Christian-Albrechts-Universität zu Kiel
Wintersemester 2000/2001
Vorkurs Informatik

Übung 1:

Ausgabetermin: 10. Oktober 2000

Abgabe: Anwesenheitsübung

Die Übungen sind zum Vertrautwerden mit den System gedacht. In dem Vorlesungsteil sind die Lösungen meist nicht vorgegeben, d.h., in der Regel werden wird man in den Dokumentationen (vor allem den Manpages) ein wenig nach dem richtigen Befehl bzw. vor allem den passenden Optionen suchen müssen.

In den Aufgaben werden Sie manchmal eine sich eine beliebige sinnfreie Datei erzeugen oder sich beschaffen müssen. *Editoren* (wie emacs/xemacs, vi oder viele andere mehr, die man in der Regel zum (sinnvollen) erzeugen und Ändern von Dateien verwendet, stehen erst später auf dem Programm des Kurses. Um sich für die Aufgaben ohne Editor schnell eine Datei zum experimentieren zu *beschaffen,* können Sie eine der folgenden Dinge tun

- eine Datei von woanders her *kopieren*
- `touch datei`: erzeugt eine *leere* Datei mit dem Namen `datei`
- `echo "Ein paar Buchstaben" > datei.txt`. Zur Erklärung: `echo` wiederholt, wie der Name andeutet, sein Argument, in diesem Fall `"Ein paar Buchstaben"`. Die Hochkommas bedeuten, daß es sich um eine sog. *Zeichenkette* (oder *string*) handelt, der geechot wird. Die Ausgabe des `echo`-Kommandos wird mit `>` in die Datei `datei.txt` geschrieben („umgeleitet“ sagt man, denn normalerweise, ohne das umleitende `>` geht die Ausgabe des Echos standardmäßig in die Shell).

Zum *Anschauen* einer Datei kann man (neben verschiedenen ausgewachsenen Editoren) folgende, primitive Befehle verwenden:

- `more datei.txt`
- `less datei.txt`
- `pg datei.txt`

Aufgabe 1: [Verzeichnisstruktur und Rechte]

Legen sie ein paar Verzeichnisse und neue Dateien an.

1. Stellen Sie fest, welche Voreinstellungen Ihnen die Systemadministration bzgl. den Rechten von

- neuen Unterverzeichnissen
- neuen Dateien

vorgegeben hat

2. Können Sie feststellen, welche Dateien Ihr *Nachbar* in *seinem* Home hat (einschließlich der versteckten Dateien?)
3. Setzen Sie die Rechte Ihres Homeverzeichnisses so, daß *jeder* lesen kann, welche Dateien und Unterverzeichnisse Sie besitzen. Stellt das eine Gefahr da? Impliziert diese Entscheidung, daß jeder den *Inhalt* der Dateien lesen kann? Oder den Inhalt der Unterverzeichnisse?
4. Erzeugen Sie in Ihrem Home ein „privates" Verzeichnis an, welches nur Sie (d.h., der *account* der Ihnen und Übungspartner zugeteilt wurde) lesen kann.[1] Welche Rechte würden Sie dabei empfehlen, was die „*Ausführbarkeit*" des neuen Verzeichnisses betrifft?
5. **optional** (für den Anfang nicht wichtig/für den etwas Fortgeschritten): Unter Punkt 1 haben Sie herausgefunden, was die Administration für Ihren account als Standardrechte für angemessen hält. der Befehl `umask` erlaubt, sich diese Voreinstellung anzuschauen (wenn man die Ausgabe von `umask` zu interpretieren versteht ...) und zu ändern. Setzen Sie Ihre Einstellung (zeitweise) so, daß alles was Sie neu anlegen, ausschließlich für Sie zugreifbar ist und so, daß alles für alle *lesbar* ist.[2]

Aufgabe 2: [Verzeichnisstruktur ändern]

Spielen Sie ein wenig mit der Verzeichnisstruktur und Dateien, z.B.

1. Legen Sie ein paar neue Verzeichnis an.
2. Löschen Sie ein Verzeichnis.
3. Erzeugen Sie Dateien in den Verzeichnissen.
4. Kopieren Dateien von einem Verzeichnis in ein anderes.
5. Kopieren Sie eine Datei in eine neue Datei. D.h. gibt es einen Unterschied zwischen den beiden Operationen?
6. Verlagern Sie eine Datei in ein anderes Verzeichnis.
7. Benennen Sie eine Datei in eine Datei mit einem neuen Namen um. Gibt es einen Unterschied?
8. Verlagern Sie ein *Verzeichnis* an einen neuen Ort. Benennen Sie es um? Was passiert dabei mit Dateien, die sich in einem Verzeichnis befinden? Was mit eventuellen Unterverzeichnissen?

[1] Beachte natürlich: der sogenannter "Superuser" (root) kann das Verzeichnis auf jeden Fall lesen, welche Rechte man auch immer wählt. Zur Not, falls es die Rechte (sinnloserweise) so eingeschränkt wurden, daß *niemand*, nichteinmal man selbst, die Daten lesen kann, kann *root* die Rechte ändern und dann lesen. Vor *root* gibt es (fast) keinen Schutz.

[2] Seien Sie vorsichtig bei Dieser Übung. Es ist nicht gut, wenn man für Fremde Verzeichnisse/Dateien auf "schreibbar" setzt (warum nicht?). Sie können natürlich die Rechte immer nachträglich ändern: `umask` dient dazu, die Rechte, die Sie *in der Mehrzahl der Fälle* bevorzugen, zu standardisieren.

9. Tippen Sie an einem geeigneten Ort

   `ls -lag > datei.txt`

   ein. Was passiert? Was passiert wenn man es nochmals macht? Was passiert, wenn man danach

   `ls -lag >> datei.txt`

   macht?

10. Nach all den Operationen: Stellen Sie fest, wieviel Speicherplatz sie bereits verbraucht haben? Wieviel haben Sie überhaupt insgesamt zur Verfügung?

Aufgabe 3: [Dateiformate]

1. Unter `~kurs001/tmp/datei1.txt.gz` liegt eine Datei. Was für eine Datei ist das? Was ist ihr Inhalt?
2. Was ist der Inhalt von `~kurs001/tmp/datei2.ps`?
3. Und was steht in `~kurs001/tmp/datei3.tar`? Hinweis: schauen Sie sich die Manpages zum Befehl `tar` an. Da `tar` recht vielseitig ist, ist die Beschreibung allerdings bereits recht lang und die Beispiele schon recht esoterisch.

Aufgabe 4: [Disketten]

1. Kopieren Sie eine Datei auf Diskette (unter einem anderen Namen.)
2. Nehmen Sie an, die Datei sei zu groß, um auf die Diskette zu passen, komprimieren sie die Datei, damit sie draufpaßt.
3. Wie würden sie ein *Verzeichnis* mit mehrere Dateien auf Diskette kopieren?
4. **Für Fortgeschrittene:** Nehmen Sie an, die Datei sei so groß, daß sie nicht einmal komprimiert auf eine Diskette passt? Was macht man? (Schauen sie sich die Befehle `split` oder `shar` an, `split` ist vermutlich einfacher)

Aufgabe 5: [Prozesse]

Ein Prozeß ist (grob) ein *Programm in Ausführung.* Rufen sie ein beliebiges Programm auf, sagen wir, die X-Version der `Manpages`:

1. Rufen Sie in einer Shell
   (a) `xman` bzw.
   (b) `xman &`

auf. Beobachten Sie den Unterschied. Beobachten Sie, was im Fall a) passiert wenn man `^Z` = "*Control-Z*" eingibt. Und was, wenn man danach `bg` eingibt? Beobachten Sie, was im Fall b) passiert, wenn man hinterher `fg` eingibt.

2. Finden Sie heraus, *wo* im System sich das Programm, d.h., die ausführbare Datei `xman` befindet.[3]
3. Terminieren Sie das Programm
   (a) mittels `^C` = "*Control C*"[4]
   (b) mittels des Befehls `kill`, wobei Sie vorher vermutlich die *Nummer* des Prozesses herausfinden müssen
   (c) in beiden Fällen: mittels des Fenstersystems

Aufgabe 6: [Wildcards/Suchen]

1. Zeigen Sie in Ihrem Home-verzeichnis alle Dateien, die ein `b` enthalten. Achten Sie auf die versteckten Dateien.
2. Suchen Sie mit dem Befehl `find` alle Dateien unterhalb ihres Homeverzeichnis mit der Endung/Extension `txt` (Beachte: `find` nicht auf das globale root-Verzeichnis `/` anwenden.
3. Unter `~kurs001/tmp` befinden sich zwei Textdateien `worte1.txt` und `worte2.txt`. Schauen Sie sich den Befehl an, um in `worte1.txt` alle Worte zu finden, die die Buchstaben "`ei`" oder "`Ei`"enthalten.[5] Wie findet man (mit nur einem Befehlsaufruf) alle derartigen Worte in *beiden* Dateien.
4. **Für die Harten (optional, war nicht auf den Folien):** die Vorige Aufgabe suchte nach dem direkten String `"ei"`. Wie kann man alle Worte finden, die ein `"e"` besitzen und danach irgenwann ein `"ng"`? Wie die Worte, die mit `"e"` anfangen und mit `"ng"` aufhören? Falls Sie das versuchen wollen: das Stichwort sind *reguläre Ausdrücke*

[3] Eine Möglichkeit, den Ort zumindest einzugrenzen, wäre mittels eines globalen `find`. **Bitte nicht machen!** Neben der Tatsache, daß dies nicht direkt die Version gibt, die tatsächlich ausgeführt wird, wird ein *globale* Suche mittels `find` von den Systemadministratoren stark mißbilligt. Der Grund ist nicht, daß `find` eine schlechter Befehl ist (er ist oft nützlich), sondern an der Netzstruktur des Rechnerpools. Näheres in den des Instituts. Morgen allgemein zu dem Thema Netze mehr.

[4] Ein Standardtaste zur Terminierung

[5] Der Befehl `grep` sucht genauer gesagt nach passenden *Zeile,* nicht Worten, aber in diesem Fall enthalten die Dateien ein Wort pro Zeile.