

EMES: Eigenschaften mobiler und eingebetteter Systeme

Betriebssysteme: Windows CE

Dr. Felix Salfner, Dr. Siegmund Sommer
Wintersemester 2010/2011



Windows CE: Die Anfänge

(Anmerkung: Alle Bilder ©Microsoft)

- Windows CE als Microsofts Weg in andere Märkte:
 - PDAs
 - Eingebettete Systeme
- Kompatibilität zum Standard-Windows (WIN32 API)
- Integration in existierende IT-Welten
- Konkurrenz zu Palm und PalmOS

Windows CE: Geschichte I

- 1996: Windows CE 1.0
 - Wenige Applikationen
 - Probleme (kein Drucken, kein dial-up)
- Anfang 1998: Windows CE 2.0
 - Viele CPUs (x86, 82x PPC, SH3, SH4, StrongArm, MIPS)
 - Internationalisierung
 - 24-Bit-Farbe
 - OS-integrierte Internet-Anbindung
 - Weiterentwickelte Applikationen
 - ActiveSync zur Synchronisation mit Desktop
 - Modularer Aufbau (OEM konfiguriert OS)
 - “Auto-PC” als Konzept
 - Hardware: Handhelds mit Tastatur

Windows CE: Geschichte II

- 1999: Windows CE 2.11
 - Aufteilung in Palm-size und Handheld:
 - * Palm-size: kleiner, lediglich Bildschirmtastatur
 - * Handheld: größer, Hardware Tastatur
 - Weiterentwicklung der Anwendungen
 - Problem: Windows-angelehnte Oberfläche
 - Problem: Trotz hoher Rechenleistung (SH3, MIPS mit 80+ MHz) deutlich langsamer als PalmOS-Rechner, hoher Energieverbrauch
- 2000: Windows PocketPC (auf Basis von Windows CE 3.0)
 - Vereinfachtes Nutzerinterface (weg vom Desktop-Windows, Stiftbedienung)
 - Optimierte Performance
 - Neue und verbesserte Anwendungen
 - Multimedia

Windows CE: Geschichte III

- 2002: Windows CE.NET (Codename „Talisker“), Windows Pocket PC 2002, ebenfalls auf Basis von Windows CE 3.0
 - Verbesserte Netzwerkkonnektivität
 - Adhoc Networking
 - Erweiterte vordefinierte Konfigurationen
 - 4 Architekturen (ARM, Mips, SH, x86)
 - Verbesserungen am Userinterface und den Applikationen
 - Verbessertes Speichermanagement
 - Shared Source
 - Eingefügt in .NET-Konzept

Windows CE: Geschichte IV

- 2003: Microsoft Windows Mobile 2003 für Pocket PC (Windows CE 4.2)
 - Nur noch ARM als Architektur
- 2004: Microsoft Windows Mobile 5.0 (Windows CE 5.0)
 - Integriertes WLAN
 - Unterstützung für 3D-Grafikchips
 - Vereinigung Handheld / Smartphone
 - Architekturen: x86, ARM, MIPS, SuperH
- 2007: Microsoft Windows Mobile 6 (Windows CE 5.2)
 - Verbessertes Speichermanagement
 - Mehr Prozesse
 - Architekturen: x86, ARM, SH4, MIPS

Windows CE: Geschichte V

- 2010: Windows Phone 7
- Übergang von Stift- zur Fingerbedienung
- Weiter erhöhte Mindestanforderungen:
 - Min. 256 MB RAM
 - Min. 8GB Flash
 - Kompass, GPS, Beschleunigungs-, Lagesensor
 - Auflösung 800x480 Pixel
 - Min. Taktfrequenz: 1Ghz + DirectX-9 Grafikprozessor

Windows CE: Überblick I

- Modulares Betriebssystem
 - Adaptierbar für Produkt (Platform-Builder)
 - OS-Kern braucht weniger als 200 KB ROM
- Bietet Interrupt-Behandlung und Interrupt-Priorisierung
- Läuft auf einer Reihe von Plattformen:
 - CE 2.11: AMD X5, MIPS R4101, ARM 720T, Motorola MPC823, ARM SA-1100, NEC VR4111, Hitachi SH3 und SH4, NEC VR4300, PPC 403GC, PPC 821, MIPS 4102, QED 5230, MIPS R3910 und R3912, x86
 - CE 3.0 und ab 5.0: ARM, Mips, SH, x86
 - CE 4: ARM und x86
- Unterstützt mehr als 1000 Funktionen der Win32-API
- Applikationen und Betriebssystem aus ROM gestartet

Windows CE: Überblick II

- Verschiedene User-Interfaces, u.a.:
 - Touch-Screen
 - Tastatur
 - Farb-Bildschirme mit bis zu 32 Bit Farbtiefe
- Kommunikation
 - Serielle Kommunikation und IrDA
 - TCP/IP
 - Mobile Channels für Webangebote für CE-Nutzer
 - Component Object Model
 - Interprozeß-Kommunikation (z.B. IR-Sockets)

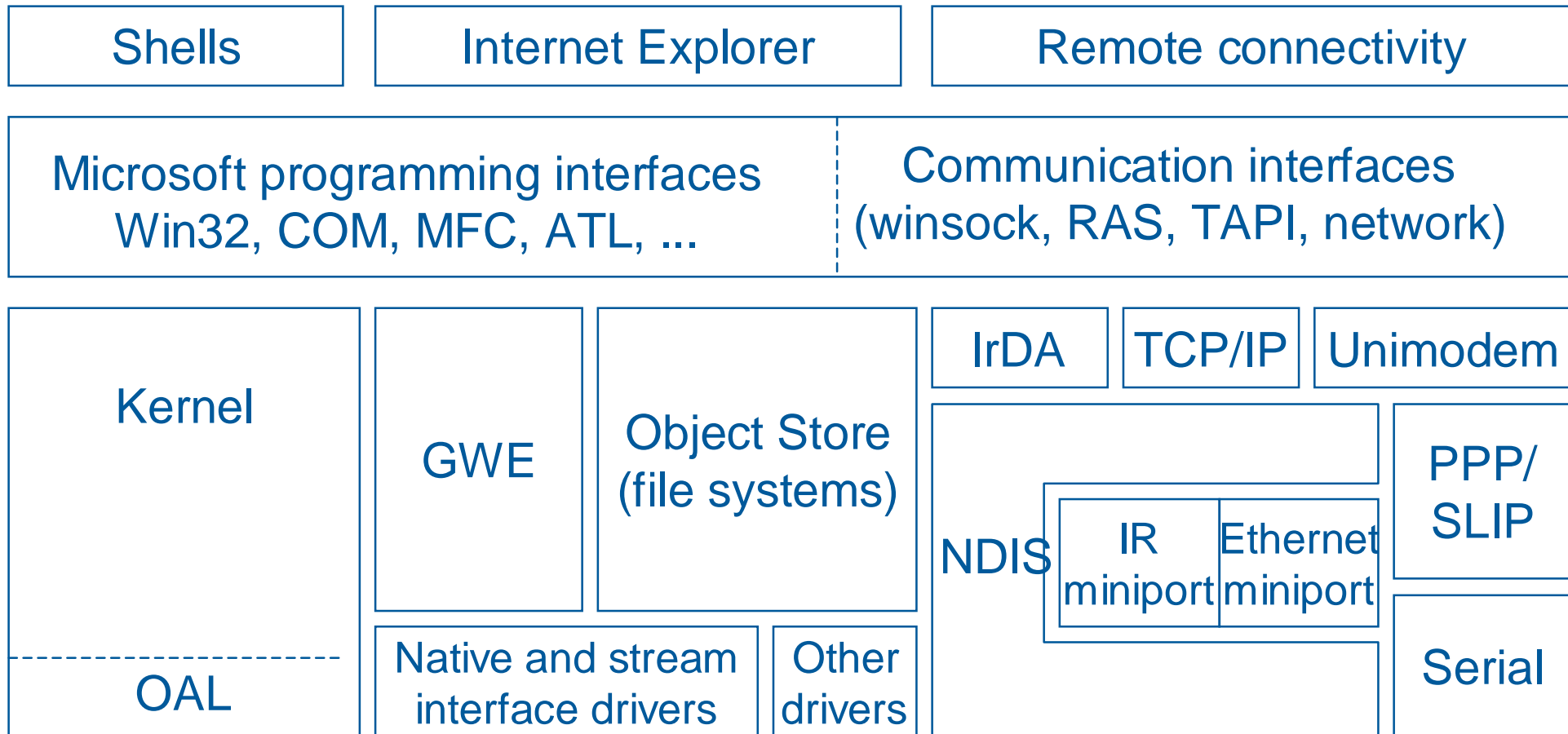
Windows CE: Architektur I

Windows CE hat vier grundlegende Module/Modulgruppen:

- Kernel mit grundlegenden Diensten
 - Prozeß- und Threadbehandlung
 - Speicher-Management
- File-System für permanente Speicherung von Informationen
- Graphics, Windowing, Event Subsystem (GWE)
 - Steuert Grafik und fensterbezogene Features
- Informationsaustausch mit anderen Geräten
- Zusätzliche Module
 - Verwaltung von installierbaren Gerätetreibern
 - Unterstützung von COM/OLE

Windows CE: Architektur II

Windows CE-based applications



Windows CE: Kernel

- Kern des OS, File: Nk.exe, wird vom OEM zusammengestellt.
- Windows CE Kern implementiert:
 - Scheduling, Thread Synchronisation
 - Behandlung von Interrupts und Ausnahmen
 - Virtuelles Speichermanagement
- “Execution in place” (XIP) aus dem ROM
 - Bei Bedarf ersetzt durch Version im RAM
- Portabel
 - 32 Bit CPUs, Little Endian
 - Unterstützung von TLB (Translation Lookaside Buffer) für Adress-Übersetzung virtuell nach physikalisch

Windows CE: Scheduling

- Win32 Prozeß- und Threadmodell
 - Round-Robin mit Prioritäten
 - 32 Prozesse, beliebig viele Threads (ab CE 6.0: 32768 Prozesse)
 - 8 Prioritäten (256 in CE 3.0)
 - Round-Robin pro Prioritätsebene
 - 25 ms Quantum in CE 3.0, einstellbar
 - Thread-Prioritäten sind fest, kein Aging
- Synchronisation
 - Wartefunktionen (WaitForSingleObject(),...)
 - Mutex-Objekte
 - Event-Objekte
 - ...
- Implementation des Priority-Inheritance-Protokolls

- Geänderte Prozess-, Thread- und Speicherverwaltung
- Aufteilung von Nk.exe in kernel.dll (Standardfunktionen) und Nk.exe realisiert *OEM Application Layer (OAL)*
- Thread-Scheduling, 256 Prioritäten (einschränkbar von OEM auf 16, seit 3.0), Fibers
- Virtuelle Speicherverwaltung mit Paging (4k Seiten auf 64k-Regionen), max 512 MB
- Sowohl ROM-Ausführung als auch komprimierte Applikationen
- Emulation für FP-Hardware, monotoner Zähler (unabhängig von Systemzeit)
- Full Kernel Mode - für hohen I/O Durchsatz, keine Sicherheit

Windows CE: Virtueller Speicher I

- Muss auch auf CPUs ohne Protected Mode lauffähig sein
- Geringe Kosten beim Kontextwechsel
- Unterstützung von XIP
- Wiederverwendung bereits geladener DLLs
- Unterstützung von Shared Memory für Interprozesskommunikation

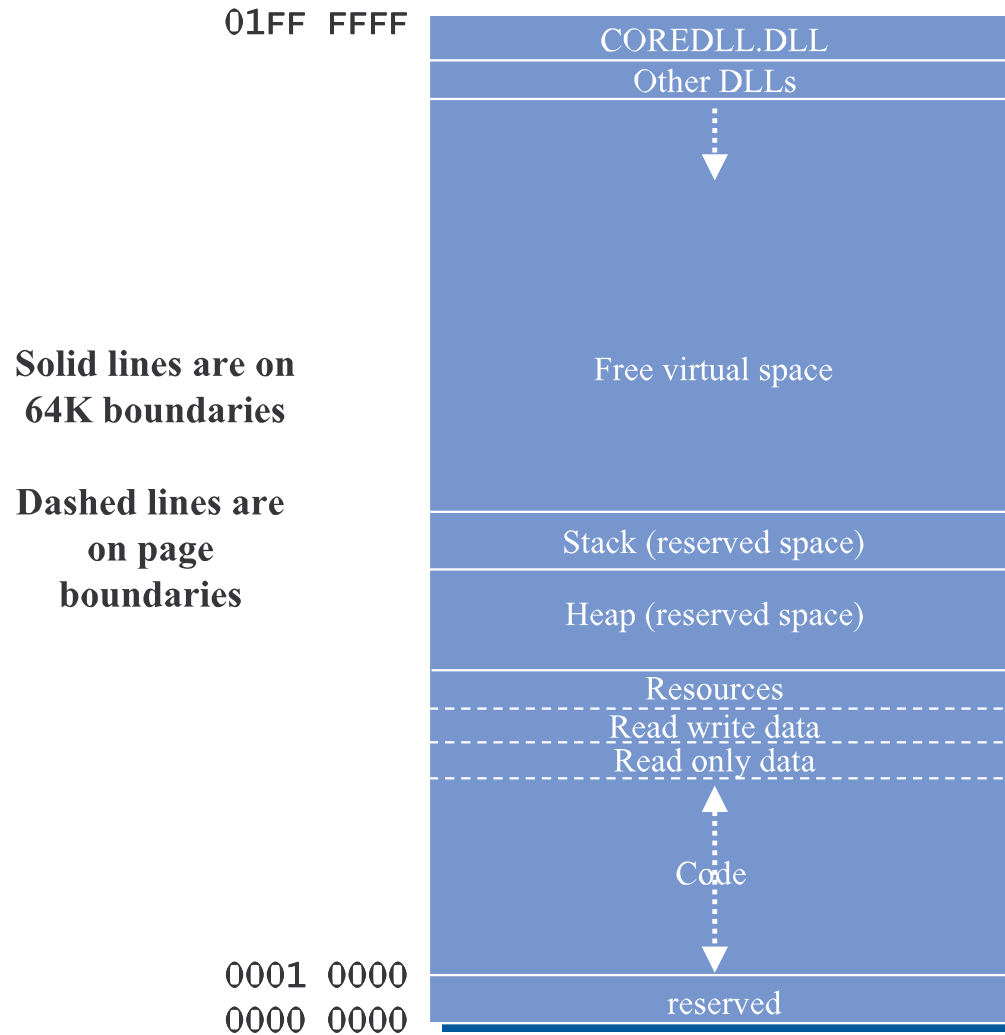
Windows CE: Virtueller Speicher II

- Bis CE 3.0:
- 32 MB vom gesamten virtuellen Speicher pro Prozeß
 - 2 GB Adressraum unterteilt in Slots zu 32 MB
 - Zielgeräte: 4..64 MB RAM, weniger als 16 MB ROM
- Ein Slot pro Prozeß
 - Slot ist eingeteilt in 512 Blöcke von 64 KB
 - Blöcke sind in Seiten von 1 oder 4 KB geteilt (architekturabhängig)
- Zuweisung der Slots
 - Slot 0: Aktiver Prozeß
 - Slot 1: Kernel
 - Slot 2: GWE (Graphics, Window Manager, Event Manager)
 - Slot 3: Filesystem
 - Slot 4: Shell
 - Neue Prozesse bekommen niedrigsten verfügbaren Slot

Probleme mit der Speicherverwaltung

- Vorgestelltes Modell gilt bis CE 3.0
- Probleme:
 - 32 MB ist für manche Anwendungen zu wenig
 - 32 MB müssen für Code, Daten und DLLs geteilt werden
 - DLL-Allokation
 - Speicherallokation an 64 KB Grenzen (ungeschickter Umgang erschöpft 32 MB schnell)

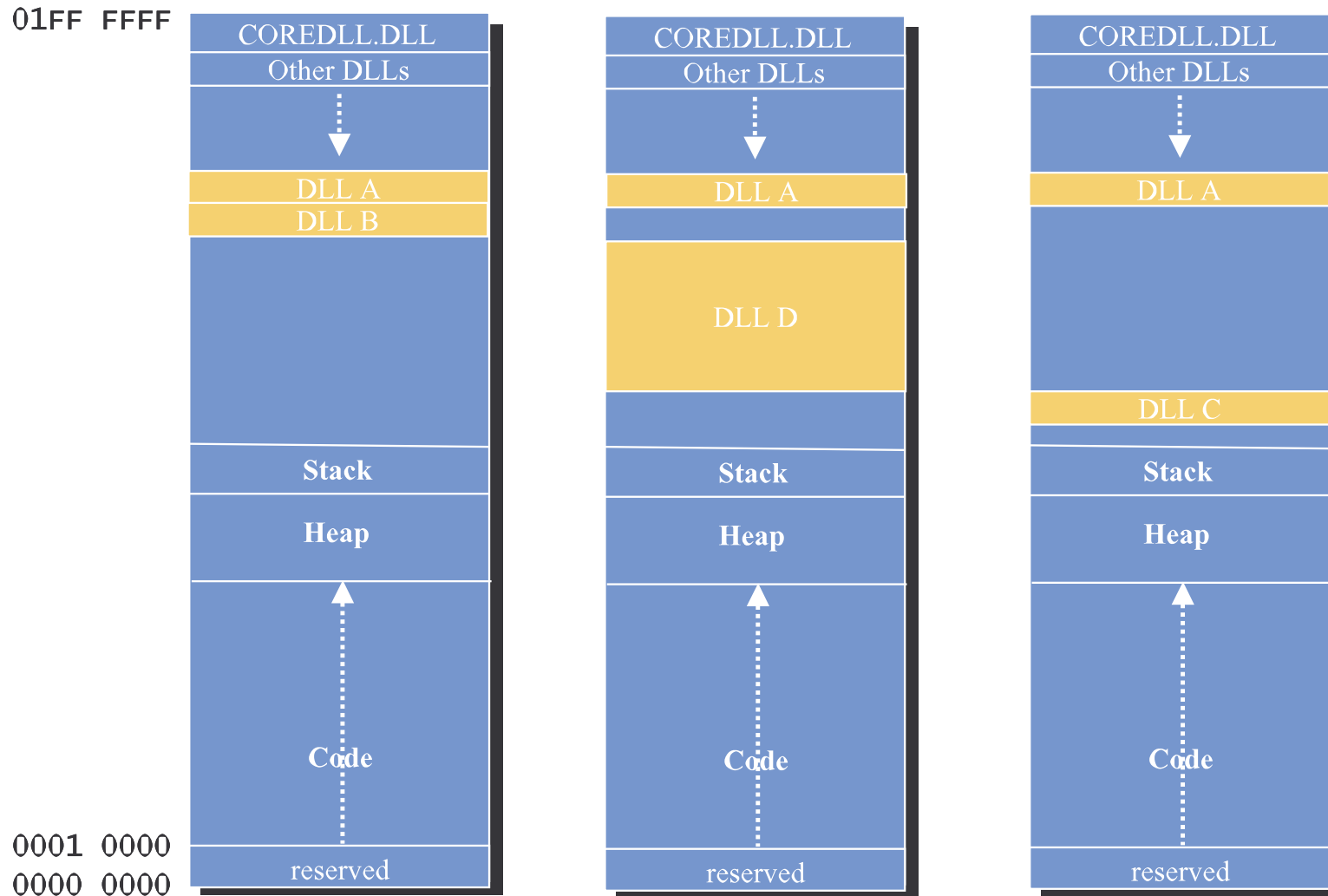
Speicherlayout einer Applikation



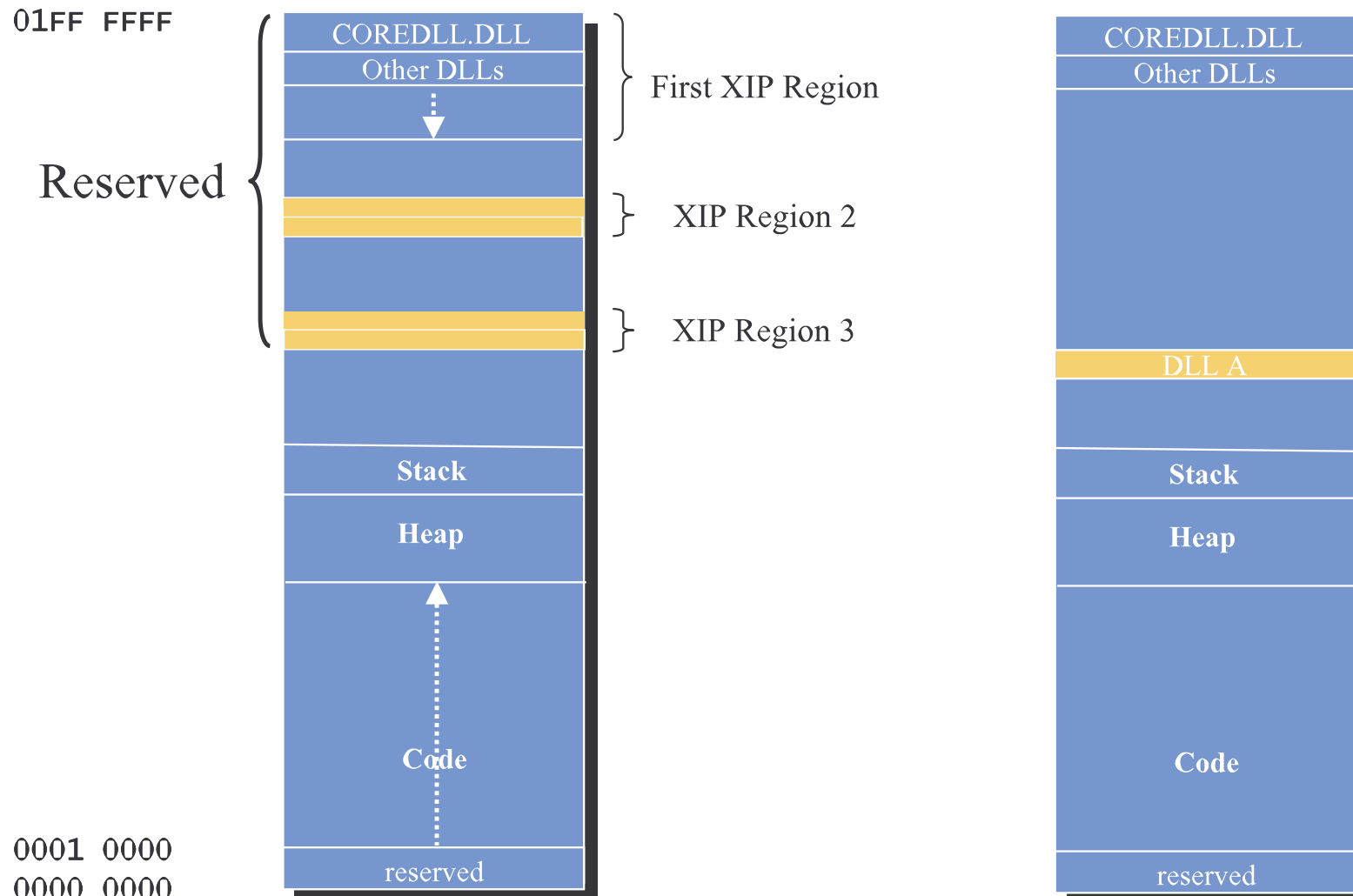
Positionierung von DLLs I

- XIP (execution in place) führt dazu, daß sich DLL-Ladebereiche verschiedener Anwendungen beeinflussen (keine 'Relocation')
 - Lücken sind nicht nutzbar
 - Offene Applikationen erschweren Vorhersagbarkeit
- DLLs werden auf 64 KB Grenzen geladen
- CE 3.0 bietet mehrere XIP-Regionen
 - Bis zu 16 MB Platzbedarf
- Resultat: von 32 MB bleibt unter Umständen weit weniger als die Hälfte

Positionierung von DLLs II



Benutzung mehrerer XIP-Bereiche



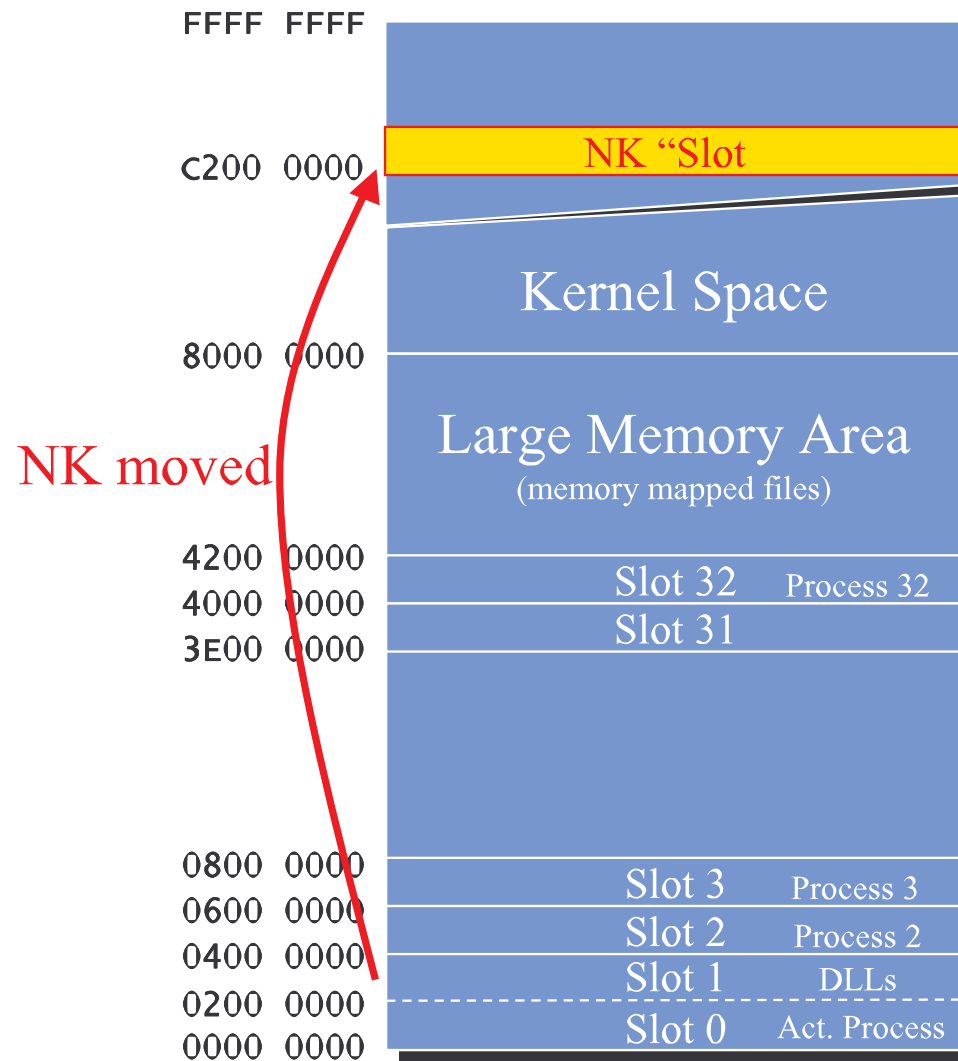
Workarounds für CE 3.0

- Eingebettete Systeme:
 - Wenige XIP-Bereiche benutzen
 - So dicht wie möglich packen
- PocketPC 2002:
 - So wenig DLLs wie möglich benutzen
 - Speichernutzung optimieren
 - * Speicher außerhalb der 32 MB Box allozieren
 - * Speicher „per Hand“ positionieren

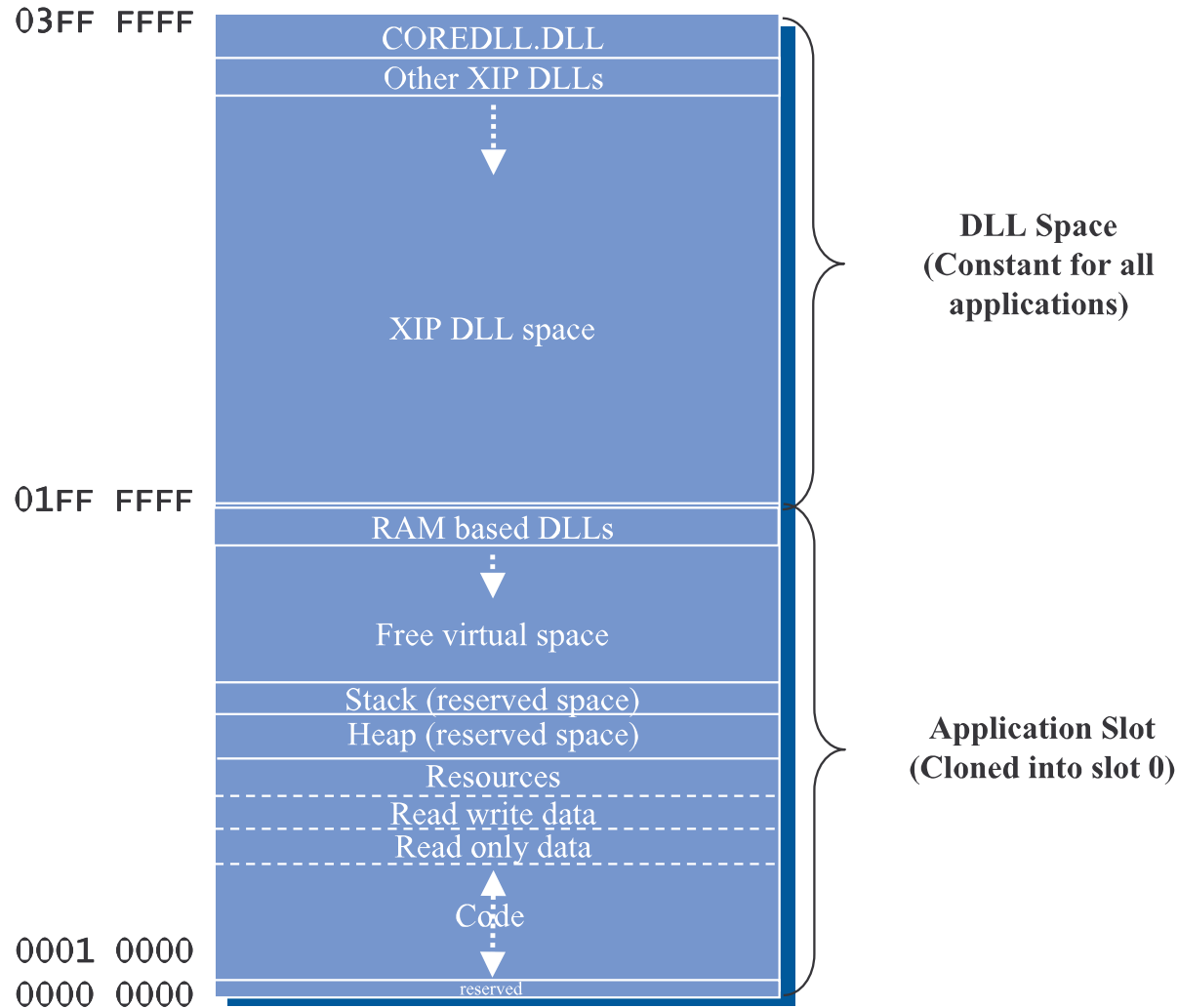
Windows CE .NET: Lösungen I

- Änderung des systemweiten Speicherlayouts
- Verschiebung des Kerns (NK.EXE) aus Slot 1 in den Kernspace (0xC200 0000)
- Slot 1 wird für DLLs benutzt
- Ergibt zusammen mit Slot 0 (aktive Applikation) 64 MB pro Applikation
- Applikation hat 32 MB für Code, Daten und RAM-basierte DLLs
- Keine Möglichkeit, Speicher in oberen 32 MB zu allokkieren

CE.NET: Systemweites Speicherlayout



CE .NET: Speicherlayout einer Applikation



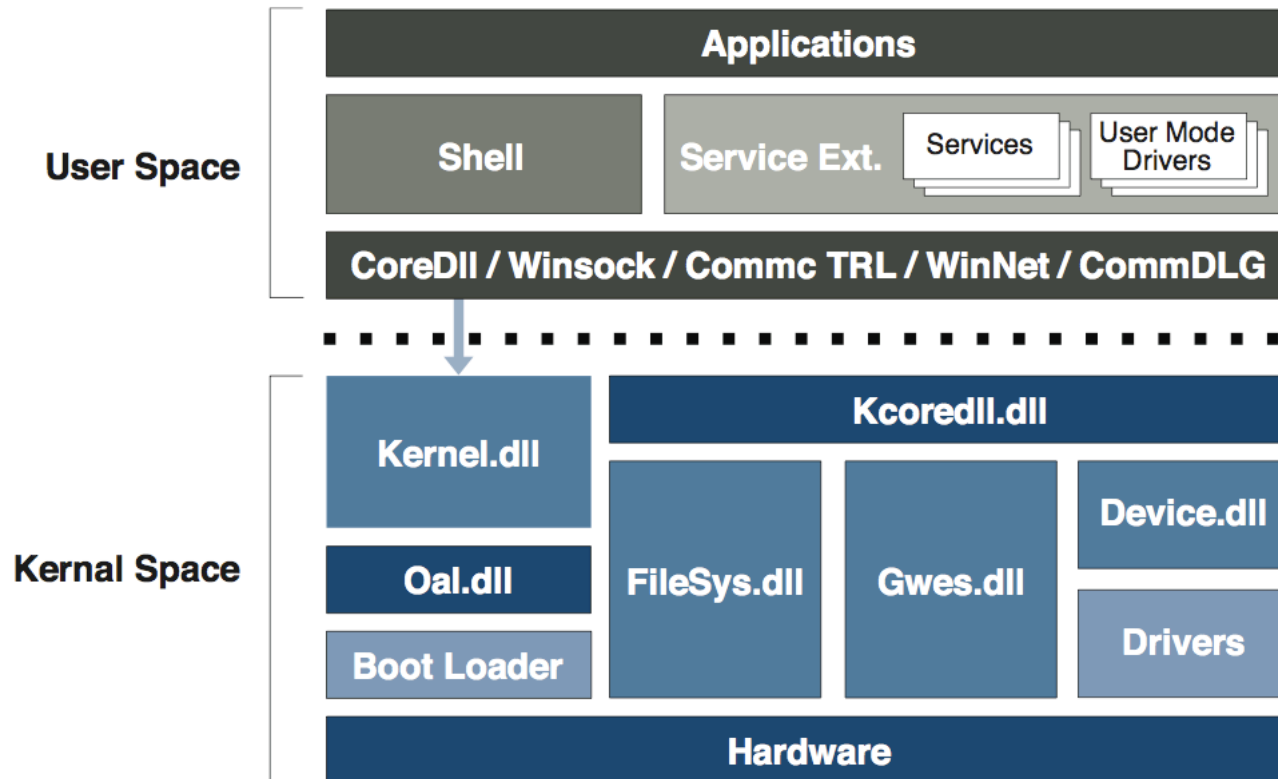
Speicherverwaltung ab CE 6.0

- Komplette Umstellung - keine Slots mehr, dadurch 32768 Prozesse statt bislang 32 Prozesse
- 2 GB (unterer) virtueller Speicher pro Prozess
- 2 GB (oberer) virtueller Speicher für Kernel

CE 6 Kernel - Systemruf

- Systemruf ist ein Funktionsruf in einen anderen Prozess, vermittelt durch Nk.exe
- Wechsel durch Trap oder Ausnahme für unbekannte Adresse, je nach CPU
- Stack und Register bleiben gleich - Thread 'migriert' von User-Prozess in System-Prozess

CE 6 Kernel - Systemruf



- Nk.exe, filesys.exe, gwes.exe, device.exe sind jetzt alle als DLL realisiert, spart IPC-Aufrufe

Windows CE: Dauerhafter Speicher (Filesystem)

- Filesystem bietet dauerhafte Speicherung von Informationen
- FAT-Filesysteme mit bis zu 9 FAT-Volumes
- Transaktionsbasiertes Filehandling
- Demand Paging (je nach Hardware)
- Mirroring zum Schutz vor Datenverlusten bei Ausfall der Energieversorgung oder Kaltstarts
- Installierbare Gerätetreiber für Blockdevices

Windows CE: Kommunikationsinterface

- Unterstützung für serielle Kommunikation, inklusive Infrarot
- Unterstützung für Internet Client Applikationen
- “Common Internet Filesystem” (CIFS) Redirektor für Zugriff auf entfernte Filesysteme über das Internet
- Teilmenge von Windows Sockets (Winsock) Version 1.1
 - Unterstützung für Secure Sockets
- TCP/IP Transportschicht konfigurierbar für drahtlose Netze
 - IrDA für robuste Infrarot-Kommunikation
 - PPP und SLIP für seriellen Netzzugriff
 - Ab CE 5.0 WLAN im System integriert
 - Bluetooth
- Netzwerkzugriff über “network driver interface specification” (NDIS)
 - Management von Telefon-Verbindungen über TAPI
 - Remote Access-Service über Modem

Windows CE: Graphics, Windowing, Event Subsystem (GWE)

- Unterstützung für viele Window-Styles
- Große Anzahl von konfigurierbaren Controls
- Eingabe über Tastatur oder Stift
- Command bar als Vereinigung der Funktionalität einer Toolbar und einer Menu bar
- “Out of Memory”-Dialogbox für Userinteraktion
- Volle UNICODE-Unterstützung

Windows CE: Graphics Device Interface (GDI)

Multiplattform GDI mit folgenden Features

- Farbe und Graustufen, bis zu 32 Bit pro Pixel
- Palettenmanagement
- True Type Fonts und Rasterfonts
- Kontexte für Drucker, Speicher und Displays
- Shape Drawing und Bit Block Transfer

Windows CE: Gerätetreiber

- Gerätetreiber in Usermode Prozessen
- Nur ein kleiner Teil des Gerätetreibers läuft im Kernelmode
 - Führt zu kurzen Interrupt Service Routinen
- Keine verschachtelten Interrupts
 - Interrupts werden in Service Routinen maskiert

Windows CE: Echtzeitfähigkeit

Anforderungen:

- Multithreaded und preemptiv
- Unterstützung von Threadprioritäten
- Vermeidung von Prioritäteninvertierungen
- Vorhersagbare Threadsynchronisation
- Vorhersagbares zeitliches Verhalten des OS:
 - Maximale Zeit der Sperrung von Interrupts
 - Maximale Zeit, die die Behandlung eines Interrupts dauert
 - Maximale Interrupt Latency
 - Maximale Ausführungszeit von Systemrufen
müssen bekannt sein

Windows CE & RT: Prioritäten

- 8 Prioritätsstufen (256 in CE 3.0)
- Level 0 und 1: Real Time und Gerätetreiber
- Level 2 bis 4: Kernelthreads und normale Applikationen
- Level 5 bis 7: Anwendungen, die jederzeit unterbrochen werden können

Priority level	Constant and Description
0 (highest)	THREAD_PRIORITY_TIME_CRITICAL (highest priority)
1	THREAD_PRIORITY_HIGHEST
2	THREAD_PRIORITY_ABOVE_NORMAL
3	THREAD_PRIORITY_NORMAL
4	THREAD_PRIORITY_BELOW_NORMAL
5	THREAD_PRIORITY_LOWEST
6	THREAD_PRIORITY_ABOVE_IDLE
7 (lowest)	THREAD_PRIORITY_IDLE (lowest priority)

Windows CE & RT: Scheduling

- Preemption basiert auf Thread-Priorität
 - Threads mit hoher Priorität laufen zuerst
 - Threads gleicher Priorität: Round Robin
 - Quantum: 25 ms (CE 3.0: einstellbar)
 - Threads niedriger Priorität laufen nicht, bevor alle mit höherer Priorität fertig sind (oder blockieren)
 - Ausnahme: Threads auf Level 0 laufen nichtpreemptiv
- Thread-Prioritäten sind fest
 - Kein Priority-Aging
- Behandlung von Prioritäten-Invertierung
 - Priority-Inheritance-Protokoll

CE 6 Kernel - Scheduler

- 256 Prioritäten
 - 0-96: RT-Treiber ('above'), 97-152: Windows CE Treiber, 153-247 RT-Treiber('below'), 248-255: Nicht-RT
 - Standard ist 252
 - Beispiele: Power Management Resume (99), USB (100), Touch (109), IR (148), Batterieüberwachung (251)
- Prioritätenvererbung um eine Stufe im Scheduler
- Quantum pro Thread einstellbar, Standard durch OAL, 0 für unbeschränkt

Windows CE & RT: Thread-Synchronisation

- Objekte für Thread-Synchronisation
 - kritische Sektion, Mutex, Ereignisse
- Wartequues für Mutex, Events und kritische Sektionen in FIFO-Priority-Order
 - Pro Priorität eine Queue, FIFO innerhalb der Queue
 - Anpassung der Queues mit Hilfe von Priority Inheritance
- Standard Win32 timer API-Funktionen
 - Zeitintervalle über Software-Interrupts
 - “GetTickCount” — Zugriff auf Systemzeit mit Auflösung von ms
 - Performance-Counter für besser auflösende Zeitinformationen

Windows CE & RT: Virtueller Speicher

- Paging I/O läuft auf einem niedrigeren Level als RT-Threads
 - Paging innerhalb der RT-Threads ist möglich
 - Speicherverwaltung im Hintergrund kann RT-Threads nicht beeinflussen
- RT-Threads sollten gelockt werden (ausgenommen vom Paging)
- Windows CE erlaubt Memory Mapping
 - Mehrere Prozesse können gleichen physikalischen Speicher benutzen
 - Schnelle Datentransfers zwischen Prozessen
 - Erhöhung der RT-Performance

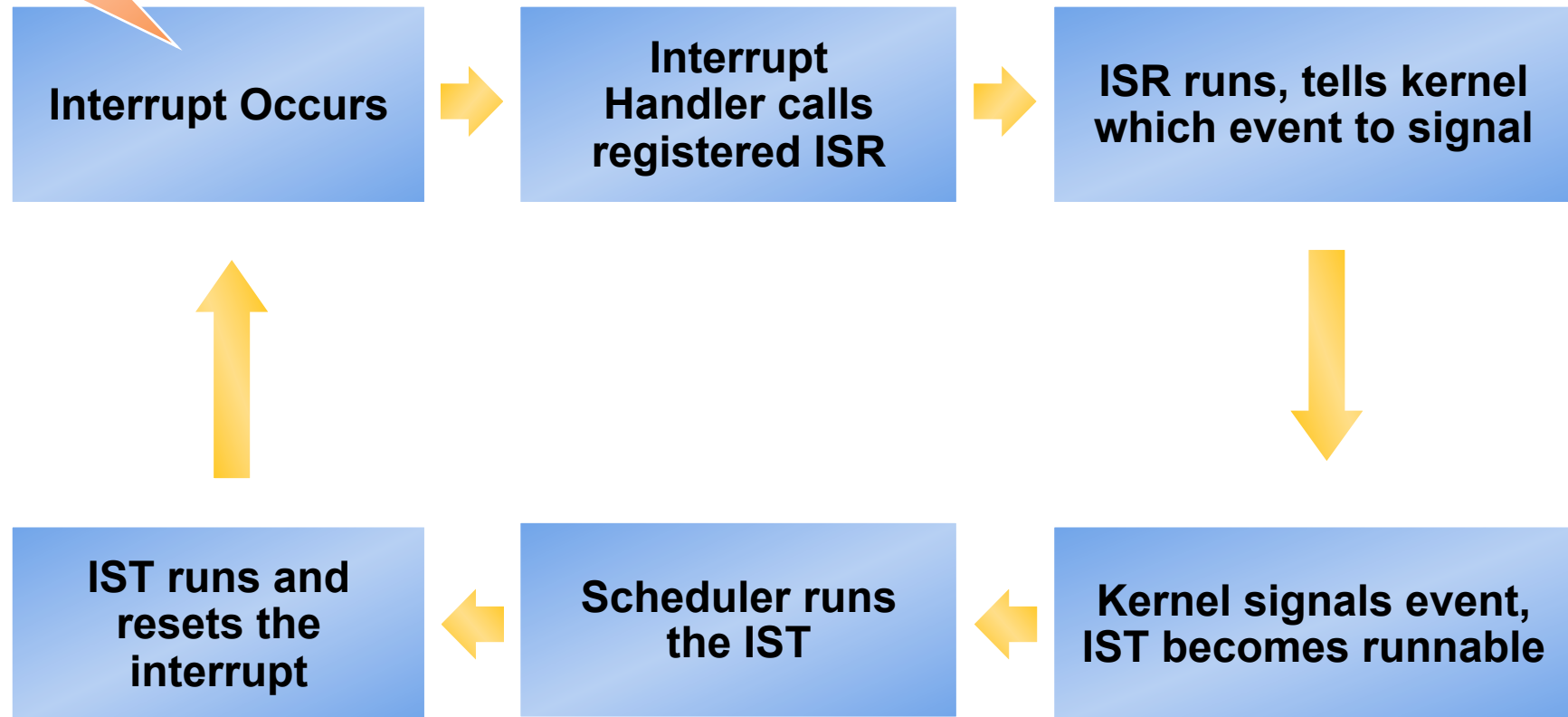
Windows CE & RT: Interrupt-Handling

- CE zerlegt Interrupt-Behandlung in Interrupt Service Routine (ISR) und Interrupt Service Thread (IST)
- ISR sind Hardware-IRQ-Leitungen zugeordnet
 - Auftreten eines Interrupts führt zum Start der ISR
- ISR ist minimal und gibt eine Interrupt ID an den Kern zurück
- Kern wertet ID aus und erzeugt ein entsprechendes Ereignis
- IST wartet auf dieses Ereignis
 - Auftreten des Ereignisses läßt IST die restliche Interruptbehandlung durchführen
 - IST läuft meist auf Prioritätslevel 0 oder 1

CE 6 Kernel - Interruptlatenz

- ISR-Latenz: Zeit von Signalisierung bis ISR-Start
 - Konstant für Threads, die im Speicher gesperrt sind (kein Paging)
 - Zeitaufwand für Kern (Register speichern etc.)
 - Berechnung basiert auf höher-priorisierten Interrupts (Anzahl Auftreten) und ISR-Laufzeit
- Während ISR-Ausführung keine anderen Threads aktiv
- IST-Latenz: Normale Analyse, plus Zeit ohne Unterbrechungen im Kern ('KCall', z.B. Prioritätenvererbung)

Interrupt-Behandlung



Windows CE & Echtzeit

- Windows CE ist Echtzeitbetriebssystem-Familie von Microsoft
- Win32 API konform aber doch grundlegende Unterschiede
- Behandlung von Prioritäteninvertierung
- Unterstützung von verschachtelten Interrupts, obere Grenze für ISR-Ausführung
- 1ms System-Timer
- Thread-Scheduling, Synchronisationsprimitiven

Windows CE & RT: Zusammenfassung

- CE Kernel Design erfüllt die Mindestanforderungen an ein RT-OS
 - OEM hat volle Kontrolle über Interruptbehandlung durch Implementation von ISR und IST
 - OEM hat volle Kontrolle über die Hardware-Interrupts (durch Zuweisung an entsprechende Behandlungs-Software — ISR und IST)
- Interrupt Latency ist bedingt vorhersagbar und begrenzt
- Ausführungszeiten für Systemrufe sind vorhersagbar und unabhängig von der Systemlast
 - Kann unter Benutzung des instrumentierten Kerns untersucht werden
- Keine analytischen Grenzen bekannt

Windows Embedded Familie

- Windows Embedded CE
- Windows Embedded NavReady (CE für Navigationsgerät)
- Windows Embedded Standard (Windows XP / Windows NT für eingebettete Nicht-RT Systeme)
- Windows Embedded POSReady 2009
- Windows Mobile 6 baut auf Windows Embedded CE auf