

Windows 2000-Scheduler

Referent: Viktor Witt

am 13.06.05

Überblick

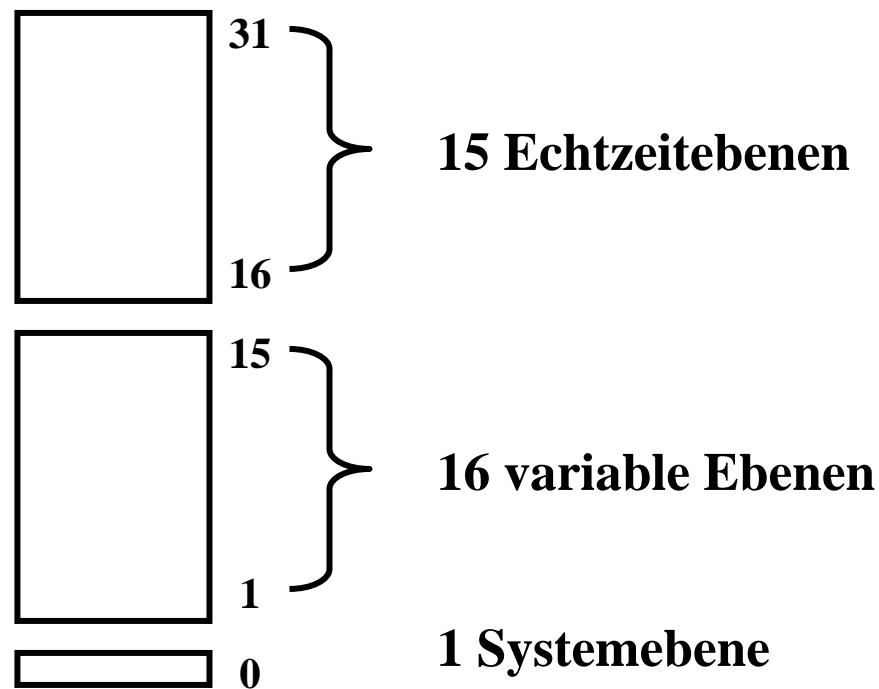
- **Funktionsweise und Begriffe**
- **Prioritätsstufen**
(Welche Prioritäten gibt es? Wie kommen sie zustande?)
- **Threadzustände**
- **Praxisbeispiele**
- **Prioritätsanhebungen**
(Wann wird die Priorität eines Threads angehoben?)
- **Zusammenfassung**

Funktionsweise und Begriffe

- Windows 2000 implementiert einen **prioritätsgesteuerten** Scheduling-Algorithmus
- Es werden nur **Threads** geschedult
- Grundidee:
 - Threads bekommen **Prioritäten** zugewiesen
 - Thread mit der **höchsten** Priorität darf ablaufen
- **Präemptiv**
 - **Kontextwechsel** kann jeder Zeit auftreten
 - keine Garantie über Rechenzeit

Prioritätssstufen

□ Windows 2000 **interne** Prioritäten



Prioritätsstufen

- **Win32-API-Prioritäten:**
- ***SetPriorityClass***
 - weist jedem Prozess eine Prioritätsklasse zu, die
 - zugleich Basispriorität für Threads des Prozesses ist
- ***SetThreadPriority***
 - jeder Thread bekommt eine relative Priorität innerhalb eines Prozesses zugewiesen

Prioritätssstufen

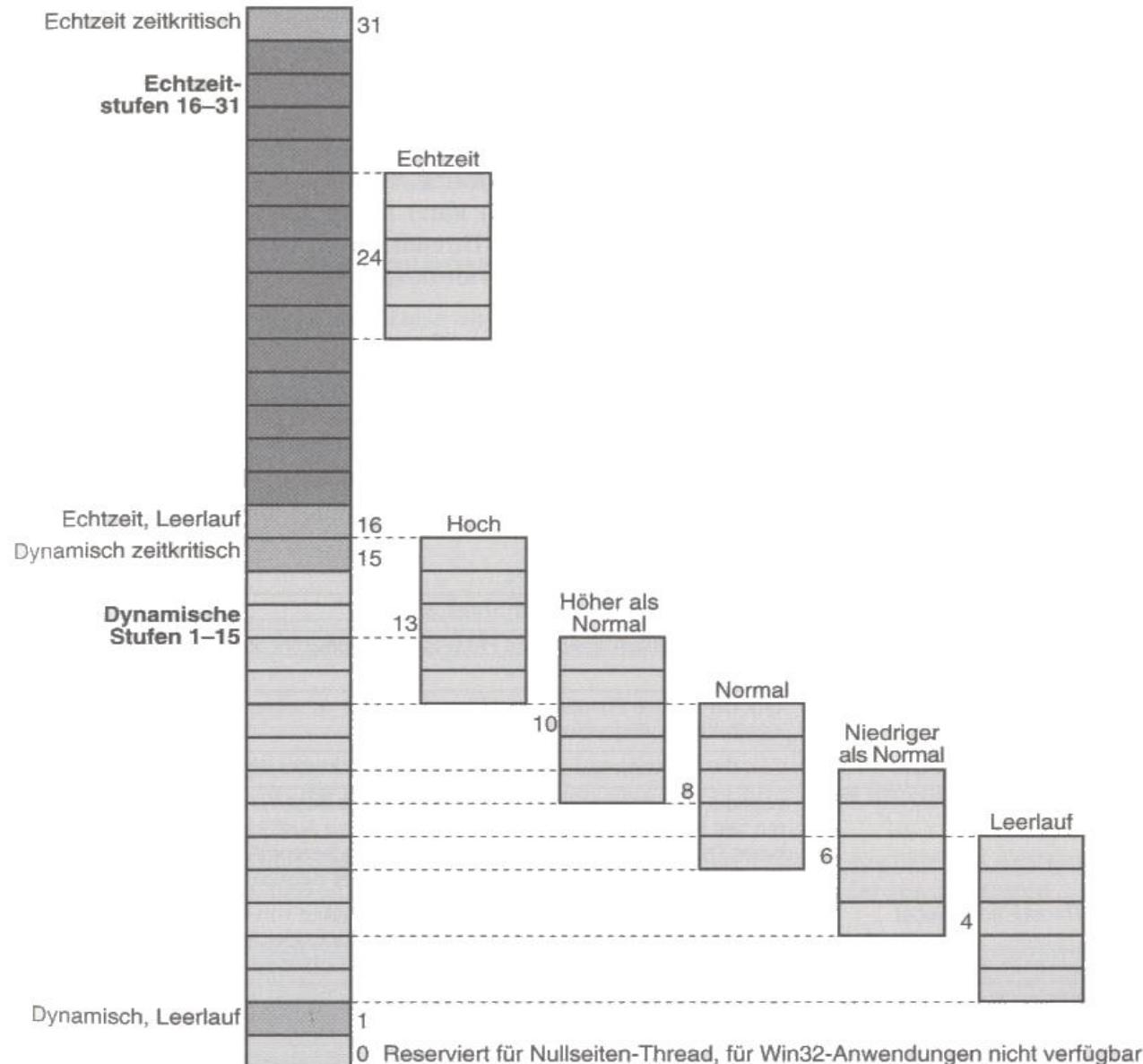
- Win32-API-Prioritäten werden auf die Windows 2000 interne Prioritäten abgebildet

Win32-Prozessklassen-Prioritäten

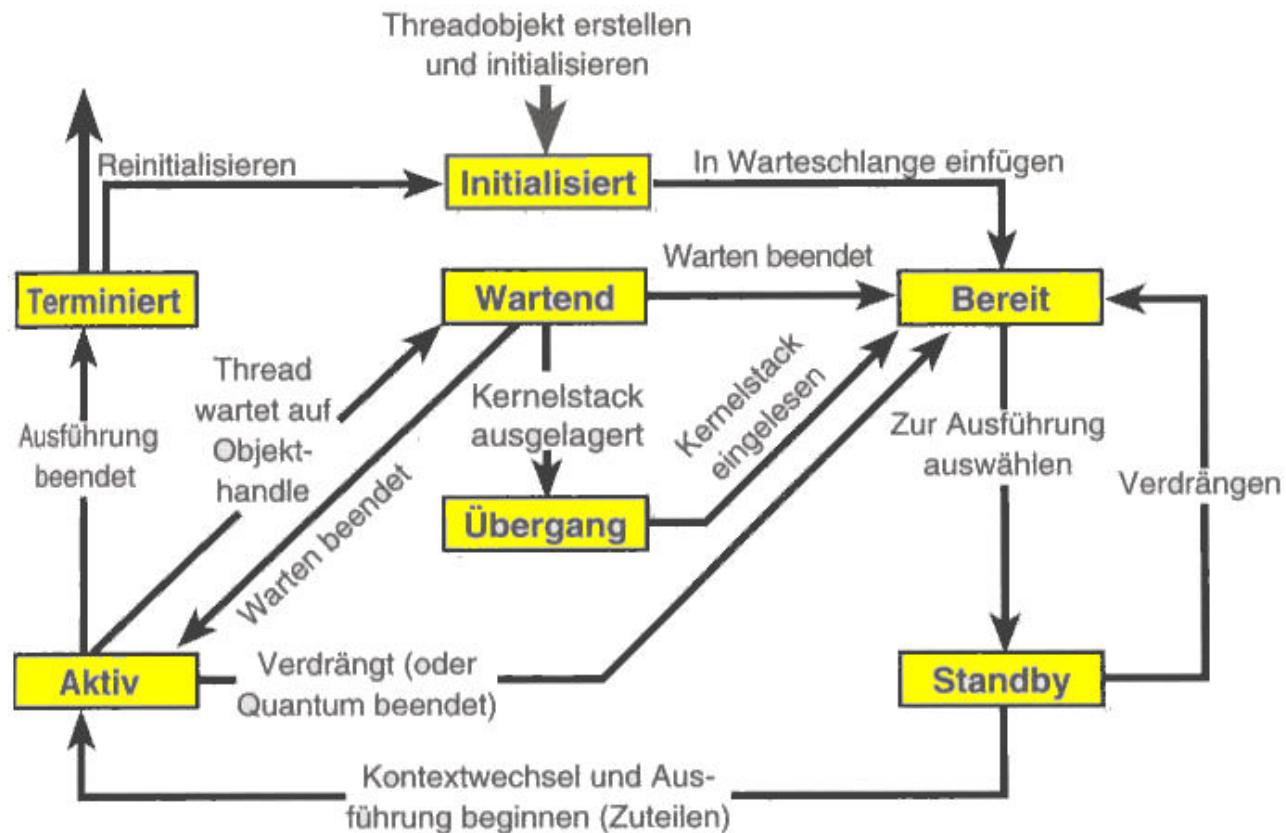
**Win32-
Thread-
Priori-
täten**

	Echtzeit	Hoch	Über normal	Normal	Unter normal	Leerlauf
Zeitkritisch	31	15	15	15	15	15
Höchste	26	15	12	10	8	6
Über normal	25	14	11	9	7	5
Normal	24	13	10	8	6	4
Unter normal	23	12	9	7	5	3
Niedrigste	22	11	8	6	4	2
Leerlauf	16	1	1	1	1	1

- **aktuelle Priorität:**
 - wird beim Scheduling verwendet
 - **dynamische Anpassung** im Bereich der variablen Stufen



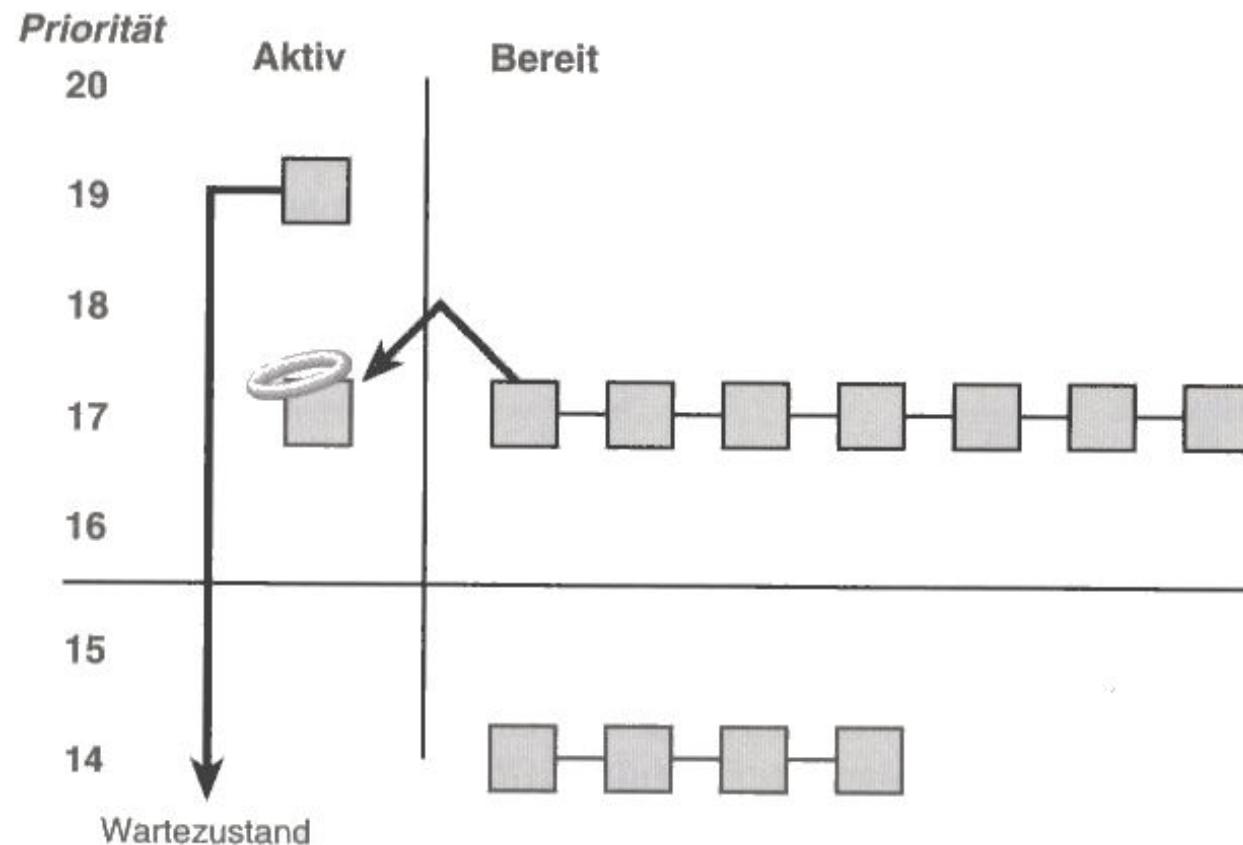
Threadzustände



Freiwilliger Kontextwechsel

- freiwilliges Wechselt in den Wartezustand mittels Win32-API-Funktionen
- Anderer Thread wird zur Ausführung ausgewählt
- das Warten auf ein Objekt beendet:
 - Thread ans Ende der Bereit-Warteschlange gestellt
 - meist Prioritätserhöhung

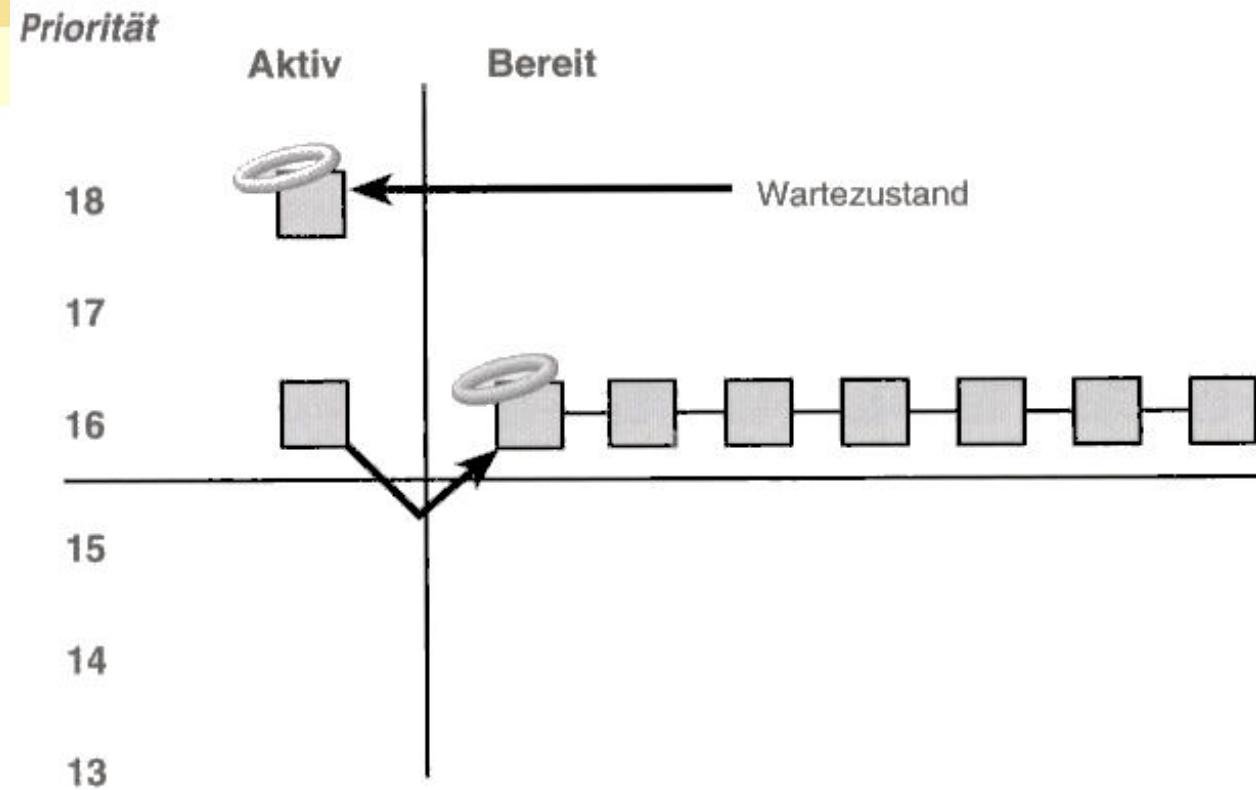
Freiwilliger Kontextwechsel



Verdrängung

- Threads niedriger Priorität werden unterbrochen, wenn ein Thread höherer Priorität ausführungsbereit wird
- Verdrängter Thread wird an den Anfang der Bereit-Warteschlange gestellt
- Er darf seine Zeitscheibe zu Ende laufen, sobald der unterbrechende Thread seine Ausführung beendet hat

Verdrängung



Aufgebrauchtes Quantum

- Round-Robin-Prinzip
- Scheduler entscheidet über Senkung der Priorität
- Thread bekommt einen neuen Quantumswert zugewiesen
- Wenn keine bereiten Threads höherer oder gleicher Priorität vorhanden
=> Thread darf für eine weitere Zeitscheibe ablaufen

Aufgebrauchtes Quantum

Priorität

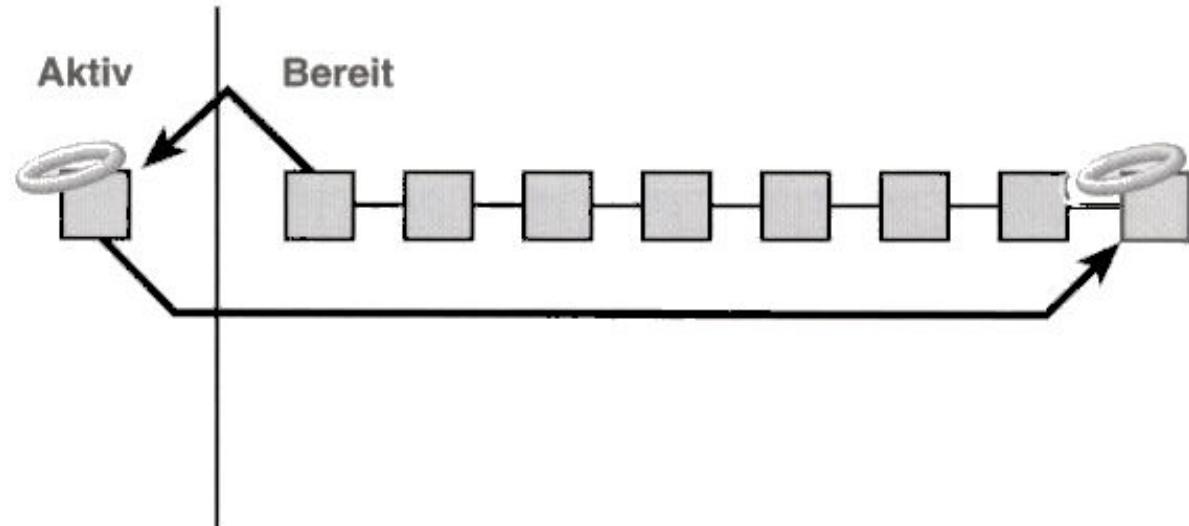
15

14

13

12

11



Überblick

- Funktionsweise und Begriffe**
- Prioritätsstufen**
(Welche Prioritäten gibt es? Wie kommen sie zustande?)
- Threadzustände**
- Praxisbeispiele**
- Prioritätsanhebungen**
(Wann wird die Priorität eines Threads angehoben?)
- Zusammenfassung**

Anhebung nach E/A-Operationen

- Anhebungen nur bei dynamischen Prioritätsstufen möglich
- Wert der Erhöhung hängt vom **Gerätetreiber** ab

Gerät	Prioritätsanhebung
Festplatte, CD-ROM, parallele Schnittstelle, Grafik	1
Netzwerk, Mailslot, Named Pipe, serielle Schnittstelle	2
Tastatur, Maus	6
Sound	8

- Mit jeder abgelaufenen Zeitscheibe wird die aktuelle Priorität um 1 Prioritätsstufe niedriger, bis Basispriorität erreicht ist

Anhebung nach Ereignissen und Semaphoren

- Erhöhung um 1 Prioritätsstufe nach dem Abwarten eines Ereignisses oder Semaphorobjektes

- schrittweises Absenken der aktuellen Priorität
(identisch zum Absenken bei E/A-Operationen)

Anhebung als Mittel gegen das „Verhungern“

- Thread hält sich seit längerer Zeit im Bereit-Zustand auf
- **sprungartige** Prioritätserhöhung auf Stufe 15
- darf auf dieser Stufe für **zwei Quanten** ausgeführt werden
- danach **unmittelbares** Zurückkehren zur Basispriorität

Anhebung von GUI-Threads und Vordergrundthreads

□ GUI-Threads

- agieren als Besitzer von Fenstern
- Prioritätserhöhung um 2 nach ihrem Erwachen

□ Vordergrundthreads

- werden von der Fensterverwaltung ermittelt
- Erhöhung wird auf aktuelle Priorität angewendet

□ Begünstigung von interaktiven Prozessen

Zusammenfassung

- Windows 2000 hat einen prioritätsgesteuerten Scheduling-Algorithmus
- Grundlage für Scheduling bilden die Threads mit ihren Zuständen
- Verhalten von Threads in Standardsituationen
- jeder Thread besitzt eine aktuelle und eine Basispriorität
- Fälle für Prioritätsanhebungen