

Aufgabenblatt 4

Systemprogrammierung (WiSe 2018/19)

Institut: Beuth Hochschule für Technik Berlin
Dozent: Prof. Dr. Christian Forler
Url: <https://lms.beuth-hochschule.de/>
Email: cforler@beuth-hochschule.de

- Die Skripte sind mit dem Editor `vim` zu erstellen.
- Die Verwendung von anderen Programmiersprachen wie `awk` ist nicht erlaubt.
- Versuchen Sie die Aufgaben mit Hilfe der Vorlesungsunterlagen zu lösen (und nicht Lösungen von Stackoverflow zu kopieren).

Aufgabe 1 (4 Punkte) Interaktives Backup

Schreiben Sie ein Shell-Skript, welche Dateien wie im Folgenden beschrieben kopiert. Als Kommandozeilenparameter soll ein Postfix (Dateiendung) übergeben werden. Für jede Datei aus den aktuellen Verzeichnes bei denen die angegebene Dateiendung passt soll der Benutzer gefragt werden, ob die Datei gesichert werden soll. Bei einer Bejahung soll die Datei in das Verzeichnis `backup` kopiert werden. Das Verzeichnis soll angelegt werden, falls nicht existent.

Aufgabe 2 (4 Punkte) Rangliste

Schreiben Sie ein Shell-Skript `rls.sh` mit der Syntax `rls.sh <min> <max>`. Das Skript soll alle Dateien im aktuellen Verzeichnis auflistet, deren Namen eine Länge von mindestens `<min>` Zeichen UND weniger als `<max>` Zeichen hat.

Geben Sie eine Usage-Meldung auf `stderr` aus und beenden Sie das Skript mit dem Rückgabewert 1, falls das Programm das Programm nicht sachgemäß aufgerufen wurde.

Aufgabe 3 (4 Punkte) Präfixmove

Schreiben Sie ein Shell-Skript `premv.sh` `<old-prefix>` `<new-prefix>` welches bei allen Dateinamen im aktuellen Verzeichnis die mit Präfix `<old-prefix>` beginnen durch den Präfix `<new-prefix>` ersetzt.

Beispiel

```
$ ls
foo.dat foobar premv.sh test.txt
$ ./premv.sh foo zzz
$ ls
premv.sh test.txt zzz.dat zzzbar
```

Aufgabe 4 (4 Punkte) Password Recovery

Schreiben Sie ein Bash-Skript welches als ersten Parameter einer Wörterliste W (Textdatei mit einem Wort pro Zeile) und als zweiten Parameter eine Liste von gehashten Passwörtern H übergeben bekommt. Das Programm soll für jedes Wort $w \in W$ testen ob der zugehörige MD5-Hash in H enthalten ist. Falls ja soll, das Skript angeben wie oft das Passwort w verwendet wurde. Die Ausgabe der Passwörter soll nach Häufigkeiten sortiert sein.

Verwenden Sie zum Bearbeiten der Aufgabe die Datei `battlefield-hashes.md5` (siehe Moodle). Es handelt sich hierbei um die MD5 Passworthashes von über 500.000 Battlefield-Spielern. :-)

Beispiel

```
$ cat word.list
12345678
password
$ ./md5_password_recovery word.list battlefield-hashes.md5
hits,password,MD5 hash
710,"password",5f4dcc3b5aa765d61d8327deb882cf99
302,"12345678",25d55ad283aa400af464c76d713c07ad
```

Hinweis: Der MD5-Hash eines Passworters lässt sich mit dem Programm `md5sum` berechnen.

Aufgabe 5 (4 Punkte) Nicht ausführbare Dateien

Schreiben Sie ein Bash-Skript `nxls.sh` `<directory>` welches in dem übergebenen Verzeichnis `directory` nur regulärer Dateien anzeigt, welche nicht ausführbar sind.

Aufgabe 6 (4 Punkte) Taschenrechner

Schreiben Sie ein Bash-Skript `calc.sh` `<int>` `<operator>` `<int>` welches als Taschenrechner fungieren soll. Es sollen die folgenden Operatoren unterstützt werden: `MULT`, `ADD`, `SUB`, `DIV`, `MOD` und `EXP`. Verwenden Sie die Case-Anweisung.

Beispielaufrufe

```
$ ./calc.sh 10 ADD 3
13
$ ./calc.sh 10 EXP 3
1000
$ ./calc.sh 10 MULT 3
30
```