CISC 322

Software Architecture

Lecture 10:

Linux Architecture

Emad Shihab

Paper by: Ivan T. Bowman, Richard C. Holt and Neil V. Brewster

Recap of Last Class

- Layered Style
  - Virtual Machine
  - Client-Server

Architecture Recovery

- Why architecture recovery?
  - Many existing systems do not have documented architecture
  - Can help you better understand your system to support maintenance and evolution

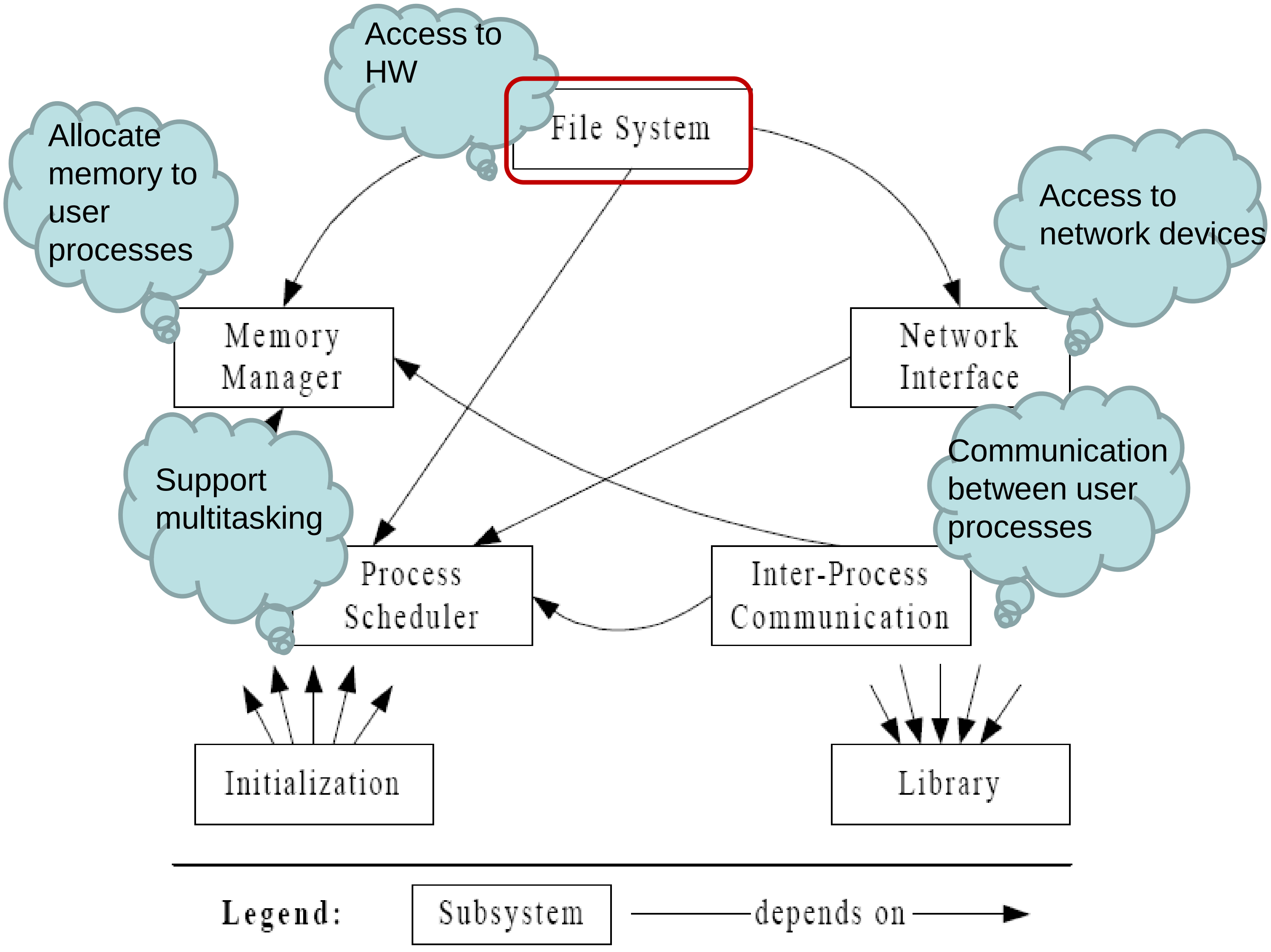
Conceptual vs. Concrete

- **Conceptual architecture**:
  - shows how developers think about a system
    - Only ‘meaningful’ relations
- **Concrete architecture**:
  - shows the actual relationships in the system
    - Harder to understand
    - Needed to make decisions related to implementation

How to Obtain Conceptual Architecture

- Examining existing documentation
  - About related or similar systems
  - Documentation provided by the project itself
  - Sometimes more reading is required to iron out inconsistencies

Conceptual Architecture

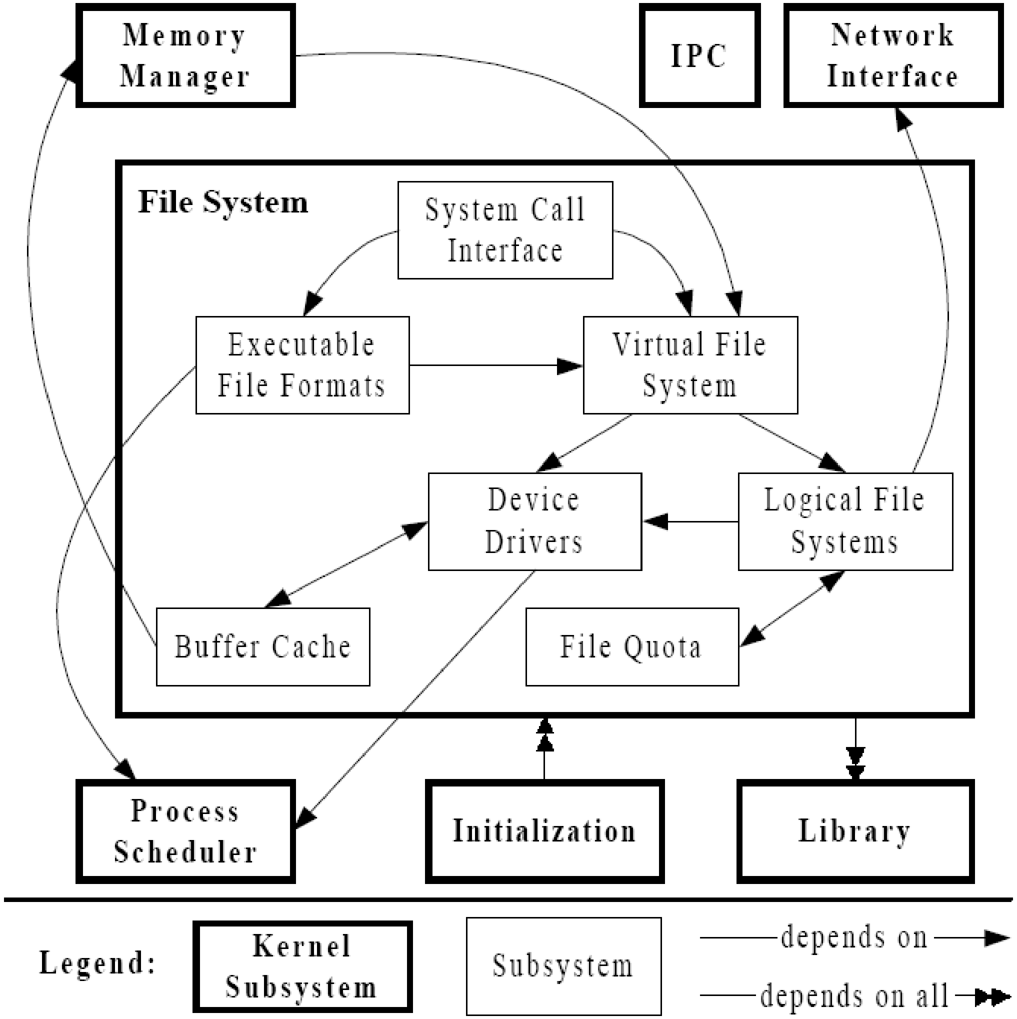


File system conceptual architecture

1. Provide access to HW devices

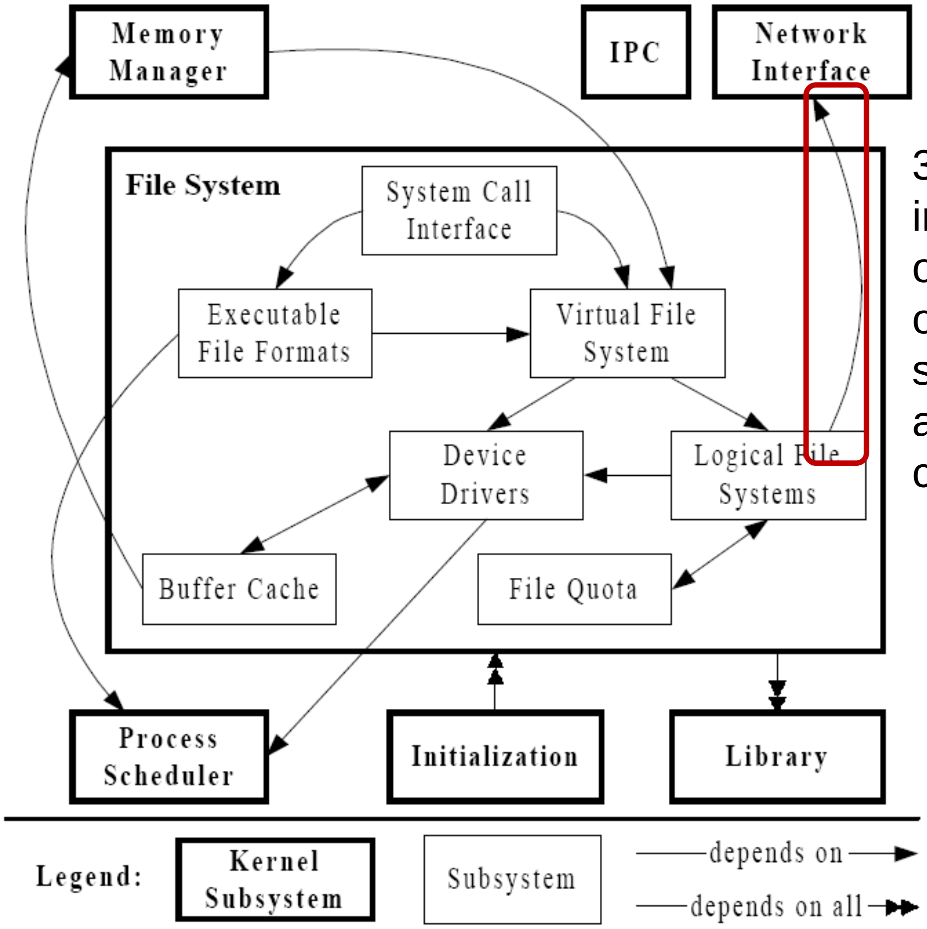
2. Support different file formats and their mapping on physical locations

3. Allow programs to be stored in several executable formats



File system conceptual architecture

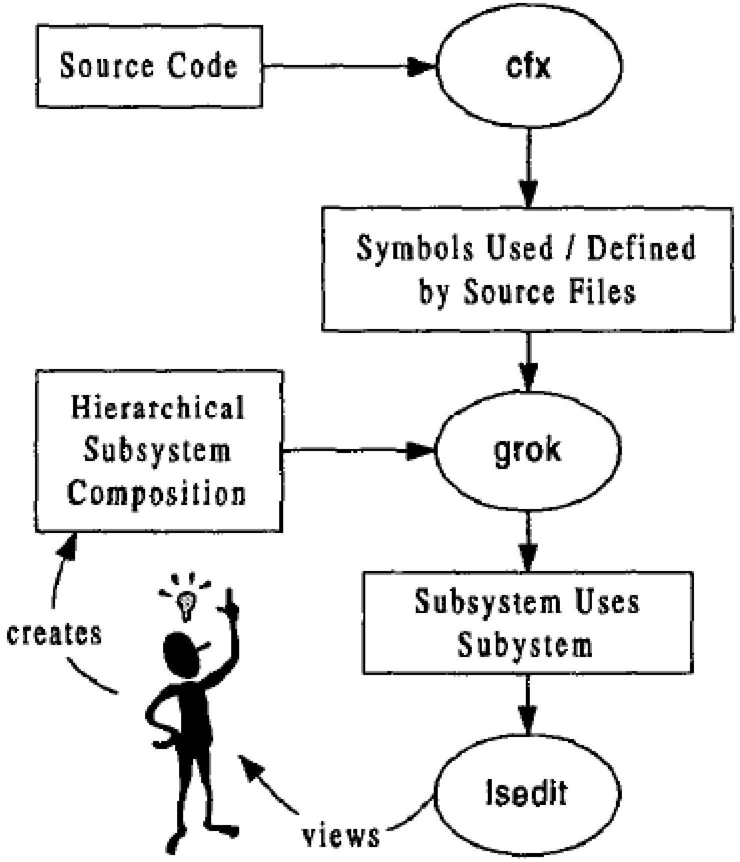
- Uses Façade design pattern
  - Other parts of kernel use File System through a single interface
  - Minimizes dependencies
- Architectural style?



3 file system implementations depend on files stored on another computer

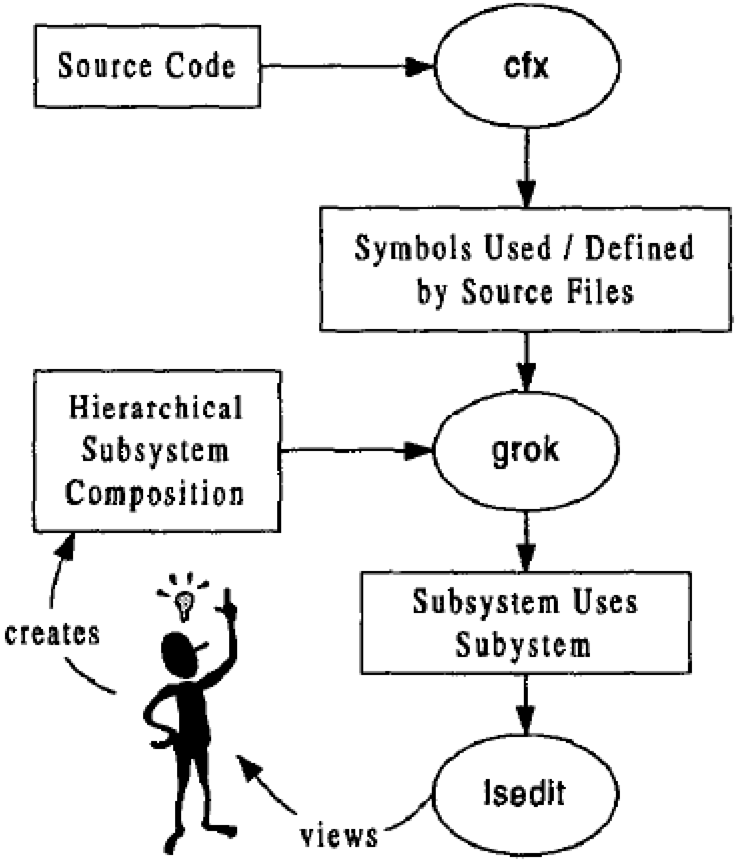
Concrete Architecture – Extraction Methodology

- Determine source code
- Source code extractor (cfx) to extract relations
  - E.g. function y calls function x
  - Extracts functions calls and variable access relations
  - Control flow and data flow dependencies


Source Code
cfx
Symbols Used / Defined by Source Files
Hierarchical Subsystem Composition
grok
Subsystem Uses Subsystem
creates
lsedit
views

Concrete Architecture – Extraction Methodology

- Use grok to determine relations between files
  - Each file was manually assigned to a subsystem
  - Used directory structure, file naming, source code comments documentation, source code (last resort)
  - Used grok to determine subsystem relations
- Used lsedit to visualize extracted system architecture

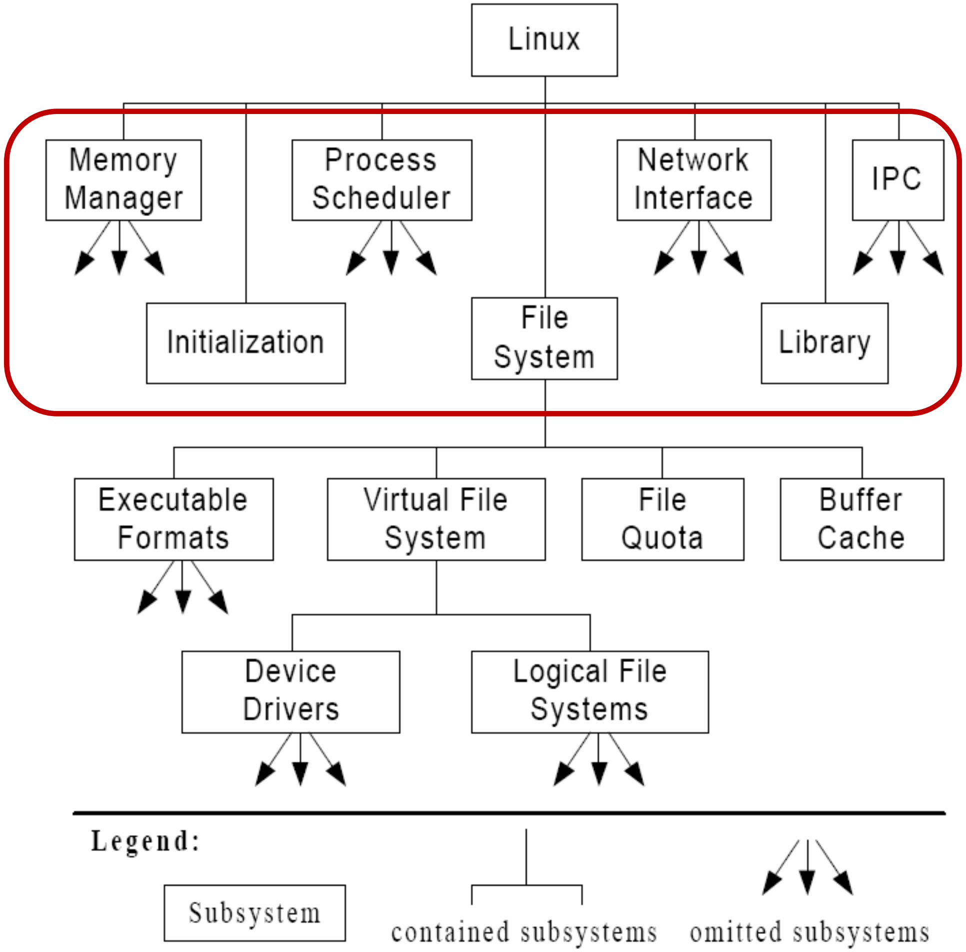

Source Code
cfx
Symbols Used / Defined by Source Files
Hierarchical Subsystem Composition
grok
creates
Subsystem Uses Subystem
lsedit
views

Concrete Architecture – Hierarchical Decomposition

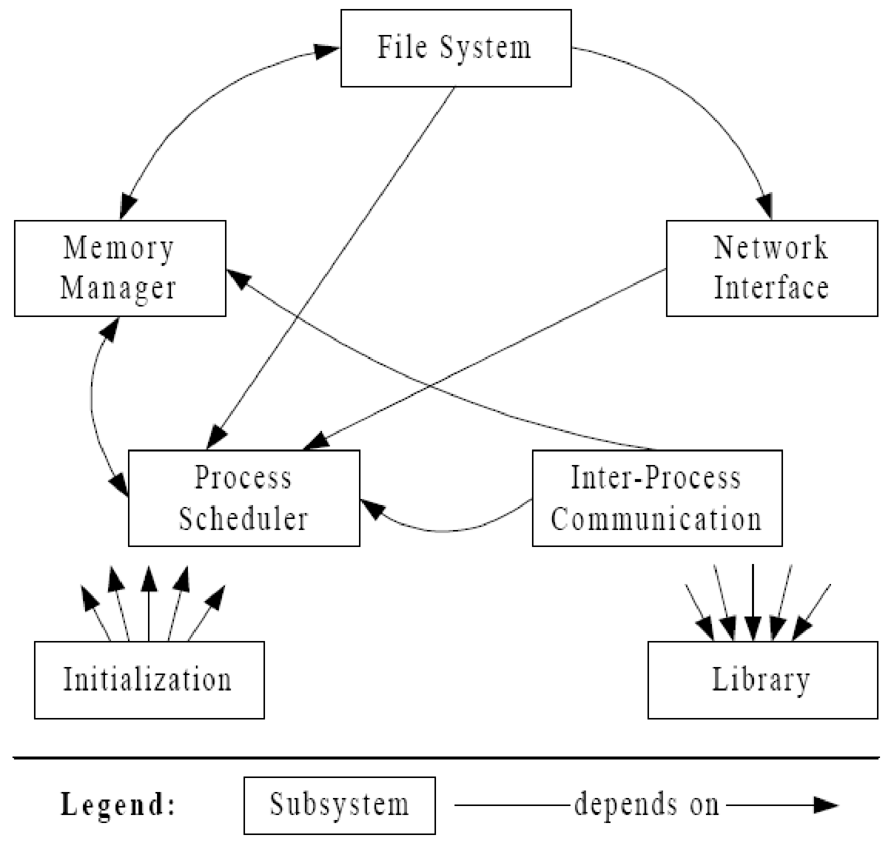
- Manually created
- Source files are assigned to subsystems
- Subsystems hierarchically assigned to subsystems
- Used to view relationships between subsystems instead of files

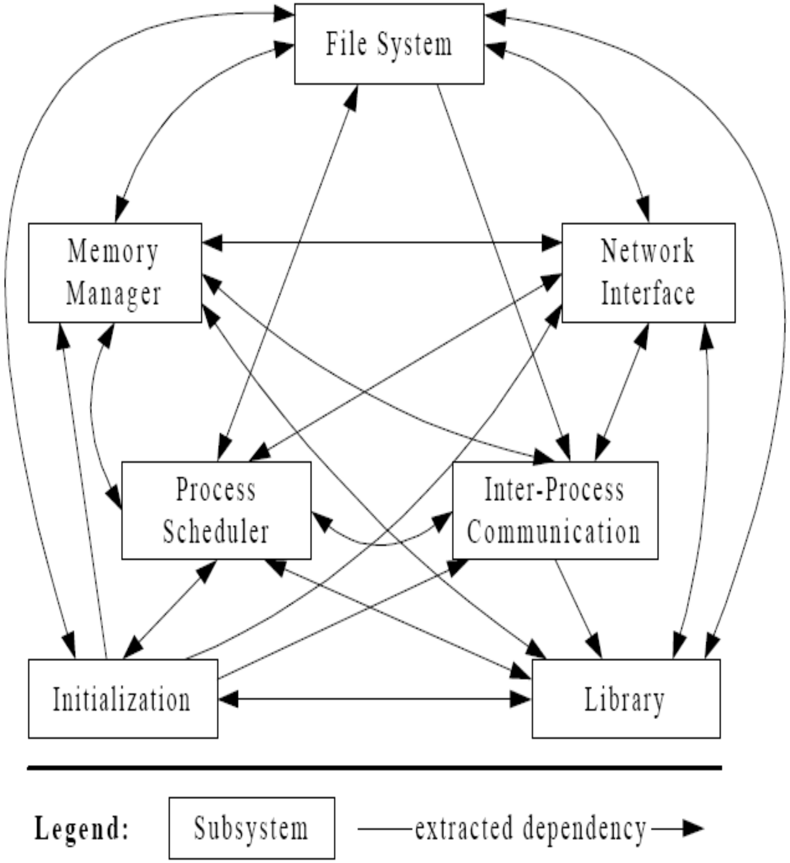
Concrete Architecture – Hierarchical Decomposition

- Seven major subsystems from conceptual arch
- Subsystems had corresponding directories in code implementation
- Used directory structure, naming convention to assign files to subsystems



Conceptual vs. Concrete Architecture



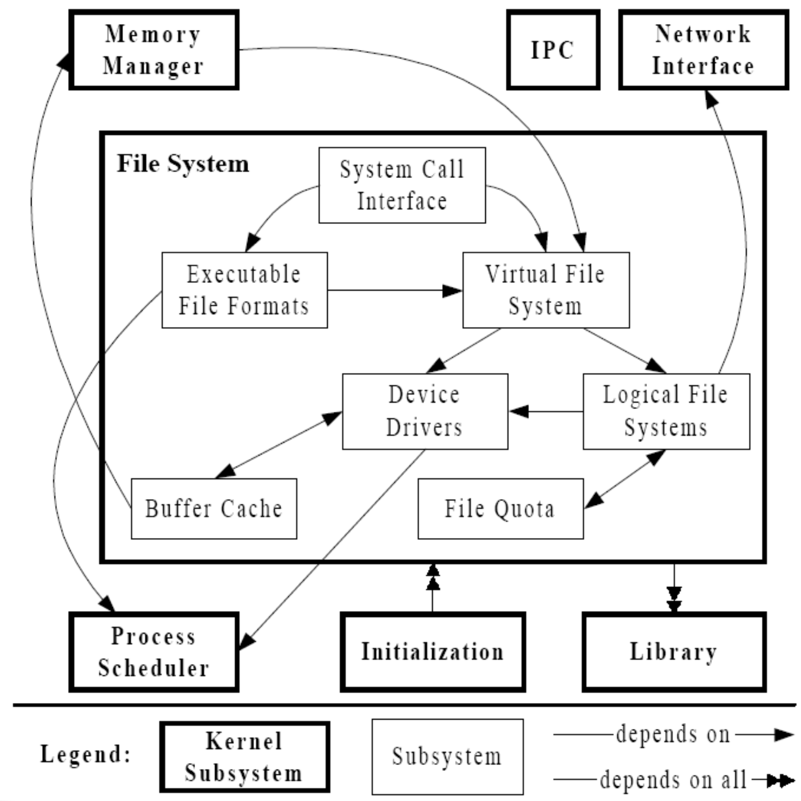


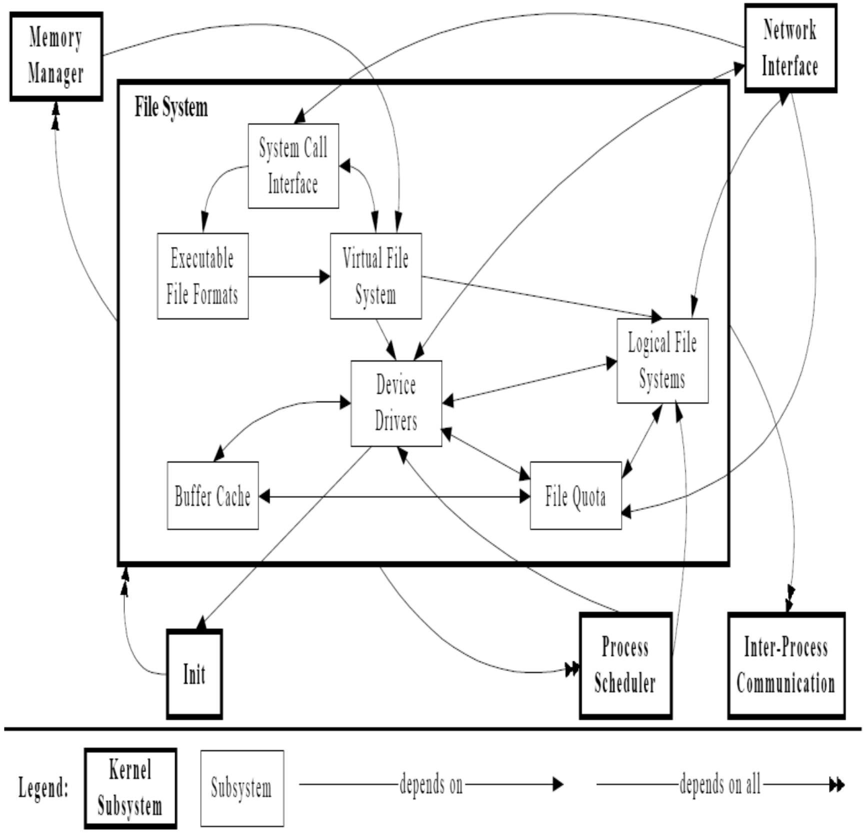
Same subsystems, more dependencies (19 vs. 37)

Why the Extra Dependencies

- Developers avoid existing interfaces to achieve better efficiency
- Expediency
- (Almost) always expect more dependencies in concrete architecture

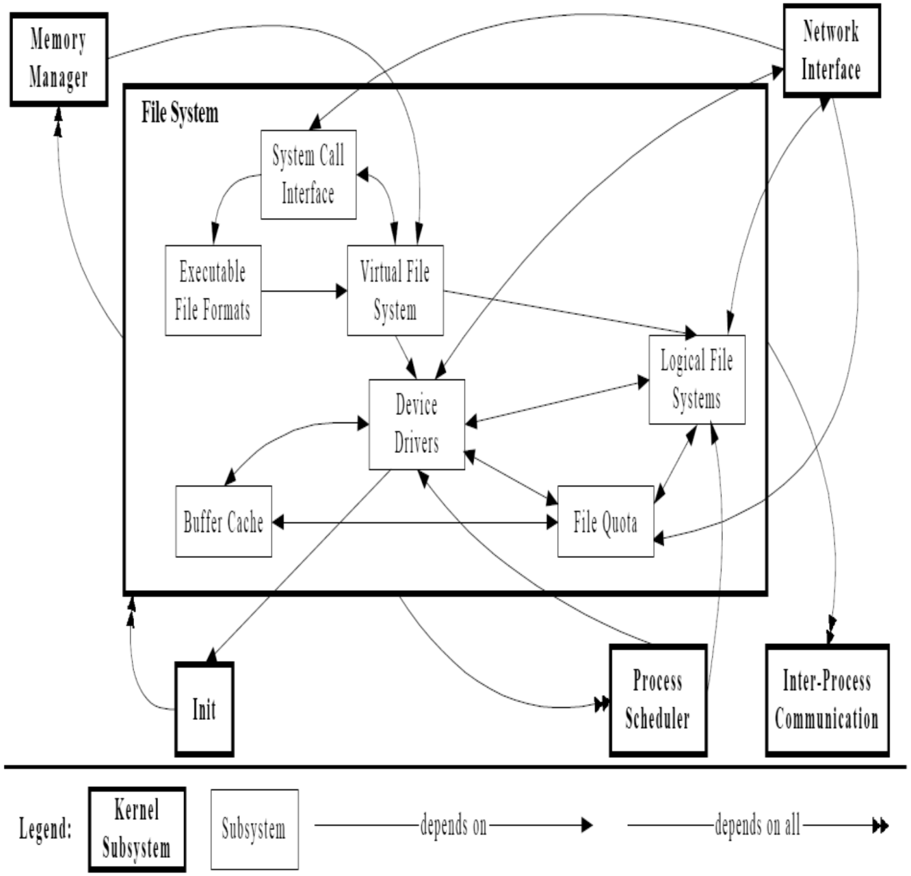
File System Concrete Architecture





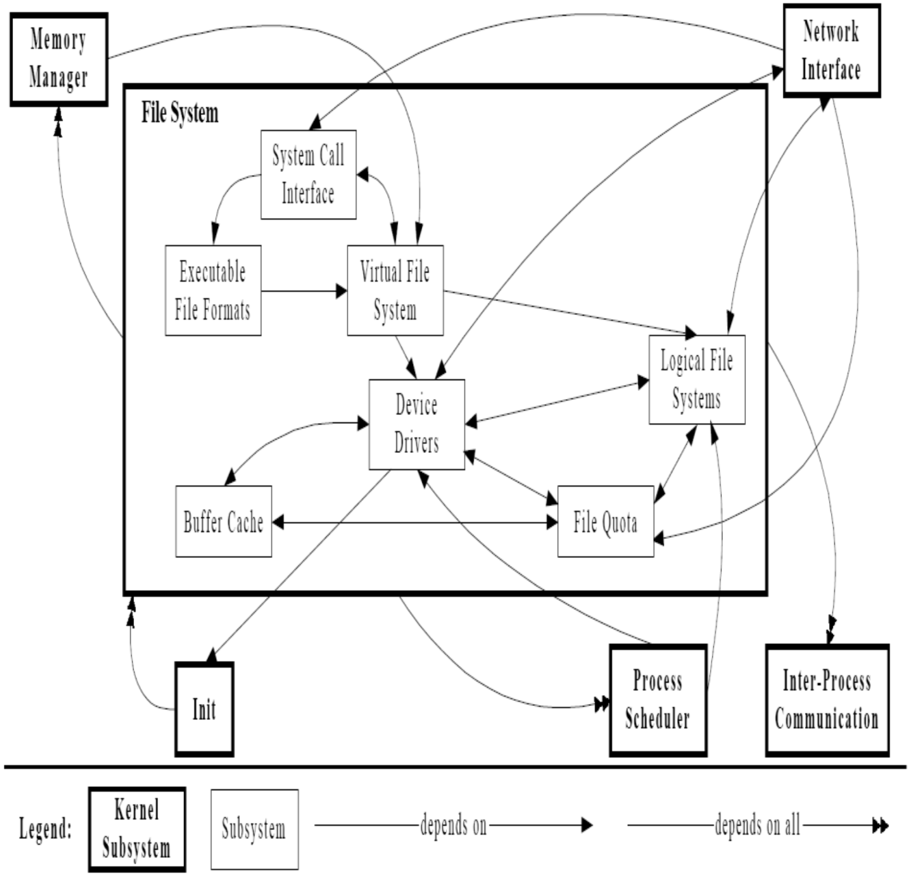
File System Concrete Architecture

- Network interface depends on logical file systems
  - File systems (NCPFS and SMBFS) that use the network were implemented by having the Network Interface directly call functions in the implementation of these logical file systems



File System Concrete Architecture

- Process Scheduler depends on Device Driver
  - (printk) in Process scheduler prints msgs to console. Calls routines implemented in device drivers subsystem



Lessons Learned

- Human judgment is needed to determine appropriate hierarchical decomposition
- Conceptual Architecture shows Control flow in one manner, but concrete showed implementation might use a different mechanism (e.g., network interface)
- Developers improved system efficiency by bypassing existing interfaces

Lessons Learned

- “The read-only stuff doesn’t really belong here, but any other place is probably as bad and I don’t want to create yet another include file.”
- Concrete architecture should be used to refine conceptual architecture
  - Not desirable to add all relations. May hinder understanding
- Best to develop concrete and conceptual architecture in an iterative manner