilde-Zeichen (~) schon als Kürzel für Ihr Privat-Verzeichnis. Aber Sie können es auch als Kürzel für das Privat-Verzeichnis von jemand anderem verwenden, wenn Sie einfach den Account-Namen desjenigen daran anhängen. Möchten Sie zum Beispiel in das Privat-Verzeichnis von Miho wechseln, lautet der Befehl `cd ~Miho`.

Spezialtasten

Die *tosh*-Shell bietet Dutzende von speziellen Navigationskombinationen. Viele von ihnen erkennen Sie aus den nützlichen, nicht dokumentierten Kürzeln wieder, die in Cocoa-Programmen funktionieren, aber unter Terminal gibt es noch mehr von ihnen:

Tastenkombination	Wirkung
Ctrl-U	Löscht die gesamte Kommandozeile, an der Sie gerade arbeiten
Ctrl-A	Bewegt die Einfügemarke an den Zeilenanfang

Ctrl-E	Bewegt die Einfügemarke an das Zeilenende
Ctrl-T	Vertauscht die letzten beiden Zeichen
Esc-F	Bewegt die Einfügemarke an den nächsten Wortanfang
Esc-B	Bewegt die Einfügemarke an den vorhergehenden Wortanfang
Esc-Delete	Löscht das vorhergehende Wort (dieses definiert sich als »alles, was mit einem Leerschritt, Schrägstrich oder den meisten anderen Satzzeichen endet, mit Ausnahme von Punkten und Sternchen«). Sie müssen die Esc-Taste drücken und loslassen und <i>danach</i> die Löschtaste drücken; wiederholen Sie dies für einzelne Wörter.
Esc-D	Löscht das Wort, das auf die Einfügemarke folgt
Esc-C	Schreibt das Wort nach der Einfügemarke groß
Esc-U	Schreibt das nächste Wort komplett in Großbuchstaben
Esc-L	Schreibt das nächste Wort komplett in Kleinbuchstaben

Mit Dateien und Verzeichnissen arbeiten

Auf den vorangegangenen Seiten haben Sie gesehen, wie Sie mit Hilfe von Unix-Befehlen durch Ihre Verzeichnisse navigieren können. Das Wandern durch Verzeichnisse ist jedoch nicht übermäßig produktiv. Glücklicherweise sehen Sie in diesem Abschnitt, wie Sie etwas mit den Dateien, die Sie sehen, *machen* können: Sie können Dateien und Verzeichnisse kopieren, verschieben, erzeugen und löschen.

Tip: Jetzt geht es ans Eingemachte, wo Sie theoretisch ein ganzes Verzeichnis voller wertvoller Arbeit mit einem falschen Tastendruck löschen können. Als Sicherheitsmaßnahme könnten Sie sich überlegen, die Administratorprivilegien für Ihren Account beim Durcharbeiten dieses Abschnitts abzuschalten, damit Sie wenigstens außerhalb Ihres Privat-Verzeichnisses keinen Unsinn anstellen können.

Hierzu müssen Sie Terminal verlassen. Öffnen Sie Benutzer in den Systemeinstellungen. Wählen Sie Ihren Account-Namen, klicken Sie auf BENUTZER BEARBEITEN, geben Sie Ihr Paßwort ein, und drücken Sie Enter. Schalten Sie die Option DER BENUTZER DARF DIESEN COMPUTER VERWALTEN aus. Klicken Sie auf OK (andererseits muß *ein* Account stets Administratorrechte besitzen, weshalb Sie vielleicht erst einen anderen Administrator-Account anlegen müssen, wie auf Seite 364 beschrieben).

cp (copy)

Mit dem Unix-Befehl *cp* können Sie eine Datei in einem Aufwasch kopieren und umbenennen (versuchen Sie *das* einmal im Finder!)

Der Grundbefehl lautet folgendermaßen: *cp pfad1 pfad2*, wobei die Platzhalter *pfad* jeweils die Originaldatei und ihre Kopie bezeichnen.

An dieselbe Stelle kopieren

Möchten Sie zum Beispiel eine Datei namens *Thesis.doc* kopieren, so tippen Sie *cp Thesis.doc Thesis2.doc* (zwischen den zwei Namen ist ein einfacher Leerschritt). Sie müssen die Kopie nicht unbedingt *Thesis2* nennen – Sie können einen beliebigen Namen wählen; aber nun haben Sie zwei identische Dateien im selben Verzeichnis. Denken Sie daran, einen Backslash vor jedem Leerzeichen einzufügen, wenn Sie die Kopie mit zwei oder mehr Worten benennen möchten (zum Beispiel *Thesis\Backup*).

Tip: Funktioniert dieser Befehl scheinbar nicht, denken Sie daran, daß Sie den *vollen* Namen von Dateien tippen müssen, die Sie verschieben – inklusive Suffix wie *.doc* oder *.gif*, die unter Mac OS X ja normalerweise verborgen sind. Mit dem Befehl *ls* können Sie vor dem Kopieren den richtigen, vollständigen Dateinamen herausfinden. Oder Sie verwenden einfach die Tabulatorergänzung, und Terminal tippt den ganzen Namen für Sie.

Kopieren und umbenennen

Möchten Sie dieselbe Datei zum Beispiel gleichzeitig in Ihren Documents-Ordner kopieren, dann müssen Sie im letzten Ausdruck einfach nur den Pfad angeben, und zwar so: `cp Reviews.doc ~/Documents/Reviews2.doc`.

Tip: Beachten Sie, daß *cp* identische Dateinamen ohne Warnung ersetzt. Über die Option *-i* (also `cp -i`) können Sie sich warnen lassen, bevor *cp* eine solche Datei ersetzt.

FREQUENTLY ASKED QUESTION

Schrägstrich und Doppelpunkt

OK, jetzt bin ich echt durcheinander. Sie sagen, daß Schrägstriche verschachtelte Verzeichnisse kennzeichnen. Aber in Mac OS 9 hab ich immer gelesen, daß die Mac-interne Ordnerschreibweise über Doppelpunkte (:) abgewickelt wird, und daß ich deshalb keine Doppelpunkte in meinen Symbolnamen verwenden darf. Was ist nun Sache?

Auf dem Schreibtisch verwendet der Mac statt des Schrägstrichs immer noch Doppelpunkte als Pfadtrennzeichen. Deshalb dürfen Sie im Finder Schrägstriche in den Dateien

verwenden, nicht aber Doppelpunkte.

In Terminal hingegen dürfen Sie Doppelpunkte verwenden, nicht aber Schrägstriche!

Hinter den Kulissen wandelt Mac OS X automatisch die eine Schreibweise in die andere um, je nach Bedarf. Eine Datei namens *Letter 6/21/2002* im Finder erscheint als *Letter 6:21:2002* in Terminal. Entsprechend erscheint ein Verzeichnis namens *Attr: Jon* in Terminal unter dem Namen *Attr/ Jon* im Finder. Seltsam – und lustig!

Kopieren ohne umbenennen

Möchten Sie etwas in ein anderes Verzeichnis kopieren, ohne den Namen zu verändern, so können Sie einfach einen Pfadnamen (ohne Dateinamen) als letzten Ausdruck verwenden. Möchten Sie also die Datei *Reviews.doc* zum Beispiel in Ihren Documents-Ordner kopieren, tippen Sie `cp Reviews.doc ~/Documents`.

Tip: Sie können das ».«-Verzeichniskürzel (für das aktuelle Arbeitsverzeichnis) zum Kopieren von Dateien aus einem anderen Verzeichnis in das Arbeitsverzeichnis verwenden, und zwar so: `% cp ~/Documents/Reviews.doc .` (man beachte das Leerzeichen und den Punkt hinter *Reviews.doc*).

Mehrere Dateien

Sie können sogar mehrere Dateien oder Verzeichnisse auf einmal kopieren. Geben Sie einfach dort, wo Sie normalerweise die Quelldatei angeben, die entsprechenden Pfadnamen jeweils durch Leerschritte getrennt an, wie in Abbildung 15-6 zu sehen.

Auch mit dem *-Joker können Sie mehrere Dateien auf einmal kopieren. Nehmen wir zum Beispiel an, Sie haben folgende Dateien in Ihrem Verzeichnis iMovie Projects: Tahoe 1.mov, Tahoe 2.mov, Tahoe 3.mov, Tahoe 4.mov, Script.doc und Tahoe Project File. Nehmen wir jetzt an, Sie möchten nur alle QuickTime-Filme in ein Verzeichnis namens FinishedMovies kopieren. Sie brauchen einfach nur `cp *mov ../FinishedMovies` einzugeben und Enter zu drücken; Mac OS X kopiert sogleich.

Abbildung 15-6:

Das erste Argument dieses Befehls enthält zwei verschiedene Dateien. Der letzte Ausdruck gibt immer an, wo Sie alle Dateien hinbewegen möchten (in diesem Fall in das Verzeichnis FinishedMovies).

```
cp Tahoe1.mov Tahoe2.mov ../FinishedMovies
```

Die Dateien, die kopiert werden sollen.

Der Zielort

Möchten Sie *alle* Dateien (nicht nur die Filme) in ein anderes Verzeichnis kopieren, so verwenden Sie hierfür nur das *-Zeichen, und zwar so: `cp * ../Finished Movies`.

GEHEIME SCHLEICHWEGE

Forks auf Kollisionskurs: Programme mit CpMac und MvMac kopieren

Könnten Sie bestimmte traditionelle Macintosh-Dateien mit der Software-Entsprechung eines Röntgenapparats inspizieren, dann würden Sie feststellen, daß jedes Symbol tatsächlich zwei Teile mit Computercode enthält, und zwar die *Daten-Fork* und die *Resource-Fork*. (Daten bezieht sich auf – naja – Daten halt oder den Programmcode, aus dem ein Programm besteht. Resources sind Programmkomponenten wie Symbole, Fenster, Symbolleisten und andere Anwendungsbauteile.)

Diese ganze Fork-Geschichte ist für Unix aber neu. Wenn Sie den Befehl `cp` (oder `mv`) verwenden, kopiert oder verschiebt Unix nur die Daten-Fork – die Fork, die es kennt. Der Effekt auf die kopierte oder verschobene Datei könnte vernachlässigbar oder aber fürchterlich sein, je nach Dateiart. Die meisten Nicht-Cocoa-Programme (AppleWorks, Word und so weiter) sind dann zum Beispiel außer Gefecht gesetzt und haben viele ihrer lebenswichtigen Bestandteile verloren. Viele bekommen auch leere Symbole verpaßt und verlieren ihre *Type-* und *Creator-Codes* (Seite 132).

Dieses Problem behob Apple mit den Kommandozeilenprogrammen *CpMac* und *MvMac*, die genau wie `cp` und `mv` funktionieren, nur daß sie alle Forks der entsprechenden Datei

kopieren oder verschieben.

Die Standardinstallation von Mac OS X enthält diese Befehle aber nicht; Sie finden sie nur, wenn Sie die Developer Tools installieren (siehe Seite 331).

Haben Sie dies erledigt, dann können Sie sowohl *CpMac* als auch *MvMac* direkt am %b-Prompt aufrufen. Da es sich hier um Unix-Zusatzprogramme handelt, müssen Sie leider stets den ganzen Pfad angeben. Ein solcher Befehl sieht deshalb oft so aus: `/Developer/Tools/CpMac -r MacWrite| Folder ~/Applications` (Sie können es so einrichten, daß das Eintippen des Pfadnamens wegfällt, aber das ist Stoff für ein eigenes Buch).

Sie können in allen Beispielen dieses Kapitels `cp` durch *CpMac* (und `mv` durch *MvMac*) ersetzen. Es gibt nur zwei kleine Unterschiede. Erstens verwendet *CpMac* statt eines großen *R* ein kleines *r* für das rekursive Kopieren (*MvMac* benötigt keine Optionen; es wird stets der gesamte Ordnerinhalt verschoben). Zweitens können Sie bei `cp` und `mv` angeben, daß Sie vor dem Überschreiben von gleichnamigen Dateien gewarnt werden möchten, bei *CpMac* und *MvMac* gibt es keine solche Warnfunktion.

Enthielt das Verzeichnis iMovie Projects weitere *Verzeichnisse* und nicht nur Dateien, so würde dieser Befehl leider eine Fehlermeldung erzeugen; der Unix-Befehl `cp` kopiert keine Verzeichnisse innerhalb von Verzeichnissen, wenn Sie dies

nicht ausdrücklich angegeben haben, und zwar mit der Option `-R`. Hier ist der fertige Befehl, der alles im aktuellen Verzeichnis – sowohl Dateien als auch Verzeichnisse – in das Verzeichnis `FinishedMovies` kopiert: `cp -R * ../FinishedMovies`.

Und noch ein Beispiel, das alle Objekte (Dateien und Verzeichnisse), die im Namen `Tahoe` aufweisen, in das Drop Box-Verzeichnis (Briefkasten) eines anderen Benutzers kopiert: `cp -R *Tahoe* ~/miho/Public/Drop\ Box`.

***mv*: Dateien und Verzeichnisse verschieben und umbenennen**

Da Sie jetzt schon wissen, wie man Dateien kopiert, möchten Sie vielleicht auch noch erfahren, wie man sie verschiebt oder umbenennt. Dies geschieht mit dem Unix-Befehl `mv` – die Verwendung ist fast genau die gleiche wie bei `cp` (mit der Ausnahme, daß Verzeichnisse innerhalb von Verzeichnissen stets mitverschoben werden und Sie hier nicht `-R` anzugeben brauchen).

Die Syntax sieht so aus: `mv altername neuername`. Möchten Sie zum Beispiel den Namen Ihres Movies-Verzeichnisses in `Filme` umbenennen, tippen Sie `mv Movies Filme`. Sie können sowohl Dateien als auch Verzeichnisse auf diese Weise umbenennen.

Dateien und Verzeichnisse verschieben

Möchten Sie eine Datei umbenennen und gleichzeitig in ein anderes Verzeichnis verschieben, dann brauchen Sie einfach nur den letzten Befehlsteil durch einen Pfadnamen zu ersetzen. Wollen Sie die Datei `Tahoe1` in Ihr Documents-Verzeichnis verschieben – und sie gleichzeitig in `LakeTahoe` umbenennen –, tippen Sie folgendes: `mv Tahoe1.mov ~/Documents/LakeTahoe.mov`.

Alle üblichen Kürzel funktionieren, auch der Joker. So könnten Sie zum Beispiel alles in Ihrem Arbeitsverzeichnis, was das Wort `Tahoe` enthält (Dateien und Ordner), in Ihr Documents-Verzeichnis verschieben: `mv *Tahoe* ~/Documents`.

Optionen

Dem Befehl `mv` können Sie eine der beiden folgenden Optionen hinzufügen:

- **-i**. Nun fragt Terminal um Erlaubnis, bevor es eine Datei durch eine gleichnamige Datei ersetzt.
- **-f**. Überschreibt gleichnamige Dateien, ohne zuerst um Erlaubnis zu fragen (so funktioniert `mv`, wenn Sie nichts anderes angeben).

Hinweis: Dieser Befehl ersetzt nie ein Verzeichnis durch ein gleichnamiges anderes (oder durch eine Datei gleichen Namens). Ist ein solches Objekt bei mehreren Quellobjekten dabei, dann werden nur die verschiedenartigen Objekte aus allen Unterverzeichnissen im Zielverzeichnis ersetzt.

***mkdir*: Neue Verzeichnisse erstellen**

Im Finder erzeugen Sie einen neuen Ordner über `ABLAGE` → `NEUER ORDNER`. In Terminal verwenden Sie hierzu den Befehl `mkdir` (für *make directory*, Verzeichnis erstellen).

Dem Befehl selbst folgt der Name, den Sie dem neuen Verzeichnis geben möchten, zum Beispiel: `mkdir 'Early iMovie Attempts'` (durch die einfachen Anführungszeichen in diesem Beispiel können Sie es sich sparen, vor jedem Leerschritt einen Backslash einzugeben).

Mit dem Befehl `mkdir` erzeugen Sie das neue Verzeichnis im aktuellen Arbeitsverzeichnis, genauso einfach geht dies aber auch woanders. Fügen Sie einfach den Pfadnamen an Ihr Argument an. Möchten Sie zum Beispiel ein neues Verzeichnis im Verzeichnis Documents→Finished anlegen, so geben Sie `mkdir '~/Documents/Finished/Early iMovie Attempts'` ein.

Tip: Gibt es kein Verzeichnis namens Finished in Ihrem Documents-Verzeichnis, dann erhalten Sie einfach eine Fehlermeldung – es sei denn, Sie verwenden die Option `-p`, die alle Verzeichnisse erstellt, die zum Ausführen Ihres Befehls erforderlich sind. Der Befehl `mkdir -p '~/Documents/Finished/Early iMovie Attempts'` erzeugt also sowohl das Verzeichnis Finished als auch das darin enthaltene Verzeichnis Early iMovie Attempts.

touch: Leerdateien erzeugen

Um eine neue, leere Datei zu erzeugen, tippen Sie `touch dateiname`. Die Datei `practice.txt` erzeugen Sie in Ihrem Arbeitsverzeichnis zum Beispiel durch `touch practice.txt`.

Und warum das? Im Moment verwenden Sie solche neuen Leerdateien wahrscheinlich hauptsächlich als Ziel beim Ausprobieren des nächsten Befehls.

rm: Dateien und Verzeichnisse entfernen

Unter Unix kann man Dateien und Verzeichnisse auf extrem effiziente Weise entsorgen. Mit einem einzigen Befehl, `rm`, können Sie beliebige Dateien und Verzeichnisse löschen – oder *alles*, worauf Sie durch Ihre Account-Rechte Zugriff haben.

Die Gefahr an dieser Sache sollte klar sein, besonders angesichts der Tatsache, daß *Löschungen* unter Unix *unmittelbar* sind. Es gibt kein Zurück, keinen PAPIERKORB ENTLEEREN-Befehl, keinen »Sind Sie sicher?«-Dialog. Unter Unix ist alles endgültig.

Der Befehl `rm` steht für »remove« (entfernen), könnte aber auch genauso gut für »respect me« stehen. Machen Sie eine kleine Pause, bevor Sie diesen Befehl aufrufen. Überprüfen Sie vor dieser Einführung in `rm` noch einmal, daß die Administratorrechte für Ihren Account auch wirklich ausgeschaltet sind.

Sie geben für diesen Befehl einfach `rm`, ein Leerzeichen und danach den genauen Namen der Datei ein, die Sie aus dem Arbeitsverzeichnis löschen möchten. Möchten Sie zum Beispiel die Datei `practice.txt` löschen, die Sie mit `touch` erzeugt haben, tippen Sie `rm practice.txt`.

Zum Löschen eines Verzeichnisses mitsamt allen Inhalten fügen Sie die Option `-r` an, und zwar so: `rm -r Beispielverzeichnis`.

Wenn Sie sich gerade besonders stark fühlen (und Ihre Risikobereitschaft groß ist), dann können Sie im `rm`-Befehl sogar Jokerzeichen verwenden. Aber ehrlich gesagt, verwenden viele erfahrene Unix-Benutzer den `rm`-Befehl *kategorisch* nicht mit Jokerzeichen, solange sie als Administrator angemeldet sind, da ein falscher Tastendruck den gesamten Inhalt eines Verzeichnisses ausradieren kann. Hier aber kommt – nur zu Anschauungszwecken – die Atombombe aller Befehlszeilen, die Zeile, die *alles* löscht, was sich im Arbeitsverzeichnis befindet: `rm -r *`.

Tip: Seien Sie doppelt wachsam, wenn Sie Jokerzeichen zusammen mit `rm` in Kommandozeilen verwenden, und dreifach wachsam, wenn Sie dabei als Administrator angemeldet sind.

Direkt nach den Buchstaben `rm` können Sie folgende Optionen einfügen:

- **-d** (für *delete*) löscht außer Dateien auch alle Leerverzeichnisse, die gefunden werden (ansonsten erzeugen Leerverzeichnisse eine Fehlermeldung).
- **-i** (für *interaktiv*) läßt den Mac nachfragen, bevor eine Datei oder ein Verzeichnis in die ewigen Jagdgründe geschickt werden.

Tip: Verwenden Sie sowohl `-f` (siehe Seite 514) als auch `-i`, was sich ja widerspricht, so hat der letzte Befehl (der sich am weitesten rechts befindet) Vorrang.

- **-P** löst einen »sicheren« Löschvorgang aus, bei dem der Speicherort der Datei auf der Festplatte drei Mal gesäubert wird, wodurch nicht einmal Norton Utilities oder das Bundeskriminalamt mehr herausfinden können, was sich dort befand.

echo: Das Sicherheitsnetz

Sie können `rm` weniger riskant gestalten, wenn Sie den Befehl *echo* vorausschicken. Dadurch tippt Terminal den Befehl ein zweites Mal, und diesmal mit einer praktischen Liste aller Objekte, die Sie gleich löschen werden. Haben Sie Joker verwendet, sehen Sie die Namen aller Dateien, die vom `*`-Zeichen betroffen sind. Wenn Sie zum Beispiel `echo rm -r *` eingeben (was ohne den *echo*-Befehl bedeuten würde »lösche alles in diesem Verzeichnis«), dann sehen Sie eine solche Liste:

```
rm -r Reviews.doc Tahoe Footage Picnic Movie Contract.doc
```

Haben Sie sich die Liste angesehen und sind einverstanden damit, was Terminal gleich durchführen wird, *dann* können Sie den Befehl erneut ohne den *echo*-Teil erteilen.

Hinweis: `rm` funktioniert nicht mit Datei- oder Verzeichnisnamen, die mit einem Bindestrich (`-`) beginnen. Diese Objekte können Sie aus Ihrem Arbeitsverzeichnis entfernen, indem Sie ihnen einen Punkt und einen Schrägstrich (`./`) voranstellen, zum Beispiel: `rm ./-Recipes.doc`.

Online-Hilfe

Mac OS X enthält fast 900 Unix-Programme wie die in diesem Kapitel beschriebenen. Wie soll man um Himmels willen wissen, was sie jeweils bewirken? Glücklicherweise bietet fast jedes Unix-Programm eine eigene kleine Hilfedatei. Diese erscheint vielleicht nicht in einem eleganten Aqua-Fenster – genaugenommen ist sie sogar ausgesprochen schlicht gehalten – aber sie bietet viel mehr Informationen als das normale Mac-Hilfe-Center.

Diese Hilfedateien nennen sich Benutzerhandbuchseiten (user manual pages), oder *Manpages*, und bieten Beschreibungen fast aller verfügbaren Befehle und Programme. Mac OS X selbst ist mit Handbuchseiten zu über 4000 Themen ausgestattet – ungefähr 9000 Druckseiten voll. Leider bieten die Manpages sehr selten den klaren Schreibstil und die lernbezogene Ausrichtung des Hilfe Centers. Es handelt sich normalerweise um eher trockene, rein faktenorientierte Beschreibungen. Sie werden feststellen, daß Sie bestimmte Abschnitte öfter durchlesen müssen. Die enthaltenen Informationen sind jedoch sowohl für Neulinge als auch für erfahrene Benutzer ungemein wertvoll, und normalerweise ist der Aufwand, sie zu durchforsten, absolut gerechtfertigt.

man verwenden

Sie erreichen die Manpage für einen bestimmten Befehl, indem Sie *man*, gefolgt von dem gewünschten Befehlsnamen, eingeben. Die Manpage für *ls* erreichen Sie zum Beispiel so: *man ls*.

Tip: Die Option *-k* ermöglicht die Suche über Schlüsselbegriffe. Geben Sie zum Beispiel *man -k appletalk* ein, erscheint eine Liste aller Manpages, die mit AppleTalk zu tun haben, und Sie können aus der Liste einen Namen auswählen und diesen Seitennamen über *man* erreichen.

Nun erscheint das Handbuch seitenweise auf dem Bildschirm, wie in Abbildung 15-7 zu sehen.

Eine normale Manpage beginnt mit folgenden Abschnitten:

- **Name.** Der Name und eine Kurzbeschreibung des Befehls.
- **Synopsis.** Liefert die Befehlssyntax mit allen Optionen und Argumenten in kompakter, umfassender Form. Die Synopsis für *du* (disk usage, Festplattenauslastung) lautet zum Beispiel so: *du [-H | -L | -P] [-a | -s] [-ckrx] [file ...]*.

Hier sehen Sie alle Optionen, die für den Befehl *du* verfügbar sind, und wie Sie sie verwenden können:

Eckige Klammern ([]) umgeben die *optionalen* Argumente. (Alle Argumente von *du* sind optional.)

Vertikale Balken nennen sich *Pipes* (|) und geben an, daß Sie (aus einer Gruppe von Einträgen, die mit Pipes getrennt sind) nur jeweils einen Eintrag verwenden können. Wählen Sie zum Beispiel Optionen für *du*, so können Sie *entweder* *-H*, *-L* oder *-P* verwenden – nicht zwei von ihnen oder alle drei auf einmal.

Das Wort *file* in der Synopsis bedeutet »hier Pfadnamen angeben«. Die Auslassungspunkte (...) danach zeigen an, daß Sie hier mehrere Pfadnamen angeben können.

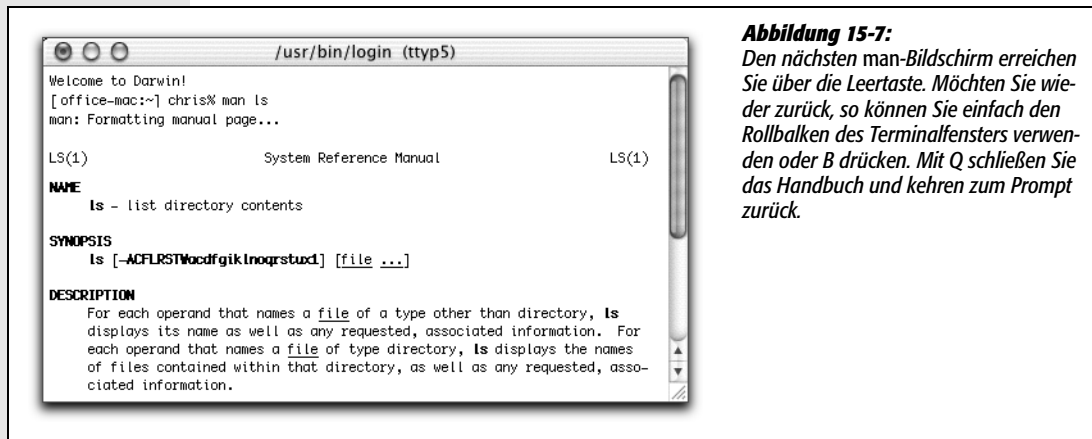


Abbildung 15-7:
Den nächsten man-Bildschirm erreichen Sie über die Leertaste. Möchten Sie wieder zurück, so können Sie einfach den Rollbalken des Terminalfensters verwenden oder B drücken. Mit Q schließen Sie das Handbuch und kehren zum Prompt zurück.

- **Description.** Beschreibt genau, was der Befehl bewirkt und wie er funktioniert. Die Beschreibung enthält auch eine komplette Liste aller Optionen des Befehls.

Näheres zur Verwendung von *man* finden Sie in den eigenen Manpages von *man* mit – was sonst? – *man man*.

Tip: Das kostenlose Programm ManOpen, das Sie von www.missingmanuals.com herunterladen können, ist ein Cocoa-Handbuchprogramm, mit dem Sie Ihre Manpages in einem hübschen, leicht bedienbaren Fenster lesen können.

Weitere Online-Hilfen

Manchmal setzt Ihnen Terminal ein kleines Stück Handbuch direkt vor die Nase – wenn es der Ansicht ist, daß Sie Schwierigkeiten haben. Verwenden Sie zum Beispiel den Befehl *mkdir* ohne einen Pfadnamen, dann unterbricht *mkdir* den Vorgang und stellt als freundliche Gedankenstütze seine eigene Synopsis dar (zwischen den Zeilen: »Äh – und so solltest du mich eigentlich *wirklich* verwenden«), und zwar so: *usage: mkdir [-p] [-m mode] dirname ...*

Terminal-Einstellungen

Wie die meisten Programme hat auch Terminal einen EINSTELLUNGEN-Befehl. Sie greifen jedoch auf die meisten Terminal-Einstellungen aus dem Befehl FENSTEREINSTELLUNGEN zu (direkt unter dem EINSTELLUNGEN-Befehl). Verbringen Sie wie die meisten Unix-Junkies Stunden über Stunden vor dem Terminal-Bildschirm, dann sind Sie wahrscheinlich irgendwann einmal froh über die Einstellungsmöglichkeiten, mit denen Sie bestimmen können, wie Terminal aussieht und reagiert.

Die meisten Optionen hier erklären sich von selbst, aber ein paar von ihnen sind beachtenswert.

Terminal → Einstellungen

Dieses Dialogfenster bietet ein paar ausgereifte Funktionen:

- **Beim Erstellen eines neuen Terminal-Fensters.** Wie auf Seite 501 beschrieben wurde, ist die Standard-Shell für Mac OS X *tcs*h, was im EINSTELLUNGEN-Dialog als */bin/tcs*h ausgewiesen ist. Wenn Sie als frischgebackener Unix-Schlaumeier bereit zum Umstieg auf eine andere Shell sind (und sich darüber im Klaren sind, daß die meisten der Beispiele in diesem Kapitel nur mit *tcs*h funktionieren), dann können Sie hier den entsprechenden Pfad angeben (vergessen Sie nicht, auch den unteren Radiobutton anzuklicken).
- **Beim Starten von Terminal eine gesicherte .term-Datei öffnen.** Möchten Sie eine *.term-Datei* verwenden (siehe Kasten auf weiter unten), dann geben Sie hier deren vollen Pfadnamen an.

SPRECHSTUNDE FÜR POWER USER

.term-Dateien

Falls Sie einmal ein paar Fenster offen haben – Ihr Haupt-Terminalfenster, ein paar *man*-Fenster (Handbuch), ein *top*-Fenster, in dem alle laufenden Programme zu sehen sind (Seite 541) und so weiter –, dann wird Ihnen vielleicht fast schwindlig beim Anblick all dieser schwarzen Buchstaben auf weißem Hintergrund. Eine einfache Lösung besteht darin, die Farben, Schriften und Positionen Ihrer Lieblingsfenster anzupassen und Terminal dann einen Schnappschuß dieser Konfiguration erstellen zu lassen.

In einer *.term-Datei* speichert Terminal die Größe, Position und den sichtbaren/unsichtbaren Status aller Fenster, die beim Erzeugen der *.term-Datei* offen sind. Die Datei enthält auch die Einstellungen jedes Fensters, zum Beispiel Schrift, Titelleiste und Farbeinstellungen (der Text in diesen Fenstern wird nicht hier gespeichert). Haben Sie einmal eine Startdatei gespeichert, dann können Sie diese jederzeit wieder öffnen und so Ihr gespeichertes Fensterlayout wieder in Terminal zum Vorschein bringen.

Sie erzeugen eine *.term-Datei*, indem Sie zunächst das Terminal-Fenster so einstellen, wie Sie möchten, und danach

ABLAGE → EINSTELLUNGEN ALS STANDARD VERWENDEN wählen.

Dann öffnen Sie Ihre Terminal-Fenster und arrangieren *diese* auf die gewünschte Weise, wobei Sie für verschiedene Fenster unter FENSTEREINSTELLUNGEN jeweils andere Einstellungen wählen können.

Als nächstes wählen Sie ABLAGE → SICHERN. Im Pop-Up-Menü des SICHERN-Dialogs können Sie wählen, ob die Einstellungen nur für das Hauptfenster (das aktive Fenster) oder für alle Fenster gelten sollen. Im gleichen Dialog können Sie auch einstellen, daß die gesicherte *.term-Datei* beim Aufrufen von Terminal geöffnet wird (in diesem Fall überschreiben die Einstellungen in der *.term-Datei* alle Einstellungen, die Sie unter FENSTEREINSTELLUNGEN vorgenommen haben).

Sichern Sie Ihre *.term-Dateien* am besten unter Library/Application Support/Terminal in Ihrem Privatverzeichnis (das Terminalverzeichnis müssen Sie eventuell selbst erstellen). Die dort gesicherten Dateien erscheinen für den Schnellzugriff automatisch im Menü ABLAGE → LIBRARY).

Fenstereinstellungen

Wählen Sie TERMINAL → FENSTEREINSTELLUNGEN, so erscheint das neue Fenster Terminal-Information (Abbildung 15-8).

Tip: Die Änderungen, die Sie unter Terminal-Information vornehmen, wirken sich nur auf das aktive Fenster aus, wodurch Sie jedes Fenster separat einstellen können. Klicken Sie hingegen auf ALS STANDARD VERWENDEN, wirken sich die Änderungen auf das aktive und alle weiteren Terminal-Fenster aus, die Sie öffnen.

Über das Pop-Up-Menü im oberen Fensterbereich haben Sie Zugriff auf folgende Einstellmöglichkeiten:

Shell

- **Beim Beenden der Shell.** Wenn Sie genug mit Terminal herumgefummelt haben, beenden Sie die Sitzung, indem Sie entweder das Fenster schließen, oder korrekterweise am Prompt *exit* eingeben (oder Control-D drücken). Diese Einstellung regelt, was passiert, wenn Sie dies tun.

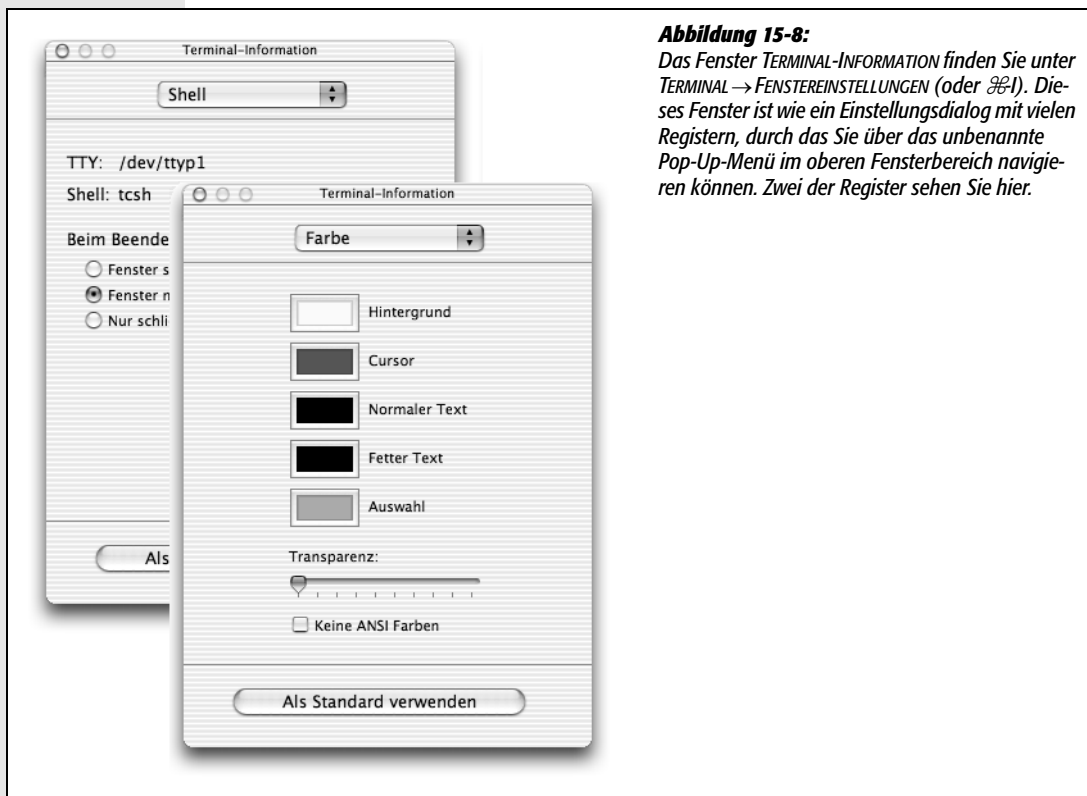


Abbildung 15-8:

Das Fenster `TERMINAL-INFORMATION` finden Sie unter `TERMINAL` → `FENSTEREINSTELLUNGEN` (oder `⌘I`). Dieses Fenster ist wie ein Einstellungsdialog mit vielen Registern, durch das Sie über das unbenannte Pop-Up-Menü im oberen Fensterbereich navigieren können. Zwei der Register sehen Sie hier.

Prozesse

Manchmal dauert es etwas, bis Shell-Befehle beendet sind. Wenn Sie ein Terminal-Fenster schließen, bevor es mit der Arbeit fertig ist, dann unterbrechen Sie den Vorgang und verlieren Ihre Arbeit. Über die Optionen hier können Sie Terminal so einstellen, daß aktive Prozesse überwacht werden und Sie eine Warnung erhalten,

wenn das Schließen eines Fensters oder das Beenden von Terminal zum Verlust von unbeendeter Arbeit führt.

Emulation

Diese Optionen wirken sich auf das von Terminal dargestellte traditionelle Hardware-Terminal aus.

- **Wahltaste+Klicken, um Cursor zu positionieren.** Hiermit können Sie Ihre Einfügemarke auf der Kommandozeile über Option-Klick bewegen.
- **Ausgabeoptionen.** Standardmäßig läßt Terminal es Sie über einen Piepton wissen, falls Sie etwas eintippen, was es nicht versteht. Nerven Sie Ihre Zimmergenossen mit Ihrem ständigen Gepiepe, können Sie hier VISUELLER EFFEKT wählen, wodurch das Terminal-Fenster statt dessen still aufblinkt.

Puffer

- **Zeilenpuffer.** Füllen Sie so nach und nach mit Ihren Kommandozeilenaktivitäten das Terminal-Fenster mit Text, dann verschwinden die älteren Zeilen oben aus dem Fenster. Damit Sie diese älteren Zeilen wieder zur Ansicht, zum Kopieren oder Drucken zum Vorschein bringen können, bietet Terminal Ihnen den *Zeilenpuffer*, der eine bestimmte Speichermenge bereitstellt – und einen Rollballen einfügt.

Tip: Allzu groß ist die Verführung, diesen Puffer auf UNBEGRENZT zu setzen, aber halten Sie sich zurück; Sie vergrößern damit nur den Speicherbedarf von Terminal. 10000 Zeilen sollten doch locker genügen.

Monitor

Hier können Sie einstellen, wie die Einfügemarke aussieht, und außerdem Einstellungen wie die folgenden vornehmen:

- **Text.** Welche Schriftart Sie auch für zukünftige Terminal-Sitzungen wählen, beachten Sie, daß Zeichen in aufeinanderfolgenden Textzeilen so aneinandergesetzt werden, als wäre eine Nichtproportionalschrift wie Monaco oder Courier angegeben.
- **Texte glätten.** Wählen Sie hier Textglättung aus, wenn Sie geglätteten Text besser lesen können.

Farbe (und Transparenz)

Wer viele Stunden mit Terminal verbringt, wird davon begeistert sein, daß Terminal wie ein Software-Chamäleon sein Aussehen verändern und so auch ausgefallene Geschmäcker (und müde Augen) zufriedenstellen kann. Sie können zum Beispiel die Schrift- und Hintergrundfarbe einfach über die Farbwahlfelder ändern und eine Farbe aus der Farbpalette wählen.

Tip: Sie können den Hintergrund des Terminal-Fensters in den Fenstereinstellungen ändern, aber es gibt eine unterhaltsamere Möglichkeit. Wählen Sie **SCHRIFT** → **FARBEN**; entscheiden Sie sich für eine Farbe. Ziehen Sie schließlich das Farbrechteck oben an eine leere Stelle im Terminal-Fenster. Und zack! Die Farbe wird angepaßt (Sie können das Farbmuster sogar auf den Text im Terminal-Fenster ziehen und so die Textfarbe ändern).

Sie können Ihre Terminal-Fenster auch halbtransparent gestalten, wodurch jeder Unix-Neuling garantiert vor Ehrfurcht erbebt. Ziehen Sie einfach den Schieberegler nach rechts, und sehen Sie dabei zu, wie der Hintergrund des aktiven Fensters langsam fast verschwindet und nur noch der Text übrigbleibt (in der 10.2-Version von Terminal brauchen Sie hierfür keinen Shareware-Hack und keine geheimen Unix-Befehle mehr).

Und wozu soll das gut sein? Da muß man seine Phantasie schon etwas spielen lassen. Trotzdem ist der Effekt wirklich hübsch, besonders, wenn man sich die Schlag Schatten und anderen Lichteffekte anschaut, die an den Fensterrändern noch zu sehen sind. (Beachten Sie, daß Sie Classic-Programme durch Ihr halbtransparentes Fenster hindurch nicht sehen können.)

Tip: Dieser Effekt sieht besonders cool aus, wenn Sie das Terminal-Fenster schwarz machen und die Schrift weiß oder gelb.

Fenster

Die Größenfelder wirken sich auf die Breite und Höhe neuer Terminal-Fenster in Zeichen (Spalten) und Zeilen aus. (Natürlich können Sie ein bestehendes Fenster jederzeit an seiner unteren rechten Ecke verändern. Beim Ziehen zeigt die Titelleiste die aktuellen Abmessungen an.)

Schalten Sie im Titel-Feld die Elemente ein, die Terminal jeweils in der Titelleiste des Fensters darstellen soll. Da Sie ja für jedes Terminal-Fenster andere Einstellungen vornehmen können, wäre es eventuell interessant, die Unterschiede in der jeweiligen Titelleiste zu verdeutlichen.

Tipps und Tricks für Terminal

Nachdem Sie Terminal jetzt eine Weile verwendet haben, sind Sie vielleicht bereit für ein paar Profi-Tipps.

Fenster wechseln

Sie können über **⌘-1**, **⌘-2** (bis zu **⌘-9**) und so weiter zwischen Ihren verschiedenen Terminal-Fenstern hin- und herschalten. Sie können die Fenster leicht voneinander unterscheiden, wenn Sie die Befehlstaste in den Titelleisten unterbringen (im Abschnitt **FENSTER** unter Terminal-Information).

.term-Standarddatei

Haben Sie die Einstellungen von Terminal angepaßt, dann können Sie die Standardeinstellungen nicht so ohne weiteres wiederherstellen – es sei denn, Sie haben sie vorher in einer eigenen .term-Datei gesichert.

Falls Sie bereits Änderungen an den Terminal-Einstellungen vorgenommen haben, dann verlassen Sie Terminal wieder und entfernen vorübergehend die *Eigenschaftsliste (property list)*, eine Datei namens *com.apple.Terminal.plist*, aus dem Ordner Privat→Library→Preferences. Ziehen Sie sie aus dem Fenster heraus – zum Beispiel auf den Schreibtisch.

Öffnen Sie danach Terminal erneut, und Sie werden sehen, daß alle Originaleinstellungen wieder vorhanden sind. Sichern Sie dieses Fenster als .term-Datei unter dem Namen Standard. Verlassen Sie schließlich Terminal, und ersetzen Sie die neue Eigenschaftslistendatei (*com.apple.Terminal.plist*) durch die, die Sie vorher entfernt hatten. Beim nächsten Öffnen von Terminal haben Sie wieder Ihre alten Einstellungen vor sich – und eine neue Standardoption in Ihrem LIBRARY-Menü, über die stets ein fabrikneues Terminal-Fenster geöffnet werden kann (vielleicht wäre es sogar gut, die Standard-Startdatei im INFORMATION EINBLENDEN-Dialog im Finder als »Geschützt« zu markieren).

