Kapitel 2

Einloggen, Filesystem und Prozesse

2.1 Benutzer

Unix bietet die Möglichkeit, dass (quasi) gleichzeitig mehrere Personen auf demselben Rechner arbeiten können. Darüberhinaus schützt es die Programme (Prozesse) der verschiedenen Benutzer voreinander. Als ein grundlegendes Konzept unter Unix gibt es daher den *Benutzer* (User). Ein Benutzer kann Prozesse starten, Files oder Directories erzeugen, die ihm zugeordnet sind. Intern werden Benutzer durch numerische User-IDs identifiziert. Um es dem (menschlichen) Benutzern einfacher zu machen, kann man einen Benutzer auch durch seinen Namen, den sog. *User-Namen* identifizieren. Dieser wird dann von den entsprechenden Programmen in die numerische ID übersetzt. Damit das System weiss, welche Benutzer gerade das System benutzen, muß man sich *anmelden* (einlogen). Dieser Vorgang wird im Detail im nächsten Abschnitt erläutert.

Ein weiteres wichtiges Konzept sind die *Gruppen*. Gruppen sind Ansammlungen von Usern und werden ebenfalls über einen Namen[1] und die damit assoziierte numerische Group-ID angesprochen. Jeder Benutzer ist in mindestens einer Gruppe, kann aber auch Mitglied mehrerer Gruppen sein. Unter Unix wird das Komzept der Gruppe dazu benutzt, Zugriffsrechte einfacher zu verteilen. Man kann z.B. eine Datei für alle Mitglieder einer Gruppe lesbar machen. User, die nicht Mitglied dieser Gruppe sind haben damit immer noch keinen Zugriff auf den Inhalt der Datei.

[1] In der Datei `/etc/groups`.

2.2 Einlogen

Das Unix System unterscheidet Benutzer nur anhand ihrer User-IDs. Damit sichergestellt ist, dass sich Benutzer auch wirklich nur mit ihrer eigenen User-ID im System aufhalten[2], verlangt das Unix System eine Authentifizierung des Benutzers, bevor es ihm erlaubt, Prozesse zu starten, etc. Für die Authentifizierung gibt es viele verschiedene Verfahren, deren Sicherheitsgrad der Sensibilität der Daten folgt (etwa Retinascanner, Fingerabdruckanalysatoren, Chipkartenleser, etc.).

2.2.1 Passwörter

Am häufigstem eingesetzt (jedenfalls im universitären Bereich) wird immer noch die Authentifizierung über ein *Passwort*. Das Passwort ist eine (möglichst lange) Sequenz von Zeichen, die nur dem Benutzer und dem System bekannt ist (oder sein sollte). Bei Einlogen fragt nun das System nach dem Passwort, der Benutzer gibt es ein und wenn das eingegebene Passwort und das Passwort, welches in der Systemdatenbank für diesen Benutzer hinterlegt ist, gleich sind, dann nimmt das System an, dass wirklich der richtige Benutzer an der Tastatur sitzt.

> In der Realität ist das System etwas komplizierter. Das System speichert aus Sicherheitsgründen nicht die Passwörter aller Benutzer im Klartext ab, da sonst durch (versehentliches) Lesen dieser Datenbank alle Passwörter komprimitiert wären.
>
> Um dies zu umgehen, wird eine *One-Way Funktion* auf das Passwort angewandt. Eine One-Way Funktion ist eine Funktion, die nur sehr schwer umkehrbar ist. Wenn man also das Funktionsergebnis hat, kann man von diesem nur sehr schwer[3] wieder das Argument bekommen. Die Theorie dahinter ist, dass die Funktion so schwierig ist, dass es am effizientesten ist, alle Passwörter der Reihe nach auszuprobieren, um das richtige herauszubekommen, siehe [Sch96] für die Details.
>
> Das System wendet also beim Einlogen diese One-Way Funktion auf das Passwort an, das der Benutzer eingibt. Stimmt das Ergebnis mit dem gespeicherten Wert in der Systemdatenbank überein, wird davon ausgegangen, dass das Passwort richtig ist und der Benutzer derjenige ist, dem die angegebene User-ID gehört.

Das Verfahren der Authentifizierung über Passwörter ist solange relativ sicher, wie die Benutzer ihr Passwort geheimhalten. Die Notiz des Passwortes auf einem Stück Papier, das man am Rechner liegenlässt ist deshalb nicht zu empfehlen.

[2] Damit man etwa sicherstellen kann, dass Mail nur von dem jeweiligen Account-Inhaber gelesen wird.

[3] D.h. mit enorm hohem Rechenaufwand.

Ein weiteres theoretisches Problem der Passwörter besteht darin, dass man sie im Prinzip durch Aufzählen und Probieren aller möglichen Passworter knacken kann. Auf Unix Systemen sind Passwörter meistens auf eine Länge von 8 Zeichen beschränkt. Benutzt man eine Mischung von Klein-/Großbuchstaben[4], Ziffern und Satzzeichen (also aus einer Menge von ca. 80 Zeichen), dann gibt es bei 8 Zeichen Passwortlänge $80^8 \approx 1.6 * 10^{15}$ Möglichkeiten. Könnte man in der Sekunde etwa eine Million Passwörter durchprobieren, bräuchte man für alle Kombinationen also etwa $1.6 * 10^9$ Sekunden, was etwa 19418 Tagen oder 53.2 Jahren entspricht[5]. Natürlich muss man im Schnitt nur die Hälfte dieser Kombinationen prüfen, bräuchte also etwa 26.6 Jahre. Damit ist es relativ unwahrscheinlich, Passwörter in vertretbarer Zeit durch Aufzählen herauszufinden.

Ein gewichtigeres Problem besteht aber darin, dass viele Benutzer Passwörter suchen, die sie sich leicht merken können, den Vornamen der Freundin etwa. Passwörter, die auf diese Art ausgedacht sind, bieten *keine Sicherheit*! Sie sind durch Ausprobieren relativ schnell erratbar[6]. Auch Namen von Städten, Filmen, etc. sind nicht geeignet. Eine relativ sichere Methode, ein gutes Passwort zu finden, ist folgende:

- Schlagen sie ein Buch zufällig irgendwo in der Mitte auf.
- Suchen sie sich aus dem Buchstaben irgendwelcher Wörter auf dieser Seite eine Sequenz aus.
- Arbeiten sie die Seitenzahl an einer (nicht fixen!) Stelle ein.
- Notieren sie sich dieses Passwort *an einem sicheren Ort*.

Beim Eingeben ihres Passwortes sollten sie darauf achten, dass ihnen niemand auf die Finger schaut und sich so das Passwort merkt[7]. Verwahren sie den Zettel mit ihrem Passwort gut. Nicht an einer Stelle, die sie täglich erreichen, etwa im Notizbuch[8].

Wenn andere Leute ihr Passwort eingeben, sollte man höflicherweise in eine andere Richtung schauen[9].

[4] Die Schreibweise in Passwörtern ist signifikant!

[5] Mein Pentium II 450 schafft pro Sekunde etwa 10000 Möglichkeiten, bräuchte also ca. 5320 Jahre, um alle Kombinationen auszuschöpfen.

[6] Wenn man ein bischen Information über die Person hat.

[7] Alles schon passiert.

[8] Man kann natürlich das Passwort auch versteckt in einem Adressbuch notieren, etwa indem man es über mehrere Einträge verteilt...

[9] Manche sagen, aus diesem Grund würden die Decken in Informatikerräumen häufiger geputzt, als anderswo.

Versuchen sie *niemals* das Passwort eines anderen herauszufinden. Solches Tun fordert nur eine unfreundliche Antwort ihres Systemadministrators heraus (siehe auch die Benutzungsbedingungen für die Rechner des CIP-Pools!).

2.2.2 Praxis

Nachdem wir nun das Prinzip der Authentifizierung über Passwörter und die Etikette besprochen haben, folgt nun noch ein Überblick über den tatsächlichen Vorgang des Einlogens.

Vorausgeschickt sei, dass man bei der Eingabe des Passwortes nicht verwundert sein sollte, dass es nicht auf dem Bildschirm dargestellt wird. Das ist absichtlich so gehandhabt, damit ein zufälliger Beobachter das Passwort nicht vom Bildschirm ablesen kann.

Auf den Rechnern des CIP-Pools kann man sich auf mehrere Arten einlogen, je nach Rechner:

1. Auf den X-Terminals (SPARC ELCs) logt man sich per *xdm* ein (s.u.)

2. Auf den SPARC 4 Rechnern `kapxx.studcs.uni-sb.de` und den FreeBSD PCs `xcipx.studcs.uni-sb.de` kann man sich über das Netz oder den xdm einlogen.

Die SPARC ELCs in den CIP-Räumen dienen nur als Terminals: Sie sind ausschliesslich für die graphische Ausgabe und Eingaben über Tastatur/Maus zuständig. Logt man sich an einem solchen Rechner ein, wird man automatisch an einen der FreeBSD PCs weiterverbunden, die in den Räumen stehen. Alle Befehle, Prozesse, die man startet laufen also auf den PCs, nur die Ausgabe wird auf den SPARC ELCs dargestellt; die Eingaben werden entsprechend an den PC weitergereicht.

xdm

Das Einlogen per xdm[10] sieht etwa wie in Abbildung 2.1 auf Seite 10 aus. Auf dem Bildschirm erscheint ein kleines Fenster, in welches man den Usernamen[11] und das Passwort eingibt.

Nach erfolgreicher Anmeldung wird automatisch die graphische Benutzeroberfläche gestartet. Was gestartet wird, steht in der Datei `.xsession` im Homedirectory. Diese kann man sich etwa mittels `cat .xinitrc` ausgeben lassen[12]

[10]Siehe auch die *xdm(1)* Manualseite.

[11]Nicht die numerische User-ID.

[12]Was ein Directory und eine Datei ist, wird allerdings erst in Abschnitt 2.4 auf Seite 11 erklärt.

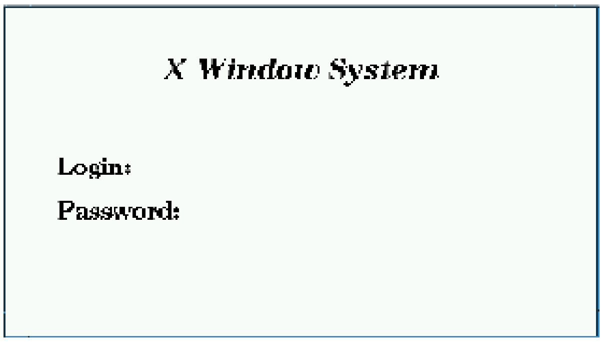

Abbildung 2.1: Einlogen per xdm

Per Default wird beim Starten der graphischen Oberfläche ein `xterm` Fenster gestartet, in diesem kann man Kommandos eingeben und Ausgaben erscheinen dort. Ein `xterm` ist ein Terminal (siehe Abschnitt 2.5 auf Seite 11), in welchem die Login-Shell des Benutzers läuft (Kapitel 3 auf Seite 12 enthält weiterführende Informationen zu den Shells).

Netzwerk

Man kann sich auch von anderen Rechnern über das Netzwerk (Internet) in anderen Rechnern einlogen. Das in Unix verwendete Konzept des *Terminals* macht es möglich, dies ohne sichtbare Unterschiede zu einem Login über die Tastatur zu bewerkstelligen.

Beim Einlogen über das Netzwerk[13], startet ein sog. *Daemon* auf dem Zielrechner, der die Netzwerkverbindung annimmt, den Usernamen und das Passwort anfordert und die Richtigkeit des letzteren verifiziert. Danach wird einfach die Netzwerkverbindung über ein sog. *Pseudoterminal* geleitet und als Ein-/Ausgabe an die Login-Shell des Benutzers verwendet. Die Eingaben werden also vom Quellrechner zum Zielrechner transportiert und dort als Eingaben der Shell interpretiert. Die Ausgaben auf dem Zielrechner nehmen den Weg in die entgegengesetzte Richtung.

Console

Die traditionelle Weise, sich in ein Unix System einzulogen besteht darin, sich über die *Console* anzumelden. Die Console ist ein spezielles Terminal, das früher der Tastatur und dem Bildschirm der Mainframes entsprach, während anderer Terminals über serielle Leitungen an diesen Hauptrechner angeschlossen wurden.

[13]Mittels `telnet` z.B., siehe Kapitel 6 auf Seite 15.

Auch heutige Systeme bieten diese Schnittstelle noch. Wenn man zu Hause einen Linux, FreeBSD oder NetBSD PC hat, dann kann man sich dort auch über einen Textbildschirm einlogen. Der einzige Unterschied zum Einlogen über das Netzwerk besteht nur darin, dass das Terminal direkt die Console ist.

Die Bedeutung der Console liegt auch darin, dass auf ihr wichtige Systemfehlermeldungen ausgegeben werden und sie das einzige Terminal ist, das beim Booten des Rechners von Anfang an zur Verfügung steht.

2.2.3 Nach dem Einlogen

Nachdem man sich im System eingelogt hat, kann man Kommandos in der Shell eingeben, Prozesse starten, etc.pp.

2.3 Dokumentation

2.4 Filesystem

2.5 Prozesse