

Die Benutzerverwaltung

In diesem Teil wird die Benutzerverwaltung von Unix beschrieben.

Inhalt

1.	Das User System von Unix.....	3
2.	Das Passwort System.....	5
2.1.1.	Die Gruppenzugehörigkeit	5
3.	Rechte von Benutzern	7
3.1.	Voreingestellter Zugriffsmodus.....	10
4.	Die Datei /etc/passwd	12
5.	Die Datei /etc/shadow.....	14
6.	Die Datei /etc/group	17
7.	Die Datei /etc/gshadow	19
8.	Das Anlegen von neuen Usern.....	21
8.1.1.	Das manuelle Anlegen von neuen Usern	21

9. Hörsaalübung24

1. Das User System von Unix

Unix hat als **Mehrbenutzersystem** bietet die Möglichkeit benutzerindividuell Rechte zu vergeben oder zu entziehen. Die Frage, nach welcher Philosophie dabei vorgegangen wird bleibt dem Systemverwalter überlassen, es gibt keine grundsätzlichen Vorgaben des Systems dafür. Es ist also möglich, allen alles zu erlauben, außer das, was sie wirklich nicht können sollen, oder umgekehrt allen alles zu verbieten, außer dem, was sie wirklich können müssen. Zwischen diesen beiden extremen Positionen sind stufenlos alle Zwischenschritte möglich.

Der grundsätzliche Mechanismus ist dabei der, dass Unix die einzelnen User unterscheidet, jeder User muss sich am Anfang seiner Sitzung mit Username anmelden (engl.: to log in) und mit Passwort ausweisen.

Die Rechte eines Users sind - zumindest was das Schreiben anbelangt - in der Regel auf seine eigenen Dateien beschränkt, also die Dateien, die er selbst angelegt hat. Jeder User hat ein so genanntes **Heimatverzeichnis** (engl. home-directory), das ihm gehört und innerhalb dessen er beliebige Dateien und Verzeichnisse anlegen darf. Meist trägt dieses Verzeichnis den Namen des Users und befindet sich im Verzeichnis /home.

Jeder User gehört **mindestens** einer **Gruppe** an. Gruppen sind nichts anderes als Verwaltungsmechanismen, die verschiedene User zusammenfassen. User können Mitglieder beliebig vieler Gruppen sein. Jede Datei im System ist mit einer Gruppenzugehörigkeit ausgestattet, für deren Mitglieder spezielle Zugriffsbestimmungen herrschen.

Der Username und der Name der Gruppe sind intern jeweils Nummern, die so genannte UserID (**UID**) und GroupID (**GID**). Der User mit der UID 0 hat grundsätzlich eine Sonderbedeutung, es handelt sich um den Systemverwalter, meist mit Usernamen **root**. Dieser User steht außerhalb

jedes Sicherheitsmechanismus. Er **darf alles**, nichts schränkt ihn ein, allerdings kann er auch damit Fehler machen und das System mit einem Befehl komplett zerstören.

2. Das Passwort System

Ein User hat in der Regel ein Passwort, mit dem er sich beim System anmeldet und damit beweist, dass er derjenige ist, der er vorgibt zu sein. Dieses Passwort ist also der Schlüssel zum System, nur mit einer passenden Kombination aus Username und dazugehörigem Passwort ist die Arbeit an einem Unix-System möglich.

Das Passwort wird grundsätzlich verschlüsselt gespeichert, also hat auch der Superuser keinen Zugriff auf das klar lesbare Passwort. Er kann zwar die Passwörter von Normalusern löschen oder verändern, er kann aber nicht ein vergessenes Passwort wieder herstellen.

Jeder User kann sein Passwort mit dem Kommando **passwd** verändern, der Superuser kann mit diesem Befehl auch die Passwörter anderer User verändern.

Ein User kann sich - falls er das entsprechende Passwort weiß - auch in einen anderen User verwandeln, ohne sich abzumelden und neu anzumelden. Der Befehl dazu ist **su** (substitute user). Unix unterscheidet dann zwischen zwei verschiedenen User-IDs, der echten **UID**, das ist die UID des ursprünglichen Users, die er beim Login erhalten hat, und der effektiven UserID (**EUID**). Das ist ein notwendiger Mechanismus, damit bestimmte Programme immer wissen, um welchen echten User es sich handelt, auch wenn er seine EUID gerade verändert hat.

2.1.1. Die Gruppenzugehörigkeit

Jeder User gehört mindestens einer Gruppe an, er kann aber auch in beliebig vielen Gruppen Mitglied sein. Eine Gruppenmitgliedschaft bringt zusätzliche Rechte auf bestimmte Dateien. Es existiert immer eine voreingestellte Gruppe, das ist die, die in der `/etc/passwd`-Datei angegeben wurde.

Die voreingestellte Gruppe ist beim Login gleichzeitig auch die aktuelle Gruppe, also die Gruppe, der Dateien zugeordnet werden, die der User neu anlegt. Der User muss NICHT als Mitglied seiner voreingestellten Gruppe in der Datei `/etc/group` aufgeführt sein, er erhält seine Mitgliedschaft durch den Eintrag in `/etc/passwd`.

Ein User muss für jede weitere Gruppe, deren Mitglied er sein soll einen Eintrag in der Datei **`/etc/group`** besitzen. Diesen Eintrag kann sowohl der Systemverwalter, als auch der Gruppenadministrator ausführen.

Mit dem Befehl **`newgrp`** kann ein User seine aktuelle Gruppe wechseln, das heißt, dass alle neuen Dateien, die er dann anlegt, dieser Gruppe zugeordnet werden. In modernen Unixen kann ein User mit diesem Kommando auch Mitglied einer Gruppe werden, ohne wirklich als Mitglied der Gruppe eingetragen zu sein. Dazu muss er aber dann das Gruppenpasswort kennen.

In diesen modernen Unixen hat jede Gruppe auch einen **Gruppenadministrator**, der selbstständig die Verwaltung einer Gruppe vornehmen kann, ohne Superuser sein zu müssen. Insbesondere das Aufnehmen von neuen Usern in die Gruppe, das Streichen von Mitgliedern oder das Ändern des Gruppenpassworts gehört zu diesen Verwaltungsaufgaben. Das Programm, mit dem ein Gruppenadministrator seine Aufgaben erledigen kann heißt **`gpasswd`**.

3. Rechte von Benutzern

Jede Datei hat neben den normalen Angaben die Attribute Eigentümer, Gruppenmitgliedschaft und Zugriffsmodus.

Eigentümer

Jede Datei im System (wirklich jede, auch Verzeichnisse, Gerätedateien usw.) hat einen Eigentümer. Dabei handelt es sich bei Systemdateien immer um den Systemverwalter, normale Dateien gehören dem User, der die Datei angelegt hat. Der Eigentümer ist über seine UserID (UID) angegeben, d.h. jede Datei hat eine 16 Bit Zahl, die den Eigentümer bezeichnet.

Gruppenmitgliedschaft

Jede Datei gehört genau einer Gruppe an. In der Regel ist das die Standardgruppe des Users, der die Datei angelegt hat, das ist aber nicht zwingend. Es ist nicht einmal vorgeschrieben, dass der Eigentümer einer Datei Gruppenmitglied der Gruppe ist, der die Datei angehört. Auch die Gruppenzugehörigkeit wird über eine 16 Bit Zahl, die GroupID (GID) dargestellt.

Zugriffsmodus

Jede Datei hat eine 16 Bit Zahl, die den Zugriffsmodus bestimmt, wie auf sie zugegriffen werden kann und welchen Typ sie hat. Diese Zahl ist in 5 Oktalziffern aufgeteilt, die den Typ und die einzelnen Rechte darstellen.

Jede Datei hat ein rwx-triple, für den Eigentümer, für ein Gruppenmitglied der Gruppe, der die Datei angehört und für den Rest der Welt. Dabei steht r für Lesen und hat den Wert 4, w für Schreiben mit dem Wert 2 und schließlich x für Ausführen (execute) mit dem Wert 1.

Es existieren drei wesentliche Befehle zum Manipulieren dieser Dateieigenschaften.

- chown zum Wechseln des Eigentümers einer Datei
- chgrp zum Wechseln der Gruppenzugehörigkeit einer Datei
- chmod zum Wechseln des Zugriffsmodus einer Datei

Neben den drei üblichen Zugriffsrechten für User, Group und Rest gibt es noch ein viertes, das auch ausgesprochen wichtig für die Systemverwaltung ist. Die normale Darstellung eines Zugriffsrechts geschieht ja numerisch mit den Oktalziffern. Normalerweise werden drei Oktalziffern verwendet, es gibt aber noch eine vierte, führende Ziffer. Eigentlich müsste der Zugriffsmodus 640 eigentlich also 0640 heißen.

Diese führende Ziffer hat nur Bedeutung für Programme bzw. Unterverzeichnisse. Die einzelnen Werte bedeuten:

Ziffer	Bedeutung
4	Das Substitute UserID-Bit ist gesetzt
2	Das Substitute GroupID-Bit ist gesetzt

Im Einzelnen haben diese drei Bits folgende Bedeutung:

Substitute UID (SUID)

Ein Programm, das dieses Bit gesetzt hat wird nicht unter der UID des Users ausgeführt, der das Programm aufgerufen hat sondern unter der UID des Eigentümers des Programms. Ein kurzes Beispiel mag das verdeutlichen.

Das Programm passwd erlaubt es jedem User sein Passwort zu verändern. Das ist vernünftig, sonst müßte jedesmal der Systemverwalter diese Aufgabe erledigen und der sollte die Passwörter ja auch nicht kennen. Die Passwörter werden aber verschlüsselt in der Datei /etc/shadow gespeichert, auf die ein Normaluser weder lesenden noch gar schreibenden Zugriff hat. Damit ein Normaluser trotzdem sein Passwort ändern kann, hat das Programm passwd als Eigentümer root und es hat das Substitute UserID Bit gesetzt. Das Programm wird also nicht unter der UID des Users ausgeführt, der es aufruft sondern unter der UID des Users, dem das Programm gehört - hier also root. Damit kann das Programm schreibend auf die Datei /etc/shadow zugreifen.

In diesem Fall kann kein Unfug mit dieser Fähigkeit getrieben werden, weil das Programm passwd ja sowieso nur das Passwort des Users ändern kann, der es aufruft. Schlimm wird es, wenn ein Programm mit Ausgang zur Shell mit dem SUID-Bit läuft. Dann hätte jeder User root-Rechte, der das Programm ausführt - sofern das Programm root gehört.

Substitute GroupID (SGID)

Dieses Bit erfüllt die exakt gleiche Aufgabe wie das SUID-Bit, nur daß es sich nicht um die UserID sondern um die GroupID handelt. Ein User, der ein Programm ausführt, dessen SGID-Bit gesetzt ist, führt dieses Programm nicht unter seiner GID aus, sondern unter der GID des Programmes.

Sticky-Bit

Früher, auf alten Großrechneranlagen gab es das Problem, dass ein Programm nur sehr langsam in den Arbeitsspeicher geladen wurde (Magnetbänder). Damit ein häufig benutztes Programm nicht jedesmal wieder geladen werden musste, konnte man es **sticky** (klebrig) machen, es blieb also nach der Ausführung im Speicher erhalten.

Heute hat das Sticky-Bit eine andere Aufgabe, die ausschließlich Verzeichnisse betrifft. In einem Verzeichnis, dessen Sticky-Bit gesetzt ist, kann ein Normaluser - auch wenn er Schreibrechte in diesem Verzeichnis besitzt - Dateien nicht löschen.

3.1. Voreingestellter Zugriffsmodus

Um festzulegen, welche Zugriffsberechtigungen beim Anlegen einer Datei verwendet werden kennt Unix das Kommando **umask**, dessen Anwendung aber gewöhnungsbedürftig ist.

Der Befehl umask erwartet eine Maske als Parameter, die sich auf den Zugriffsmodus bezieht, den neu zu erstellende Dateien bekommen sollen. Das Wort Maske bedeutet, dass nicht die Werte eingegeben werden, die gesetzt werden sollen, sondern umgekehrt, die Werte, die nicht gesetzt (maskiert) werden sollen. Die einfachste Möglichkeit besteht darin, die gewünschten oktalen Werte jeweils von 7 abzuziehen:

Will man z.B. dafür sorgen, dass alle unsere Dateien die Zugriffsberechtigung

`rw-r-----`

bekommen, also Lese- und Schreibrecht für den Eigentümer, Leserecht für Gruppenmitglieder und keine Rechte für den Rest der Welt, dann entspräche das der oktalen Darstellung 640.

Die dafür notwendige umask wäre dann:

$$7 - 6 = \underline{1}$$

$$7 - 4 = \underline{3}$$

$$7 - 0 = \underline{7}$$

Mit dem Befehl

```
$ umask 137
$
```

würden also alle Dateien, die wir anlegen die gewünschte Zugriffsberechtigung 640 bekommen. Das hat noch einen kleinen Haken, weil die Verzeichnisse, die wir erstellen würden auch diesen Modus bekämen. Damit wäre für uns selbst das Durchsuchungsrecht (x) nicht gesetzt. Unix hat aus diesem Grund dafür den Mechanismus entwickelt, dass selbst wenn im umask-Kommando das x-Recht gesetzt ist, beim Erzeugen von normalen Dateien dieses Recht nicht gesetzt wird, beim Erzeugen von Verzeichnissen hingegen schon. Ein vernünftiges umask-Kommando setzt also zumindestens für den Eigentümer auch das x-Recht. Damit wäre ein typischer Wert für umask z.B. **022** (rwxr-xr-x) oder **027** (rwxr-x---).

4. Die Datei /etc/passwd

Diese Datei ist die zentrale Userdatenbank aller Unix-Systeme. Der Name rührt daher, dass früher hier auch die verschlüsselten Passwörter beinhaltet waren, das hat sich heute geändert. Diese Datei muss für alle Welt lesbar sein, denn jedesmal, wenn ein Unix-System die Zuordnung zwischen UID und Username vornehmen soll (etwa beim Anzeigen des Inhaltsverzeichnisses) muss sie eingelesen werden.

Die Datei /etc/passwd ist, wie fast alle Unix-Konfigurationsdateien, eine einfache Textdatei. Jede Zeile repräsentiert einen Usereintrag und besteht aus verschiedenen Feldern, die jeweils durch einen Doppelpunkt getrennt sind.

Die Bedeutung der einzelnen Felder ist einfach, im Folgenden werden sie genauer erklärt:

`Username:Passwort:UserID:GroupID:Beschreibung:Homeverzeichnis:Startshell`

Username

Der Username, mit dem sich der User einloggen kann.

Passwort

Hier stand früher das verschlüsselte Passwort. In heutigen Unix-Systemen steht hier jeweils nur ein x, das signalisiert, dass das Passwort in einer anderen Datei (/etc/shadow) gespeichert ist.

UserID

Die numerische UserID, eine 16 Bit Ganzzahl, die den User eindeutig identifiziert. Der Systemverwalter root hat hier immer die Null, alle anderen sind beliebig wählbar.

GroupID

Die Identifikation der voreingestellten Gruppe, in der der User Mitglied ist. Jeder User muß mindestens in einer Gruppe Mitglied sein, die GroupID, die hier angegeben ist, ist die Gruppe, der eine Datei gehört, die der User neu anlegt.

Beschreibung

Eine genauere Beschreibung des Users, die aus mehreren Worten bestehen darf. Meist enthält sie den echten Vor- und Zunamen. Beliebig wählbar.

Home-Verzeichnis

Dieses Feld enthält die Angabe, welches Verzeichnis das Heimatverzeichnis des Users ist. Das ist das Verzeichnis, in dem jede Sitzung beginnt.

Startshell

Das Programm, das als Kommandointerpreter geladen werden soll, wenn sich der User eingeloggt hat. Meist ist es eine so genannte Shell, es kann aber auch jedes beliebige Anwendungsprogramm sein.

5. Die Datei `/etc/shadow`

Früher stand das verschlüsselte Passwort in der Datei `/etc/passwd`, die für alle User lesbar sein muss. Das warf ein Sicherheitsproblem auf, denn jeder User konnte jetzt alle Passwörter einsehen, zwar nur in verschlüsselter Form, aber immerhin. Manche User versuchten jetzt etwa alle Wörter eines Wörterbuchs mit dem Algorithmus zu verschlüsseln, mit dem auch das Passwort verschlüsselt wird und dann die Ergebnisse mit denen dieser Datei zu vergleichen. Damit konnten Passwörter erspäht werden.

Alle User benötigen lesenden Zugriff auf die Datei `/etc/passwd`. Die Abspeicherung des Passworts selbst muss nicht zwangsweise in dieser Datei stattfinden. Aus diesem Grund begann man schon früh, eine eigene Datei anzulegen, die die Passwörter der User enthält und die nicht für alle lesbar ist. Diese Datei wurde **`/etc/shadow`** (Schatten) genannt. Sie ist zwar vom Login-Prozess lesbar, aber nicht von allen Usern. Wie die `/etc/passwd` Datei ist sie eine einfache Textdatei in der jeder User eine Zeile belegt. Auch hier sind die einzelnen Felder der Datensätze mit Doppelpunkt getrennt.

```
Username:Passwort:Alter:min.Alter:max.Alter:Warnzeit:Pufferzeit:Ungültigkeit: Reserviert
```

Die einzelnen Felder haben folgende Bedeutung:

Username

Der Username, wie in `/etc/passwd`

Passwort

Das verschlüsselte Passwort. Wenn hier statt eines verschlüsselten Passworts nur ein Sternchen steht, so bedeutet das, daß es kein gültiges Passwort gibt, mit dem sich dieser User einloggen kann. Das ist meist bei Verwaltungusern (wie etwa bin, daemon, lp, ...) der Fall. Nur der Systemverwalter kann sich mit dem su-Befehl in diese User verwandeln.

Alter

Tage vom ersten Januar 1970 zu dem Tag, an dem das Passwort das letzte Mal geändert wurde.

min. Alter

Anzahl der Tage, bis ein Passwort geändert werden darf.

max. Alter

Anzahl der Tage, bis wann ein Passwort geändert werden muss.

Warnzeit

Anzahl der Tage, vor dem Auslaufen der Gültigkeit des Passworts, ab denen der User auf das baldige Auslaufen seines Passworts hingewiesen wird.

Pufferzeit

Anzahl der Tage, die nach dem Auslaufen des Passworts verstreichen, bis der Account tatsächlich ungültig wird.

Ungültigkeit

Tage vom ersten Januar 1970 zu dem Tag, an dem das Passwort ungültig wird.

Reserviert

Ein reserviertes Feld für zukünftige Erweiterungen.

Die beiden ersten Angaben (Username und Passwort) müssen zwingend gemacht werden. Mit den folgenden Angaben können sicherheitsrelevante Einstellungen vorgenommen werden.

Das Passwort darf nicht geändert werden, bis die entsprechende Anzahl von Tagen (min.Alter) seit der letzten Änderung vergangen sind und muss geändert werden, sobald die maximale Anzahl von Tagen (max.Alter) seit der letzten Änderung verstrichen sind. Wenn das Feld *min.Alter* eine größere Zahl als *max.Alter* enthält, so darf ein User sein Passwort gar nicht ändern.

Ein Passwort wird ungültig, wenn es nach der angegebenen Anzahl der maximalen Gültigkeitsdauer (max.Alter) und der anschließenden Pufferzeit nicht geändert wurde.

Ein Passwort wird auch grundsätzlich ungültig, wenn das letzte gültige Feld (Ungültigkeit) ausgefüllt ist und die entsprechende Anzahl von Tagen erreicht ist. Damit kann ein begrenzter Zugriff implementiert werden, der an einem bestimmten Tag endet.

6. Die Datei /etc/group

Diese Datei enthält Informationen über die Gruppen, die dem System bekannt sind und deren Mitglieder. Wenn ein User über seinen Eintrag in der /etc/passwd einer Gruppe zugeordnet ist, so steht er NICHT automatisch in dieser Liste, er ist aber automatisch Mitglied dieser Gruppe.

Wie die Datei /etc/passwd ist auch /etc/group eine Textdatei, deren Zeilen die Einträge repräsentieren. Die Felder sind auch hier wieder durch Doppelpunkte getrennt.

Die Zeilen haben folgendes Format:

```
Gruppenname: Passwort: GroupID: Mitgliedsliste
```

Gruppenname

Der Name der Gruppe

Passwort

Entsprechend der /etc/passwd Datei standen hier früher die Passwörter für die Gruppen. Heute steht hier, wie bei /etc/passwd, ein x - die eigentlichen Passwörter stehen heute in der Datei /etc/gshadow

GroupID

Hier steht die numerische GID

Mitgliedsliste

Eine durch Kommas getrennte Liste von Usernamen. Wessen Username hier steht ist Mitglied der Gruppe und hat die entsprechenden Rechte.

7. Die Datei /etc/gshadow

In manchen modernen Unixen hat auch die Gruppenverwaltung ihre Passwörter in einer separaten Datei, eben /etc/gshadow. Die Felder sind wie üblich durch Doppelpunkte voneinander getrennt:

```
Gruppenname: Passwort: Gruppenverwalter: Mitgliedsliste
```

Die einzelnen Felder haben folgende Bedeutung:

Gruppenname

Der Username, wie in /etc/passwd

Passwort

Das verschlüsselte Passwort. Falls hier nur ein * oder ein ! steht, bedeutet das, dass diese Gruppe kein Passwort hat.

Gruppenverwalter

Jede Gruppe kann einen Verwalter haben, der das Recht hat, andere Mitglieder aufzunehmen, Mitgliedschaften zu löschen oder das Passwort zu verändern. Dieser Verwalter muss NICHT der Systemverwalter sein.

Mitgliedsliste

Eine durch Kommas getrennte Liste von Usernamen wie in /etc/group

Der Hauptvorteil dieser neuen Architektur ist die Tatsache, dass jede Gruppe ihren eigenen Verwalter haben kann und so eine gewisse Eigenständigkeit aufweisen kann, ohne immer den Systemverwalter zu brauchen um eine kleine Veränderung vorzunehmen. Das Programm *gpasswd* ermöglicht es dem Gruppenverwalter, diese Änderungen durchzuführen.

Wenn ein User in der Mitgliedsliste der Gruppe steht, muss er dieses Passwort weder wissen, noch braucht er es, um auf Dateien zuzugreifen, die dieser Gruppe angehören - er ist automatisch in der Gruppe.

Nur wenn ein User, der nicht als Gruppenmitglied eingetragen ist, dessen Username also nicht in der Datei */etc/group* und */etc/gshadow* in der Mitgliedsliste der Gruppe auftaucht, auf Dateien zugreifen will, die nur für Gruppenmitglieder dieser Gruppe lesbar sind, dann muss er sich mit dem Befehl *newgrp* kurzzeitig in ein Gruppenmitglied verwandeln. Dazu braucht er das Passwort, sonst könnte das ja jeder und die Sicherheit wäre dahin.

Wenn eine Gruppe in */etc/gshadow* ein *!* oder *** im Passwortfeld aufweist, statt eines Passworts, dann gibt es für diese Gruppe kein Passwort.

8. Das Anlegen von neuen Usern

Grundsätzlich kann nur der Systemverwalter neue User im System anlegen. Dazu gibt es verschiedene Möglichkeiten, welche davon benutzt werden hängt von den Vorlieben des Verwalters ab.

8.1.1. Das manuelle Anlegen von neuen Usern

Um einen neuen User von Hand anzulegen reicht als Minimalwerkzeug wieder, wie unter Unix üblich, ein Texteditor. Die wichtigsten Schritte sind

Eintrag des Users in `/etc/passwd`

Hier werden nur die üblichen Angaben gemacht, wie vorher beschrieben. Wichtig ist hier die Überlegung, welche UserID der neue User bekommen soll und welcher Gruppe er standardmäßig angehören soll. Das kommt natürlich stark auf das jeweilige System an und auf die Frage, was der User können soll. Nehmen wir als Beispiel an, der neue User as soll die UserID 500 bekommen und Mitglied der Gruppe users (GID 100) werden. Dann würde der Eintrag in `/etc/passwd` lauten:

```
as:x:500:100:::/home/as:/bin/bash
```

Eintrag des Users in `/etc/shadow`

Hier ist manuell nur der erste Eintrag zu machen, aber die fehlenden acht Doppelpunkte dürfen nicht vergessen werden. Ein typischer Eintrag wäre also z.B.

```
as:::::::::
```

Damit hat der User as einfach kein Passwort. Der Systemverwalter kann es entweder dem User überlassen, sich jetzt selbst ein Passwort zu vergeben oder er kann mit dem Befehl *passwd as* ein Passwort anlegen. Dieses Passwort wird dann verschlüsselt in */etc/shadow* gespeichert und gleichzeitig wird auch das Alter (Feld 3) errechnet und gesetzt.

Evt. Eintrag in */etc/group* und */etc/gshadow*

Falls der User noch Mitglied anderer Gruppen werden soll, müssen noch die Einträge in */etc/group* bzw */etc/gshadow* gemacht werden. Auch das kann entweder mit einem Texteditor oder mit dem Befehl *gpasswd* vorgenommen werden

Anlegen eines Homeverzeichnis

In der Regel tragen die Homeverzeichnisse den Namen des Users und befinden sich im Verzeichnis */home*. Mit dem Befehl

```
$ mkdir /home/as
```

würde etwa für den oben genannten Beispieluser ein Homeverzeichnis erstellt. Damit könnte er aber noch nicht viel anfangen, denn dieses neue Verzeichnis würde ja noch dem Systemverwalter gehören, der es angelegt hat. Wichtig ist also auch, daß das Verzeichnis dem neuen User übereignet wird und die Gruppenzugehörigkeit der Gruppe bekommt, die die Standardgruppe des neuen Users ist. Wenn die Gruppe unseres Beispielusers hans die Gruppe users wäre, so würde der Befehl also lauten:

```
$ chown as:users /home/as
```

Kopieren der Standard-Dateien in das neue Verzeichnis

In der Regel befinden sich einige versteckte Dateien und Verzeichnisse in jedem Heimatverzeichnis, die als persönliche Konfigurationsdateien des Users dienen. Im Verzeichnis `/etc/skel` stehen all diese benötigten Dateien, sie sollten jetzt noch in das neu erstellte Verzeichnis kopiert werden. Danach müssen die Kopien wieder dem neuen User übereignet werden. Die dazu notwendigen Befehle wären also:

```
$ cp -r /etc/skel/* /home/as
$ chown -R hans:users /home/as/*
```

Damit ist der User komplett angelegt, er kann sich jetzt einloggen und mit dem System arbeiten.

9. Hörsaalübung

Ein Programmiererteam bestehend aus 2 Mitgliedern (Adam, Eva) soll eine Anwendung entwickeln.

Was muss am Unix-System getan werden, damit das Team ungestört von anderen entwickeln kann?
