

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

Praktikum Rechnerarchitektur

Entropie (A406)

Projektaufgabe – Aufgabenbereich Theorie der Informationsverarbeitung

1 Organisatorisches

Auf den folgenden Seiten finden Sie die Aufgabenstellung zu Ihrer Projektaufgabe für das Praktikum. Die Rahmenbedingungen für die Bearbeitung werden in der Praktikumsordnung festgesetzt, die Sie über die Praktikumshomepage¹ aufrufen können.

Wie in der Praktikumsordnung beschrieben, sind die Aufgaben relativ offen gestellt. Besprechen Sie diese innerhalb Ihrer Gruppe und konkretisieren Sie die Aufgabenstellung. Die Teile der Aufgabe, in denen Assembler-Code anzufertigen ist, sind für die 64-Bit x86-Architektur (x86-64) unter Verwendung der SSE-Erweiterungen zu schreiben; alle anderen Bestandteile der Hauptimplementierung sind in C nach dem C11-Standard anzufertigen.

Der **Abgabetermin** ist der **16. Juli 2021, 11:59 Uhr (MESZ)**. Die Abgabe erfolgt per Git in das für Ihre Gruppe eingerichtete Projektrepository. Bitte beachten Sie die in der README.md angegebene Liste von abzugebenden Dateien.

Die **Abschlusspräsentationen** finden in der Zeit vom **09.08.2021 – 03.09.2021** statt. Weitere Informationen werden noch bekannt gegeben. Beachten Sie, dass die Folien für die Präsentation am obigen Abgabetermin im PDF-Format abzugeben sind.

Sofern die Rahmenbedingungen (Hygiene-, Abstandsregeln etc.) für eine Präsenzprüfung nicht vorliegen, kann gemäß §13a APSO die geplante Prüfungsform auf eine virtuelle Präsentation (Videokonferenz) oder eine kurze schriftliche Fernprüfung umgestellt werden. Die Entscheidung über diesen Wechsel wird möglichst zeitnah, spätestens jedoch 14 Tage vor dem Prüfungstermin nach Abstimmung mit dem zuständigen Prüfungsausschuss bekannt gegeben.

Bei Fragen/Unklarheiten in Bezug auf den Ablauf und die Aufgabenstellung wenden Sie sich bitte an Ihren Tutor.

Wir wünschen Ihnen viel Erfolg und Freude bei der Bearbeitung Ihrer Aufgabe!

Mit freundlichen Grüßen
Die Praktikumsleitung

¹<https://gepasp.in.tum.de/eraweb/>

2 Entropie

2.1 Überblick

Die Informationstheorie ist ein Feld der Mathematik, in welchem man sich allgemein mit dem Kodieren von Nachrichten aufgrund von statistischen Gegebenheiten beschäftigt. Sie werden in Ihrer Projektaufgabe einen bestimmten Teilbereich näher beleuchten und einen bestimmten Algorithmus in Assembler implementieren.

2.2 Funktionsweise

In der Informationsverarbeitung ist die Entropie H ein Maß für den mittleren Informationsgehalt einer Nachricht. Für eine Wahrscheinlichkeitsverteilung einer Zufallsvariablen X lässt sich die Entropie berechnen als:

$$H(X) = \sum_{x \in X} -P(X = x) \cdot \log_2(P(X = x)) \quad (1)$$

Sie erhalten eine Eingabedatei mit Werten einer Wahrscheinlichkeitsverteilung mit einer unbestimmten Anzahl an Fließkommazahlen, welche durch Leerzeichen getrennt sind. Ihre Aufgabe ist es, die Entropie der Verteilung zu bestimmen.

2.3 Aufgabenstellungen

Ihre Aufgaben lassen sich in die Bereiche Konzeption (theoretisch) und Implementierung (praktisch) aufteilen. Sie können (müssen aber nicht) dies bei der Verteilung der Aufgaben innerhalb Ihrer Arbeitsgruppe ausnutzen. Antworten auf konzeptionelle Fragen sollten an den passenden Stellen in Ihrer Ausarbeitung in angemessenem Umfang erscheinen. Entscheiden Sie nach eigenem Ermessen, ob Sie im Rahmen Ihres Abschlussvortrags auch auf konzeptionelle Fragen eingehen. Die Antworten auf die Implementierungsaufgaben werden durch Ihrem Code reflektiert.

Wichtig: Sie dürfen im Assembler-Code nur solche arithmetischen Operationen verwenden, die grundlegende Berechnungen durchführen (im Zweifel: die vier Grundrechenarten), nicht jedoch Instruktionen, die komplexere Berechnungen durchführen (z.B. Wurzel, Logarithmus, Exponentiation).

2.3.1 Theoretischer Teil

- Man kann zeigen, dass für die Entropie die folgenden Grenzen gelten:

$$0 \leq H(X) \leq \log_2(|X|) \quad (2)$$

$|X|$ bezeichne dabei die Kardinalität von X . Bestimmen Sie, für welche Arten von Verteilungen die Gleichheitszeichen gelten.

- Welche Eigenschaften muss eine Menge von Zahlen erfüllen, um eine Wahrscheinlichkeitsverteilung zu sein?
- Leiten Sie mathematisch drei Möglichkeiten her, die Funktion zu berechnen. Verwenden Sie dazu sowohl eine reine Reihendarstellung als auch eine Methodik, die auch einen Tabellen-Lookup benutzt.
- Beschreiben Sie, wie der Logarithmus in der C-Standardbibliothek *glibc* implementiert ist.
- Untersuchen Sie Ihre fertige Implementierung auf Genauigkeit und Performanz. Vergleichen Sie Ihre beiden Verfahren und ziehen Sie als weiteren Vergleich eine C-Implementierung sowie eine Assembler-Implementierung unter Nutzung von komplexeren Instruktionen, die beispielsweise eine Wurzelberechnung durchführen, heran.
- Nutzen Sie Ihre Implementierung, um die Entropie des Zufallsgenerators der C-Standardbibliothek (`rand`) sowie des Betriebssystems (`/dev/urandom`) zu bestimmen. Generieren Sie dazu eine große Zahl von Zufallszahlen und messen Sie dessen Entropie. Welche Werte würden Sie erwarten? Welche Werte erhalten Sie?

2.3.2 Praktischer Teil

- Implementieren Sie im Rahmenprogramm I/O-Operationen in C, mit welchen Sie Ihrem Assemblerprogramm einen Pointer auf die Inhalte einer eingelesenen Datei übergeben können. Das Konvertieren der Eingabedaten in ein Array mit Fließkommazahlen kann in C implementiert werden.
- Implementieren Sie in der Datei mit dem Assemblercode für jedes beschriebene Verfahren jeweils eine Funktion mit der Signatur

```
float (size_t len, const float data[len])
```

welche die Entropie des Arrays von Single-Precision-Werten `data` berechnet und zurückgibt. Die Funktion soll außerdem überprüfen, ob die Werte des Eingabearrays die Eigenschaften einer Wahrscheinlichkeitsverteilung erfüllen. Der Parameter `len` gibt die Länge des Eingabearrays an.

2.4 Allgemeine Bewertungshinweise

Die folgende Liste soll Ihnen als Gedächtnisstütze beim Bearbeiten der Aufgaben dienen. Beachten Sie ebenfalls die in der Praktikumsordnung angegebenen Hinweise.

- Stellen Sie unbedingt sicher, dass *sowohl* Ihre Implementierung *als auch* Ihre Ausarbeitung auf der Referenzplattform des Praktikums (`1xhalle`) kompilieren und vollständig korrekt bzw. funktionsfähig sind.
-

- Die Implementierung soll mit GCC/GNU as kompilieren. Verwenden Sie keinen Inline-Assembler und keine x87-FPU- oder MMX-Instruktionen. Sie dürfen alle SSE-Erweiterungen bis SSE4.2 benutzen. Andere ISA-Erweiterungen (z.B. AVX, BMI1) dürfen Sie nur benutzen, sofern Ihre Implementierung auch auf Prozessoren ohne derartige Erweiterungen lauffähig ist.
 - Sie dürfen die angegebenen Funktionssignaturen (nur dann) ändern, wenn Sie dies (in Ihrer Ausarbeitung) begründen.
 - I/O-Operationen dürfen grundsätzlich in C implementiert werden.
 - Denken Sie daran, das Laufzeitverhalten Ihres Codes zu testen (Sichere Programmierung, Performanz) und behandeln Sie *alle möglichen Eingaben*, auch Randfälle. Ziehen Sie ggf. alternative Implementierungen als Vergleich heran.
 - Eingabedateien, welche Sie generieren, um Ihre Implementierungen zu testen, sollten mit abgegeben werden; größere Eingaben sollten stattdessen stark komprimiert oder (bevorzugt) über ein abgegebenes Skript generierbar sein.
 - Stellen Sie Performanz-Ergebnisse nach Möglichkeit grafisch dar.
 - Vermeiden Sie unscharfe Grafiken und Screenshots von Code.
 - Geben Sie die Folien für Ihre Abschlusspräsentation im PDF-Format ab. Achten Sie auf hinreichenden Kontrast (schwarzer Text auf weißem Grund!) und eine angemessene Schriftgröße. Verwenden Sie 16:9 als Folien-Format.
-