2.6.A Skripte/Schleifen – Arbeitsblatt

Aufgabe 1: Mehrere Jahreskalender in einem Befehl

Der Befehl `cal` kann leider immer nur einen Monat oder ein Jahr ausgeben. Sie aber möchten in einem einzigen Aufruf mehrere Jahre ausgeben.

a) `calvonbis1.sh`
Dieses Skript soll alle Jahreskalender von 2030 bis 2040 nacheinander ausgeben.

b) `calvonbis2.sh`
Der erste Parameter soll die erste, der zweite Parameter die letzte Jahreszahl sein. Der Aufruf `cal 2050 2052` soll die Jahre 2050, 2051 und 2052 ausgeben.

c) `calvonbis3.sh`
Jedes Jahr, das als Parameter angegeben ist, soll ausgegeben werden. Der Aufruf `cal 2060 2070 2080` soll die Jahre 2060, 2070 und 2080 ausgeben.

Aufgabe 2: Umwandeln mehrerer Dateien

Dieses Skript soll jede als Parameter angegebene Text-Datei mit dem Namen `x.txt` nach `x.html` kopieren. Die Kopie soll einen HTML-Vorspann und einen HTML-Nachspann bekommen, damit sie in einem Browser genauso aussieht wie das Original in einem Texteditor. Der Vorspann sieht so aus:

```
1 <html><head></head><body><pre>
```

Der Nachspann sieht so aus:

```
1 </pre></body></html>
```

a) Ergebnis: `txttohtml1.sh`
Realisieren Sie dieses Skript mit einer kopfgesteuerten Schleife!

b) Ergebnis: `txttohtml2.sh`
Realisieren Sie dieses Skript mit einer Zählschleife!

Aufgabe 3: Erweiterung des Start-Skriptes

Erweitern Sie das Start-Skript des vorherigen Arbeitsblattes, so dass mehrere Dateien mit einem Befehl geöffnet werden können! Hinweis: Zum gleichzeitigen Öffnen ist es nötig, die Programme im Hintergrund zu starten.

a) Ergebnis: `mstart1`
Realisieren Sie dieses Skript mit einer kopfgesteuerten Schleife!

b) Ergebnis: `mstart2`
Realisieren Sie dieses Skript mit einer Zählschleife!

Aufgabe 4: Schleife mit Parameter aus einer Datei

Das Skript `persinfo` soll für alle Nutzer, deren Name in der Datei `persinfo.txt` steht (ein Name je Zeile), mit Hilfe der Programme `who`, `w` und der Dateien `/etc/group` und `/etc/passwd` möglichst viele Informationen herausfinden und anzeigen.

a) Ergebnis: `persinfo`
Realisieren Sie das Skript mit Hilfe einer Zählschleife!

Aufgabe 5: Überall-Befehl

Schreiben Sie ein Skript `ueberall`, das einen Befehl in allen Verzeichnissen eines Teilbaums des Dateisystems ausführt. Es soll also in jedes Verzeichnis gewechselt werden, dort soll dann der Befehl ausgeführt werden.

Ein Beispiel: Die folgende Befehlszeile soll allen Backup-Dateien im Home-Verzeichnis des Benutzers (und in allen Unterverzeichnissen davon) ein neues Datum geben.

Terminal
```
schueler@debian964:~$ ueberall ~ touch '*.bak'
```

Ein weiteres Beispiel: Die nun folgende Befehlszeile soll den berühmten Befehl `make` im Verzeichnis `/usr/local/source` und in allen Verzeichnissen darunter ausführen.

Terminal
```
schueler@debian964:~$ ueberall /usr/local/source make
```

`make` erwartet nämlich im jeweils aktuellen Verzeichnis eine Datei mit dem Name `Makefile`, in der dann weitere Arbeitsschritte beschrieben sind.

a) Ergebnis: `ueberall1`
Realisieren Sie dieses Skript mit einer kopfgesteuerten Schleife!

b) Ergebnis: `ueberall2`
Realisieren Sie dieses Skript mit einer Zählschleife!

Hinweis: Mit dem Befehl `pushd xyz` kann man in das Verzeichnis `xyz` wechseln, wobei das bisherige Verzeichnis zwischengespeichert wird. Mit `popd` kommt man wieder zurück auf das bisherige Verzeichnis. Mit mehreren Aufrufen von `pushd` und `popd` kann man so einen Verzeichnisstapel (LIFO) auf- und wieder abbauen.

Aufgabe 6: Erreichbarkeits-Seite

Das Skript `erreichbar.sh` soll regelmäßig (alle zwei Sekunden) abfragen, ob die Adresse 9.9.9.9 (=WAN) oder zumindest die Adresse 10.92.0.1 (=das LAN) erreichbar ist. Je nach Fall soll an der Stelle `/tmp/erreichbar.html` entweder eine Datei mit der Meldung „Router und Netz nicht erreichbar“ oder mit der Meldung „Router erreichbar, Netz nicht“ oder mit der Meldung „Netz erreichbar“ abgelegt werden.

a) `erreichbar1.sh`: Schreiben Sie das Skript! Die Meldungsdateien sollen nur Textformat haben. Rufen Sie das Skript auf und testen Sie das Ergebnis (w3m file:///tmp/erreichbar.html)!

b) `erreichbar2.sh`: In diesem Fall sollen die Meldungsdateien HTML-Format haben. Rufen Sie das Skript auf und testen Sie das Ergebnis (firefox file:///tmp/erreichbar.html)!

c) `erreichbar3.sh`: Nun sollen die Meldungsdateien so beschaffen sein, dass der Browser von selbst alle fünf Sekunden diese Datei neu lädt.