

Assignment 6

Abgabe bis 11. Woche

Je nach Halbgruppe 15./16. Mai,

Achtung!

Jeder Studierende muss in der 11. Woche eine gelöste Aufgabe abgeben.

Anforderungen:

- Jede Aufgabe wird in einer eigenständigen Datei (.c) gelöst .
- Jede Halbgruppe lädt die Lösungen in das ihrer Halbgruppe entsprechende Assignment.
- Die Dateien werden nach folgendem Format benannt und hochgeladen:

Name_Vorname_Gruppe_X.c

wobei **X** die Nummer der Aufgabe ist, die Sie lösen müssen.

- Wer andere als die zugewiesenen Aufgaben löst, wird mit Note 0 bewertet.

Achtung! Die in dieser Assignment zugewiesene Aufgabe muss unbedingt mithilfe eines C-Programms gelöst werden.

Achtung! Lösungen, die nur einen Prozess verwenden, in dem die gesamte Anforderung gelöst wird, werden nicht berücksichtigt. (Mindestens 3 Prozesse werden zum Lösen verwendet.)

Zuordnung der Aufgaben: Die zur Lösung vorgeschlagenen Aufgaben werden in der Laborstunde der 9. Woche verteilt.

Aufgabe 1

Sei eine Datei, die die ersten N natürlichen Zahlen der Reihe nach enthält. Man definiert P_i als den Prozess, der alle Vielfachen von i im Bereich von 2 bis N entfernt. Schreiben Sie ein Programm, das alle Primzahlen zwischen 1 und N bestimmt. N wird vom Benutzer bereitgestellt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 2

Eine natürliche Zahl ist perfekt, wenn sie gleich der Summe ihrer Teiler (alle außer sich selbst) ist. Schreiben Sie ein Programm, das alle perfekten Zahlen kleiner als N bestimmt. N wird vom Benutzer bereitgestellt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 3

Sei eine Datei, die N ganze Zahlen enthält. Seien zwei Prozesse: ein Prozess bestimmt den Minimalwert und der zweite den Maximalwert aus einer Folge von Zahlen. Schreiben Sie ein Programm, das das k -te Element (wenn die Zeichenfolge in aufsteigender Reihenfolge sortiert wurde) aus einer Folge von ganzen Zahlen bestimmt. N und k werden vom Benutzer bereitgestellt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen. In der Lösung der Aufgabe ist es NICHT erlaubt, die Folge vorher zu sortieren.

Aufgabe 4

Man definiert P_i als den Prozess, der prüft, ob i ein Teiler von N ist, und der bestimmt der Rang (Ordnung) der Multiplizität dieses Teilers. Schreiben Sie ein Programm, das die Zahl N in Primfaktoren zerlegt. N wird vom Benutzer bereitgestellt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Beispiel:

$$180 = 2 \cdot 5 \cdot 3 \cdot 2 \cdot 3 = 2^2 \cdot 3^2 \cdot 5^1$$

(2 ist ein Teiler von 180 mit der Ordnung der Multiplizität 2, weil es 2 hoch der Potenz 2 bei der Zerlegung von 180 in Primfaktoren erscheint.)

Standardausgabe:

2 ist ein Teiler für 180 und der Ordnung der Multiplizität ist 2.

3 ist ein Teiler für 180 und der Ordnung der Multiplizität ist 2

5 ist ein Teiler für 180 und der Ordnung der Multiplizität ist 1.

Aufgabe 5

Man definiert P als den Prozess, der den größten gemeinsamen Teiler zweier Zahlen bestimmt. Schreiben Sie ein Programm, das den größten gemeinsamen Teiler von N Zahlen bestimmt. N und die N Zahlen werden vom Benutzer bereitgestellt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 6

Man definiert P_i als den Prozess, der bestimmt, wie oft der Buchstabe i in einem Text vorkommt (die Häufigkeit des Auftretens). Schreiben Sie ein Programm, das eine Statistik darüber erstellt, wie oft jeder Buchstabe in einem vom Benutzer bereitgestellten Text vorkommt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 7

Sei eine Folge von Zahlen. Man definiert P_i als den Prozess, der die dritte Potenz (der Kubikzahl) der Zahl i bestimmt. Schreiben Sie ein Programm, das die Summe der Kubikzahlen der Zahlen aus der Folge bestimmt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 8

Sei eine Folge von Zahlen (x_1, x_2, \dots, x_n) . Schreiben Sie ein Programm, das die Teilsummen $x_1, x_1 + x_2, x_1 + x_2 + x_3, \dots, x_1 + x_2 + \dots + x_n$ ermittelt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 9

Sei eine Zeichenfolge. Schreiben Sie ein Programm, das alle Vokale aus dieser Zeichenfolge entfernt. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 10

Sei eine Folge von Zahlen. Schreiben Sie ein Programm, das die Summe der Zahlen aus der Folge mit der Methode divide et impera berechnet: Ein Prozess teilt die Folge in zwei Teilfolgen, die er anderen Prozessen (Kindprozesse) zur Berechnung der Summe gibt, wonach er die beiden erhaltenen Ergebnisse addiert. Die Kindprozesse wenden weiterhin die gleiche Technik an. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.

Aufgabe 11

Sei eine Folge von Zahlen. Schreiben Sie ein Programm, das das Produkt der Zahlen aus der Folge mit der Methode `divide et impera` berechnet: Ein Prozess teilt die Folge in zwei Teilfolgen, die er anderen Prozessen (Kindprozesse) zur Berechnung des Produkts gibt, wonach er die beiden erhaltenen Ergebnisse multipliziert. Die Kindprozesse wenden weiterhin die gleiche Technik an. Das Programm verwendet eine geeignete Anzahl von Prozessen, um die Aufgabe zu lösen.