

Bitte beachten Sie die allgemeinen Hinweise auf Übungszettel 1

Aufgabe 1: Diverses

Aufgabe 1.1: Carry-lookahead-Addierer

1. Lösen Sie die rekursive Berechnung von $ü_3$ dem Skript entsprechend in Und- und Oder-Verknüpfungen von g_i und p_i mit $i \leq 3$ auf.
2. Berechnen Sie g_i und p_i mit $i \leq 3$ für $a_{3..0} = 1010$ und $b_{3..0} = 0110$.
3. Berechnen Sie die ersten vier Ziffern s_i mit $i \leq 3$ der Summe von a und b aus b) entsprechend der Carry-lookahead-Addierer-Methode.

Aufgabe 1.2: Hacking

Gehen Sie davon aus, dass Sie Zugriff auf ein Array haben, das im Stack angelegt wurde.

1. Wie können Sie durch dieses Array die Rücksprungadresse der Funktion manipulieren?
2. Wie können Sie trotzdem noch die Kontrolle über das System erlangen wenn Sie nur den alten "rbp" Wert manipulieren können?
3. Würden Ihre Angriffe noch ohne Weiteres funktionieren, wenn der Stack nicht mehr *descending* sondern *ascending* ist?

Aufgabe 2: Funktionspointer

Implementieren Sie die folgenden Funktionen in Assembler:

- Die Funktion *fold* soll mit einer übergebenen Funktion alle Werte eines gegebenen Arrays miteinander verrechnen und das Ergebnis zurückgeben.

```
int64_t fold(int64_t (*func)(int64_t, int64_t), uint64_t len, int64_t a[len],  
            uint8_t dir);
```

Parameter:

func - Funktion mit der alle Einträge aus **a** verrechnet werden sollen

len - Länge von Array **a**

a - Array mit den zu verrechnenden Werten

dir - Richtung in die *fold* arbeiten soll (0 - von vorne nach hinten, 1 - von hinten nach vorne)

- Die Funktion *zipWith* soll aus zwei gegebenen Arrays jeweils die Elemente mit dem gleichen Index mittels einer übergebenen Funktion verrechnen und das Ergebnis an die gleiche Position in einem dritten Array schreiben.

```
void zipWith(int64_t (*func)(int64_t, int64_t), uint64_t len, int64_t a[len],  
            int64_t b[len], int64_t c[len]);
```

Parameter:

func - Funktion mit der die Einträge aus den Arrays **a** und **b** verrechnet werden sollen

len - Länge der Arrays

a - Array mit den ersten Operanden

b - Array mit den zweiten Operanden

c - Array für die Ergebnisse