

Assemblerprogrammierung

Aufgabe 1: Kontrollstrukturen

Beschreiben Sie in Textform, durch welche Befehlsfolgen die abgebildeten Konstrukte dargestellt werden können, und geben Sie anschließend entsprechenden MIPS-Assembler Befehle an:

a).	b)	c)
int i = 12; int j=0;	int a; int b; int c;	int sum = 0;
while(i > 0){	if(a ==b){	for(int i = 1; i<10; i++){
j += --i;	c = a*b;	sum += i + i/2;
}	} else {	}
	c = a - b;	
	}	
	Werte von „a“ und „b“	
	Stehen in \$a0 und \$a1	

Aufgabe 2: Kontrollstrukturen in der Anwendung

Schreiben Sie ein Programm in Assemblersprache, welches über die Zahlen von 1 bis 100 iteriert. Dabei soll die Summe aller geraden Zahlen in \$v0, die Summe aller ungeraden Zahlen in \$v1 abgelegt werden.

Aufgabe 3: Load / Store

An der Adresse 0x10000400 liege eine Liste mit Ganzzahlen ohne „0“. Das Ende der Liste sein durch ein Wort nur aus „Nullen“ gekennzeichnet.

Erstellen Sie ein Programm in Assemblersprache, welches die Summe aller Listenelemente bildet. Legen Sie das Ergebnis in \$v0 ab.

Aufgabe 4: Umsetzung von Arithmetik

Schreiben Sie ein Programm in Assemblersprache, welches die Primfaktoren zu einer in \$a0 liegenden Zahl berechnet. Speichern Sie die Primfaktoren an der Adresse 0x10000000 als Liste, wobei das erste Listenelement die Anzahl der Listenelemente (ohne sich selbst) angeben soll. Es sei anzunehmen, dass an der angegebenen Adresse genug Speicherplatz reserviert wurde.