Betriebssysteme

Operating System Structures

Prof. Dr.-Ing. Frank Bellosa | WT 2016/2017

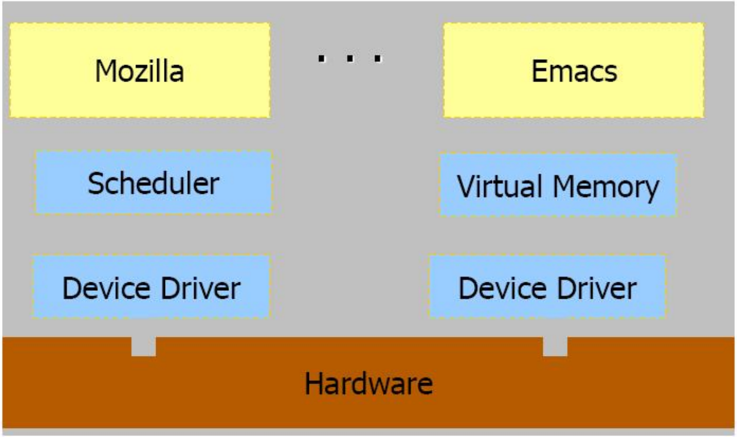
KARLSRUHE INSTITUTE OF TECHNOLOGY (KIT) – OPERATING SYSTEMS GROUP

(c) Philipp Kupferschmied

KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Monolithic Systems

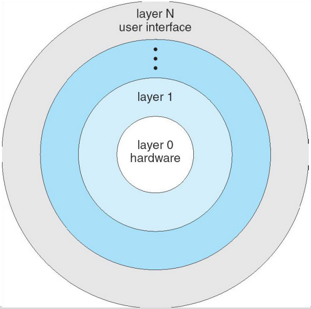


Monolithic Systems: Pros and Cons

- Advantages
  - Well understood
  - Easy access to all system data (they are all shared)
  - Cost of module interactions is low (procedure call)
  - Extensible via interface definitions
- Disadvantages
  - No protection between system and application
  - Not stable or robust
- Examples
  - uCLinux, PalmOS, VxWorks, OSEK/VDX, eCos

Layered Systems

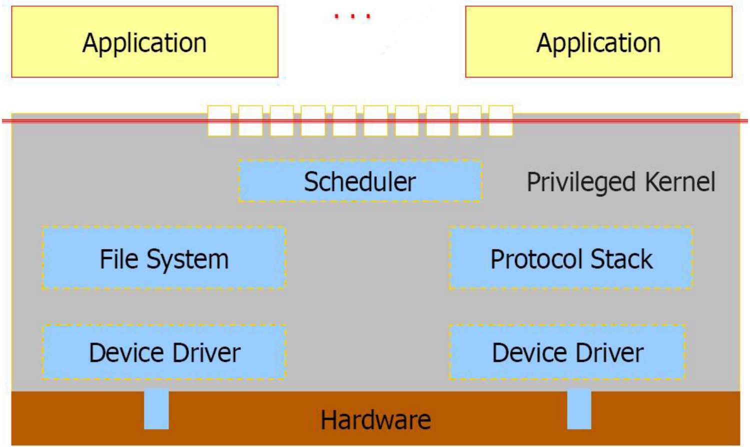
- System is divided into many layers (levels)
  - Each layer uses functions (operations) and services of lower layers
  - Bottom layer (layer 0) is hardware
    - Easier migration between platforms
    - Easier evolution of hardware platform
  - Highest layer (layer N) is the user interface
  - Lower layers implement mechanisms
  - Upper layers implement policies (mostly)



Layered Systems: Pros and Cons

- Advantages
  - Each layer can be tested and verified independently
  - Correctness of layer N only depends on layer N-1
  - → Simpler debugging/maintenance
- Disadvantages
  - Just unidirectional protection
  - Mutual dependencies (e.g., calls between process, memory and file management) prevent strict layering
    - Need to reschedule processor while waiting for paging
    - May need to page in information about tasks
    - Memory would like to use files for its backing store
    - File system requires memory services for its buffers
- Examples
  - THE (Dijkstra), Multics(GE), VOCOS(EWSD)

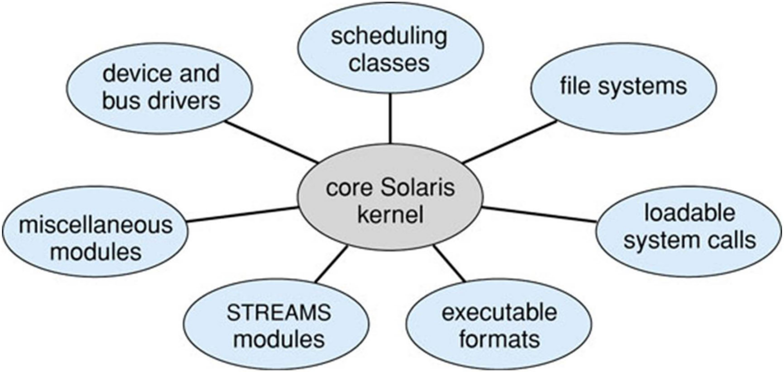
Monolithic Kernels



Monolithic Kernels: Pros and Cons

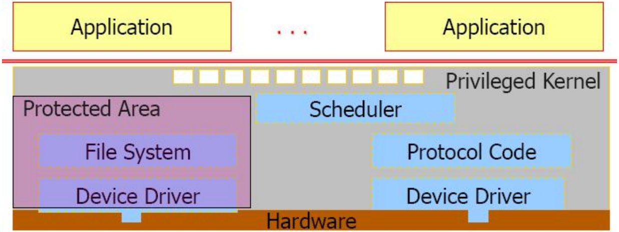
- Advantages:
  - Well understood
  - “Good¨ performance
  - Sufficient protection between applications
  - Extensible via interface definitions and static/loadable modules
    - Uses object-oriented approach
    - Each core component is separate
    - Each talks to the others over known interfaces
    - Each is loadable as needed within the kernel
- Disadvantages:
  - No protection between kernel components
  - Side-effects by undocumented interfaces
  - Complexity due to high degree of interdependency
- Examples
  - Linux, Solaris

Solaris Modular Approach



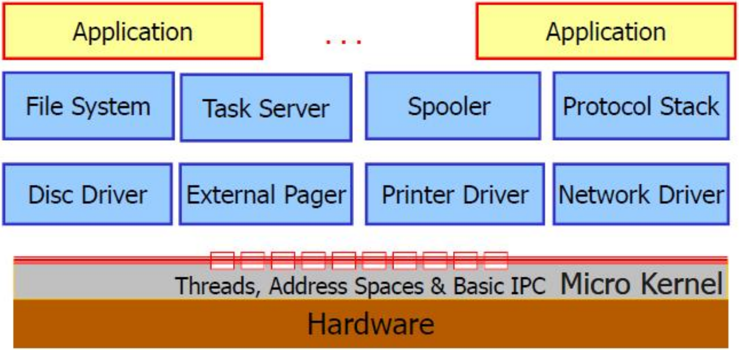
Approaches tackling Complexity and Fault Isolation

- Safe kernel extensions
  - SPIN - safe programming language (Modula 3) @ U of Washington
  - Spring - OO design @ SUN Microsystems
  - VINO - sandboxing @ Harvard

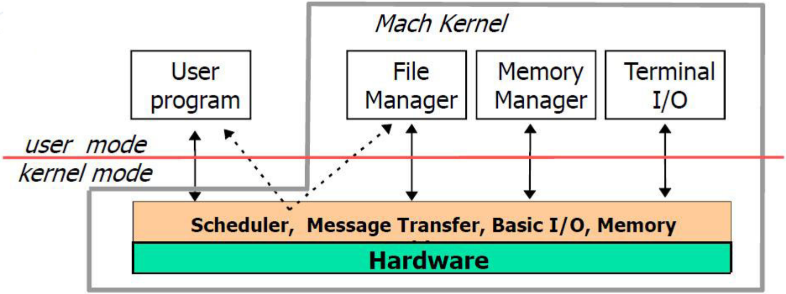


- Exokernel@MIT
  - Kernel offers multiplexing of raw HW
  - All other control is done at application level
- Microkernels
  - MACH @ CMU, L4 @ KIT, EROS, Pebbles, QNX Neutrino

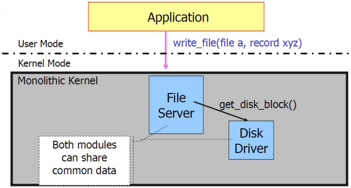
Microkernel Systems

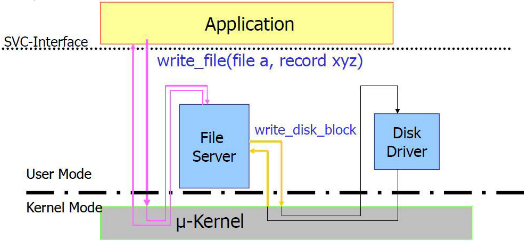


MACH Microkernel



Architectural Cost Monolithic vs. Micro-Kernel



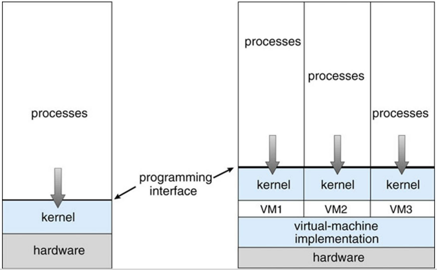


Microkernels: Pros and Cons

- Advantages:
  - Easier to test/prove/modify
  - Improved robustness & security
    (each system component in user level is protected from itself)
  - Improved maintainability
  - Coexistence of several APIs
  - Natural extensibility
    (add a new server, delete a no longer needed old server)
- Disadvantages:
  - Additional decomposing
  - Communication (IPC-) overhead → low performance
  - Bad experiences (2 B$ loss) with IBMs Workplace OS (1991-1995)
    1 kernel based on Mach 3.0 for OS/2, OS/400, AIX, Windows, · · ·

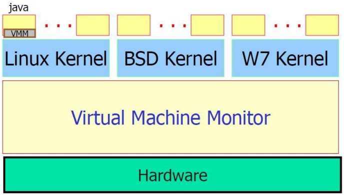
Virtual Machines

- A virtual machine takes the layered approach to its logical conclusion. It treats hardware and the operating system kernel as though they were all hardware
- A virtual machine provides an interface *identical* to the underlying bare hardware.
- The operating system host creates the illusion that a process has its own processor and (virtual memory)
- Each guest is provided with a (virtual) copy of the underlying computer

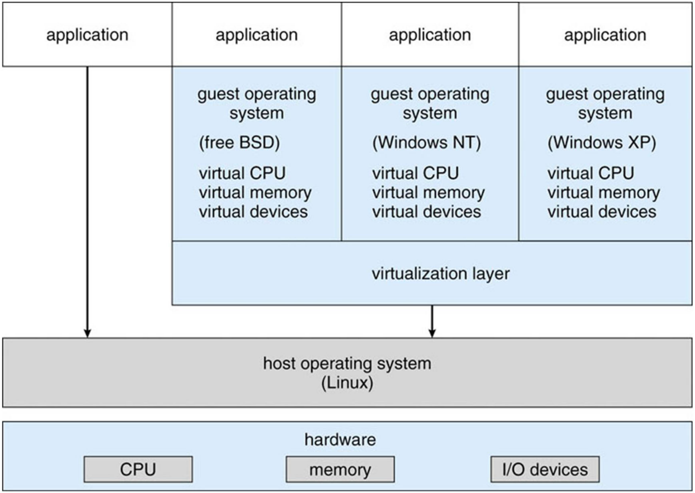


Virtual Machines Benefits

- Multiple execution environments (different operating systems) can share the same hardware
- Protect from each other
- Some sharing of file can be permitted &controlled
- Communmicate with each other & other physical systems via networking
- Useful for development, testing
- Consolidation of many low-resource use systems
- “Open Virtual Machine Format”(OVF), allows a VM to run within many different virtual machine (host) platforms

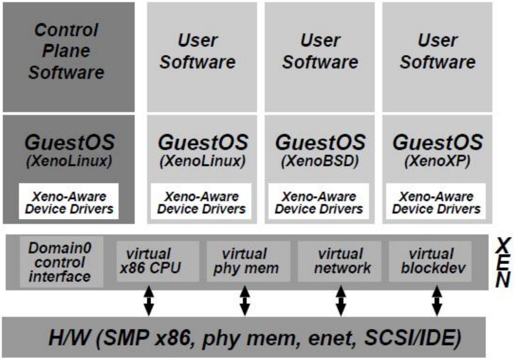


Example: VMware Architecture



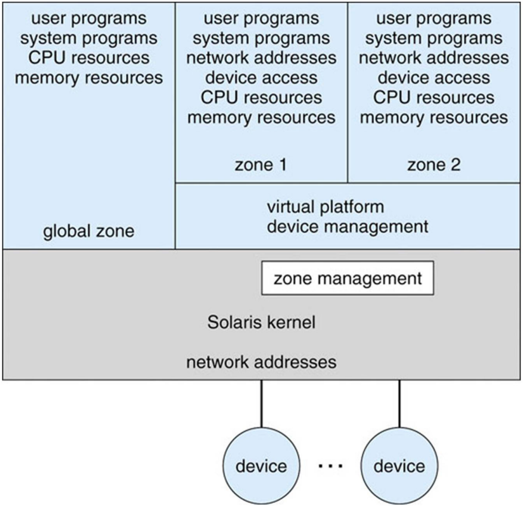
Para-Virtualization

- Presents guest with system similar but not identical to hardware
- Guest must be modified to run on paravirtualized hardware (e.g., XEN)



- Guest can be an OS, or in the case of Solaris 10 applications

Solaris 10 with 2 Containers


user programs
system programs
CPU resources
memory resources
global zone
user programs
system programs
network addresses
device access
CPU resources
memory resources
zone 1
user programs
system programs
network addresses
device access
CPU resources
memory resources
zone 2
virtual platform
device management
zone management
Solaris kernel
network addresses
device
...
device