

Grundlagen von Betriebssystemen

Aufgabenblatt 5

7 Standardeingabe und Standardausgabe

Viele Prozesse lesen und/oder schreiben Daten. So schreibt z.B. `cat datei` den Inhalt von `datei` in die Terminaldatei (s. Gerätedateien). Erinnern Sie sich, dass Sie mit `cat > datei` auch Daten in die Datei `datei` geschrieben haben. Ein weiteres Beispiel ist `rm -i`, das sowohl Daten (Meldungen) auf das Terminal schreibt als auch Eingaben von der Tastatur erwartet (`j/n`).

Zu jedem Kommando gibt es voreingestellte (Standard-) Dateien, von denen Daten gelesen und/oder in die Daten geschrieben werden. Diese nennt man:

- Standard-Eingabe (`stdin`) (meist die Tastatur)
- Standard-Ausgabe (`stdout`) (meist das Terminal des Prozesses (`tty`))
- Standard-Fehlerausgabe (`stderr`) (meist das Terminal des Prozesses)

1. Starten Sie einen Prozess, indem Sie `cat` (ohne Argument) eingeben. Geben Sie nun am Cursor einen mehrzeiligen Text ein (mit Zeilenumbrüchen). Beenden Sie den Prozess mit `^D`. Sie sehen, dass die Standardausgabe offenbar das Terminal ist. Wenn man die Eingabe in einer regulären Datei auffangen möchte, dann muss man die Standard-Ausgabe **umlenken**. Dazu dient das `>`. Probieren Sie das Kommando `cat > datei` noch einmal aus!
2. Sehen Sie sich den Inhalt mit `cat datei` an. (Hier ist `datei` Argument!) Erzeugen Sie eine weitere Datei mit dem Namen `target` und folgendem Inhalt:

```
Dies ist die Datei target.  
Gleich mache ich sie kaputt.
```

Kontrollieren Sie mit `cat target`.

3. Lenken Sie jetzt die Ausgabe von `cat datei` in die Datei `target` um.

Kommando: _____

Was ist passiert mit `target` passiert? _____

4. Benutzen Sie jetzt das Umlenkzeichen doppelt: `cat datei >> target`.

Was ist jetzt mit `target` passiert? _____

Info: Ist die Shellvariable `noclobber` gesetzt (ohne Wert), so können mit `>` keine Dateien überschrieben werden. Mit `>!` kann man dann die Umlenkung (und damit das Überschreiben der Zieldatei) erzwingen.

5. Automatisieren Sie die Kommunikation mit `rm -i`.

- (a) Weil `rm -i` die Antwort `j` erwartet (bei Einstellung des Systems auf English: `y`), erstellen Sie eine Datei `antwort`, die eine einzige Zeile mit einem `j` enthält. Schließen Sie diese Zeile mit einem Zeilenumbruch (`<ENTER>`) ab, bevor Sie speichern.

- (b) Löschen Sie jetzt Ihre Datei `datei` mit `rm -i datei < antwort`. Die Datei wurde ohne Sicherheitsabfrage gelöscht. Erklären Sie ausführlich, wie es dazu kommt!

6. Arbeiten Sie auch mit der Standard-Fehlerausgabe.

- (a) Führen Sie folgendes Kommando aus: `cat antwort gibtsnicht` Sie sehen, dass sowohl der Ausgabekanal als auch der Fehlerausgabekanal von `cat` mit dem Terminal verbunden sind. Leitet `>` die Fehlerausgabe mit um?

`cat antwort gibtsnicht > out` _____

Die Datei `out` ist entstanden. Sehen Sie sich den Inhalt an.

- (b) Probieren Sie jetzt `cat antwort gibtsnicht >& out`.

Welche Funktion erfüllt `>&`? _____

- (c) Was geschieht bei `(cat antwort gibtsnicht > out) >& err?`

Mit welchen Shell-Funktionen ist das zu erklären?

Hinweis: Kommandos, die in ein Paar runder Klammern eingeschlossen sind, werden in einer Subshell der aufrufenden Shell ausgeführt. Probieren Sie z.B. `(cd /etc; pwd)`.

Welche Ausgabe erhalten Sie? _____

In welchem Verzeichnis stehen Sie danach? _____

Warum? _____

7. Verknüpfen Sie Kommandos.

- (a) Das letzte Beispiel, das in der Subshell ausgeführt wurde, zeigt bereits, wie man Kommandos hintereinander ausführen kann. Probieren Sie, mit einem einzigen Kommando erst `who` und dann `date` auszuführen.

Kommando: _____

- (b) Die Ausgabe von `ls -l /bin` ist zu groß, um in einem Terminalfenster vollständig angezeigt zu werden. Sie haben das Kommando `more` kennengelernt, um große Dateien seitenweise anzeigen zu lassen. Warum funktioniert `ls -l /bin; more` nicht, um die Ausgabe seitenweise auszugeben? _____

- (c) Eine **Pipeline** leitet die Ausgaben eines Prozesses in den Eingabekanal eines anschließend auszuführenden Prozesses. Das Verknüpfungszeichen ist `|`.

Probieren Sie `ls -l /bin | more`.

- (d) Das Kommando `wc` (*wordcount*) gibt die Größe von Dateien aus: `wc antwort`

Ausgabe: _____

Interpretieren Sie die einzelnen Ausgabewerte. Benutzen Sie die Manualseite von `wc`.

- (e) Mit welchem Kommando können Sie die Zeilen in der Ausgabe von `ls -l /bin` zählen? (Benutzen Sie den Pipeline-Mechanismus der Shell und die richtige Option von `wc`).

Kommando: _____ Ausgabe: _____

- (f) Definieren Sie eine lokale Variable mit dem Namen `benutzer` und weisen Sie ihr als Wert die Anzahl der Ausgabezeilen von `who` zu.

Kommando: _____

Ändern Sie jetzt den Wert von `benutzer` so, dass die Zeichenkette `Benutzer:` vor dem aktuellen Wert gespeichert wird. Speichern Sie dabei den zuvor ermittelten Wert (bisheriger Wert der Variablen) *nicht* noch einmal neu!

Kommando: _____

- (g) Vergleichen Sie die Ausgaben der Kommandos `ls -l /bin /gibtsnicht | more` und `ls -l /bin /gibtsnicht |& more`.

Welche Funktion erfüllt `|&?` _____

8 Einige UNIX-Werkzeuge

UNIX- und LINUX-Distributionen stellen einen umfangreichen Satz an Werkzeugen zur Verfügung, die gewisse Dienste bei der Arbeit im System erbringen. Dazu gehören Editoren wie z.B. der `vi` oder einfache Dienste wie `wc` bis zu Compilern höherer Programmiersprachen wie `C`. Anhand einiger Beispiele sollen Sie Ihre Übersicht über UNIX-Werkzeuge erweitern.

8. Suchen Sie Dateien im Dateisystem.

Ein Kommando der Form `find Verzeichnis Suchkriterien` gibt alle Dateien in *Verzeichnis* und allen seinen Unterverzeichnissen aus, die die *Suchkriterien* erfüllen.

- (a) Laden Sie von der Webseite der Lehrveranstaltung die Datei `Suchanfragen.pdf` herunter und informieren Sie sich über einige Suchkriterien bei `find`.
- (b) Probieren Sie aus und erklären Sie mit eigenen Worten, welche Suchanfrage jeweils ausgeführt wird.

`find /usr/local/etc -type d -print` _____

`find /usr/bin -name '*sh' -print` _____

`find /usr/bin ! -user root -print` _____

`find /usr/bin ! -user root -exec ls -l {} \;` _____

- (c) Lassen Sie beim zweiten Kommando einmal die Apostrophe weg. Warum funktioniert die Suchanfrage so nicht? (*Denken Sie an die Shell-Substitutionen!*)

- (d) Lassen Sie sich alle pdf-Dateien anzeigen, die in Ihren persönlichen Verzeichnissen liegen.

Kommando: _____

- (e) Lassen Sie sich (mit einem Kommando!) alle pdf-Dateien anzeigen, die in den persönlichen Verzeichnissen aller Benutzer des Systems liegen. (*Sie dürfen unterstellen, dass in `/home` keine pdf-Datei liegt.*)

Kommando: _____

- (f) Wiederholen Sie diese Aufgabe, aber leiten Sie diesmal die korrekten Ausgaben in eine reguläre Datei um, während Sie die Fehlermeldungen nach `/dev/null` umleiten (*entspricht dem Wegwerfen der Fehlermeldungen*).

Kommando: _____

9. Durchsuchen Sie Dateiinhalte und andere Datenströme.
 Ein Kommando der Form `grep Muster Datei_1 Datei_2 ... Datei_n` gibt alle Zeilen in *Datei_1*, *Datei_2* oder *Datei_n* aus, die das **Muster** enthalten.
- (a) Leiten Sie die Ausgabe von `man ls` in eine Datei mit dem Namen `ls.info`. Führen Sie dann aus: `grep option ls.info` und überzeugen Sie sich, dass alle ausgegebenen Zeilen das Wort `option` enthalten.
- (b) Benutzen Sie `grep` mit nur einem Argument: `grep n`. Geben Sie nach dem Aufruf von `grep e` einige Wörter, z. B. Vornamen, ein, die Sie jedesmal mit `<ENTER>` abschließen (also zeilenweise eingeben). Beenden Sie mit `^D`. Welche Zeilen werden hier von `grep` nach dem Muster `n` durchsucht? _____
- (c) Erzeugen Sie die gleiche Ausgabe wie bei `grep option ls.info` mit einem Kommando, das die Datei `ls.info` nicht benutzt und statt dessen den Pipeline-Mechanismus der Shell verwendet.
 Kommando: _____
- (d) Wechseln Sie ggf. in Ihr Login-Verzeichnis. Lassen Sie sich alle Zeilen in versteckten Dateien Ihres Loginverzeichnisses anzeigen (`.*`), die das Wort `alias` enthalten.
 Kommando: _____
- (e) Informieren Sie sich über Optionen von `grep` (Manual-Seiten!). Geben Sie mit eigenen Worten die Bedeutung folgender Optionen wieder.
`-n` _____
`-l` _____
`-s` _____
`-i` _____
- Geben Sie jetzt die *Namen* aller versteckten Dateien in Ihrem Loginverzeichnis aus, die das Wort `alias` enthalten.
 Kommando: _____
- (f) Lassen Sie sich mit einem einzigen Kommando alle Zeilen von Werten der Umgebungsvariablen ausgeben, die die Zeichenkette `usr` enthalten. (*Denken Sie wieder an den Pipeline- Mechanismus.*)
 Kommando: _____
10. Arbeiten Sie mit **regulären Ausdrücken**. Die Suche nach (Teil-) Worten in den Zeilen einer Datei ist bei `grep` nur ein Spezialfall. Im Allgemeinen sucht `grep` nach Zeilen, die Zeichenketten enthalten, welche auf ein Muster passen, das durch einen UNIX-regulären Ausdruck definiert ist.
- (a) Sehen Sie sich in `Suchanfragen.pdf` die Bedeutung der Zeichen an, die zur Angabe von UNIX-regulären Ausdrücken verwendet werden können.
- (b) Kopieren Sie die Dateien `/home/rlehre/adressen` und `/home/rlehre/bdresses` in Ihr Arbeitsverzeichnis. (Geht das mit nur einem Kommando?)
 Sehen Sie sich die Dateiinhalte an.
- (c) Betrachten Sie die Ausgaben folgender Suchanfragen und notieren Sie die Teilwörter der ausgegebenen Zeilen, die auf das Anfragemuster des Kommandos passen:
`grep 'W...r' adressen` _____
`grep 'K.r' adressen` _____
`grep 'K.*r' adressen` _____

grep '.er' adressen _____

grep '[bk]er' adressen _____

grep '[b-k]er' adressen _____

grep '[bkA-M]er' adressen _____

grep 'in' adressen _____

grep 'in\$' adressen _____

- (d) Bei welchen dieser Suchanfragen kann man die Apostrophe weglassen?

Probieren und Begründen Sie. _____

- (e) Achten Sie auf die Unterschiede zwischen den Mechanismen der Shell-Substitution und regulären Ausdrücken! Wechseln Sie dazu in das Verzeichnis `/usr/bin`. Wiederholen Sie nun folgende Aufgabe zur Dateinamenexpansion durch die Shell:

Geben Sie mit Hilfe des `echo`-Kommandos alle Dateinamen in `/usr/bin` aus, die

i. die Länge 2 haben, Kommando: _____

ii. mit einem Vokal beginnen und auf `d` oder `s` enden, Kommando: _____

iii. mindestens eine Ziffer enthalten, Kommando: _____

Das `echo`-Kommando soll jeweils nur ein Argument haben!

Geben Sie die jeweils gleichen Dateinamen aus, indem Sie das Kommando

`ls -1 | grep`

geeignet ergänzen. (Die Option `-1` sorgt dafür, dass die Ausgabe von `ls` einspaltig erfolgt, jede Zeile der Ausgabe also genau einen Dateinamen enthält.)

(i) _____ (ii) _____ (iii) _____

Überprüfen Sie mit dem Kommando `wc`, dass jeweils die gleiche Anzahl von Dateinamen wie bei Aufgabe 1 ausgegeben wird.

- (f) Wechseln Sie in `/usr`. Geben Sie nun alle direkten Unterverzeichnisse, die (mindestens) Lese- und Ausführungsrechte für `group` und `others` haben (Hinweis: Benutzen Sie `grep`.)

Kommando: _____

11. Finden Sie Unterschiede zwischen verschiedenen Versionen einer Datei.

- (a) Führen Sie aus: `diff adressen badressen` und danach `diff badressen adressen`. Welche Symbole in der Ausgabe von `diff` haben welche Bedeutung?

(*Hinweis: Denken Sie an die englischen Wörter change, append und delete.*)

- (b) Probieren Sie jetzt dasselbe Kommando mit der Option `-y`.

Bedeutung: _____