

KV Betriebssysteme	Übung #1	SS 2005
Name:	Matr-Nr:	Gruppe:
Abgabe: 05.04.2005		Tutor:

Beispiel 1: Defragmentierung

Schreiben Sie in Java ein Programm, welches die Fragmentierung und Defragmentierung einer FAT (File Allocation Table) simuliert.

Erforderliche Benutzereingaben:

- Anzahl der verfügbaren Cluster (von Sektoren wird abstrahiert)
- Belegungsfaktor (Eingabe: 70% → 70% aller Cluster sind belegt)
- Durchschnittliche Anzahl von Cluster, die von einer einzelnen „Datei“ belegt werden sollen.
- Mindestgrad der Fragmentierung. Überlegen Sie sich einen Algorithmus, mit welchem Sie eine Maßzahl für Fragmentierung einer FAT berechnen können.

Vorgehensweise für Fragmentierung

1. Erstellen und Initialisieren der leeren FAT
2. Initialisieren des Inhaltsverzeichnisses, welches die Zeiger auf die Starteinträge enthält.
3. „Beispieldateien“ erstellen (Einträge im Inhaltsverzeichnis erstellen, Zeiger auf Starteinträge setzen, FAT Einträge erstellen).
4. Fragmentierung durch Löschen von Einträgen.
5. Schritt 3 und 4 wiederholen, bis erwünschte Belegungs- und Fragmentierungsfaktoren erreicht sind.

Visualisieren Sie den schrittweisen Fragmentierungsprozess!

(8 Punkte)

Defragmentierung

Überlegen Sie sich einen Algorithmus für die Defragmentierung. Dokumentieren, implementieren und visualisieren Sie diesen. Achten Sie hierbei auf folgende Punkte und dokumentieren Sie diese:

- Achten Sie auf Sicherheit. Sollte der Defragmentierungsprozess plötzlich abbrechen (Stromausfall, etc.), sollten keine Daten verloren gehen. Wie stellen Sie dies sicher?
- Achten Sie auf Effizienz der Implementierung. Geben Sie die Laufzeitkomplexität an (Anzahl Durchläufe, Anzahl Tauschoperationen) und spezifizieren Sie jeweils den Bezugspunkt (Anzahl fragmentierte Cluster, Anzahl Cluster gesamt, ...).

(12 Punkte)

Tests

Geben Sie mindestens 3 verschiedene, dokumentierte Testfälle ab (inklusive aller notwendigen Eingaben / Screenshots / Ausgaben). Nehmen Sie als Anzahl der verfügbaren Cluster 30, 100 und 300 an und als Belegungsfaktor 80%, 50% und 90%. Arbeitet Ihr Algorithmus auch bei einer 99%igen oder 100%igen Belegung?

(4 Punkte)

Bonus

Wenn Sie sich für eine graphische, interaktive Visualisierung mit Swing entscheiden, erhalten Sie 4 Bonus-Punkte.

Hinweise

- Dokumentieren Sie den Sourcecode.
- Dokumentieren Sie Ihre Überlegungen für die einzelnen Algorithmen separat in einem Textdokument.
- Geben Sie Ihre Übungsausarbeitung inkl. Programmcode sowohl ausgedruckt als auch in WeLearn ab:

<http://welearn5.fim.uni-linz.ac.at>

Die WeLearn-Abgabe muss zusätzlich auch die compilierte Klassen enthalten. Abgegeben werden muss eine ZIP-Datei, die alle Dateien enthält. Der Name dieser Datei muss der Matrikelnummer entsprechen, z. B. 9812345.zip

Der Abgabeordner ist `/Courses/KV Betriebssysteme SS2005/Abgaben/Übung 1`

WeLearn Accounts:

Bei allen denjenigen, die schon aus früheren LVAs einen Account auf WeLearn 5 hatten, bleibt das Passwort unverändert. Für alle, die noch keinen Account auf WeLearn 5 hatten, wurde ein neuer Account mit der jeweiligen Matrikelnummer als Benutzernamen und Passwort eingerichtet. Sollten Sie ein Problem mit Ihrem Account haben, schreiben Sie bitte an Fr. Dr. Loidl (loidl@fim.uni-linz.ac.at).