Systemsimulation am Beispiel von Palm OS

erarbeitet von Lorenz Knies und Swen Meyer

SoSe 2002

Übersicht

1. Palm PDA, Hardware, OS, Software
2. Hardwaresimulation
3. Programmentwicklung
4. Palm Entwicklungswerkzeuge
5. Ausblick

Palm PDA

Übersicht

- Palm PDA - Einleitung
- das Gerät
- Betriebssysteme
- Software und Graffiti®
- Dragonball™ Prozessor
- HotSync®

Palm PDA Einleitung

- Personal Digital Assistant
- Firmengründung im Januar 1992
  - Hauptprodukt war die Software Graffiti®
- Strategie: die Funktionalität eines Desktop-PCs nicht auf PDAs abbilden, sondern neue an die Bedürfnisse angepasste Geräte schaffen
- Entwicklung eines PDA durch Palm Computing der kostengünstig, kompakt und einfach zu bedienen ist
- 1995, U.S. Robotics kauft Palm Computing auf
- Anfang 1996 erscheint der erste PDA von Palm Computing, der Pilot 1000 mit dem PalmOS 1.0
- ende 1996 erscheint der Nachfolger der Palm 5000 mit PalmOS 1.0.3

Palm PDA Einleitung

- der PalmPilot Professionell (´97) besitzt einen eingebauten TCP/IP-Stack, der ihn über die serielle Schnittstelle und eine PPP-Verbindung netzwerkfähig macht
- Juni 1997, 3com übernimmt PDA-Entwicklung
- ab OS 3.2 gibt es die drahtlose Verbindung des Palm ins Internet über ein integriertes Funkmodem mit Sende- und Empfangseinrichtung
- des weiteren wurden Möglichkeiten für Einbindung von Erweiterungskarten geschaffen, wie die Sony MemoryCards etc.
- das aktuellste PalmOS trägt die Versionsnummer 4.0

Palm PDA das Gerät

- Dragonball CPU (DB, EZ, VZ)
- 160x160 Pixel Display (36 cm^2)
- 8 MB RAM (ab VZ max 32 MB)
- Schnittstellen: seriell, IrDA, (USB)
- Graffiti®
- HotSync®
- Slot für Cards

Palm PDA

Vergleich

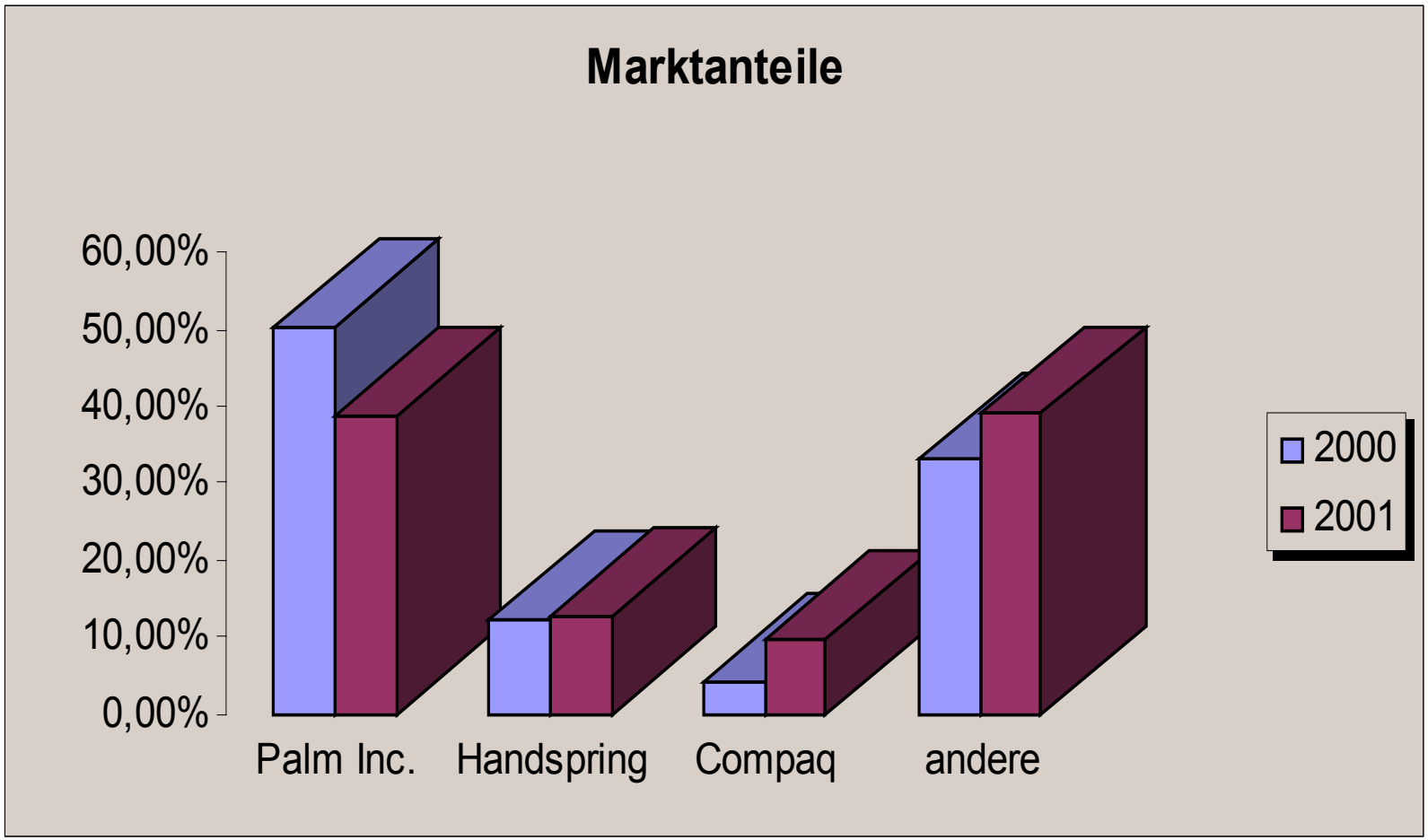
Palm™ Pilot 1000 handheld		128K	32K	1.0	DB	16		E	A
Palm Pilot 5000 handheld		512K	32K	1.0	DB	16		E	A
Palm Pilot 1MB upgrade		1MB	32K	1.0	DB	16		E	A
PalmPilot Personal handheld		512K	32K	2.0	DB	16		EG	A
PalmPilot Pro handheld		1MB	64K	2.0	DB	16	ᵀᴾ	EFG	B
PalmPilot Pro upgrade		1MB	64K	2.0	DB	16	ᵀᴾ	EFG	B
Upgrade for Pilot 1000, Pilot 5000, and PalmPilot Personal									
Palm III handheld	**Go**	2MB	96K	3.0	DB	16	ᵀᴾ 📡 ⚡	EFGS	B
Palm 2MB upgrade		2MB	96K	3.0	DB	16	ᵀᴾ 📡 ⚡	EFGS	B
Upgrade for Pilot 1000, Pilot 5000, and PalmPilot Personal									
Palm IIIc handheld	**Go**	8MB	256K	3.5	EZ	20	ᵀᴾ 📡 ⚡	EFGSJ	E
Rechargable lithium ion batteries									
Palm IIIx handheld	**Go**	4MB	128K	3.1	EZ	16	ᵀᴾ 📡 ⚡	EFGS	C
Palm IIIe handheld	**Go**	2MB	128K	3.3	EZ	16	ᵀᴾ 📡	EFGS	C
English=Palm OS 3.1. Non-English=Palm OS 3.3. SE versions have clear cases									
Palm IIIxe handheld	**Go**	8MB	256K	3.5	EZ	16	ᵀᴾ 📡 ⚡	E	C
Palm V handheld	**Go**	2MB	128K	3.1	EZ	16	ᵀᴾ 📡 ⚡	EFGSI	D
Rechargable lithium ion batteries. Supports web clipping using OmniSky service									
Palm Vx handheld	**Go**	8MB	**F1**	3.3 / 3.5	EZ	20	ᵀᴾ 📡 ⚡	EFIGSJ	D
Rechargable lithium ion batteries. Supports web clipping using OmniSky service									
Palm VII handheld	**Go**	2MB	128K	**F2**	**F2**	16	ᵀᴾ 📡 ⚡	E	C
Built-in wireless connections for web clipping									
Palm VIIx handheld	**Go**	8MB	256K	3.5	EZ	20	ᵀᴾ 📡 ⚡	E	G
Built-in wireless connections for web clipping									
Palm m100 handheld	**Go**	2MB	128K	3.5.1	EZ	16	ᵀᴾ 📡	EJZ	F
Includes two extra silkscreen buttons for contrast and clock.									
Palm m105 handheld	**Go**	8MB	256K	3.5.1	EZ	16	ᵀᴾ 📡	EJZ	F
Includes two extra silkscreen buttons for contrast and clock.									
Palm m500 handheld	**Go**	8MB	256K	4.0	VZ	33	ᵀᴾ 📡 ⚡	EJZ	K
Expansion Card slot and Universal Connector									
Palm m505 handheld	**Go**	8MB	256K	4.0	VZ	33	ᵀᴾ 📡 ⚡	EJZ	L
Expansion Card slot and Universal Connector									

Palm PDA Betriebssysteme

- Palm OS (3Com)
- WinCE (PocketPC Microsoft)
- Epoc (PSION) Symbian
- Linux

Palm PDA

Markt



Software

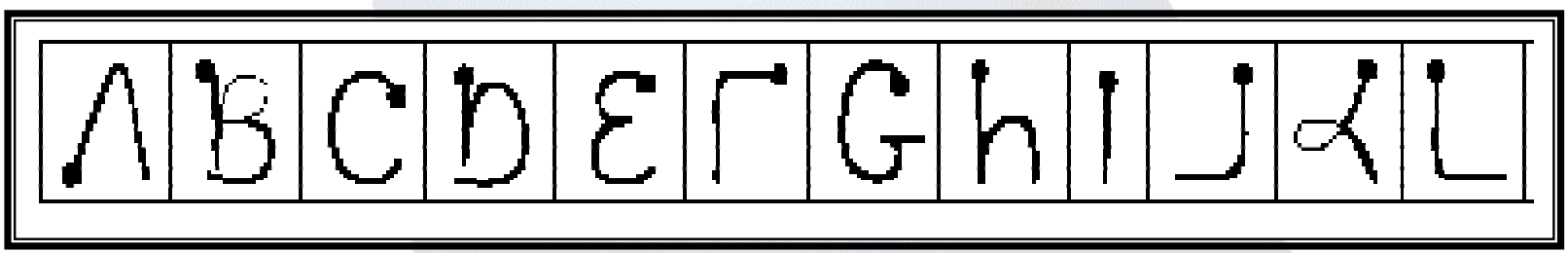
Jeder PDA wird mit mindestens folgender Software ausgeliefert:

- Kalender,
- Aufgabenliste,
- Taschenrechner,
- Kostenabrechnung,
- Desktop PIM (=Personal Information Manager),
- TCP/IP Stack (ermöglicht Netzwerkverbindung und Surfen)
- Adressbuch,
- Notizen,
- Email-Client,

Palm PDA

Graffiti®

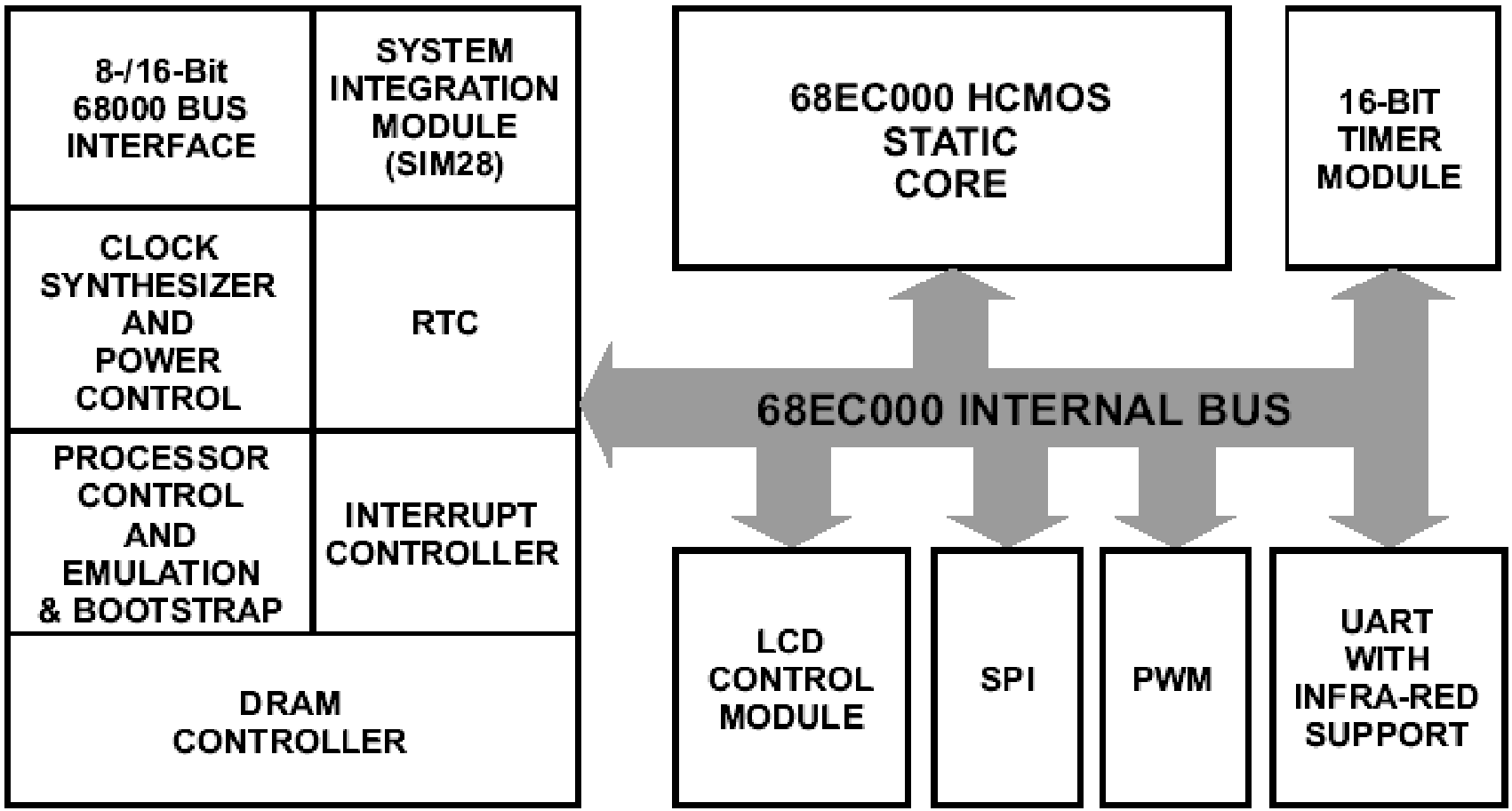
- feste Implementierung der Sprache
  - Jedes Zeichen muss einzeln eingegeben werden
  - die Eingabe eines Zeichens muss ohne Unterbrechung erfolgen (mit einer Eingabelinie)
  - Schrifterkennung verlässlich



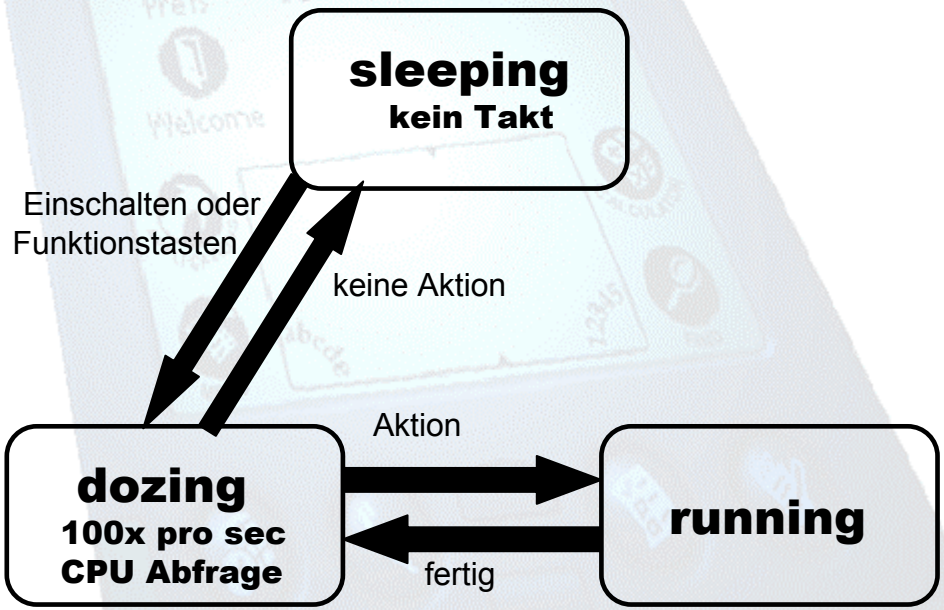
Palm PDA Dragonball

- 16,58 MHz (20 MHz möglich)
- kompatibel zum Motorola 68000
- keine Fließkommaeinheit
- integrierter Interrupt-, DRAM- und LCD Controller
  - Der LCD Controller unterstützt 4 Graustufen
  - (ab VZ Farbe)
- optimiert auf Einsparung von Energie

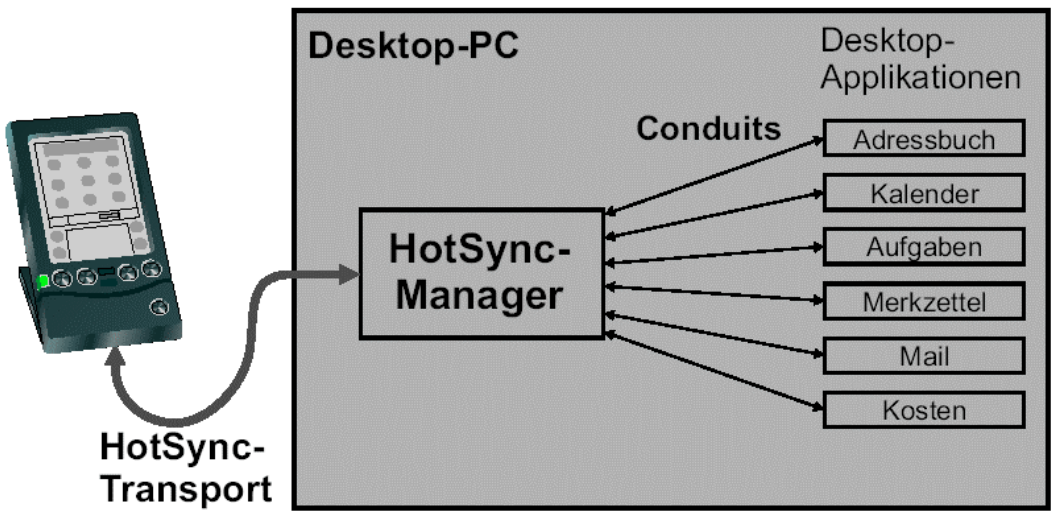
Palm PDA Dragonball



Der DB kennt 3 Grundzustände



- **Conduits**
  - für jede Applikation, die Daten abgleichen will, muss ein eigener Conduit erstellt werden
  - HSM ruft Conduits auf



Übersicht

1. Palm PDA, Hardware, OS, Software
2. **Hardwaresimulation**
3. Programmentwicklung
4. Palm Entwicklungswerkzeuge
5. Ausblick

HW-Simulation

Übersicht

- Softwaremodell des Systems
- Binary Translation
- Hardwarenachbau

Simulation Softwaremodell

- Kernstück ist meistens ein Simulator für den Prozessor (Palm: 68000) und den Hauptspeicher des Systems.
- Aufrufe auf I/O-Geräte werden im Simulator erkannt, abgefangen, und durch entsprechende Aktionen auf dem Hostcomputer ersetzt (z.B. Bildschirmausgabe).
- Grundidee ist meistens ein Array für den Hauptspeicher, und Variablen für die Register der Prozessors.
- Es wird stur ein Befehl aus dem (simulierten) Hauptspeicher geholt, und in einem grossen Case-Statement der passende Befehl simuliert

HW-Simulation Softwaremodell

```
int memory[ PALM_MEMORY_SIZE ];

   int pc; // 68000 program counter
   int regs[]; // 68000 register
   ...

   main() {
       init();
       simulate();
   }

   init() {
       loadMemory( ROMFileName );
       pc = 0;
   }
```

HW-Simulation Softwaremodell

```
simulate() {
   while(true) {
     instruction = memory [ pc ];        // Befehl holen
     pc = pc + 1;                        // PC inkrementieren
     opcode = getOpcode( instruction ); // Befehl dekodieren
     addr   = getAddress( instruction );
     ...
     switch( opcode ) {
       case ADD:                         // code für ADD-Befehl
       ...
       case MULT:                        // code für MULT-Befehl
       ...
       case WRITE:     // write register to memory: IO-Zugriffe
                       // abfangen
       addr = getAddress( instruction );
       data = regs
       if (isIOAddress()) handleIOWrite( instruction );
       else memory[ addr ] = data;
       ...
       case JUMP:
       pc = getJumpAddress( instruction );     ...}}}
```

HW-Simulation Softwaremodell

```
void handleIOWrite( instruction, regs ) {
     int addr = getAddress( instruction );
     if (addr == UART1) writeToUART( ... )
     else if (addr == DISPLAY) writeToDisplay( ... )
...
}
```

- **Vorteil**
  - perfekter Nachbau des gesamten Systems möglich,
  - originales Zeitverhalten
- **Nachteil**
  - langsamer Ablauf (vgl. Java-Interpreter)
  - typ. 10..100 Takte/Host für einen Takt/Target
  - mühsames Nachbauen aller IO-Spezialchips des Targetsystems

Simulation BinaryTranslation

- Rekompilation von Target-Code in entsprechenden Host-Code beim ersten Aufruf,
- Beibehalten des Originalcodes für Notfälle (z.B. selbstmodifizierendes Code),
- Umsetzung von Target-Systemaufrufen in entsprechende Host-Systemaufrufe.

z.B. DEC FX/32 (erlaubt x86-Windows-Programme unter Alpha)
z.B. Java JIT-Compiler

Simulation

Nachbau

- Nachbau der kompletten Hardware
- heute vor allem in FPGA-Technik

(Thema des Vortrags von Björn)

Übersicht

1. Palm PDA, Hardware, OS, Software
2. Hardwaresimulation
3. **Programmentwicklung**
4. Palm Entwicklungswerkzeuge
5. Ausblick

Übersicht

- Besonderheiten
- OS Struktur
- Speicher, Handles
- Forms und Oberflächenelemente
- Events

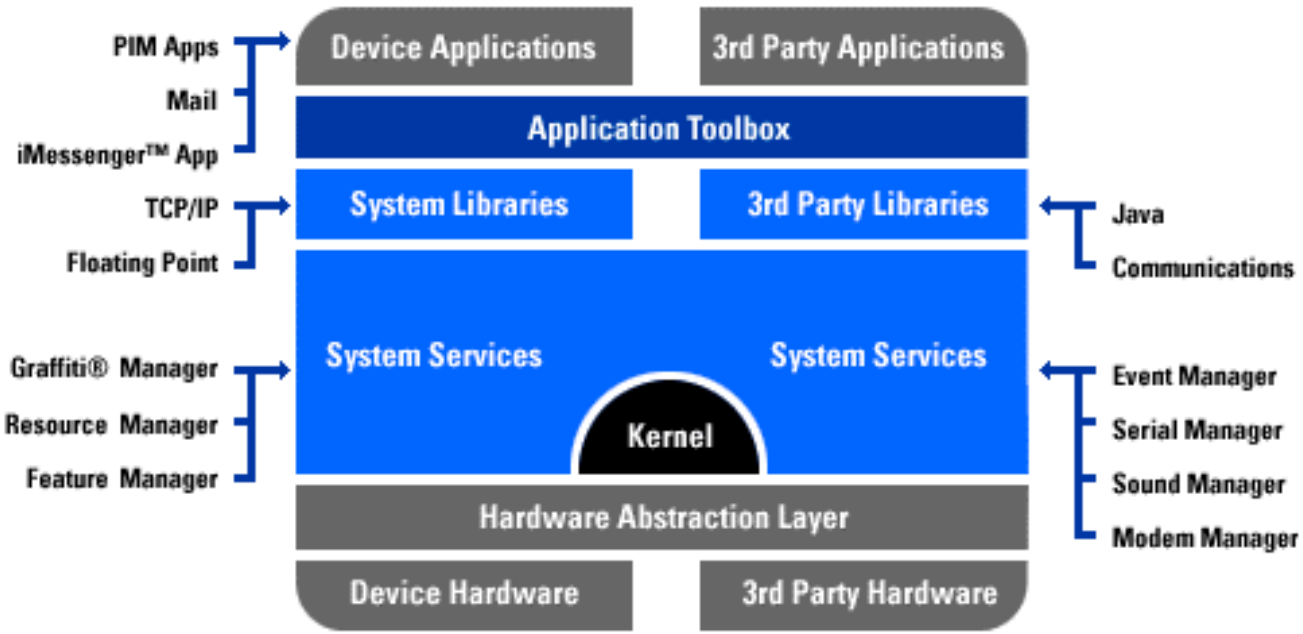
Entwicklung Besonderheiten

- Berücksichtigung des kleinen Bildschirms
- eingeschränkte Texteingabe
- Synchronisation mit Desktop PC
- möglichst kleine Programme
- möglichst effiziente Programme

Entwicklung OS Struktur

- Singletask-Betriebssystem
  - es kann immer nur ein einziger Nutzerprozess laufen, aber weitere Systemprozesse sind erlaubt
- PalmOS kennt für Nutzerprozesse also weder echtes Multitasking, noch Multithreading
- der Kernel selbst ist eigentlich Multitasking tauglich
- Die Betriebssystemfunktionen sind in Manager unterteilt

Entwicklung OS Struktur

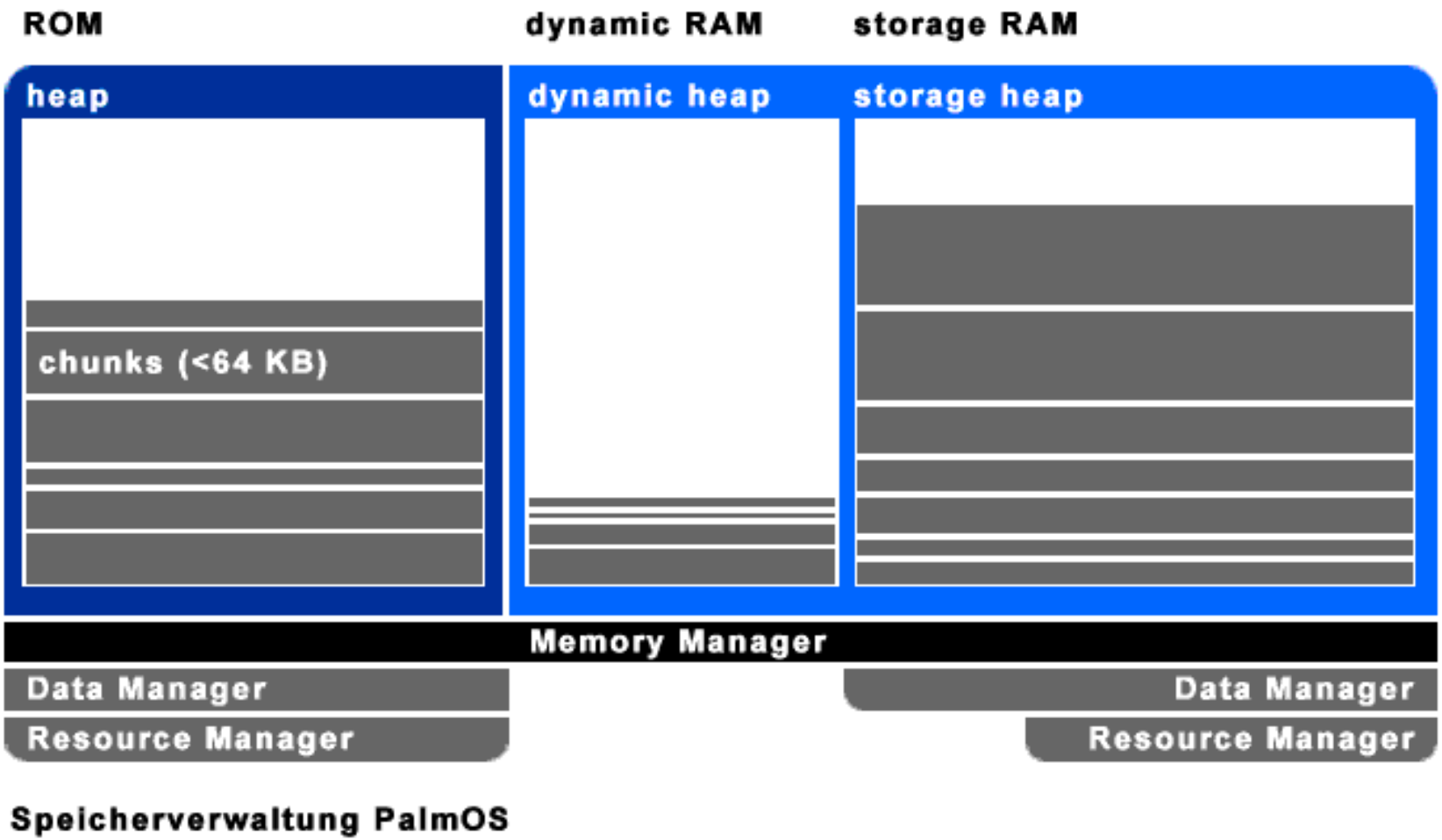


Speicher

- 32-bit Architektur und nutzt daher 32-bit Adressen; Standarddatentypen sind 8, 16 und 32 bits groß
- Speichergrößen bis 4 GB verwaltbar, aber Palm OS wurde entwickelt um effizient mit extrem wenig Speicher (<1MB) arbeiten zu können
- PalmOS unterteilt Speicher in *Heaps*
- Heap kann sich nicht über mehrere Speicherkarten erstrecken
- ein Heap umfasst einen zusammenhängenden Adressbereich; Heaps selber müssen nicht direkt aufeinanderfolgen
- Palm OS besitzt keinen *virtuellen Speicher*, was zur Folge hat das ein Programm im worst case nicht laufen kann, wenn nicht genügend dynamischer Speicher vorhanden ist

Palm PDA

Speicher



Speicherverwaltung PalmOS

Speicher

- Dynamic Memory
  - entspricht RAM auf Desktop Rechnern
  - ein einziger 64KB - 128KB großer Heap (dynamic heap), der für dynamische Speicherzuweisungen genutzt werden kann
  - bei Beendigung eines Programms werden alle auf dem dynamischen Heap angelegten Strukturen gelöscht
  - dynamic heap wird genutzt um z.B. globale Variablen zu speichern, den TCP/IP-Stack oder IrDA-Stack zu laden.
  - Speicheranforderungen über den Memory Manager

Speicher

- Storage Memory
  - enthält alle Daten, die nach dem Ende eines Programms oder nach einem Reset nicht gelöscht werden sollen
  - zu vergleichen mit „Harddisk“ beim Desktop PC
  - da es sich aber um RAM handelt, müssen zu verarbeitende Daten nicht in den Dynamic Memory kopiert werden.
  - lesender Zugriff ist also direkt möglich
  - schreibender Zugriff nur über Betriebssystemfunktionen
  - Speicheranforderungen über den Data Manager

Speicher

- Chunks sind zusammenhängende Speicherbereiche
- Folge: bei stark fragmentierten Heaps können (im worst case) Speicheranforderungen nicht erfüllt werden
- Lösung: es wird ein Defragmentier-Algorithmus gestartet, welcher Speicherblöcke zwischen den Heaps hin und her schiebt, um möglichst große freie Speicherbereiche zu erzeugen

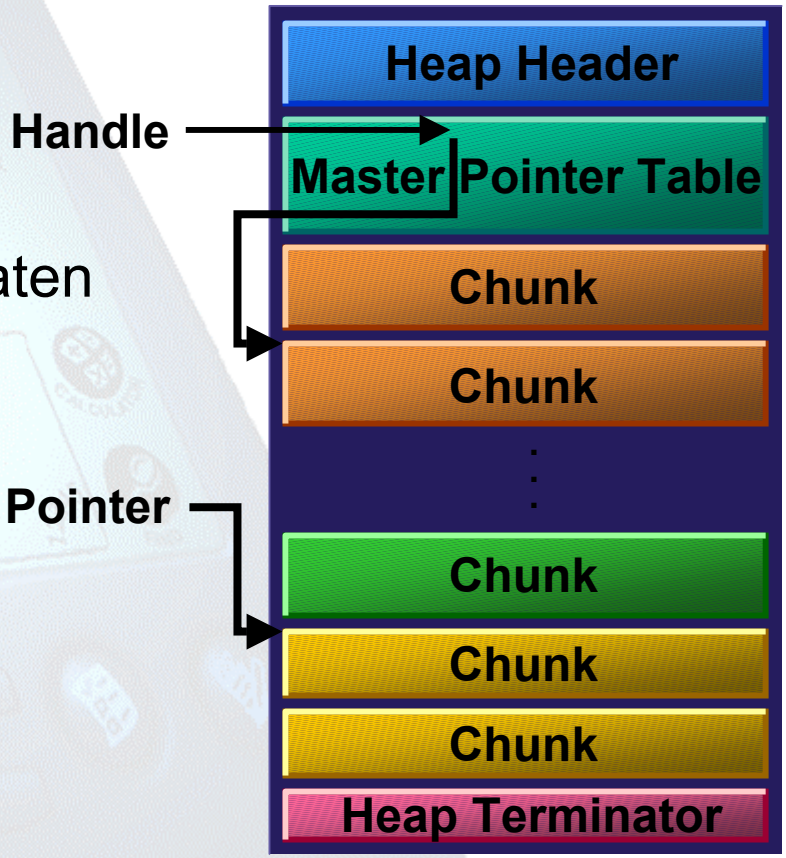
Entwicklung Handles

- Verschiebbarer Speicher wird durch 'Handles' belegt.
- Nicht-verschiebbarer Speicher wird mit Hilfe von 'Pointern' belegt.
- Möchte man auf den Speicher zugreifen, so muss das Handle durch `MemHandleLock` „gelockt“ werden, wodurch man einen Pointer auf den belegten Speicher erhält.
- Solange das Handle gelockt ist, kann der Speicher nicht mehr verschoben werden.

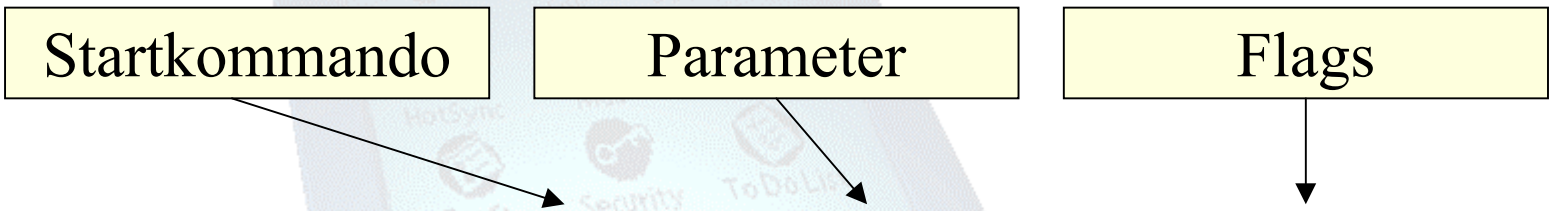
Entwicklung

Handles

- Pointer zeigen direkt auf die Daten
- Handles referenzieren master pointers
- Durch „locken“ des Handles bekommt man den Pointer



Entwicklung main-Routine




```
Uint32  PilotMain(Uint16 cmd, Ptr cmdPBP, Uint16
   launchFlags)
{  ...
   if (cmd == sysAppLaunchCmdNormalLaunch)
   {   error = StartApplication();   // Initialisierung
       ...
       FrmGotoForm(MainForm);  //Start Form Aufrufen
       AppEventLoop();         //Nachrichtenschleife
       StopApplication ();     // Aufräumen
   }
   ...
}
```

- Forms
  - haben etwa die Funktion von Fenstern
  - bilden Grundlage für Menus, Buttons, Checkbox ...
  - Größe beim Compilieren festgelegt
  - Forms können über aktuellem Form gelegt werden

```
void FrmPopupForm (UInt16 formId)
```

  - es kann zu einem bestimmten Form gewechselt werden

```
void FrmGotoForm (UInt16 formId)
```

Entwicklung — Events

- Palm OS Programme sind Eventgesteuert
- Eventtypen
  - System Events (Funktionstasten, Stift, HotSync)
  - Menü Events
  - Application Events

– können im Event-Loop des Tasks vom Event-Manager abgefragt werden

Entwicklung

Events

- Aktive Anwendung erhält alle Nachrichten
- Fertige Handler für System- und Menunachrichten müssen von der Anwendung aufgerufen werden
- Registrierung von Nachrichtenbehandlungsroutinen für Forms möglich
- Zum Beenden wird appStopEvent an die Anwendung geschickt

Übersicht

1. Palm PDA, Hardware, OS, Software
2. Hardwaresimulation
3. Programmentwicklung
4. **Palm Entwicklungswerkzeuge**
5. Ausblick

4 Werkzeuge

Übersicht

- POSE
- IDE
- SDK
- PRC-Tools
- CodeWarrior

Werkzeuge

POSE

- Palm OS Emulator
- entwickelt aus Greg Hewgills copilot / xcopilot
- nach wie vor freie Software
- Sourcecode erhältlich
- Aktuelle Version 3.5 (www.palmos.com)
- Versionen für Windows, MacOS und Unix

Werkzeuge

POSE

Vorteile:

- emuliert mit *ROM-Images* verschiedene Devices
- Debugschnittstelle für Entwicklungsumgebungen
- ermöglicht profiling
- erweiterte Testfunktion (Gremlins)
- Screenshots für Handbücher o.ä. leicht zu erstellen

Werkzeuge

IDE

- Integrated Development Environment
- Softwarepaket, das die Entwicklung von Programmen erleichtert
- Hauptkomponenten:
  - Editor
  - Compiler/Assembler/Linker
  - Debugger
- darüber hinaus:
  - Projektmanager, Versionskontrolle, Wizards und Tools

Werkzeuge

SDK

- Software Development Kit
- SDK enthält include Dateien für
  - Deklaration von Systemfuntionen
  - Definition von Systemkonstanten usw.
- SDK 4.0 aktuelle Version

Werkzeuge

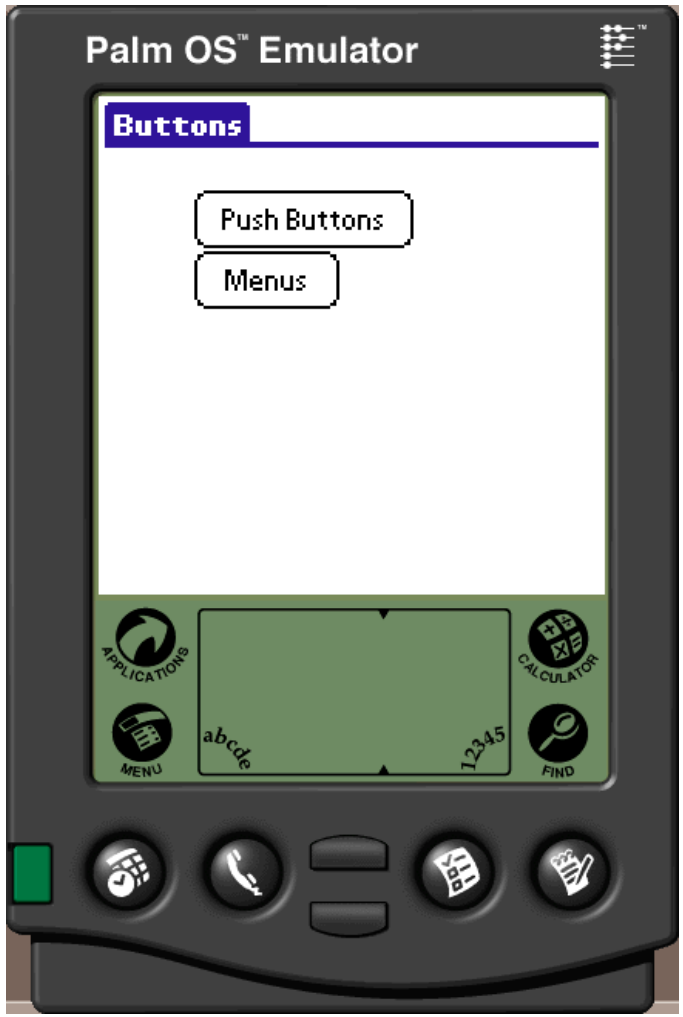
PRC-Tools

- PRC-Tools 2.0
- GCC basiert (benötigt unter Windows CygWin)
- enthalten spezielle m68k-palmos Compiler, Assembler und Linkerversionen des GCC
- reine Kommandozeilen-Tools

Werkzeuge

PilRC

- PilRC 2.8
- Text-Ressource-Definitionen
  → binäre Ressourcedateien (PRC)




```
Beispiel:

FORM ID ButtonsFormID AT (2 2 156 156)
USABLE
MODAL
BEGIN
  TITLE "Buttons"
  BUTTON "Push Buttons" ID  PBButtonID AT (30 30
AUTO AUTO)
  BUTTON "Menus" ID  MButtonID AT (30 50 AUTO AUTO)
END
```

Werkzeuge CodeWarrior

- Aktuell: Version 8.0 (www.metrowerks.com)
- Komplette IDE bestehend aus:
  - Editor
  - Compiler/Linker
  - Debugger
  - Drag&Drop Formulardesigner (Palm Constructor) mit Ressourcecompiler
- Läuft unter MacOS und Windows
- Unterstützte Prozessoren:
  - M68328 Dragonball, M68EZ328 (PalmV/VII), M68VZ328 (Handspring Visor)
- Unterstützt auch 3rd Party SDKs: Symbol, Handspring, Handera, Sony Clié

Übersicht

1. Palm PDA, Hardware, OS, Software
2. Hardwaresimulation
3. Programmentwicklung
4. Palm Entwicklungswerkzeuge
5. **Ausblick**

Palm Pilot - Ausblick

- Intel, Motorola, Texas Instruments und ARM entwickeln optimierte ARM Powered®-CPU-Lösungen für die Palm OS-Plattform
- Einführung des ARM-cored Dragonball™ MX1 (Ende 2002)

Palm Pilot - Ausblick

- XScale: Nachfolger der StrongARM-Prozessoren.
- skalierbare Chip-Mikroarchitektur speziell für PDAs und Handys
  - bei minimalem Stromverbrauch
  - bis zu 1 GHz
  - je nach Belastung zw. 10 mW - 1,6 W
  - kompatibel zu ARM 5.0