

Praktikum Digitale Kodier- und Kompressionsverfahren

DKK-00: Vorbereitungsversuch

1 Ziel

In diesem *freiwilligen* Versuch richten Sie Ihre Entwicklungsumgebung in und verarbeiten Binärdaten aus beliebigen Dateien. Damit werden Arbeitsschritte vorbereitet und geübt, welche Sie auch im weiteren Verlauf des Praktikums benötigen.



Die Umsetzung kann grundsätzlich mit einer beliebigen Programmiersprache und auf einem beliebigen Betriebssystem erfolgen. In diesem Praktikum wird die Nutzung von C++ empfohlen.

2 Durchführung

2.1 Implementierung Zeichenzähler

Erstellen Sie ein Programm, das die Häufigkeit des Auftretens aller 8-Bit-Zeichen in beliebigen Dateien ermittelt, mit folgenden Eigenschaften:

1. Das Programm soll möglichst von der Kommandozeile (Terminal) Ihres Betriebssystems aus bedienbar sein, und zwei Argumente erwarten: den Dateinamen der Quelldatei und den Dateinamen einer Zieldatei, welche in 256 Zeilen die Häufigkeit des jeweiligen Zeichens enthält (Textdateiformat, wie CSV-Format, nur eine Spalte). Beispiel zur Bedienung, falls Ihr Programm den Namen `count` trägt:
`count Faust.txt Faust_count.csv`
Die Datei `Faust.txt` ist Teil des Begleitmaterials zu diesem Versuch.
2. Die Quelldatei soll byte-weise eingelesen werden (Alternative: im Ganzen in einen Zwischenspeicher einlesen und diesen Puffer byte-weise verarbeiten).
3. Das Programmsoll mit beliebigen Dateien umgehen können. Es dürfen also beispielsweise keine Annahmen getroffen werden, dass in der Quelldatei bestimmte Zeichen nicht enthalten sind.



4. Als kleine Validierungsmaßnahme soll das Programm am Ende überprüfen, ob die Summe aller Zeichenhäufigkeiten auch der Größe der Quelldatei (in Bytes) entspricht.
5. Fehler, wie z.B. eine nicht existierende Quelldatei, sollen abgefangen und dem Benutzer mitgeteilt werden.

2.2 Implementierung Programm zur Bitinvertierung

Erstellen Sie ein Programm, das alle Bits in einer Datei invertiert, mit folgenden Eigenschaften:

1. Das Programm soll möglichst von der Kommandozeile (Terminal) Ihres Betriebssystems aus bedienbar sein, und zwei Argumente erwarten: den Dateinamen der Quelldatei und den Dateinamen einer Zieldatei, in welcher die invertierten Daten gespeichert werden sollen.
2. Die Verarbeitung (Inversion) soll byte-weise erfolgen.
3. Fehler, wie z.B. eine nicht existierende Quelldatei, sollen abgefangen und dem Benutzer mitgeteilt werden.

2.3 Tests und Vergleich

Wenden Sie die beiden Programme auf alle Testdateien an, die in dem versuchsbegleitenden ZIP-Archiv `DKK_Testdateien.zip` enthalten sind. Die korrekte Funktionsweise des Invertierungsprogramms können Sie selbst überprüfen, indem Sie das Resultat erneut invertieren lassen. Eine doppelt invertierte Datei muss wieder dem Original entsprechen.

3 Protokoll

Für diesen freiwilligen Vorbereitungsversuch muss kein Protokoll angefertigt werden.

Viel Erfolg bei der Bearbeitung!