

ARBEITSBLATT – LÖSUNG

BETRIEBSSYSTEME – LINUX SHELLSCRIPTS

Partnerarbeit:

(Lest euch die Aufgaben gemeinsam gut durch und bearbeitet innerhalb von Linux folgende Fragestellungen)

1. Arbeiten mit Prozessen:

1.1. Lies die Manual-Page des Befehls **top** und finde heraus:

- Was macht der Befehl? **Er ist quasi der Task-Manager von Linux**
- Welche Informationen werden angezeigt? **PID, command program, %CPU, %MEM, ...**
- Wie kann man die Prozesse nach Speicherverbrauch in Prozent/nach CPU-Verbrauch in Prozent sortieren?

Nach Speicherverbrauch: **top O N Enter**

Nach CPU-Verbrauch: **top O K Enter**

1.2. Lies die Manual-Page des Befehls **ps** und finde heraus:

- Wie kannst Du Dir alle Prozesse anzeigen lassen (nicht nur deine)?
Befehle: ps -a (nur meine) ps -ax (alle)
- Wie kannst Du Dir alle Prozesse mit dem Namen **getty** anzeigen lassen?
Befehle: ps ax|grep getty
- Wie kannst Du Dir mit ASCII-Art die Prozesshierarchie anzeigen lassen?
Befehle: ps -ax -f pstree pstree -A

1.3. Schreibe ein Skript **runforever**, das nichts Bestimmtes macht, aber für immer weiterläuft (die Befehle true bzw. false eignen sich hervorragend zum Lösen einer solchen Aufgabe). Starte anschließend **runforever** im Vordergrund, halte die Ausführung an, und lass es dann im Hintergrund weiterlaufen.

```
Script Inhalt: #!/bin/bash
while true; do
mkdir run4ever
rmdir run4ever
done
sleep 30
done
```

Befehle: sh runforever

STRG-Z (anhalten)

sh runforever &
[1] Bckgr.Prc. 17223 PID

1.4. Finde die PID (Prozess-ID) des vorher gestarteten **runforever** Prozesses und lokalisiere es in der Ausgabe von **ps tree**.

Befehle: ps PID [1]+ 17223 Ausgabe: ganz unten: [1]+ Exit 127 sh runforever

2. Shell-Programmierung:

2.1. Rufe das Skript **linuxscript1** auf.

Befehle: cd ~/Desktop sh linuxscript1

Task 1 von 2 - Erstelle jetzt die Verzeichnisse in ~/linux1 ...

Task 2 von 2 - Erstelle jetzt die Dateien in ~/linux1 ...

Jetzt kann's losgehen!

Es erstellt ein Verzeichnis `~/linux1` und darin einen Verzeichnisbaum. Wechsle in dieses Verzeichnis und finde in dem erstellten Verzeichnisbaum alle Dateien des Dateityps `*.txt`, die den Text `ba` beinhalten.

Befehle: `cd ~/linux1`
`find ~/linux1 -type f -name *.txt | grep ba`
find = Suchprogramm
/linux1 = das Verzeichnis
-type f = durchsucht Dateien
-name = durchsucht Dateien mit einem bestimmten Namen
grep = Sucht nach einem bestimmten Suchwort

Führe dieselbe Aufgabe für alle Dateien, die einen String von der Form „Buchstabe gefolgt von einem a“ enthalten.

Befehle: `find . | grep '.a'`

- 2.2. Implementiere das Suchverfahren aus 2.1. als ein Shellsript `findintextfiles`: Ausgehend vom aktuellen Verzeichnis, soll dieses alle Dateien mit der Endung `*.txt` nach dem als Argument übergebenen Ausdruck durchsuchen.

Shell-Script in `~/linux1` platzieren

Inhalt: `#!/bin/bash`
`echo -n "Geben Sie eine oder zwei Position(en) vor a ein:";`
`while read V1;`
`do`
`echo "Var1 hat Wert $V1";`
`find ~/linux1 -type f -name *.txt | grep $V1`
`done`

- 2.3. Rufe das Skript `linuxscript2` auf. Es erstellt ein Verzeichnis `~/linux2` und darin einige Dateien mit dem Namen `linux2file_*`.

Befehle: `cd ~/Desktop` `sh linuxscript2`
Erstelle die benötigten Dateien...
Jetzt kann's losgehen!

Wechsle in dieses Verzeichnis

Befehle: `cd ~/linux2`

und kopiere mit einem neuen Script `filemove` diese Dateien in das Unterverzeichnis `target`, wobei sie neue Namen erhalten sollen: statt `linux2file` soll nun `unix2datei` stehen (z.B. `linux2file_13` wird zu `unix2datei_13`).

Befehle: `mkdir target`

Folgende Zeilen einzeln tippen oder Script schreiben daraus:

```
find . -name "linux2file_*" -type f -exec cp "{}" ~/linux2/target \;  
oder  
cp linux2file_* ~/linux2/target  
cd ~/linux2/target  
rename linux2file_ unix2datei_ linux2file_*
```