

A Shell, Wildcards und bedingte Verkettung von Befehlen

Wichtig: Bitte nicht nur die **Fragen beantworten**, sondern auch unbedingt **alle** verwendeten **Befehlszeilen** und (falls gefragt) **Ausgaben** im Protokoll angeben! Bei Schwierigkeiten bitte nicht ewig experimentieren oder abwarten, sondern Nachbarn oder Laborbetreuer fragen!

- (1) Melden Sie sich am System (Kali-Linux VM) mit dem Benutzer junioradmin an (an der WSL mit Ihrem User).
- (2) Wechseln Sie in das Verzeichnis /etc
- (3) Zeigen Sie folgende Dateien (bzw. Ordner – aber nur den Ordner-Namen, nicht den Inhalt des jeweiligen Ordners - *Tipp: -d*) an (*wie?*):
 - a. Alle Dateien und Verzeichnisse mit genau 3 Zeichen Länge
 - b. Dateien und Ordner, deren drittletzter Buchstabe ein r ist
 - c. Dateien und Ordner, deren 3. Buchstabe ein p ist (*wenn nicht vorhanden, mit g versuchen*)
 - d. Dateien und Ordner, deren 3. Buchstabe ein p ist (*wenn nicht vorhanden, mit g*), und deren Namenslänge größer als 3 ist
 - e. Dateien und Ordner, deren 3. Buchstabe ein p ist (*wenn nicht vorhanden, mit g*), und deren Namenslänge exakt 3 ist
- (4) Wie geben Sie alle Zeilen der Datei /etc/passwd aus, in denen die Zeichenkette „root“ vorkommt? (Hinweis: *grep*)
- (5) Erweitern Sie nun Ihre Suchabfrage: Falls in der Datei /etc/passwd die Zeichenkette „root“ vorkommt, geben Sie nach der Ausgabe von *grep* die Wörter „SEARCH FINISHED“ aus (verwenden Sie *eine* Befehlszeile mit bedingter Befehlsverkettung bei Erfolg).
- (6) Wiederholen Sie die Suche nun **ohne** Anzeige der Fundstellen (Option: „still“) und geben Sie nur dann „Hurra, gefunden!“ aus, wenn die gesuchte Zeichenkette auftritt (Gegenkontrolle: Suchen Sie nach „batman“!).
- (7) Zusätzlich soll zum obigen Beispiel „Oje, IHR VORNAME hat nichts gefunden!“ ausgegeben werden (also wenn es keine Fundstellen gibt), indem Sie die Verkettung der Befehle mit einem weiteren Verkettungs-Operator erweitern (= bedingte Verkettung bei Misserfolg).

B History und Completion

- (8) Lassen Sie sich eine Liste der bisher ausgeführten Befehle anzeigen – man nennt das auch die *command line history* – (*wie machen Sie das also?*), finden Sie die Befehlsnummer des Beispiels d) oben heraus, und wiederholen Sie den Befehl über die Eingabe seiner Nummer (*wie? - Tipp: !...*).
- (9) Wiederholen Sie den letzten Befehl aus der *History*.
- (10) Suchen Sie in den eingegebenen Befehlen (*History*) rückwärts (*Tipp: Strg-r*) nach ?p oder ?g – welchen Befehl finden Sie?
- (11) *Tipp:* fügen Sie komplexeren Befehlen, die Sie oft wiederverwenden, einen Kommentar hinzu – etwa so:
 - `cat /etc/services #1`
 - `grep service /etc/services | sort | wc -l #2`Was machen diese Befehle? Wie suchen Sie den Befehl mit #1 am einfachsten wieder rückwärts?
- (12) Schreiben Sie den (Teil)Befehl

`cat /etc/pa`

- a. Was passiert wenn Sie die Tabulator-Taste (mehrmals) drücken?
 - b. Schreiben Sie weiter (`cat /etc/pas`) und dann die Tabulator-Taste (mehrmals) drücken?
 - c. Funktioniert das auch für Befehle (c Tab Tab)?
- (13) Noch bessere Ergänzungen (auch für Optionen) bringt das Paket `bash-completion` (ggf. nachinstallieren)

```
sudo apt install bash-completion
```

oder gleich als Superuser:

```
su
apt install bash-completion
exit
```

Was wird jetzt z.B. nach `ls -` (Tab Tab) ergänzt? Vorher das Terminal schließen und erneut öffnen!

C Ausgabe von Dateiinhalten/ Umleitung der Ein-/Ausgabe

- (14) Zeigen Sie mit `cat` den Inhalt der Dateien `/etc/passwd` und `/etc/services` an – wie machen Sie das jeweils? Verwenden Sie anschließend statt `cat` die Befehle `more` und dann `less` zum Betrachten der Datei `/etc/services`. Welche Unterschiede stellen Sie zwischen diesen drei Befehlen fest? (**Hinweis:** verwenden Sie ein Terminal mit 24 x 80 Zeichen).
- (15) Lassen Sie sich den Inhalt des Verzeichnisses `/etc` *seitenweise* unter Zuhilfenahme von `more`, dann mit `less` ausgeben (*Tipp: Pipe: befehlszeile1 | befehlszeile2*)! Worin unterscheiden sich diese beiden Befehle?
- (16) Finden Sie mit den `man`-Pages heraus, welche Funktion die Befehle `head` und `tail` (grundsätzlich) haben!
- (17) Arbeiten mit `head` und `tail`:
- a. Lassen Sie sich die ersten 8 Zeilen der Datei `/etc/services` anzeigen!
 - b. Lassen Sie sich die letzten 5 Zeilen des *System-Logs* (`/var/log/syslog`) ausgeben (WSL: z.B. `/var/log/apt/term.log`)
 - c. Lassen Sie sich die letzten 15 Zeilen der *Kernel-Meldungen* (`/var/log/kern.log` bzw., falls leer, `/var/log/kern.log.1`) ausgeben (*nur bei VM*)!
 - d. Welche der Dateien unter `/var/log/` sind eigentlich Text-Dateien (also mit direkt in der Konsole lesbar – *Tipp: file **)
 - e. Lassen Sie sich die letzten 6 Zeilen der *zuletzt geänderten Protokolldatei*, die als “ASCII Text” gespeichert wird, aus `/var/log` anzeigen. (**Hinweis** Sie dürfen dazu den Namen der Datei zuerst mit einer eigenen Befehlszeile feststellen).
- (18) Arbeiten mit Umlenkung der Ausgabe und Wildcards:
- a. Erzeugen Sie mit den Befehlszeilen

```
echo Use > Datei1
echo Study > Datei2
echo Share > Datei3
...
```

vier (4) Dateien in Ihrem *Home-Directory* (Benutzerverzeichnis, also `~`).
 - b. Lassen Sie sich nun nur jene Dateien in Ihrem Benutzerverzeichnis mit `ls` anzeigen, deren Dateinamen mit der Zeichenfolge `Dat` beginnen.
 - c. Zeigen Sie den *Inhalt* all dieser Dateien mit *einem* Befehl an („zusammengehängt“ = *concatenated*) auf der Konsole an!

- (19) Statt das Ergebnis auf der Konsole auszugeben, erzeugen Sie mit `cat` nun eine Datei `AllesZusammen`, die (wie zuvor) die Inhalte aller Dateien enthält, die mit der Zeichenfolge `Dat` beginnen.
- (20) Suchen Sie nach der Zeichenkette `log in /etc/passwd` und speichern Sie die Fundstellen in der Datei `~/log.fundstellen`.
- Fügen Sie einen Screenshot in Ihr Protokoll ein und laden Sie diesen in Eduvidual hoch!
- (21) Suche Sie alle Vorkommen von `junioradmin` in den Dateien `/etc/*` – die vielen Fehlermeldungen wollen Sie nicht sehen, leiten Sie deshalb die Fehlerausgabe mit `... 2> /dev/null` in den digitalen Papierkorb um.
- (22) **(Bonus: Heavy – erst nach dem letzten Punkt machen!)**
- Zählen Sie mit `wc` die Anzahl der Zeichen (sonst nichts) in `~/log.fundstellen` und hängen Sie das Ergebnis `n` im Format „`--- enthält n Zeichen`“) *wieder* an die Datei an. *Tipp: man `wc` für die richtige Option!*

D Bonus: Hard- und Softlinks

- (23) Die Verzeichnisbäume
- `~/Uebungsverzeichnis25_Vorname` (*Ihr Vorname ist gemein und*
 - `~/Uebungsverzeichnis__2525__`
- aus der letzten Übung sollten noch vorhanden sein – wenn nicht, erzeugen Sie diese einfach noch einmal (*siehe letzte Laborübung – in der Shell, nicht im GUI ;-!*)
- (24) Arbeiten mit symbolischen Links (*nicht vergessen, alle verwendeten Kommandozeilen anzugeben – wie bisher*):
- Erzeugen Sie einen *symbolischen Link (Softlink)* mit dem Namen `Link_Vorname` (Vorname ist Ihr Vorname, im selben Ordner wie das Original – das sollte also `~` sein), der auf `Uebungsverzeichnis_Vorname` verweist. Prüfen Sie das Ergebnis mit `ls -lR` – sind alle Dateien unter dem Softlink erreichbar?
 - Detto mit `Link2`, dieser soll auf `Uebungsverzeichnis__2525__` verweisen.
 - Woran erkennen Sie im Datei-Listing (mittels `ls -l`), ob es sich um einen symbolischen Link (Softlink) handelt?
 - Testen Sie, was passiert, wenn Sie nun `Uebungsverzeichnis__2525__` komplett entfernen (*mit welchem Befehl?!*)
 - Ist der Link noch vorhanden (*Listing machen!?*)
 - Funktioniert er?
 - Was passiert, wenn Sie versuchen, einen symbolischen Link auf eine nicht-existente Datei oder ein nicht-existentes Verzeichnis zu erstellen (*ausprobieren – geben Sie die Befehlszeilen an!?*)
 - Erzeugen Sie nun eine Datei samt Inhalt namens `Datei25` mit der Befehlszeile
`echo Inhalt für Datei25 > Datei25`
 - Jetzt wollen wir (normale) Verzeichniseinträge (Namen) für Dateien erzeugen, sogenannte *Hardlinks*: Erstellen Sie einen *Hardlink* von `Datei25` namens `Datei99`!
 - Jede Datei unter Linux hat ein sogenannte *Inode-Nummer*, die diese eindeutig identifiziert (unabhängig von dem Namen/Ort der Datei).
 - Betrachten Sie die *Inode*-Nummern von `Datei25` und `Datei99` (mittels `ls -li`) – was fällt Ihnen auf?
 - Was bedeutet die Zahl „2“ in der 3. Spalte des Listings mit `ls -li`?

i. Verändern Sie Datei99 mit

`echo` Neuer Dateiinhalt für Datei99 > Datei99

Betrachten Sie nun den Inhalt von Datei25. Was ist passiert?

j. Löschen Sie die ursprüngliche Datei25 und betrachten Sie erneut den Inhalt von Datei99: Welchen “Unterschied im Verhalten” (beim verbleibenden Hardlink) stellen Sie gegenüber der Verwendung eines Softlinks (symbolischer Link – siehe zuvor) fest?

k. Versuchen Sie, einen *Hardlink* für ein *Verzeichnis* (etwa ~/Uebungsverzeichnis__2525__) zu erstellen – was stellen Sie fest?

(25) Wägen Sie ab bzw. recherchieren Sie: Was sind die Vorteile bzw. Nachteile von Softlinks (symbolischen Links) gegenüber Hardlinks (normalen Verzeichniseinträgen)

Viel Spaß!