

## 2. Aufgabenblatt

Ferdinand Markward Scheller, Frederick Brandenburg

1. In spezialisierten Systemen wäre es möglich, eine direkte Verbindung zwischen der Hardware und dem Prozessor zu ziehen, über die, ohne Verwendung eines Interrupt Controllers, Interrupts kommuniziert werden können.
2. Grundsätzlich sind nur Systemaufrufe, die nicht zwangsläufig im Kernel Mode ablaufen müssen für die Weitergabe per vDSO geeignet. Darüber hinaus ist es sinnvoll, nur Aufrufe zu übergeben, die von dem User nur gelesen werden, ein Schreibversuch würde fehlschlagen, da der User nicht in Kerneladressräumen schreiben kann.
3. Kooperatives Multitasking bietet den Vorteil, dass die Entwicklung von Programmen insofern einfacher ist, dass Programme nicht an jeder Stelle wiedereintrittsfähig sein müssen. Die Stellen, an denen ein Prozess unterbrochen und später wieder weitergeführt werden kann, können beliebig bei der Entwicklung gewählt werden. (<http://www.on-time.com/rtos-32-docs/rtkernel-32/programming-manual/advanced-topics/preemptive-or-cooperative-multitasking.htm> (12.05.2020, 18:20 Uhr))
4. Im PCB werden zunächst Informationen gespeichert, die für die Identifikation des Prozesses nötig sind (PID, Elternprozess, Benutzer). Dazu kommen Prozessorstatusinformationen (stack pointer, program counter, processor status word), für den User sichtbare Register und diverse Kontrollinformationen (benutzer Speicher, benutzte Prozessorzeit, I/O, Informationen bzgl. scheduling etc.). Diese Werte werden aktualisiert, wenn der Prozess den aktiven Zustand verlässt (also in ready oder blocked wechselt) und eingelesen, wenn er diesen wieder betritt.