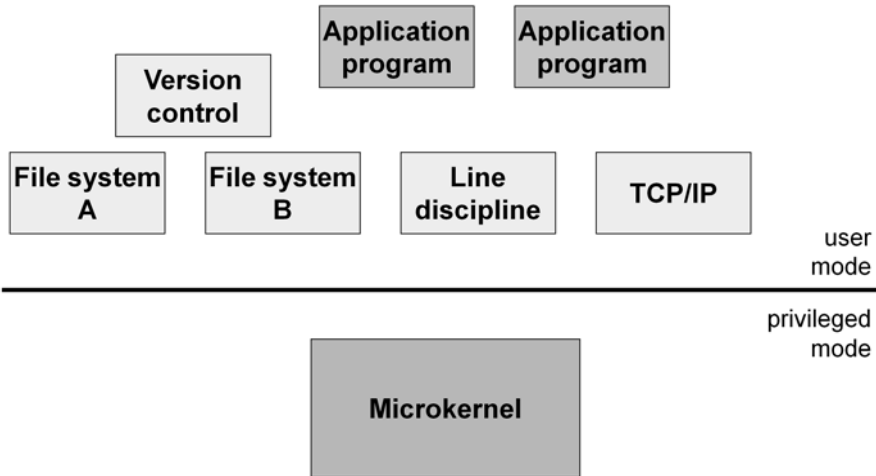


Microkernels

This material is covered in Section 4.2.2 of the textbook.

OS Services as User Apps



Why?

- **It's cool ...**
- **Assume that OS coders are incompetent, malicious, or both ...**
 - OS components run as protected user-level applications
- **Extensibility**
 - easier to add, modify, and extend user-level components than kernel components

Implementation Issues

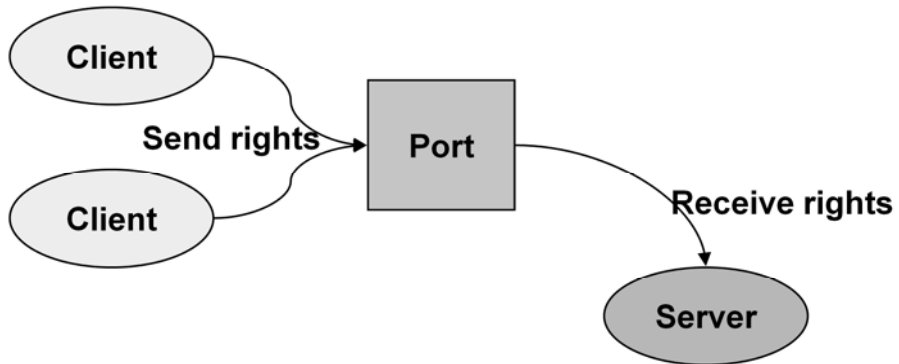
- How are modules linked together?
- How is data moved around efficiently?

Mach

- **Developed at CMU, then Utah**
- **Early versions shared kernel with Unix**
 - basis of NeXT OS
- **Later versions still shared kernel with Unix**
 - basis of OSF/1
 - basis of Macintosh OS X
- **Even later versions actually functioned as working microkernel**
 - basis of GNU/HURD project
 - HURD: HIRD of Unix-replacing daemons
 - HIRD: HURD of interfaces representing depth

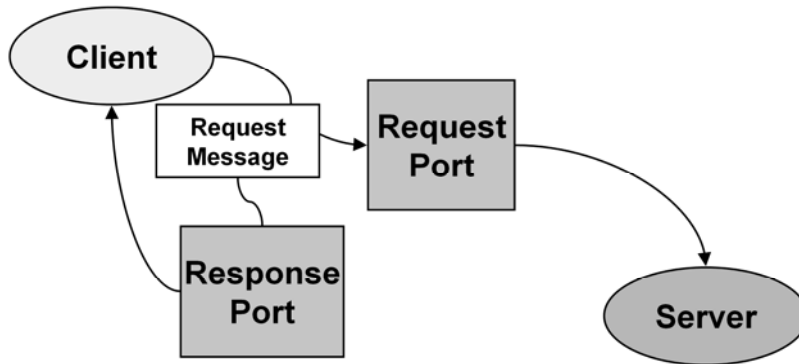
Mach Ports (1)

- Linkage construct



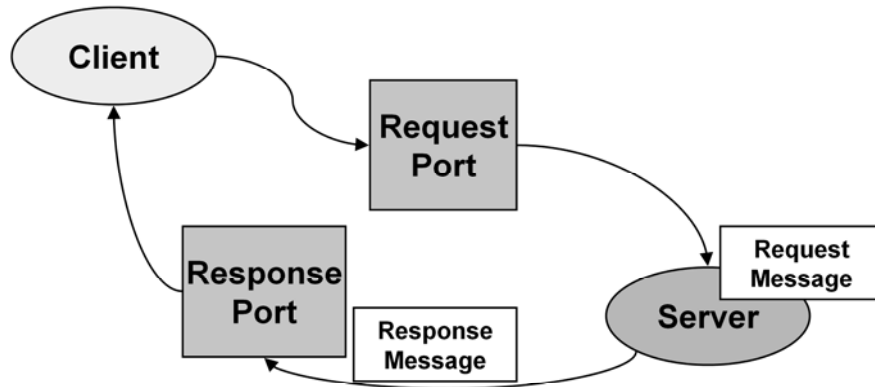
Mach Ports (2)

- Communication construct



Mach Ports (3)

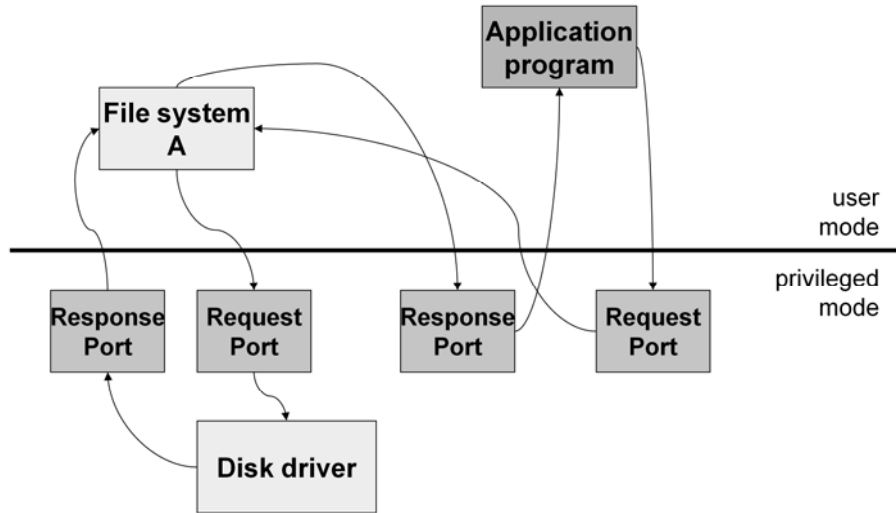
- Communication construct



RPC

- Ports used to implement *remote procedure calls*
 - communication across process boundaries
 - if procedures are on same machine ...
 - local RPC

Example



Successful Microkernel Systems

-
-
- ...

Attempts

- **Windows NT 3.1**
 - graphics subsystem ran as user-level process
 - moved to kernel in 4.0 for performance reasons
- **Macintosh OS X**
 - based on Mach
 - all services in kernel for performance reasons
- **HURD**
 - based on Mach
 - services implemented as user processes
 - no one uses it, for performance reasons ...