

Introduction to Operating Systems

What is an operating system?

"A special piece of hardware that..."

- abstracts (simplifies)

- arbitrates (manages)

...the use of a computer system.

Visual Metaphor

"An operating system is like a factory shop manager"

- Directs operational resources

- controls use of employee time, parts, tools...

- Enforces working policies

- fairness, safety, clean-up...

- Mitigates difficulty of complex tasks

- simplifies operation → optimizes performance

- Directs operational resources

- controls use of CPU, memory, peripheral devices

- Enforce working policies

- e.g. fair resource access; limits to resource usage...

- Mitigate difficulty of complex tasks

- abstracts hardware details (system calls)

the following are components of an OS:

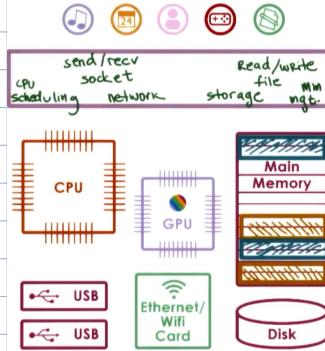
- file system

- device driver

- scheduler



What is an Operating System?



- hide hardware complexity
- resource management
- provide isolation & protection

OS Elements

Abstractions

- process, thread (corresponds to applications the OS executes); file, socket, memory page

Mechanism

- create, schedule, open, write, allocate

Policies

- least-recently used (LRU), earliest deadline first (EDF)

OS Elements: Memory Management Example

Abstractions

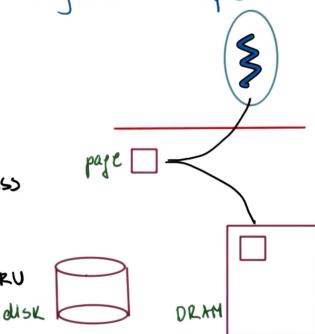
memory page

Mechanism

allocate, map to a process

Policies

least recently used - LRU



User / Kernel Protection Boundary

For an OS to achieve its goal of managing hardware on behalf of applications

it must have special privileges to have direct access to the hardware

User-level

- applications

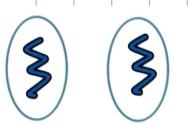
Kernel-level

- OS kernel

- privileged access

- hardware access

Unprivileged mode



user-level

User-Kernel switch is supported by hardware:

- trap instructions

- an attempt to make a privileged operation

- when in user-mode

privileged mode
kernel-level

Operating system

Mm

CPU

Privileged bit

- "System call interface": interaction between the application and the OS can be done through the system call interface. OS export a set of operations that applications can specifically invoke if an application wants the OS to perform a certain service (privileged access on behalf of the applications)

- open(file), send(socket), malloc(memory)

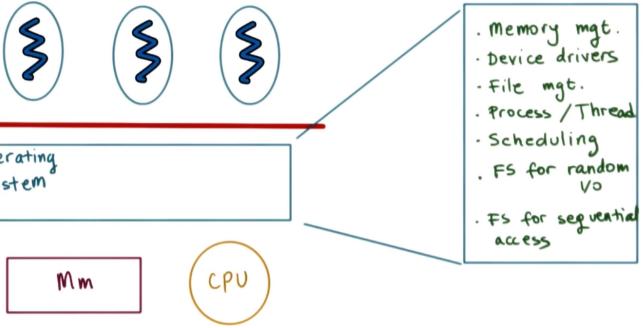
- signals

Different OS structures

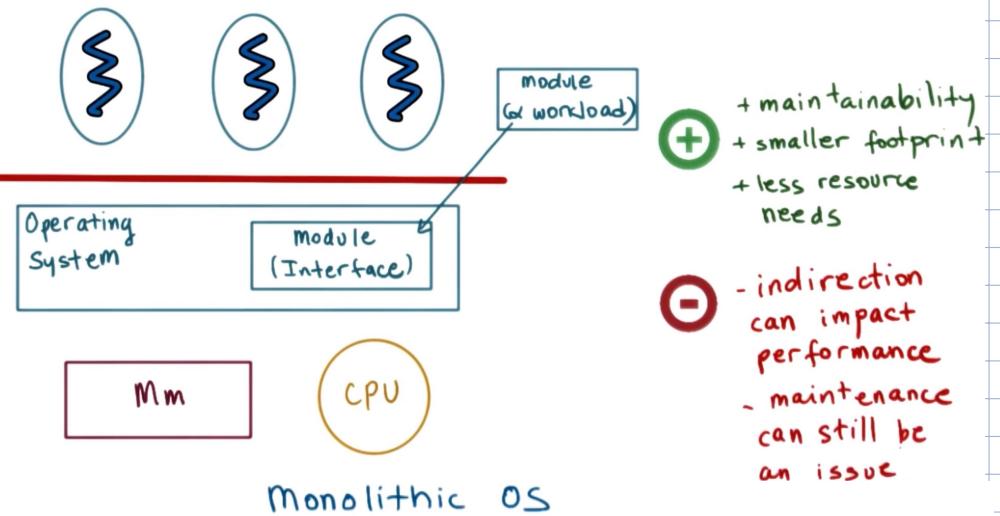
- + everything included
- + inlining, compile-time optimizations

- customization, portability, manageability ...
- memory footprint
- performance

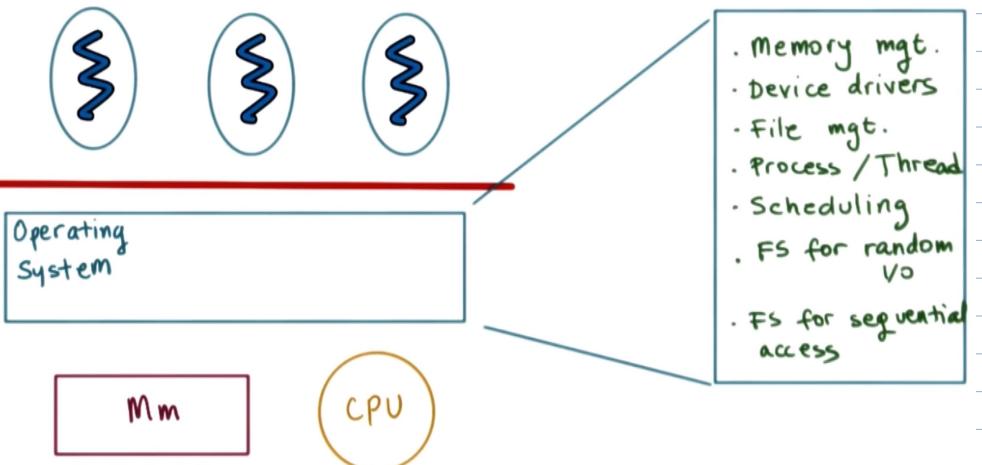
Monolithic OS



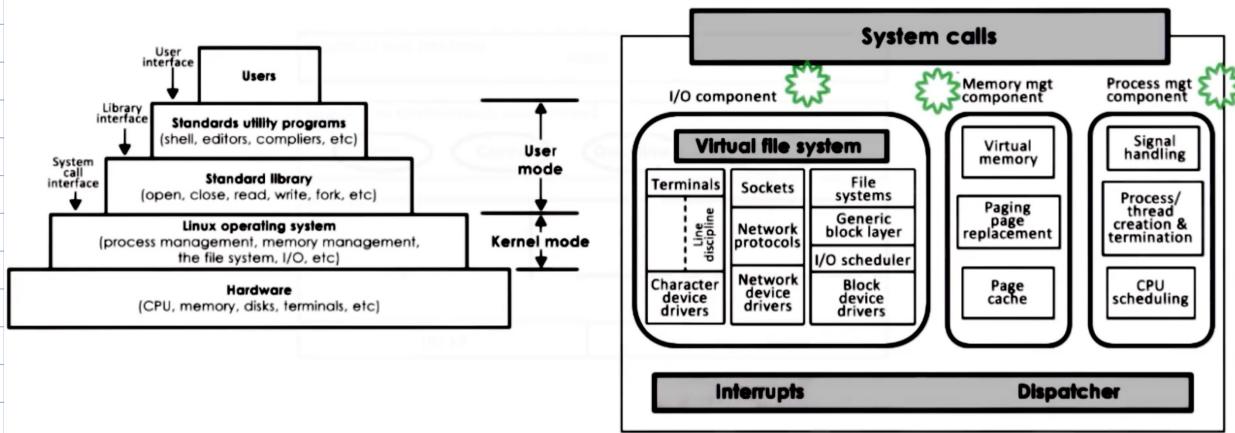
Modular OS



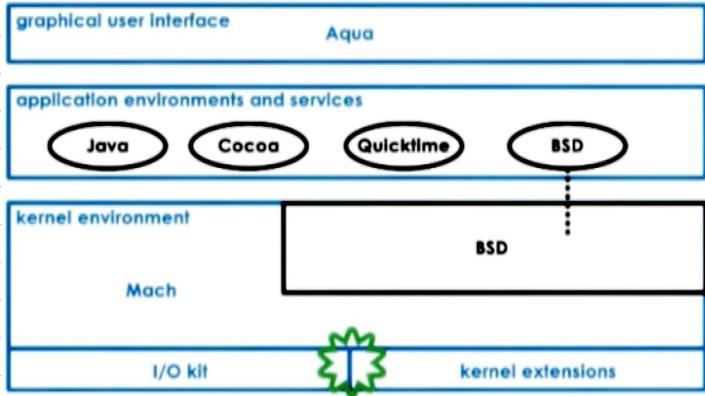
Monolithic OS

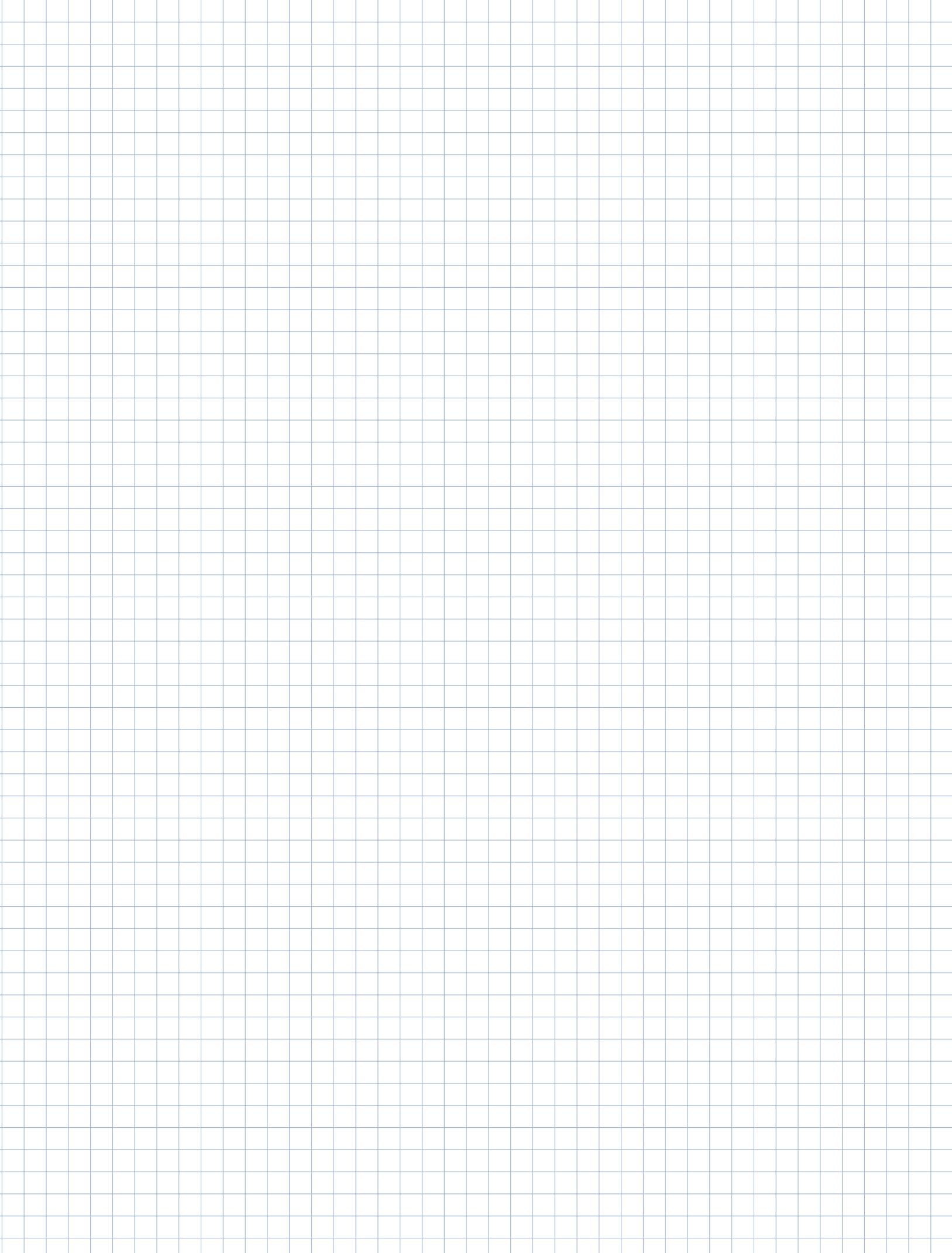


Linux Architecture

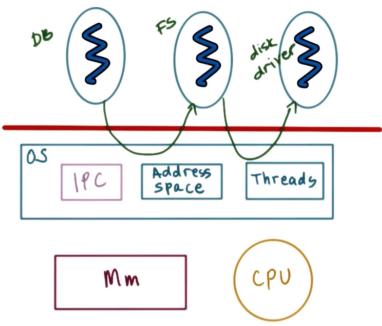


Mac OS Architecture



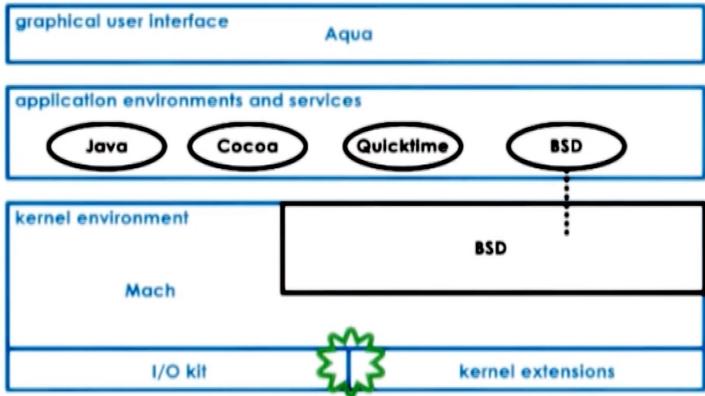


Microkernel

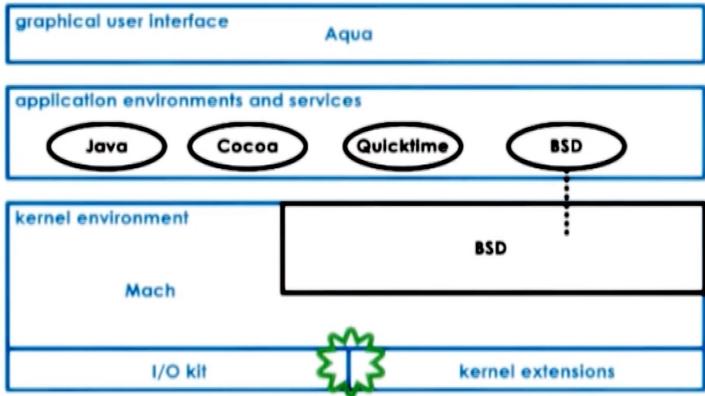


- ⊕ + size
+ verifiability
- ⊖ - portability
- complexity of software development
- cost of user/kernel crossing

Mac OS Architecture



Mac OS Architecture



Mac OS Architecture

