Übung 11

Da die Semaphore t auf 0 gesetzt ist, blockiert Prozess0 bei t.wait, da t nicht mehr weiter dekrementiert werden kann (wurde mittels Konstruktor auf 0 initialisiert). Prozess1 kann zwar s.wait ausführen und damit s dekrementieren, blockiert aber im nächsten Schritt wie Prozess0 an der Stelle t.wait, da wiederrum t nicht dekrementiert werden kann. Demnach können die Prozesse weder sich selbst noch den jeweils anderen „befreien“ (t.signal bzw. s.signal wird nie erreicht). Nur ein weiterer Prozess könnte mittels „signal“ auf eine der Semaphoren den Lock lösen.

Übung13

Der Fehler, der bei der Berechnung entsteht ist bis zu einer Intervallanzahl von 10^5 bei sequenzieller Abarbeitung und paralleler Berechnung mit geringer Prozessoranzahl gleich. Bei mehr Prozessoren (hier 30) ist der Fehler deutlich größer. Zwischen 10^6 und 10^9 Intervallen ist die parallele Berechnung gegenüber der Sequenziellen in der Größe des Fehlers im Vorteil. Ab 10^10 Intervallen pendeln sich die Parallelberechnungen auf einen Fehlerwert von etwa 10^-8 ein. Ab hier ist die sequenzielle Abarbeitung (bzgl. der Fehlergröße) deutlich besser.