HA 1	Fachgebiet Architektur eingebetteter Systeme **Rechnerorganisation** Hausaufgabe MIPS-Assembler	AES Embedded Systems Architecture WS2015/16

Ausgabe: 16.12.2016 Abgabe: 20.01.2017

Hausaufgabe: Minecraft-Texturen

In dieser Hausaufgabe sollen Sie ein MIPS-Assembler-Programm entwickeln, das die Auflösung und die Farbtiefe von digitalen Bildern reduziert.

Hinweise:

- Bitte machen Sie den Lösungsweg (z.B. durch Kommentare) aller Aufgaben deutlich nachvollziehbar. Ein fehlender Lösungsweg führt zu Punktabzug.
- Nutzen Sie die (Qt)SPIM-Dokumentation unter: `http://spimsimulator.sourceforge.net/further.html` Beachten Sie vorallem das Dokument *Appendix A: Assemblers, Linkers, and the SPIM Simulator*.
- Ihre Abgabe muss mit der QtSPIM-Installation auf den TU-Berlin-Rechnerpools (z.B. TEL 106 und TEL 206) lauffähig sein. Bitte testen Sie Ihre Abgabe dahingehend.
- Bitte beachten Sie die Registerkonventionen!

1. Bilder laden und speichern (7 Punkte)

Ein Bild kann im Speicher als Array abgelegt werden, indem die einzelnen Zeilen mit Bildpunkten kontinuierlich hintereinander gespeichert werden. Am Anfang des Bildes befindet sich außerdem ein Header, der Breite, Höhe und Farbtiefe, also die Anzahl der Farben, enthält.

Ihr Programm soll Bild-Dateien im PGM-Format (binär, P5) lesen und schreiben können. Beispielsweise beschreibt die Zeichenkette `"P5␣2␣2␣255␣\00\00\00\00"` ein Graustufenbild im Binärformat mit 2×2 schwarzen Pixeln und `"\nn"` beschreibt ein Byte mit dem hexadezimalen Wert $0xnn$.

Sie finden Informationen zum PGM-Format unter `https://de.wikipedia.org/wiki/Portable_Anymap`. Um Dateien zu lesen und zu schreiben bietet QtSPIM, ähnlich wie ein Betriebssystem, verschiedene Syscalls. Sie müssen aus einem String von Ziffern den Wert der Zahl berechnen. Dazu benötigen Sie eine ASCII-Tabelle.

1. Implementieren Sie eine Routine `load_img`, die ein Bild lädt und den Header verarbeitet.
2. Implementieren Sie außerdem eine Routine `store_img`, die als Argumente Höhe und Breite des Bildes sowie einen Zeiger auf die Nutzdaten erhält und damit eine Bilddatei erzeugt.

2. Verringern der Bildauflösung (7 Punkte)

In dieser Aufgabe soll die Auflösung eines Bildes verringert werden.

Da sich die Auflösung in jeder Dimension jeweils um den Faktor zwei ändert, können Sie jeweils aus 4 benachbarten Pixeln einen neuen Pixel errechnen. Beispielsweise kann aus dem Bild a mit einer

Auflösung von 4×4 Pixeln das Bild a' mit einer Auflösung von 2×2 Pixeln berechnet werden.

$$a = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}, a' = \begin{pmatrix} :: & :: \\ :: & :: \end{pmatrix}$$

Bedenken Sie, dass sich mit der Auflösung auch die Farbwerte ändern müssen.

1. Implementieren Sie eine Routine `interpolate2`, die die Auflösung eines Bildes um den Faktor 2 verkleinert. Die Routine erhält als Argumente die Speicheradresse sowie Höhe und Breite des Bildes.
2. Implementieren Sie eine Routine `interpolate`, die die Auflösung eines Bildes um einen Faktor 2^n verringert.

3. Verringern der Farbtiefe (3 Punkte)

In dieser Aufgabe soll die Farbtiefe eines Bildes verringert werden. Die gegebenen Bilder haben eine Farbtiefe von 8 Bit. Die Verringerung der Farbtiefe soll ebenfalls in 2er-Potenzen (8 Bit, 4 Bit, 2 Bit, 1 Bit) möglich sein. Das bedeutet, dass statt 2^8 z.B. nur noch 2^4 Graustufen zur Verfügung stehen.

Implementieren Sie eine Routine `quantize`, die als Argumente die Speicheradresse des Bildes, die Breite/Höhe des Bildes und einen Quantisierungsfaktor erhält.

4. Betrachtungsaufgabe (3 Punkte)

In dieser Aufgabe sollen die ein Vergleich der Ergebnisse von Verringerung der Auflösung und der Verringerung der Farbtiefe gezogen werden. Wenn bei einem Bild sowohl die Auflösung als auch die Farbtiefe verringern werden soll, dann gibt es zwei Möglichkeiten: entweder verringern Sie zuerst die Auflösung und danach die Farbtiefe oder andersherum.

1. Probieren Sie beide Möglichkeiten aus und berichten Sie Ihre Ergebnisse.
2. Kommt es zu Unterschieden? Warum? Begründen Sie Ihre Beobachtungen anhand Ihrer Implementierung.