

## Übungsblatt 6

### Aufgabe 1 (Dateisysteme)

1. Welche Informationen speichert ein Inode?
2. Nennen Sie drei Beispiele für Metadaten im Dateisystem.
3. Was ist ein Cluster im Dateisystem?
4. Wie kann ein UNIX-Dateisystem (z.B. ext2/3), das keine Extents verwendet, mehr als 12 Cluster adressieren?
5. Wie sind Verzeichnisse bei Linux-Dateisystemen technisch realisiert?
6. Nennen Sie einen Vorteil und einen Nachteil kleiner Cluster im Dateisystem im Gegensatz zu großen Clustern.
7. Unterscheiden DOS/Windows-Dateisysteme Groß- und Kleinschreibung?  
 Ja       Nein
8. Unterscheiden UNIX-Dateisysteme Groß- und Kleinschreibung?  
 Ja       Nein
9. Moderne Betriebssysteme beschleunigen Zugriffe auf gespeicherte Daten mit einem Cache im Hauptspeicher.  
 Ja       Nein
10. Die meisten Betriebssystemen arbeiten nach dem Prinzip...  
 Write-Back       Write-Through
11. Nennen Sie je einen Vorteil und einen Nachteil eines Caches im Hauptspeicher, mit dem Betriebssysteme die Zugriffe auf gespeicherte Daten beschleunigen.
12. Was ist ein absoluter Pfadname?
13. Was ist ein relativer Pfadname?
14. `/var/log/messages` ist ein...  
 Absoluter Pfadname       Relativer Pfadname
15. `BTS_Vorlesung/Vorlesung_05/folien_bts_vorlesung_05.tex` ist ein...  
 Absoluter Pfadname       Relativer Pfadname

16. Dokumente/MasterThesis/thesis.tex ist ein...  
 Absoluter Pfadname       Relativer Pfadname
17. /home/<benutzername>/Mail/inbox/ ist ein...  
 Absoluter Pfadname       Relativer Pfadname
18. Welche Information speichert der Bootsektor eines Dateisystems?
19. Welche Information speichert der Superblock eines Dateisystems?
20. Warum fassen manche Dateisysteme (z.B. ext2/3) die Cluster des Dateisystems zu Blockgruppen zusammen?
21. Was ist die Dateizuordnungstabelle bzw. File Allocation Table (FAT) und welche Informationen enthält sie?
22. Was ist die Aufgabe des Journals bei Journaling-Dateisystemen?
23. Nennen Sie einen Vorteil von Journaling-Dateisystemen gegenüber Dateisystemen ohne Journal.
24. Welchen Vorteil hat der Einsatz von Extents gegenüber direkter Adressierung der Cluster?
25. Was macht das Defragmentieren?
26. Welche Art der Datenverarbeitung wird durch Defragmentieren maximal beschleunigt?
27. Wann ist Defragmentieren sinnvoll?

## Aufgabe 2 (Dateisysteme)

Kreuzen Sie bei jeder Aussage zu Dateisystemen an, ob die Aussage wahr oder falsch ist.

Aussage	wahr	falsch
Inodes speichern alle Verwaltungsdaten (Metadaten) der Dateien.		
Dateisysteme adressieren Cluster und nicht Blöcke des Datenträgers oder Laufwerks.		
Je kleiner die Cluster, desto größer ist der Verwaltungsaufwand für große Dateien.		
Je größer die Cluster, desto geringer ist der Kapazitätsverlust durch interne Fragmentierung.		
Unter UNIX haben Dateieindungen schon immer eine große Bedeutung.		
Moderne Dateisysteme arbeiten so effizient, dass Puffer durch das Betriebssystem nicht mehr üblich sind.		
Absolute Pfadnamen beschreiben den kompletten Pfad von der Wurzel bis zur Datei.		
Das Trennzeichen in Pfadangaben ist bei allen Betriebssystemen gleich.		
Ein Vorteil der Blockgruppen bei ext2 ist, dass die Inodes physisch nahe bei den Clustern liegen, die sie adressieren.		
Eine Dateizuordnungstabelle (FAT) erfasst die belegten und freien Cluster im Dateisystem.		
Bei der Master File Table von NTFS ist Fragmentierung unmöglich.		
Ein Journal im Dateisystem reduziert die Anzahl der Schreibzugriffe.		
Journaling-Dateisysteme grenzen die bei der Konsistenzprüfung zu überprüfenden Daten ein.		
Bei Dateisystemen mit Journal sind Datenverluste garantiert ausgeschlossen.		
Vollständiges Journaling führt alle Schreiboperation doppelt aus.		
Extents verursachen weniger Verwaltungsaufwand als Blockadressierung.		

## Aufgabe 3 (Mustervergleiche und Datenauswertung)

1. Nennen (oder beschreiben) Sie eine sinnvolle Anwendung für das Kommando `sed`.

2. Erzeugen Sie eine Datei `sedtest.txt` mit folgendem Inhalt:

```
Zeile 1  
Zeile 2  
Zeile 3  
Zeile 4  
Zeile 5  
Zeile 6
```

Fügen Sie mit `sed` 3 Leerzeichen am Anfang jeder Zeile ein (Einrückung).

3. Geben Sie mit `sed` die Zeilen 2 bis 5 der Datei `sedtest.txt` aus.

4. Löschen Sie mit `sed` jede 2. Zeile der Datei `sedtest.txt`.

5. Erzeugen Sie eine Datei `htmlcode.html` mit folgendem Inhalt:

```
<a href="BTSSS2016/index.html">Betriebssysteme (BTS)</a><p>  
<b>Das ist eine <i>HTML-Datei</i></b><br>  
<h2>Eine Überschrift</h2>
```

Entfernen Sie mit `sed` alle HTML-Tags aus der Datei `htmlcode.html`.

6. Erzeugen Sie eine Datei `umlaute.txt` mit folgendem Inhalt:

```
Bäume, Äpfel, Bücher, Übertreibung  
Töpfe, Öffentlichkeit, Straße, Spaß
```

Ändern Sie mit `sed` alle Umlaute aus der Datei `umlaute.txt` in „ae“, „oe“, „ue“, „Ae“, „Oe“, „Ue“ und „ss“.

7. Erzeugen Sie eine Datei `bundesliga_08_0405.txt` mit den Ergebnissen des 8. Spieltags der Saison 2004/2005:

```
Schalke      - Bochum      3 : 2 61500 Zuschauer  
Bielefeld    - Stuttgart   0 : 2 22700 Zuschauer  
Dortmund     - Nürnberg    2 : 2 73500 Zuschauer  
Leverkusen   - Hamburg     3 : 0 22500 Zuschauer  
Freiburg     - Mainz       1 : 2 24000 Zuschauer  
Kaiserslautern - Berlin     0 : 2 30500 Zuschauer  
Wolfsburg    - Mönchengladbach 2 : 1 26500 Zuschauer  
Rostock      - Hannover    1 : 3 16500 Zuschauer  
Bremen       - München     1 : 2 42000 Zuschauer
```

8. Nennen (oder beschreiben) Sie eine sinnvolle Anwendung für das Kommando `awk`.

9. Ermitteln Sie mit `awk` alle Spiele, bei denen mehr als 35000 Zuschauer waren.

10. Ermitteln Sie mit `awk` alle Spiele, bei denen weniger als 50000 Zuschauer waren und bei denen es einen Sieg der Heimmannschaft gab.
11. Ermitteln Sie mit `awk` für jedes Spiel die Summe der gefallen Tore.
12. Ermitteln Sie mit `awk` in welcher Stadt die meisten Zuschauer waren und geben das Ergebnis wie folgt aus:

Die meisten Zuschauer waren in STADT (ANZAHL).