Universität Potsdam
Institut für Informatik

GdP-Rechnerübung

Aufgabenblatt 5
(zuletzt aktualisiert: 14. Oktober 2022)

7 Shell-Substitutionen

1. Arbeiten Sie mit Shell-Substitutionen!
   Sie haben bereits gesehen, dass das `$`*-Zeichen eine Sonderbedeutung in der Shell hat: Bevor ein Kommando ausgeführt wird, substituiert die Shell zunächst alle Zeichenkette, die mit* `$` *beginnen, durch die Werte derjenigen Variablen, deren Namen hinter dem* `$`*-Zeichen beginnt. Es gibt weitere Shell-Substitutionen.*

   (a) Wechseln Sie in `/usr/bin` und führen Sie aus:

   `ls *sh`; Ausgabe: _____

   `ls ?sh`; Ausgabe: _____

   `ls [az]sh`; Ausgabe: _____

   `ls [a-m]sh`; Ausgabe: _____

   Hinweis: Auf Linux-Systemen müssen Sie ggf. das Verzeichnis `/bin` *benutzen.*
   Interpretieren Sie die Ergebnisse mit dieser Übersicht zur **Dateinamenexpansion**:

Pattern	Bedeutung
`*`	beliebige Zeichenkette, auch die leere (Wildcard)
`?`	ein beliebiges Zeichen (Wildcard)
`[...]`	ein Zeichen aus der Menge, z.B. `[aeiou]`
`[`*von*`-`*bis*`]`	ein Zeichen zwischen *von* und *bis*
`[^...]`	ein Zeichen außerhalb der Menge, z.B. `[^0-9]`
`^...`	Negation des gesamten Patterns
{{*wort1*,...,*wortn*}}	eines der Wörter
∼	absoluter Pfadname des (eigenen) Login-Verzeichnisses
∼*user*	abs. Pfadname des Login-Verzeichnisses von *user*

   (b) Wechseln Sie in Ihr Login-Verzeichnis. Werden bei `echo *` alle Dateien angezeigt? Probieren Sie auch `echo .*`
   Wie können Sie sich alle Dateien in Ihrem Login-Verzeichnis mit nur einem `echo`-Kommando ausgeben lassen?

   _____

   (c) Wechseln Sie in das Verzeichnis `/usr/bin`. Geben Sie nun mit Hilfe des `echo`-Kommandos alle Dateinamen in `/usr/bin` aus, die

   i. die Länge 2 haben: _____
   ii. mit einem Vokal beginnen und auf `d` oder `s` enden: _____
   iii. mindestens eine Ziffer enthalten: _____

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

(d) Führen Sie folgende Kommandos aus, die eine **Kommandosubstitution** enthalten und notieren Sie die jeweilige Ausgabe:

i. `echo Ich arbeite auf dem System` `` `uname` ``. _____

ii. `echo Die Systemzeit ist` `` `date` ``. _____

Hinweis: Nutzen Sie den Backtick ` (Shift + Taste rechts neben ß, 2-mal drücken), nicht das einfache Anführungszeichen ' !

(e) Erklären Sie, welchen Wert die Variable `vlist` nach `vlist=`ls`` hat.

_____

(f) Welche ASCII-Zeichen haben Sie insgesamt kennengelernt, die eine Sonderbedeutung in der Shell besitzen?

_____

Um diese Zeichen vor einer Interpretation durch die Shell zu schützen, können Sie sie auf drei Arten maskieren:

`\`	schützt das direkt folgende Zeichen,
`'...'`	schützt alle eingeschlossenen Zeichen,
`"..."`	schützt alle eingeschlossenen Zeichen außer `\, $, !` und `

(g) Geben Sie mit `echo` aus: `Die Zeichen * und $HOME werden nicht interpretiert.` Maskieren Sie dabei die beiden Sonderzeichen auf die drei verschiedenen Arten.

Kommando mit 1. Maskierungsart: _____

Kommando mit 2. Maskierungsart: _____

Hinweis: Die Ausgabe sollte für beide Kommandos lauten: `Die Zeichen * und $HOME werden nicht interpretiert.`

Kommando mit 3. Maskierungsart: _____

Ausgabe bei der 3. Maskierungsart: _____

(h) Geben Sie in `/home/gdplehre/unix-uebungen` alle Dateien aus, deren Namen ein Fragezeichen enthält.

Kommando: _____

8 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 auch mit* `cat > datei` *Daten in eine 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*)

2. Starten Sie einen Prozess, indem Sie `cat` (ohne Argument) eingeben. Geben Sie nun am Cursor einen mehrzeiligen Text ein (mit Zeilenumbrüchen). Beenden Sie die Eingabe mit `CTRL+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* `>`*-Zeichen.* Probieren Sie das Kommando `cat > datei` noch einmal aus!

3. Sehen Sie sich den Inhalt mit `cat datei` an. (*Hier ist* `datei` *das 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`.

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

   Kommando: _____

   Was ist mit `target` passiert? _____

5. Benutzen Sie das Umlenkzeichen in dem Kommando doppelt, also: `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.*

6. Automatisieren Sie die Kommunikation mit `rm -i` durch Verwendung des Standardeingabekanals.

   (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` bzw. `y` 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!

   _____

   _____

   _____

7. 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 1> out 2> err`?

   _____

   _____

   Hinweis: `1>` *leitet nur die Standardausgabe (stdout) weiter,* `2>` *nur die Standardfehlerausgabe (stderr).*

8. Verknüpfen Sie Kommandos.

   (a) Probieren Sie mit einem einzigen Kommando erst `who` und dann `date` auszuführen. Verwenden Sie dafür die Konkatenation (`Befehl1;Befehl2`)

   Kommando: _____

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

(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 /usr/bin` zählen? (Benutzen Sie den Pipeline-Mechanismus und die richtige Option von `wc`).

Kommando: _____ Ausgabe: _____

9 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 z.B. Editoren oder einfache Dienste wie `wc` *bis zu Compilern höherer Programmiersprachen wie* `C`. *Anhand einiger Beispiele sollen Sie Ihre Übersicht über UNIX-Werkzeuge erweitern.*

9. 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) 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 {} \;` _____

_____

(b) Wechseln Sie in das Verzeichnis `/usr/bin` und lassen Sie beim zweiten Kommando einmal die einfachen Anführungszeichen weg. Warum funktioniert die Suchanfrage so nicht? (*Denken Sie an die Shell-Substitutionen!*)

_____

_____

(c) Lassen Sie sich mit `find` alle Dateien anzeigen, die in Ihrem Login-Verzeichnis liegen und auf '.pdf' enden.

Kommando: _____
Hinweis: Falls Sie noch keine PDF-Datei in Ihrem Login-Verzeichnis haben, können Sie sich dieses Aufgabenblatt in Ihr Login-Verzeichnis kopieren:
`cp /home/gdplehre/unix-uebungen/aufgabenblaetter/u05.pdf ~`

(d) Lassen Sie sich (mit einem `find`-Kommando!) alle Dateien anzeigen, die auf '.pdf' enden und in den Login-Verzeichnissen aller Benutzer des Systems liegen.

(*Sie dürfen annehmen, dass in* `/home` *keine pdf-Datei liegt.*)

Kommando: _____

(e) Wiederholen Sie diese Aufgabe, aber leiten Sie diesmal die Fehlermeldungen nach `/dev/null` um (*entspricht dem Wegwerfen der Fehlermeldungen*). Die korrekten Ausgaben sollen weiterhin auf dem Terminal erscheinen.

Kommando: _____

10. 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 n` einige Wörter, z. B. Vornamen, ein, die Sie jedesmal mit `<ENTER>` abschließen (also zeilenweise eingeben). Beenden Sie mit `CTRL+D`. Die Zeilen welcher „Datei“ werden hier von `grep` nach dem Muster `n` durchsucht? _____

(c) Erzeugen Sie die gleiche Ausgabe wie bei `grep option ls.info` mit einem Kommando, bei dem grep 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.
Hinweis: Verwerfen Sie wieder die Fehlermeldungen.

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: _____

Regulärer Ausdruck	Bedeutung
`Zeichen`	dieses Zeichen
`.`	beliebiges Zeichen
`[...]`	Zeichenklasse
`[^...]`	negierte Zeichenklasse
`*`	beliebige Wiederholung des letzten Ausdrucks
`+`	Wdh.: mind. ein Vorkommen
`?`	Wdh.: 0 bis 1 Vorkommen
`{n}`	Wdh.: n Vorkommen
`{n,}`	Wdh.: mind. n Vorkommen
`{n,m}`	Wdh.: n bis m Vorkommen
`^`	Vorkommen am Zeilenanfang
`$`	Vorkommen am Zeilenende
`|`	Oder Verknüpfung
`(...)`	Gruppierung

(g) 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.*

i. Sehen Sie sich in `Suchanfragen.pdf` die Bedeutung der Zeichen an, die zur Angabe von UNIX-regulären Ausdrücken verwendet werden können.

ii. Kopieren Sie die Dateien `adressen` und `bdressen` aus dem Verzeichnis `/home/gdplehre/uniex-uebungen` in Ihr Arbeitsverzeichnis. Sehen Sie sich die Dateiinhalte an.

iii. 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 '[b-k]er' adressen` _____

`grep '[bkA-M]er' adressen` _____

`grep 'in' adressen` _____

`grep 'in$' adressen` _____

iv. Bei welchen dieser Suchanfragen kann man die einfachen Anführungszeichen weglassen? Probieren und begründen Sie. _____

_____

_____

v. Achten Sie auf die Unterschiede zwischen den Mechanismen der Shell-Substitution und regulären Ausdrücken! *Denken Sie bei dieser Aufgabe daran, dass Sie gewisse Zeichen durch Maskierung vor ungewollter Interpretation durch die Shell schützen müssen!*
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

A. die Länge 2 haben, Kommando: _____

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

C. 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` *(*Minus Eins*) sorgt dafür, dass die Ausgabe von* `ls` *einspaltig erfolgt, jede Zeile der Ausgabe also genau einen Dateinamen enthält.*) Überprüfen Sie mit dem Kommando `wc`, dass jeweils die gleiche Anzahl von Dateinamen wie zuvor ausgegeben wird.

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

vi. Wechseln Sie in `/usr`. Geben Sie nun alle direkten Unterverzeichnisse, die (mindestens) Lese- und Ausführungsrechte für *group* und *others* haben *(Hinweis: Benutzen Sie* `grep`*, um die Ausgabe von* `ls` *zu durchsuchen.)*

Kommando: _____

(a) Finden Sie Unterschiede zwischen verschiedenen Versionen einer Datei. Schauen Sie sich hierzu zunächst noch einmal den Inhalt von den Dateien `adressen` und `bdressen` an.

(b) Führen Sie aus: `diff adressen bdressen` und danach `diff bdressen adressen`.

(c) Welche Symbole in der Ausgabe von `diff` haben welche Bedeutung?

_____

_____

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