Eingebettete Systeme
Echtzeitverhalten und Betriebssysteme

1. Einführung

Software development

Example software development

Trends

- Embedded CPU-Cores are becoming smaller and more powerful
- Higher computing power because of higher clock rate, deeper pipelines, branch prediction, more dynamic decisions, caches, …
- Deeper integration of CPU-cores with
  - programmable hardware
  - (analog) interfaces
  - digital signal processors
  - memory
Trends

Memory becomes a critical resource

Communication plays an important role in embedded systems
- distributed objects DCOM CORBA,
- Firewire,
- Bluetooth,
- USB,
- IrDA,
- Jini,
- TCP/IP,
- ...

Consequences

Critical components of a software development environment:
- Compiler (efficient code generation, predictable execution times of tasks and processes)
- Interfaces ((automatic) generation of device drivers for external units and between hardware and software components)
- Embedded operating systems (memory management, process management, I/O, protection mechanisms, synchronization, communication)

Interfaces

What does a device driver do?
- initialization
  - register with operating system
  - initialize interrupt service
  - allocate memory
  - initialize hardware
- data access
  - data exchange between application and hardware (e.g. memory mapped, DMA)
  - processing of protocol layers
- interrupt service
Interfaces

Automatic software generation is often part of software development environments, e.g. Aisyis DriveWay, Stenkil MakeApp, Intel ApBuilde, Motorola MCUnit, CoWare, …

- Properties of devices are usually stored in a database; software generation has access to this database
- Specific drivers between hardware and software components are synthesized

Compiler

Because of complexity of computer architectures and applications, we find more and more software written in high level languages like C++, Java, Ada.

Because of cost constraints (just a few installations, many different processor families exist), the compiler support for embedded systems is rather poor (RedHat Cygnus, TI Tartan, Green Hills). Demand for retargetable compilers is growing.

Compiler

The compilation quality for embedded microprocessors (e.g. ARM, MIPS, PowerPC, …) is satisfying whereas the code quality for DSPs is rather poor.

- Specialized data paths, e.g. multiply-accumulate
- Multiple memory banks and buses
- Specialized addressing modes, e.g. circular buffers
- Specialized instruction sets, e.g. zero overhead loops
- Specialized peripheral units

Important criteria

- Short and predictable execution times of processes
- Efficient use of memory structure
- Small code size
Compiler

Operating systems (OS)

- Software layer between application and hardware.
- **Major task** of an OS is the dynamic management of memory, communication and computing resources:
  - **Memory management** (main memory, secondary memory, virtual memory, memory protection, paging/segmentation, file system)
  - **Process management** (scheduling, task queuing, synchronization, communication)
  - **Security and protection mechanisms**
  - **I/O management** (device driver, scheduling, networking)
  - **Support for distributed** hardware resources and distributed applications
- **Desirable features**: Timeliness, design for peak load, predictability, fault tolerance, maintainability

Process management

- **Important tasks of process management**:
  - Execution of **quasi-parallel tasks** on a processor using processes or threads (lightweight process) by
    - maintaining process states,
    - process queuing,
    - preemptive tasks (fast context switching) and quick interrupt handling
    - dealing with dynamically generated and terminated processes
  - **CPU scheduling** (guaranteeing deadlines, minimizing process waiting times, fairness in granting resources such as computing power)
  - **Process synchronization** (critical sections, semaphores, monitors, mutual exclusion)
  - **Inter-process communication** (buffering)
  - Support of a **real-time clock** as an internal time reference
  - **Common use of resources**

- **Minimal set of process states**: