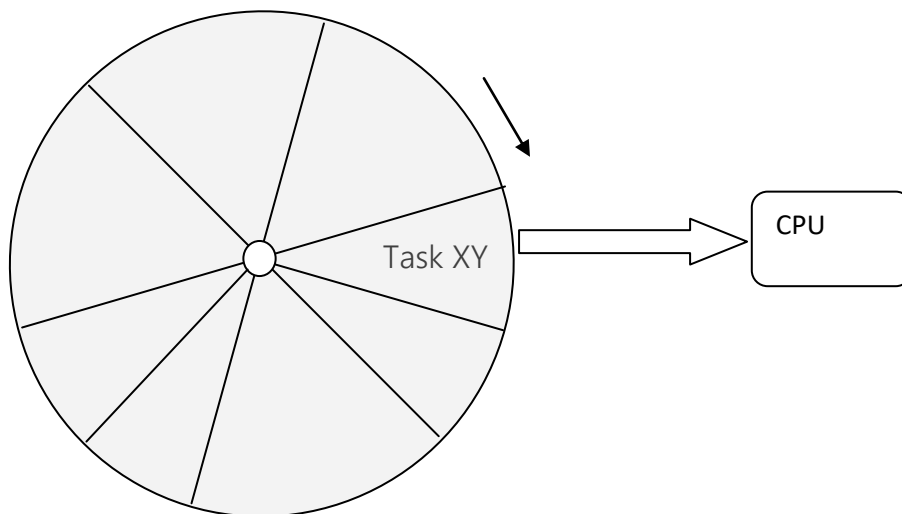




# Multitasking

---

Multitasking ist nichts anderes als der Betrieb der CPU im Zeitmultiplex. Zeitmultiplex läßt sich gut als rotierende „Zeitscheibe“ darstellen.

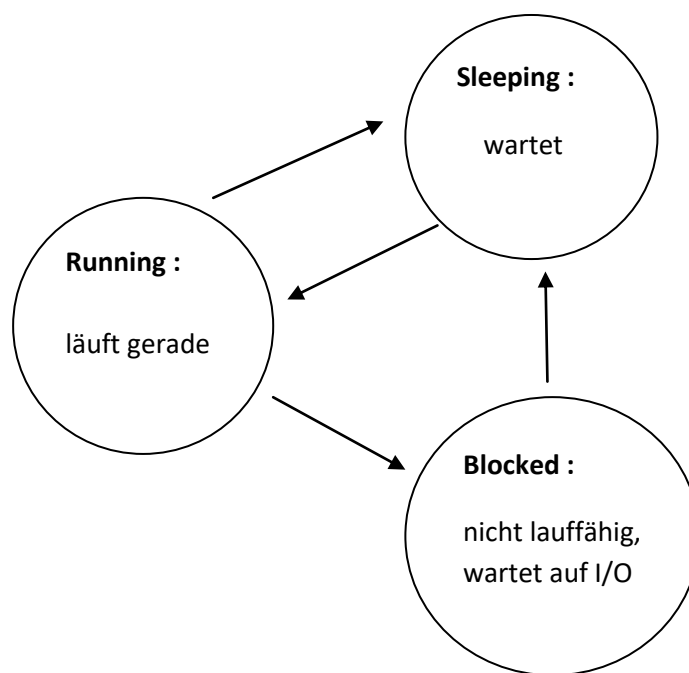


Die Zeitscheibe dreht sich, jede Task bekommt eine Zeitlang die CPU zugeteilt. Die zugeteilte Zeit heißt Quantum. Zu Beginn und am Ende jedes Quantums muß das Betriebssystem die Taskumschaltung durchführen, diese Zeiten gehen als Systemrechenzeit den Prozessen verloren.

# Taskumschaltung

---

Man kann den Zustand einer Task auch gut als state-machine (Zustandsdiagramm) darstellen. In einem der Kreise (Zustände) befindet sich die Task gerade :



Eine rechenbereite Task wird rechnerisch gesetzt („bekommt die CPU“). Nach ihrem Quantum wird sie wieder schlafen gelegt. Falls eine Task ein (im Vergleich zur CPU unendlich langsames) Peripheriegerät anspricht (Netzwerk, Festplatte usw.), wird sie blockiert („nicht rechenbereit“). Wenn die Peripherie meldet, daß sie fertig ist, wird die Task wieder rechenbereit.

## Scheduler

---

Die Funktion der Taskumschaltung erfüllen zwei wichtige Betriebssystemkomponenten. Der Scheduler legt strategisch fest, welche Task als nächste drankommt und wann. Der Dispatcher führt dann die Taskumschaltung durch.

Der Scheduler ist für das Verhalten des Betriebssystems von entscheidender Bedeutung. Es gibt eine ganze Reihe verschiedener Scheduling-Strategien. Für unser Umfeld, die Industrieanwendung, sind folgende von Bedeutung :

- Kooperatives Multitasking
- Präemptives (englisch : preemptive) Multitasking
- Echtzeit-Scheduling

## kooperatives Multitasking

Bei diesem Verfahren wirkt die running task selber bei der Entscheidung des Schedulers mit, wann sie gestoppt wird. Der optimale Zeitpunkt ist jeweils bei Übergang nach blocked, weil sie ja dann sowieso nicht mehr rechnen kann. Das funktioniert so richtig nur bei sehr interaktiven Systemen (Büro, Spiele usw.), und hat einen gewaltigen Hacken :

Wenn ein Programmfehler vorliegt, zum Beispiel eine Endlosschleife, steht das ganze System. Es findet kein Taskwechsel mehr statt, den Rechner kann man dann neu booten.

Im Industrieumfeld kann das also nie das primäre Scheduling-System sein.

## preemptive multitasking

Das Problem beim kooperativen Multitasking war das Mitspracherecht der Task. Beim preemptive multitasking wird der Taskwechsel deswegen taskunabhängig, durch einen Timer, gesteuert. Egal was eine Task tut, auch wenn sie durch Programmfehler abstürzt, die Systemstabilität ist dadurch nicht gefährdet. Das ist zwar nicht so effektiv wie kooperatives Scheduling, aber dafür stabil.

Es lohnt sich, über die Auswirkung der Quantum-Länge beim preemptive multitasking nachzudenken.

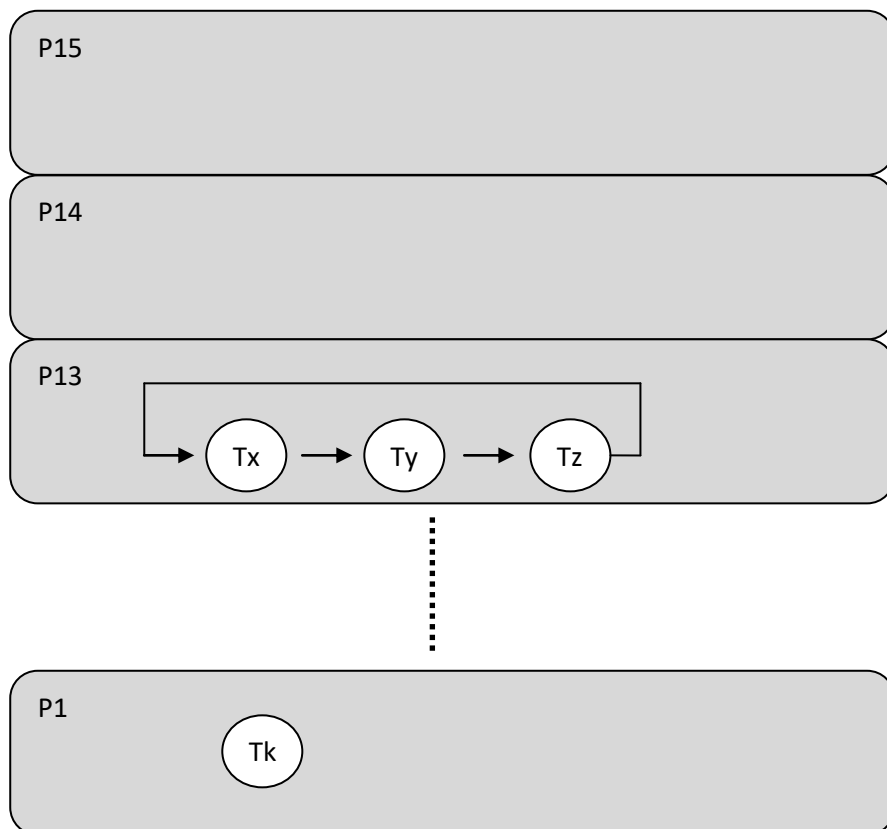
Ein langes Quantum bewirkt, daß das Verhältniss von Taskrechenzeit und Taskumschaltzeit günstiger wird. Die Netto-Rechenleistung des Systems steigt.

Ein kurzes Quantum bewirkt, daß die Prozesse schneller wieder drankommen. Das bewirkt eine schnellere Reaktionszeit des Systems z.b. auf Benutzereingaben. Der Anwender fühlt ein „schnelles System“.

## preemptive multitasking in Windows

Windows arbeitete bis Windows NT mit kooperativem Multitasking. Mit einer relativ schwachen Hardware kann so ein einigermaßen schnelles System gebaut werden, für Heim-anwender, Gamer usw. optimal. Ab Windows NT hatte Microsoft den professionellen Markt im Visier. Dazu mußte das System stabil werden, und (wie z.B UNIX) preemptive arbeiten.

Der Windows-Scheduler arbeitet mit 15 Prioritätsstufen :



P15 ist die höchste Stufe, P1 die niedrigste.

Nun werden beginnend mit P15 zunächst alle Tasks abgearbeitet, die in dieser Stufe vorhanden sind. Erst wenn diese alle beendet sind, schaut man in die nächst tiefere Prioritätsstufe.

Jedes mal, wenn eine Task in der aktuell arbeitenden Priorität mit einem Quantum nicht zu Ende rechnen kann (das Quantum also voll ausrechnet) fällt sie eine Stufe in der Priorität. (Schon nach wenigen (genau 14) Quanten kommt sie also in P1 an)

Falls eine Task, egal in welcher Stufe, interaktiv wird (also z.B. mit einem Benutzer interagiert), wird sie sofort in die höchste Priorität versetzt.

Das Quantum ist in der höchsten Priorität kurz (schnelle Reaktion), bei Windows NT waren es 20ms. In der niedrigsten Priorität ist es lang (damals 120ms), um hohe Rechenleistung zu erzielen.

Wenn man lange genug drüber nachdenkt, versteht man den Sinn ..