
1	2	3	4								Σ
---	---	---	---	--	--	--	--	--	--	--	----------

Die Bearbeitungszeit der Probeklausur ist 45 Minuten; für die richtige Klausur haben Sie 120 Minuten Zeit. Entsprechend hat diese Probeklausur reduzierten Umfang (37,5 %). Bitte bearbeiten Sie alle Aufgaben. Es sind insgesamt 45 Punkte zu erreichen (richtige Klausur: 120 Punkte).

Tipp: Lesen Sie zunächst alle Aufgaben durch und entscheiden Sie, welche Fragen Sie am leichtesten beantworten können; starten Sie dann mit diesen Aufgaben.

Viel Erfolg!

1. Dateisysteme

(17 / 45 Punkte)

a) Indirektion (2 P.)

Ein Unix-artiges Dateisystem verwendet fünffache Indirektion. Geben Sie den Faktor an, um den die maximale Dateigröße dadurch (grob) wächst – im Vergleich zu einem ansonsten gleichen Dateisystem, das ohne Indirektion arbeitet. (Die Lösung ist in der Form „XX-fache“ Größe anzugeben.) Begründen Sie Ihre Antwort.

b) Indirektion - Rechenbeispiel (6 P.)

Ein Unix-artiges Dateisystem arbeite mit folgenden Parametern:

- Größe des Dateisystems: 40 GByte
- Größe eines Datenblocks: 4 KByte
- Indirektion: 2-fach
- Im Inode: 5 direkte Verweise, 2 einfach indirekte, 2 zweifach indirekte Verweise

Berechnen Sie die nötige Größe einer Blockadresse (für die Speicherung in den Indirektionsblöcken), die Anzahl der Adressen pro Block und die maximale Dateigröße. (Die Lösung kann eine Summe von Zweierpotenzen 2^n sein.)

c) Linux-Tools (4 P.)

(i) Welche Funktion hat das Programm `fsck.ext3`? Durch welchen Mechanismus in welcher Konfigurationsdatei wird sein automatischer Aufruf beim Systemstart gesteuert?

(ii) Mit welchem Programm können Sie ein Ext3-Dateisystem erzeugen?

d) Partitionen und fdisk (5 P.)

(i) Bei Verwendung des Kommandos `n (new)` in `fdisk` erscheint folgende Rückfrage, welcher Partitionstyp erzeugt werden soll:

Command (m for help): `n`

Partition type

- `p` primary (1 primary, 1 extended, 2 free)
- `l` logical (numbered from 5)

Gibt es auf dieser Platte bereits a) primäre, b) erweiterte, c) logische Partitionen? (Die möglichen Antworten zu a, b, c sind jeweils „ja“, „nein“ oder „unbekannt“.)

(ii) Mit dem `fdisk`-Kommando `t` ändern Sie den Typ einer Partition. Was bewirkt eine solche Typ-Änderung?

2. Shell-Nutzung

(8 / 45 Punkte)

a) Betrachten Sie die folgenden Aussagen. Geben Sie zu jeder Aussage an, ob sie wahr oder falsch ist, und geben Sie eine kurze Begründung (in Stichworten).

- Alle in der Shell durch `VARNAME=wert` definierten Variablen sind auch in Kind-Shells sichtbar.
- Der Befehl `rm -r DIRNAME` löscht rekursiv das Verzeichnis `DIRNAME` (mit allen enthaltenen Dateien und Unterordnern).
- Mit dem Befehl `help PROGRAM` rufen Sie die Handbuchseite zu einem Programm auf.
- Mit dem Umleitungsoperator `2>` leiten Sie die Standardausgabe in eine Datei um.

3. Shell-Programmierung

(7 / 45 Punkte)

- a) Implementieren Sie eine For-Schleife (welche die Variable `i` von 3 bis 7 laufen lässt und im Rumpf der Schleife jeweils nur `echo $i` ausführt) auf drei unterschiedliche Arten. (3 P.)
- b) Welche Funktion hat der Befehl `continue` in einer Schleife? (1 P.)
- c) Der `find`-Befehl hat eine Option `-print0`, die dazu führt, dass bei der Ausgabe der Treffer das Zeichen mit ASCII-Wert 0 statt des Zeilenumbruchs als Trenner verwendet wird. Geben Sie ein Beispiel für eine sinnvolle Verwendung dieser Option und erläutern Sie, welchen Vorteil der Einsatz von `-print0` hat. (3 P.)

4. Shell-Programmierung: Beispielprogramm

(13 / 45 Punkte)

Betrachten Sie das folgende Shell-Skript `raetsel.sh`:

```
#!/bin/bash
i=$1; k=0
while true; do
    echo $i
    ((k += i))
    ((i += $2))
    [[ $i -gt $3 ]] && break
done
echo s=$k
```

- a) Simulieren Sie einen Skript-Lauf von `./raetsel 3 2 10`; notieren Sie alle Ausgaben, die das Programm beim Aufruf mit diesen Argumenten erzeugt. Geben Sie außerdem an, welchen Inhalt die Variable `$i` am Ende hat. (1 P.)
- b) Beschreiben Sie die Funktionalität des Skripts. (2 P.)
- c) Ersetzen Sie die Zeile mit `&&` durch einen besser verständlichen If-Then-Else-Block. (1 P.)
- d) Erstellen Sie eine angepasste Version des Skripts, welche `$1`, `$2` und `$3` in sinnvoll benannte Variablen einliest und dann nur noch diese Variablen verwendet. Durch den Einsatz der Variablen soll das Skript ohne weitere Kommentierung direkt verständlich sein. (3 P.)
- e) Das Skript funktioniert nur wie gewünscht, wenn der mittlere Parameter positiv ist. Geben Sie an, welches Problem auftritt, wenn Sie einen negativen Parameter verwenden. (2 P.)
- f) Lösen Sie das Problem aus Aufgabe e), indem Sie eine angepasste Version des Skripts bereitstellen, die auch mit negativen mittleren Parametern sinnvoll umgeht. (4 P.)