Freie Universität Berlin

Übung 03

Rechnerarchitektur
Wintersemester 21/22
Prof. Dr.-Ing. Jochen Schiller

Abgabe 19.11.2021, 10:15

Aufgabe 1: Zahlendarstellung und Rechnen

Führen Sie die folgenden vier Berechnungen in B+V Darstellung, im Einerkomplement, Zweierkomplement und in einer Exzessdarstellung durch. Gehen Sie von 8-Bit breiten Registern aus. Wählen Sie für die Exzessdarstellung einen sinnvollen Offset und begründen Sie Ihre Wahl.

Bitte geben Sie immer Ihren vollständigen Rechenweg an!

1. $23 + 81$
2. $36 - 14$
3. $72 - 87$
4. $-113 - 37$

Nennen Sie zusätzlich zwei Vorteile des Zweierkomplements gegenüber den anderen Darstellungen.

Aufgabe 2: Typkonvertierung

Implementieren Sie eine „String to Integer“ *oder* – als deutlich forderndere Aufgabe – eine „Integer to String“ Funktion. Diese sollen zwischen einer ASCII Zeichenkette der Form „`1234`“ mit einer gegebenen Basis (bspw. 10) und ihrer entsprechenden Integer Zahlendarstellung 1234 konvertieren.

Die Funktionen haben folgende Signatur:

```
int64_t strToInt(const char *str, uint8_t base);
size_t intToStr(int64_t val, uint8_t base, char *str,
                size_t len);
```

str Speicherbereich, der die Zeichenkette aus Zahlen vorhält / in den ASCII-Ziffern geschrieben werden
base Basis in der die Zeichenkette angegeben ist / werden soll
val Integer der umgewandelt werden soll
len Maximale Länge die der String haben darf

Rückgabewert: der passende Integerwert bzw. die Anzahl der geschriebenen Zeichen ohne `NUL`-Terminator.

Hinweis: Die Funktion `strToInt` *soll terminieren, sobald ein Zeichen erreicht wird, dass nicht umgewandelt werden kann. Bei angegebenen Basen* $b \leq 1$ *oder* $b > 36$ *soll die Funktion nichts tun. Sie dürfen davon ausgehen, dass der Eingabe-String nullterminiert ist. Sorgen Sie in jedem Fall dafür, dass der Ergebnis-String nullterminiert ist.*