# Operating System Architectures

## *History of OS Structuring*

- ➜ Monolithic kernels
- ➜ Microkernels
- ➜ Object-based open operating systems





# Monolithic Operating Systems

- ❏ One kernel for all OS functions
- ❏ Execuction in privileged mode

- \+ Efficient communication between kernel components
- \+ Direct access to all data structures in the kernel
- \+ Privileged instructions directly executable
- – Functionality not adaptable to individual application needs
- – Often bad structuring $\rightarrow$ error prone
- – No protection against buggy or malicious kernel components





# Microkernels

- ❏ Take OS mechanisms out of the kernel
- ❏ Reduce kernel to mechanisms which have to be executed in privileged mode

- \+ Better modularity
- \+ More flexibility with external OS modules
- \+ No privileges for external OS mechanisms

- – Communication between external modules expensive
  (RPC, factor 20 to 70 compared to a normal procedure call)





# Object-based Open Operating Systems

- ❏ Continuation of the trend ’monolithic kernel $\rightarrow$ microkernel’
- ❏ Further modularity needs less communication costs
  - ➜ no separate address spaces
  - ➜ less safety
  - ➜ protection at programming-language level

- \+ Flexibilty and modularity of microkernel combined with efficiency of monolithic kernels

# Future Operating Systems

- ❏ Adaptabilty to application needs
  - – Efficency
  - – Security
  - – Resource management strategies
- ➔ Applications must not be forced to work around OS mechanisms

- ❏ Higher level of abstraction for building OS mechanisms