

Übungen in Systemnaher Programmierung

Übungsblatt 6

Sie sollen Kapitel 6 nochmal lesen (das war schon in der vergangenen Woche auf). Auch Kapitel 7 über robuste Programme sollen Sie lesen.

Aufgabe 1

- Wie macht man aus einem Zahlenwert 0...9 das entsprechende ASCII Zeichen?
- Wie macht man im ASCII Code aus einem grossen Buchstaben einen kleinen Buchstaben?

Aufgabe 2

Schreiben Sie ein Programm in Assembler, das die Anzahl der Zeichen und die Anzahl der Zeilen in einer Textdatei bestimmt und auf den Bildschirm (stdout) schreibt. Der Name der Datei wird als Argument an das Programm übergeben.

Aufgabe 3

Wie ist das Programm im Bartlett in Kapitel 6 strukturiert? Zeichnen Sie in einem Diagramm die Abhängigkeiten der einzelnen Dateien. Mit $A \implies B$ kennzeichnen Sie, dass Datei B die Quelltextdatei A inkludiert. Mit $A + B$ kennzeichnen Sie, dass die Objektdateien A und B miteinander gelinkt werden.

Aufgabe 4

Diese Aufgaben sind eine eingeschränkte Auswahl der "Going Further" Aufgaben am Ende von Kapitel 6:

1. Schreiben Sie das Programm um, so dass Kommandozeilenargumente verwendet werden.
2. Fangen Sie Fehler ab, die beim Öffnen der Dateien auftreten können. Welche Fehlercodes es gibt, finden Sie in der man page "man 2 open". Sie können einen Schreibschutzfehler hervorrufen, wenn Sie die Datei als normaler Anwender an einem nicht erlaubten Ort öffnen, z.B. mit dem Pfad `"/meine-datei.txt"`.
3. Schreiben Sie ein Programm mit dem Sie auswählen können, welcher Datensatz ausgegeben werden soll. Sie geben auf der Kommandozeile einfach die Nummer 0, 1, 2, ... an, dann wird der jeweilige Datensatz ausgegeben. Mit dem Systemaufruf `lseek()` können Sie schnell an eine beliebige Stelle in der Datei gehen. Der Funktionscode für `lseek` in Register `eax` ist 19. Siehe "man 2 lseek".