

Programování mikrokontrolerů

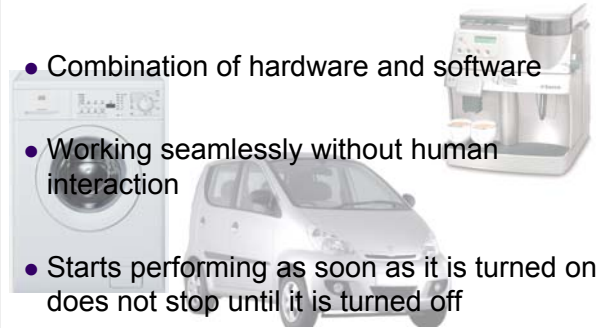
Embedded Systems



Embedded System



- Combination of hardware and software
- Working seamlessly without human interaction
- Starts performing as soon as it is turned on, does not stop until it is turned off



Typical attributes



- Single function
- Constrained
- Real-time or at least Reactive

Design process



- Software and hardware design is indivisible process
1. Requirements specification
 2. Select appropriate software architecture
 3. Select minimal hardware architecture
 4. Repeat 2. and 3. until compromise is set

Support Tools



- Hardware
 - Prototyping boards
 - Development kits
- Software
 - Emulators
 - Simulators
 - Cross-compilers
 - Monitors
 - Cross-debuggers



Software Architectures



- Endless polling
- Event handling
- Task scheduling
- Real-time operating systems
- The simplest is the best!

Polling – Round-Robin



- I/O polling in a loop, running tasks as needed
- Example – digital voltmeter
 1. check scale
 2. read voltage
 3. display voltage
 4. goto 1

7

Polling – Round-Robin II



- Pros
 - simplicity
- Cons
 - max waiting time = loop length
 - handling delays may compromise correctness

8

Round-Robin with Interrupts



- Urgent events generate interrupts:
 - high priority tasks via IRQ
 - low priority tasks in main loop
- Example – Full duplex Bridge
 - ISR1 – receive on port 1, insert into buffer 2
 - ISR2 – receive on port 2, insert into buffer 1
 - main loop:
 1. if port 1 free and data in buffer 1, forward it
 2. if port 2 free and data in buffer 2, forward it

9

Round-Robin with Interrupts II



- Pros
 - ability to handle different priority
 - possibly different IRQ priorities
- Cons
 - same priority for Round-robin tasks
 - Round-robin tasks may be overrun by IRQ tasks
 - Race conditions

10

Function Queue



- ISR only insert tasks into queue
 - may implement priorities
- Main loop activates task at queue head

11

Function Queue II



- Pros
 - Ability to define priority scheme
- Cons
 - long tasks affect response time
 - higher priority tasks must wait for current task finish

12

Scheduling

- Endless loop
- Cyclic Executive
- Time-driven Cyclic Executive
- Multi-rate Cyclic Executive
- Function Queue

+ interrupts



13

Scheduling II

- Preemptive scheduling
- Priority driven preemptive
- Earliest deadline first
- Adaptive partition

+ interrupts



14

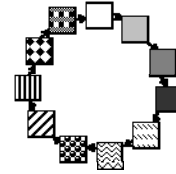
Endless loop

```
DO FOREVER
  Request Input Device make a Measurement
  Wait for the Measurement to be ready
  Fetch the Value of the Measurement
  Process the Value of the Measurement
  IF Value is Reasonable
    THEN Prepare new Result using Value
    ELSE Report an Error
  END-IF
  Request Output Device to deliver the Result
  Wait for the Result to be output
  Confirm that output is OK
END DO
```



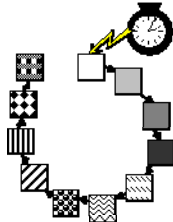
15

Cyclic Executive



16

Time-driven Cyclic Executive



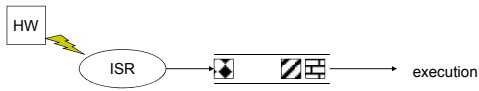
17

Multi-rate Cyclic Executive



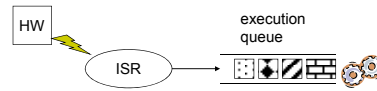
18

Function Queue



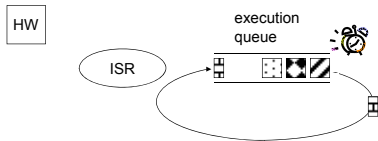
19

Preemptive scheduling



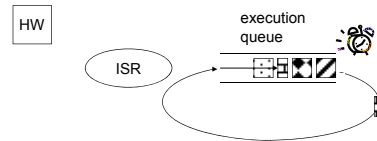
20

Preemptive scheduling



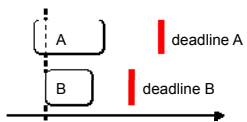
21

Priority driven Preemptive



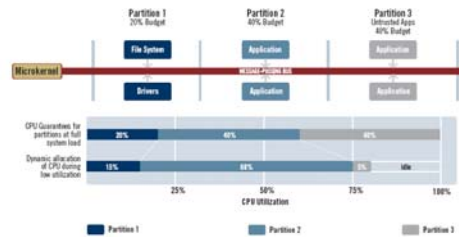
22

Earliest Deadline



23

Adaptive Partition



24

Real-Time Systems I



- Correctness of operation
 - logical correctness
 - performance time
 - response time
- Real-time \neq high performance computing

25

Real-time Systems II



- Hard
 - the completion of an operation after its deadline is considered useless
- Soft (immediate)
 - lateness tolerance
 - best effort

26

Real-time Systems III



- Pros
 - improved application development
 - improved predictability
- Cons
 - OS consumes resources
 - complex to develop the OS

27

Real-time Systems IV



Once you decide to use real-time OS,
your best design is the one
where you use it the least.

28