Christian Weiss: 445316
Florian Hoffmann: 444959
Yannick Hettinga: 445071

Aufgabe 1.1

a)

i) waiting → running: Dies kann nicht eintreten, da wenn das Ereigniss auf das gewartet wird eintritt, zuerst in den ready-Zustand gewechselt wird, bis der Scheduler dem Prozess Zeit auf dem CPU gibt.

ii) running → waiting: Dies tritt ein, wenn der Prozess entweder auf I/O oder ein anderes Ereigniss warten muss, bevor er weiterrechnen kann.

iii) ready → waiting: Dies kann nicht eintreten, da der Prozess, solange er ready ist nicht Ausgeführt wird und daher nicht signalisieren kann, dass er auf ein Ereigniss warten muss.

iv) ready → terminated: Dies kann eigentlich nicht eintreten, da der Prozess entweder sich selber beenden muss, was er nur selber kann, oder von einem externen Signal beendet wird, wobei aber der Signal Handler des Prozesses ausgeführt wird. Also muss der Prozess in running sein, um zu terminieren, aber es gibt meist auch die Möglichkeit zu "killen", was in jedem Zustand passieren kann, ohne dass der Prozess reagieren kann.

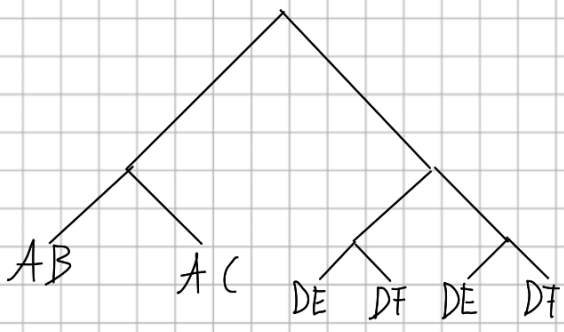
b) Eigentlich nur einer, aber bei modernen Geräten mit mehreren Kernen und hardwareseitigem Multithreading so viele wie das Gerät an Threads hat

c) Beliebig viele, bis der Hauptspeicher oder die Prozesstabelle voll ist.

d) In beiden Zuständen wird der Prozess gerade nicht auf der CPU ausgeführt, aber im ready Zustand liegt es nur daran, dass einem anderen Prozess die CPU-Zeit vom Scheduler zugeteilt wurde, während bei waiting auf ein prozess-externes Ereignis gewartet werden muss.

Aufgabe 1.2.

a)



b) "A" wird vor "B" und "C" ausgegeben. "B" wird vor "C" ausgegeben. "D" wird vor "E" und "F" ausgegeben.

Aufgabe 1.3

b) Relativ gesehen sind die Anzahl an Ausgaben etwa gleich.

c) Eine Fork-Bomb ruft wiederholt fork() auf, wobei jeder Kindprozess das selbe macht, wodurch das System überfordert und blockiert wird. Z.B. gibt es in Linux die Möglichkeit mit "ulimit -u <anzahl an Eigenen-prozessen>". Dies limitiert die Anzahl an Prozessen, die ein einziger User haben kann.