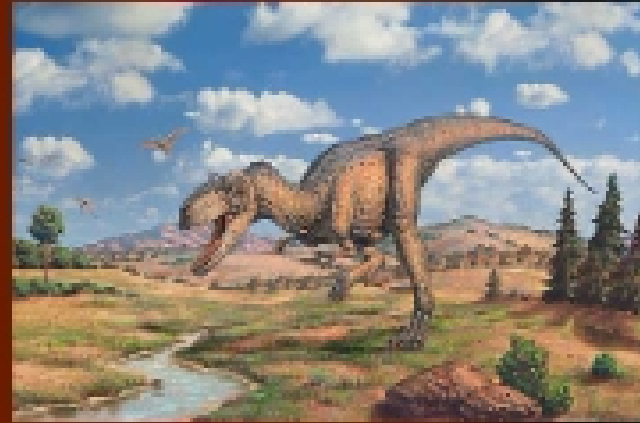


# HYDRA – The kernel of a Multiprocessor Operating System by Wulf etc.

(Presentation By Alex Kachurin and Mohamed Saad Laassel)

- Introduction To Hydra
- Design Philosophy
- Overview Of The Hydra Environment
- The Protection Mechanism
- Path Names And Walk Right
- Systems And Subsystems
- An Example (Bibliography System)
- Conclusion
- References



## Introduction to Hydra

- Designed in early 70's, Carnegie-Melon University
- Not an OS by itself, but rather a kernel base for a collection of OS'es. (e.g Linux kernel vs. Debian Linux)
- Runs on C.mmp, a multiprocessor constructed at Carnegie-Melon University (up to 16 processors, up to 32 MB of memory, crossbar switch architecture vs. bus architecture)
- The goal is "to exploit and explore the potential inherent in a multiprocessor computer system".
- Provide an environment for effective utilization of the hardware resources
- To facilitate the construction of such environment



## Design Philosophy

- Separation of mechanism and policy  
(High level policies such as scheduling and protection vs. low level mechanisms such as message dispatching)
- Multiprocessor environment.
- Multiple instances of the systems coexist together
- Integration of the design with implementation methodology (Structured programming, modular approach)
- Rejection of strict hierarchical layering. (Popular since Dijkstra's THE system, but limits flexibility)
- Protection
- Reliability (16 processors, redundancy, error recovery)



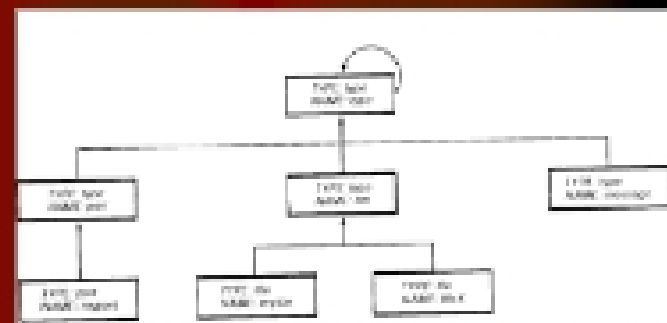
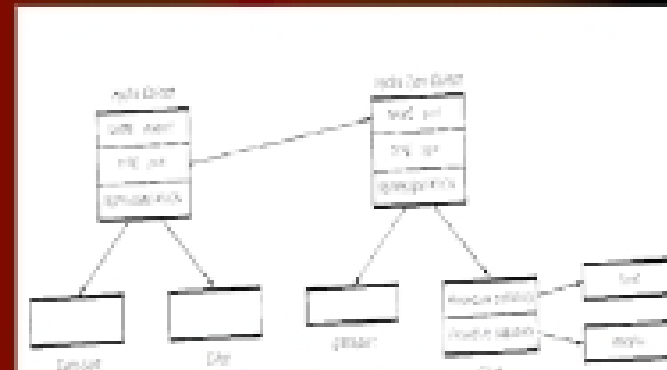
## Design Philosophy

- Need to decide what belongs to the kernel and what does not. Key principles:
- A kernel is to provide facilities for building an operating system.
- An operating system defines an "abstract machine" by providing facilities, or resources, which are more convenient than those provided by the bare hardware.
- An operating system allocates (hardware) resources in such a way as to most effectively utilize them.
- Instances of resources are called "objects". Objects belong to several distinct "types" (object based system)
- Reference count and garbage collection



## Overview of Hydra Environment

- Key terms: Procedures, Local Namespaces (LNS), Processes and Capabilities
- Hydra procedures support protection facilities via templates (formal parameters list)
- Hydra procedures are reentrant and potentially recursive
- LNS is a record of execution environment at the time of invocation
- LNS is dynamic and gets generated each time a procedure is invoked, based on capabilities
- LNS gets erased when a procedure finishes
- Capability is a reference to an object along with a collection of access rights to this object.
- Capabilities are manipulated by the kernel so they can't be forged.
- Capabilities are stored in C-lists



## Overview of Hydra Environment

- Hydra Process is the smallest entity that can be scheduled for execution
- Processes are represented as stacks of LNS, representing the cumulative state of a single sequential task.
- Synchronization primitives (Semaphores, Locks, Mutexes)

