

# Real Time Control

Chapter 15

Gunnar Lindstedt

automation 2022

# Real-time Programming Important Features

- Execution flow determined by external events
- Act on data and on external signals (sensors)
- May explicitly refer to time
- Timing constraints
- The result cannot be predicted
- Multitasking
- The program never terminates – waits for new data

automation 2022

# Important Concepts

- **Multitasking**
- Scheduling – process states
- Resource protection
- Process synchronization
- Periodic execution

automation 2022

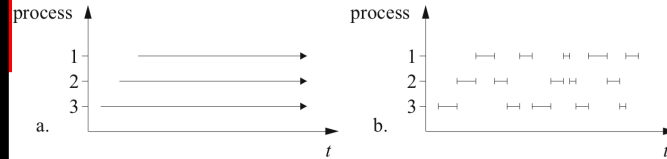
# Multitasking

Letting several processes run concurrently on one or several CPUs

The diagram shows a multitasking structure. On the left, a task graph consists of a central node with four arrows pointing to four parallel task blocks, each containing a smaller node and a rectangle. On the right, a flowchart shows a box labeled 'cobegin' at the top with four arrows pointing down to four boxes labeled 'S1', 'S2', '...', and 'Sn'. From the bottom of each 'Si' box, an arrow points down to a single box labeled 'coend'.

automation 2022

## Multitasking



automation 2022

## Managing Multitasking

- Divide the CPU between several processes (activities)
- Context switching
- A scheduler
- Pre-emptive scheduling
- Time slice

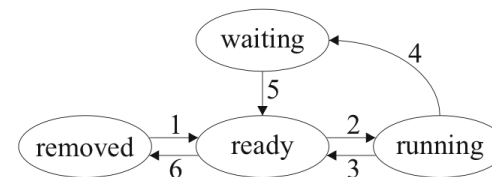
automation 2022

## Important Concepts

- Multitasking
- Scheduling – process states
- Resource protection
- Process synchronization
- Periodic execution

automation 2022

## The States of a Process



automation 2022

## Scheduling Strategies

- Cooperative scheduling – round robin
- Interrupt scheduling – priorities
- Pre-emptive scheduling
- Time-slice scheduling

automation 2022

## Important Concepts

- Multitasking
- Scheduling – process states
- Resource protection
- Process synchronization
- Periodic execution

automation 2022

## Resources

- CPU
- Disk areas, memory cells
- I/O units
- Variables
- Bus connection
- ....

automation 2022

## Mutual Exclusion

- Safety – access limits have to be respected. Protected area is not accessed by more than 1 process at a time
- Liveness – a program is supposed to do what it is meant to do. No indefinite blocking

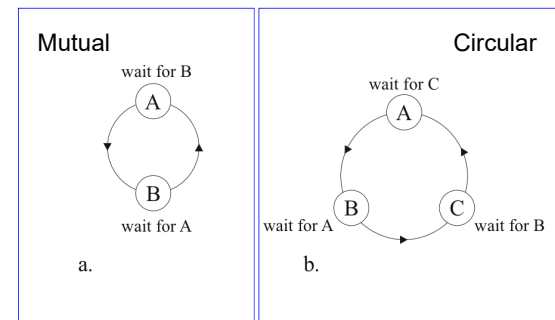
automation 2022

## Creating a Deadlock

- Two processes P1 and P2
  - Two resources r1 and r2 (a certain I/O or a certain memory position)
1. P1 reserves r1 and P2 reserves r2
  2. P1 needs also r2 and will wait for P2
  3. P2 needs r1 and will wait for P1

automation 2022

## Deadlock



automation 2022

## Conditions to Create Deadlock

- Mutual exclusion – some resources can only be reserved by 1 process at a time
- Non-preemptive allocation – a resource can only be released by the one that made the reservation
- Successive allocation
- Inverse allocation

automation 2022

## The Race Condition

{process p1}	{process p2}
load x	load x
x:=x+1	x:=x-1
store x	store x

**Some operations may not be interrupted!**

automation 2022

## Using Semaphores

Semaphore sem – an integer

**Signal:** increase the value with 1 (unconditionally)

**Wait:** if sem>0 the decrease the value with 1;  
otherwise wait

automation 2022

## Protecting Resources

```
Program sem_example  
var sem: semaphore  
begin
```

```
  sem:=1  
  cobegin
```

```
    while true do  
      begin (*Process P1*)  
        wait (sem);  
        (*protected resource*)  
        signal (sem);  
        ....  
      end; (*process P1*)
```

```
  coend;  
end
```

```
  while true do  
    begin (*process P2*)  
      wait (sem);  
      (*protected resource*)  
      signal (sem);  
      ....  
    end; (*process P2*)
```

automation 2022

## Important Concepts

- Multitasking
- Scheduling – process states
- Resource protection
- **Process synchronization**
- Periodic execution

automation 2022

## Synchronization (single-sided)

```
(*process read data*)  
while true do  
begin
```

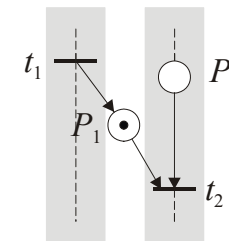
```
  ('get data*)  
  signal (data_available)
```

```
end;
```

```
(*process operate on data*)  
while true do  
begin
```

```
  wait (data_available)  
  ('operate on data*)
```

```
end;
```



automation 2022

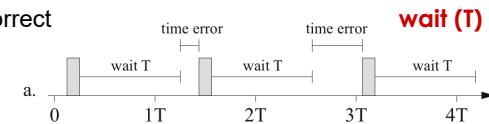
## Important Concepts

- Multitasking
- Scheduling – process states
- Resource protection
- Process synchronization
- **Periodic execution**

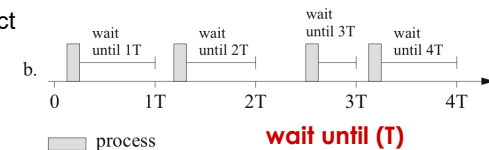
automation 2022

## Execution of Periodic Tasks

Not correct



Correct



automation 2022

## Real-time Programs

- Be ready to run
- Never terminate
- If not running – wait
- Running efficiently
- Structured (difficult to debug!)

automation 2022

## Real-time Language Features

- Definition of processes that can be executed in parallel
- Priority driven process switch
- Synchronization
- Data exchange among processes
- Direct access to external hardware
- Time related functions
- Interrupt and exception handling

automation 2022

## Summary

- Real time programs never execute deterministically
- Several tasks in parallel
- Scheduling
- Resource protection
- Process synchronization
- Periodic execution

More to learn in  
courses on  
*Real time programming*