

3.1 Erzeugen eines Kindprozesses: forkone (3 Punkte)

Schreiben Sie ein Programm `forkone`, das einen Kind-Prozess erzeugt und dann auf dessen Beendigung wartet. Der Kind-Prozess soll jeweils im Sekundentakt bis zu einer Zahl K hochzählen. K wird auf der Kommandozeile übergeben.

Zu Beginn und kurz vor dem Beenden soll der Elternprozess die jeweils aktuelle Uhrzeit ausgeben. Beim Hochzählen sollen die Prozess-ID des jeweiligen Kindprozesses und die des Elternprozesses vorangestellt werden, so dass sich folgendes Ausgabeformat ergibt.

Ausserdem soll der Kindprozess den Wert K zu seiner eigenen Prozess-ID addieren und das Ergebnis modulo 100 (also nur die Einer- und Zehnerstellen) als Rückgabewert (Exit-Code) liefern. Hinweis: C kennt den Modulo-Operator `%`. Diesen Wert soll der Elternprozess auslesen und als Ergebnis ausgeben.

```
$ ./forkone 10
Start: Tue Oct  9 22:16:22 2018
68333 68332 1
68333 68332 2
68333 68332 3
68333 68332 4
68333 68332 5
68333 68332 6
68333 68332 7
68333 68332 8
68333 68332 9
68333 68332 10
Exit-Code: 43
Ende: Tue Oct  9 22:16:32 2018
```

Syntax: `forkone <K>`

Quellen: `forkone.c`

Executable: `forkone`

3.2 Erzeugen mehrerer Kindprozesse: forkmany (7 Punkte)

Erweitern Sie das Programm von Abschnitt 3.1, dass es N Kindprozess erzeugt, die alle jeweils bis K zählen. Der Elternprozess soll sich alle erzeugten Kind-Prozesse in einer Liste merken und warten, bis sie alle beendet sind.

Schließlich sollen die Kind-Prozesse nicht alle bis K zählen, sondern bis zu einer zufälligen ganzen Zahl in $[K/2; 1,5 \times K]$, wenn der optionale Parameter “-r” auf der Kommandozeile angegeben wird. Hierfür können Sie Integer-Arithmetik verwenden. Nehmen Sie als Default-Parameter $K = 10$ und $N = 1$ an, falls der jeweilige Parameter nicht auf der Kommandozeile übergeben wird.

Das Ausgabeformat bleibt unverändert. Führen Sie das Programm mehrfach mit vielen Prozessen und $K \geq 10$ aus und beobachten Sie, ob/wie sich die Reihenfolge der Ausgaben der Prozesse bei den einzelnen Schritte aufgrund des Scheduling ändert.

Hinweis: die eckigen Klammern in der folgenden Syntax drücken optionale Parameter aus. Die Reihenfolge der Argumente kann prinzipiell beliebig sein. Verwenden Sie `getopt (3)` zum Parsieren der Kommandozeilenparameter.

Syntax: `forkmany [-k <K>] [-n <N>] [-r]`

Quellen: `forkmany.c list.c list.h`

Executable: `forkmany`