

OSEK/VDX

Markus Mayer

05.12.2002



Was ist OSEK/VDX?

- Familie von Standards
- Systemschnittstelle für Kraftfahrzeugelektronik
- Osek-OS
 - Echtzeitbetriebssystem
 - Multitasking
 - auch auf kleinsten Prozessoren lauffähig



Inhalt des Vortrags

1. Geschichte
2. Ziele
3. Struktur
4. OSEK-OS



1. Geschichte

- Vor OSEK: Zeitscheiben
- Deutschland: OSEK
- Frankreich: VDX
- Zusammenschluss 1994
- 1995 erste OSEK/VDX Spezifikation



2. Ziele

1. Senkung von Entwicklungskosten
2. Portabilität von Software
3. Optimale Anpassung an die jeweilige Hardware



3. Struktur

1. OSEK-COM: Kommunikationssystem
2. OSEK-NM: Netzwerk Management
3. OSEK-OS: Betriebssystem



4. OSEK-OS

1. Architektur
2. Tasks (Unterprogramme)
3. Events
4. Ressourcen
5. Interrupts

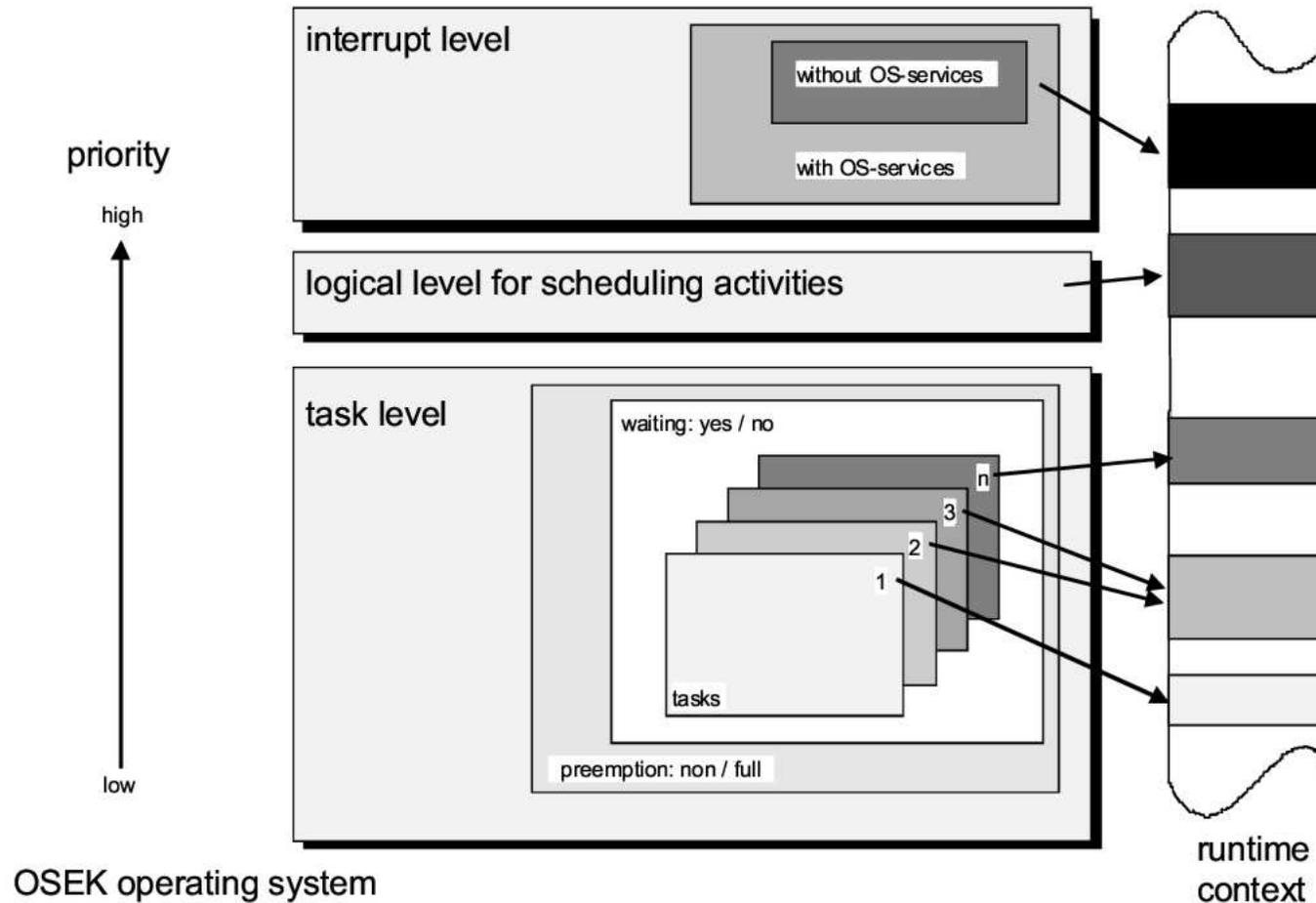


Architektur

- Während der Laufzeit nicht mehr konfigurierbar
- Konformitätsklassen
- Schnittstellen für Benutzer:
 1. Interruptroutinen
 2. Tasks

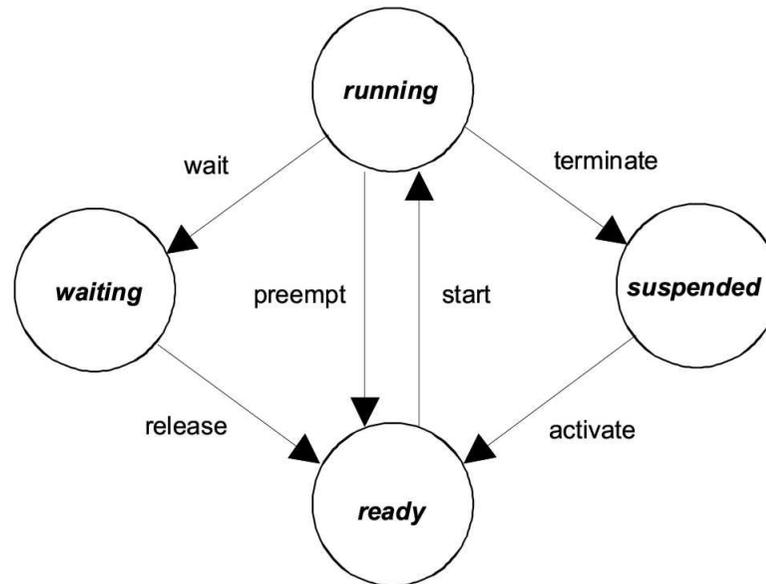


Prozessorbelegung



Tasks

- Standard- und erweiterte Tasks
- unterschiedliche Zustände:



Vor- und Nachteile der Taskkategorien

1. Standardtasks

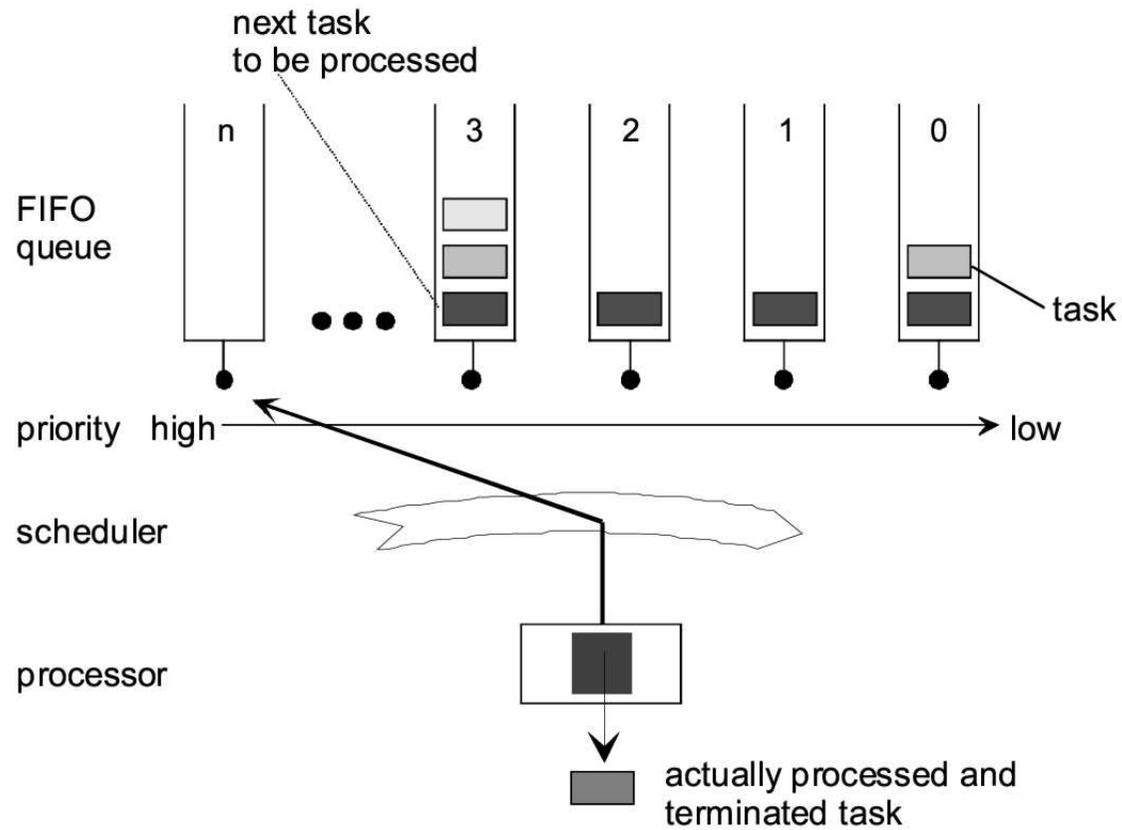
- ✓ Wenig Systemressourcen benötigt
- ✗ wenig Synchronisationsmöglichkeiten

2. Erweiterte Tasks

- ✓ Erfüllt alle Synchronisationsmöglichkeiten
- ✗ Waiting: Arbeitsspeicher benötigt!

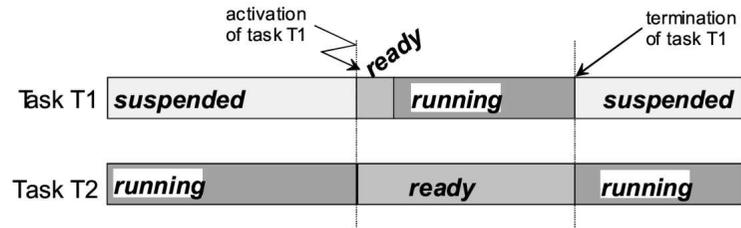


Scheduling

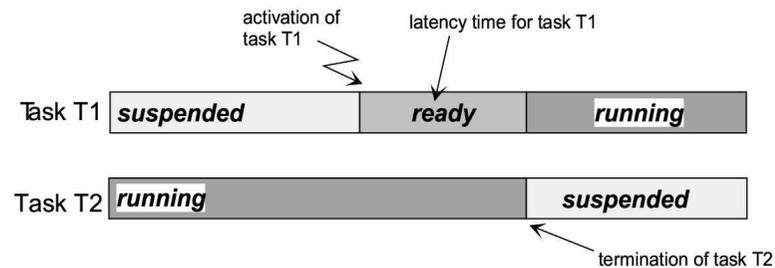


Arbeitsweisen des Schedulers:

1. Unterbrechendes (Preemptives) Scheduling:



2. Nicht unterbrechendes Scheduling:



Events

- Nur für erweiterte Tasks
- Kriterium für Zustandswechsel: “Waiting” nach “Ready”
- Objekt, vom Betriebssystem verwaltet
- *Alle* Tasks und Interruptroutinen können Event setzen
- Bedeutung hängt vom Anwendung ab



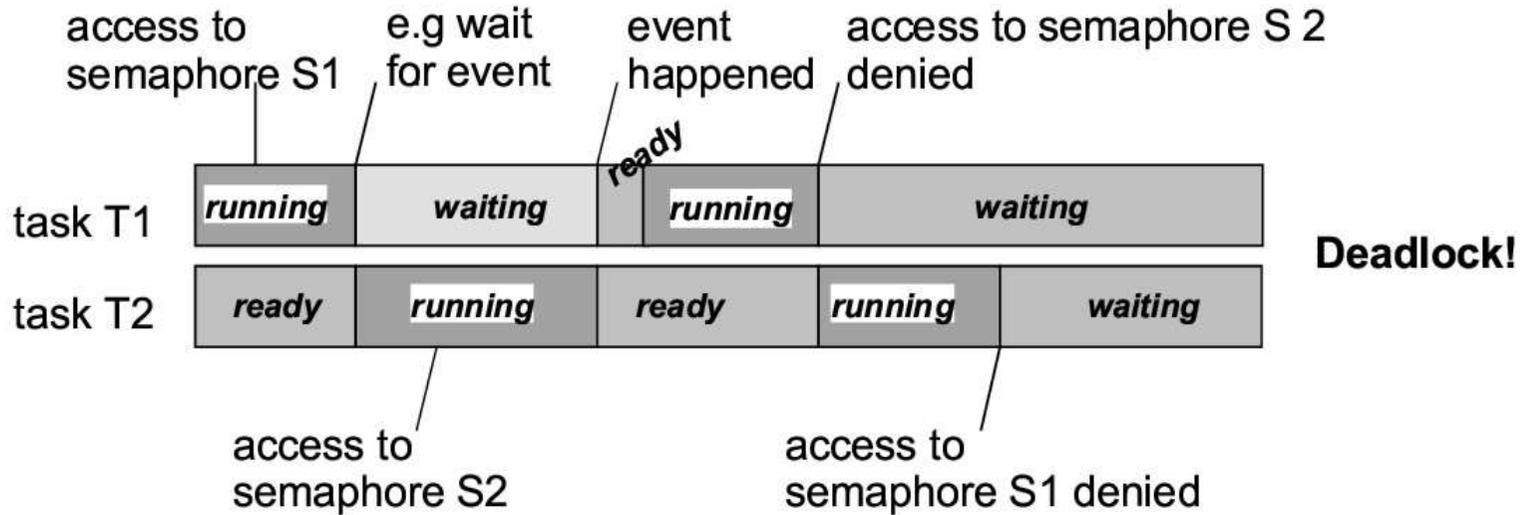
Resourcenverwaltung

- zeitgleicher Zugriff mehrerer Tasks nicht erlaubt
- keine Priotitätenumkehr, kein Deadlock möglich
- “Blockiert” Zustand: Kein Zugriff auf Ressourcen!

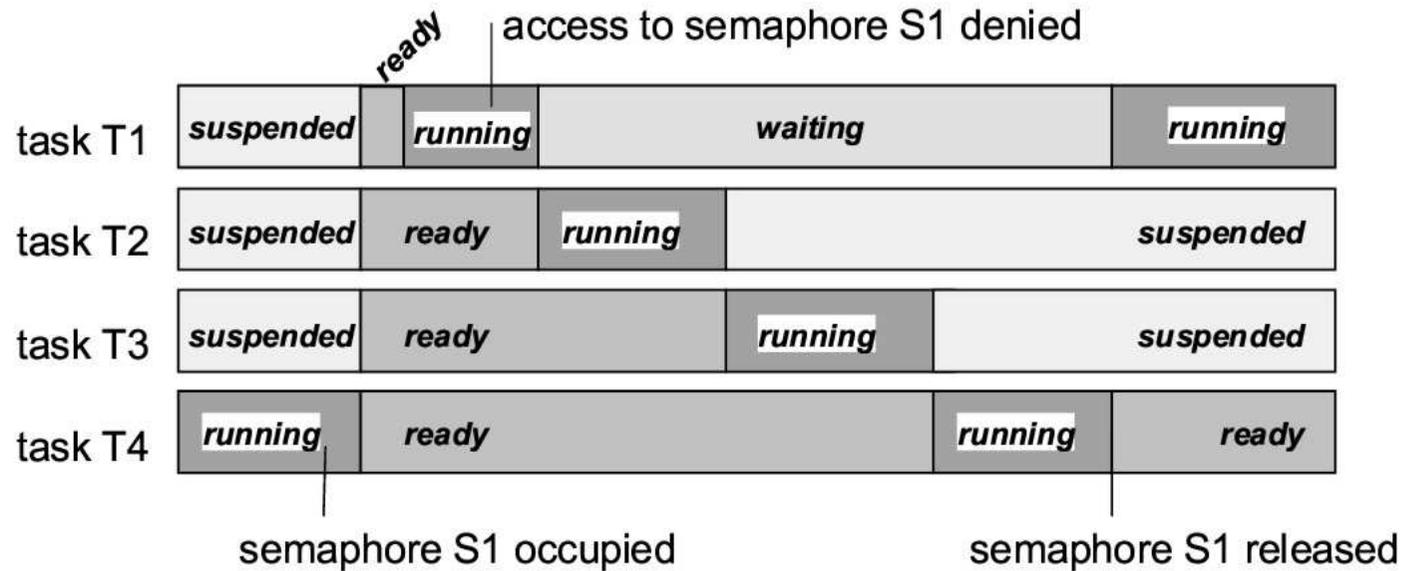
OSEK Priority Ceiling Protokoll!



Deadlock



Priority Inversion

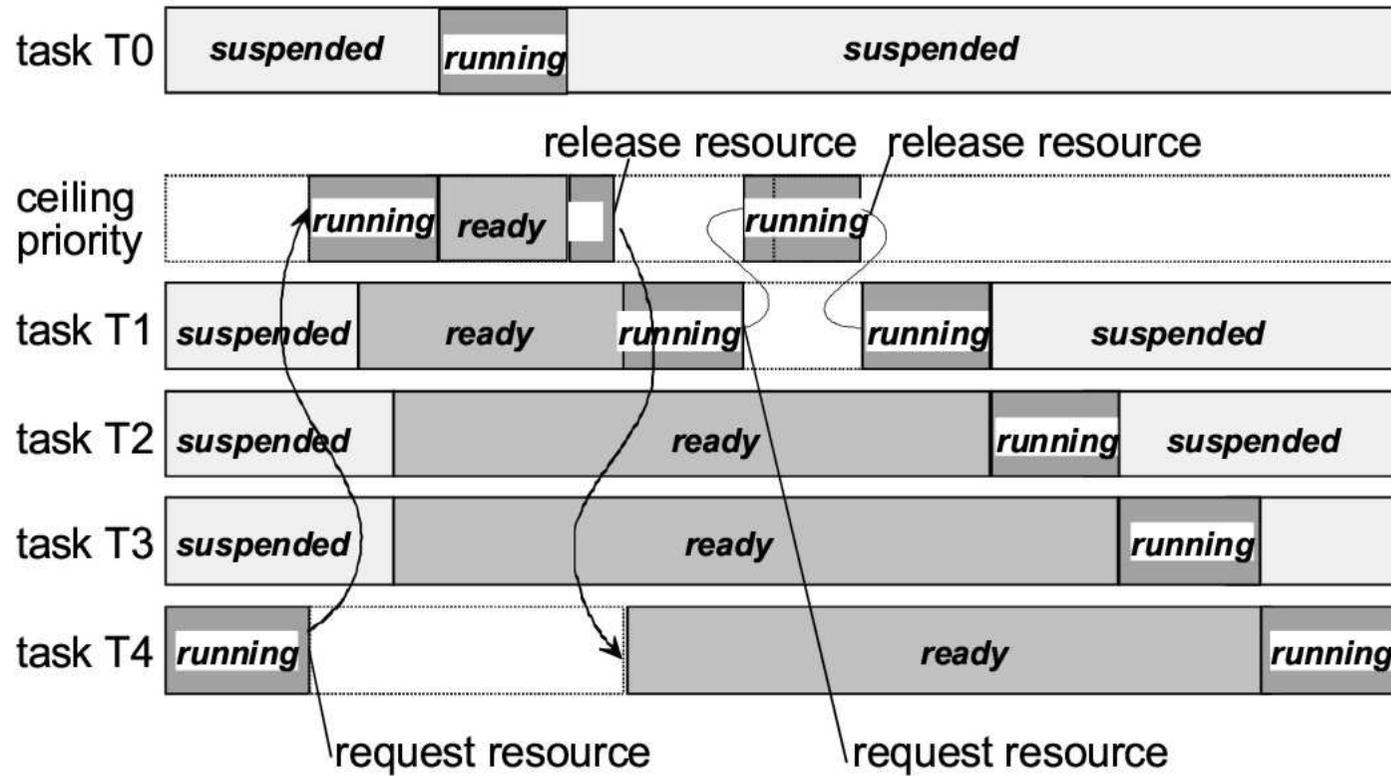


Priority Ceiling Protokoll

- Resource bekommt Priorität
- Task wird bei Zugriff auf diese angehoben
- Bei Freigabe der Resource: Alte Priorität



Beispiel zum PSP



Interrupts

- Scheduling: Hardware
- Können alle Tasks vom Prozessor verdrängen
- Während der Ausführung kein Rescheduling



Firmen, die ein OSEK-OS implementiert haben

- 3Soft (Erlangen)
- Metrowerk

Typical Memory Requirements:

	MPC555	HCS12DP256
Total OS ROM	2304 bytes	880 bytes
OS data	108 bytes	69 bytes
OS code	2196 bytes	811 bytes
Total OS RAM	466 bytes	86 bytes

Benchmark configuration:

- BCC1 with one task per priority
- No multiple activations
- 10 alarms with task activations
- 1 ISR (in addition to system timer)
- Standard status, debug level zero
- Pre-emptive Scheduling
- 10 basic tasks + 1 initialization task
- 1 16-bit (1 ms)
- No messages, no resources
- No hooks

- Softtech



References

[1] www.osek-vdx.org

[2] www.3soft.de/e/proosek

[3] www.kfzelectronic.de/b_osek.htm

