OS
D: Intermediary bw user and hardware
Mainframe
A: OS Types N: CPU idle when IO R: Multipro
Multiprogramming
A: OS
OS Advancements
A: OS
Time Sharing
A: OS Adv D: Multiple users to interact with machine using terminals T: Illusion of concurrency
Virtualisation
A: OS Adv E: Every program executes as if it has all the resources to itself E: OS gives the layer of abstraction
OS Motivation
A: OS
Abstraction
A: OS Mot E: To standardise hardware configurations E: Efficiency, Programmability, Portability
Resource Allocator
A: OS Mot E: Multiple programs should be allowed to do simultaneously
Control Program
A: OS Mot M: Malicious or Accidental use M: Ensure isolation among users X: Unclosed while loop
OS Design
A: OS T: Robust, Flexible, Maintainable, Performant
Hardware
A: OS De
Software
A: OS De R: OS
UI
A: Software
Kernel Mode
A: Software E: Allows interaction with the /Hardware directly R: OS
User Mode
A: Software E: Cannot interact with /Hardware
Syscall Interface
A: User Mode E: Allows user to interact with /Hardware via /OS T: Cannot be used by /Kernel-Mode T: Cannot use high level /Library
Library
A: User Mode E: Interacts with /OS and /Hardware
OS Types
A: OS
Monolithic OS
A: OS Types T: Kernel is one big program, with one /Syscall-Interface T: Good SE principles are possible B: Easy to call any function N: Coupling, Complicated

Microkernel OS

A: OS Types E: Minimum set of fnalities that needs to be implemented E: All unimportant parts are in /User-Mode B: If there is a bug in unimportant parts, OS can just kill it off N: Bad performance, Needs to go through IPC

Virtual Machine

A: OS E: Full control of the machine E: Can run several OS on the same hardware E: Software emulation of the hardware

Hypervisor

A: Virtual Machine E: Manages and Creates the /Virtual-Machine T: Below /OS

Hypervisor Types

A: Hypervisor

Bare Metal Hypervisor

A: Hypervisor Types T: Runs directly on hardware

Type 2 Hypervisor

A: Hypervisor Types E: Still have your own OS running, and guest OS runs inside the /Virtual-Machine