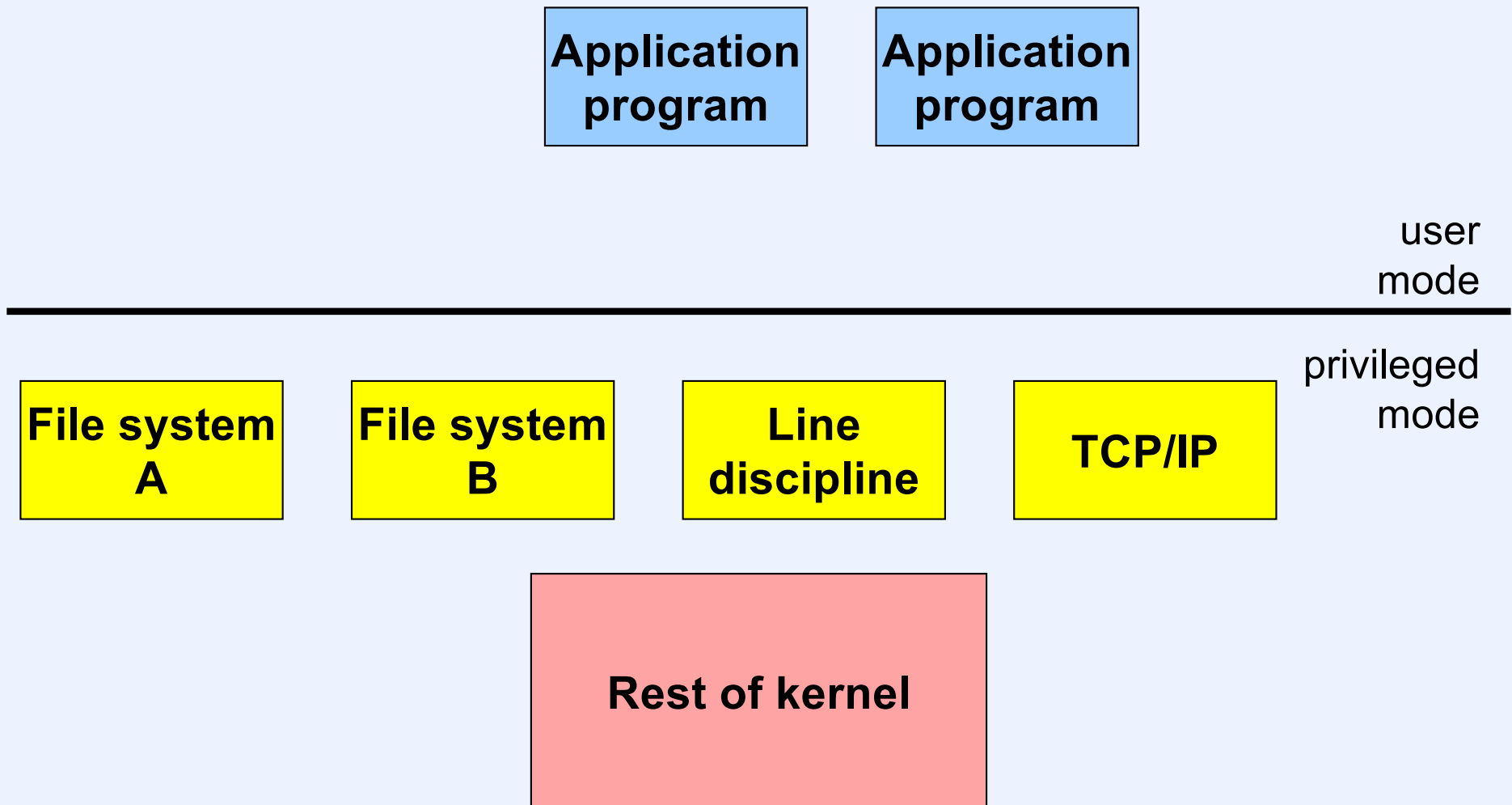


Microkernels

Traditional OS Organization

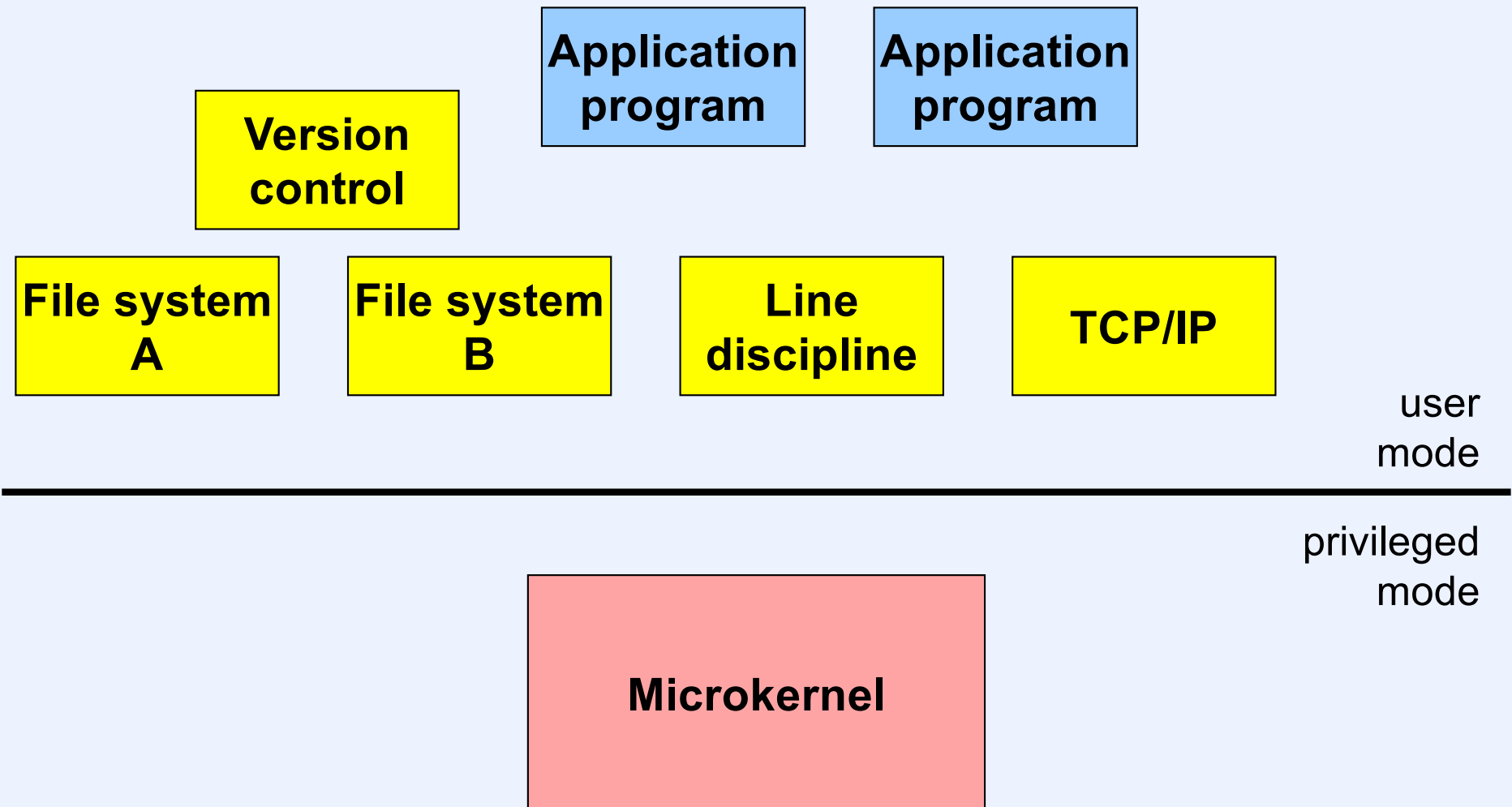


Quiz

In the previous slide, assume each of the two application programs run as separate processes. What's in the slide employs:

- a) three address spaces: one for each process, and one for the kernel**
- b) six address spaces: one for each process, one for each of the four kernel components, and one for the rest of the kernel**
- c) two address spaces: one for each process, with the kernel existing in a shared portion of the two process address spaces**

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

- **What are the building blocks?**
- **What is run in privileged mode?**

Mach

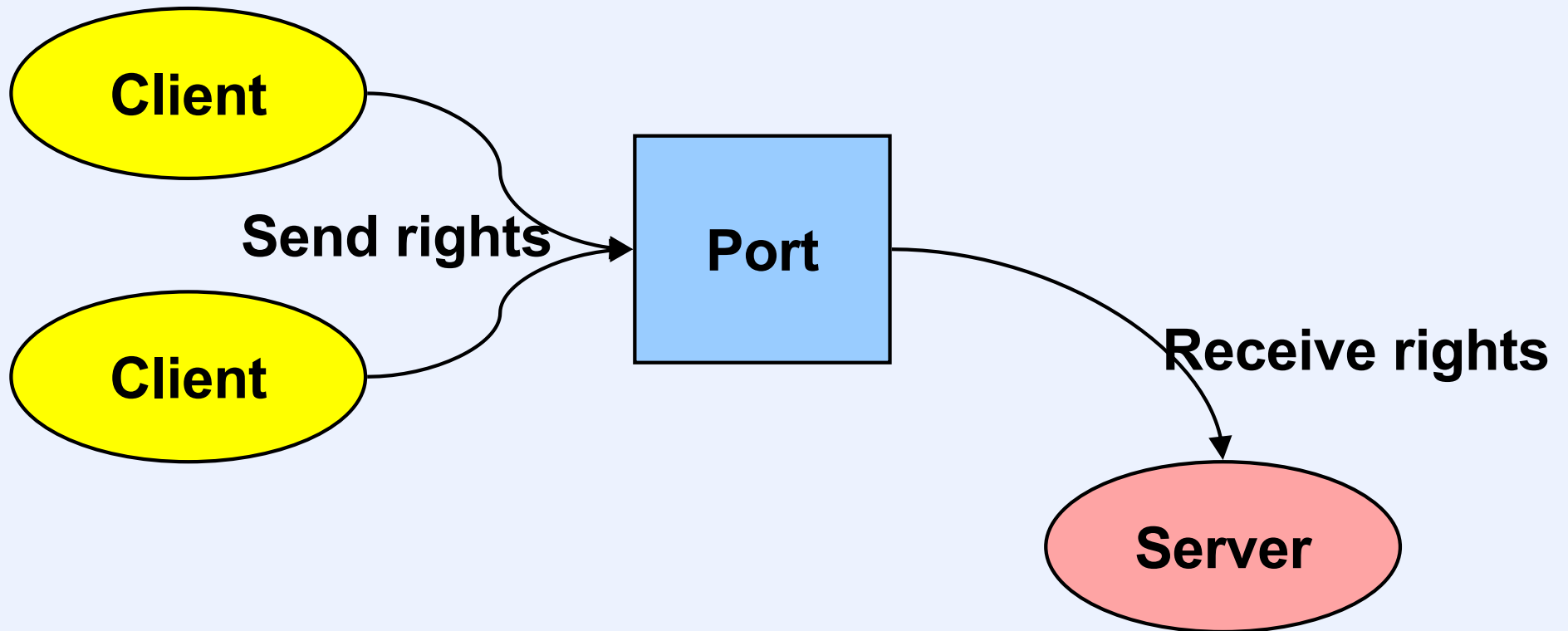
- **Developed at CMU, then Utah**
- **Early versions shared kernel with Unix**
 - **basis of NeXT OS**
 - **basis of Apple OS X**
- **Later versions still shared kernel with Unix**
 - **basis of OSF/1**
- **Even later versions actually functioned as working microkernel**
 - **basis of GNU/HURD project**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

Mach's Building Blocks

- **Tasks**
 - represent services/objects
 - holders of access rights
- **Threads**
 - represent virtual processors
- **Ports**
 - communication channels and access rights
- **Messages**
 - carriers of data and access rights

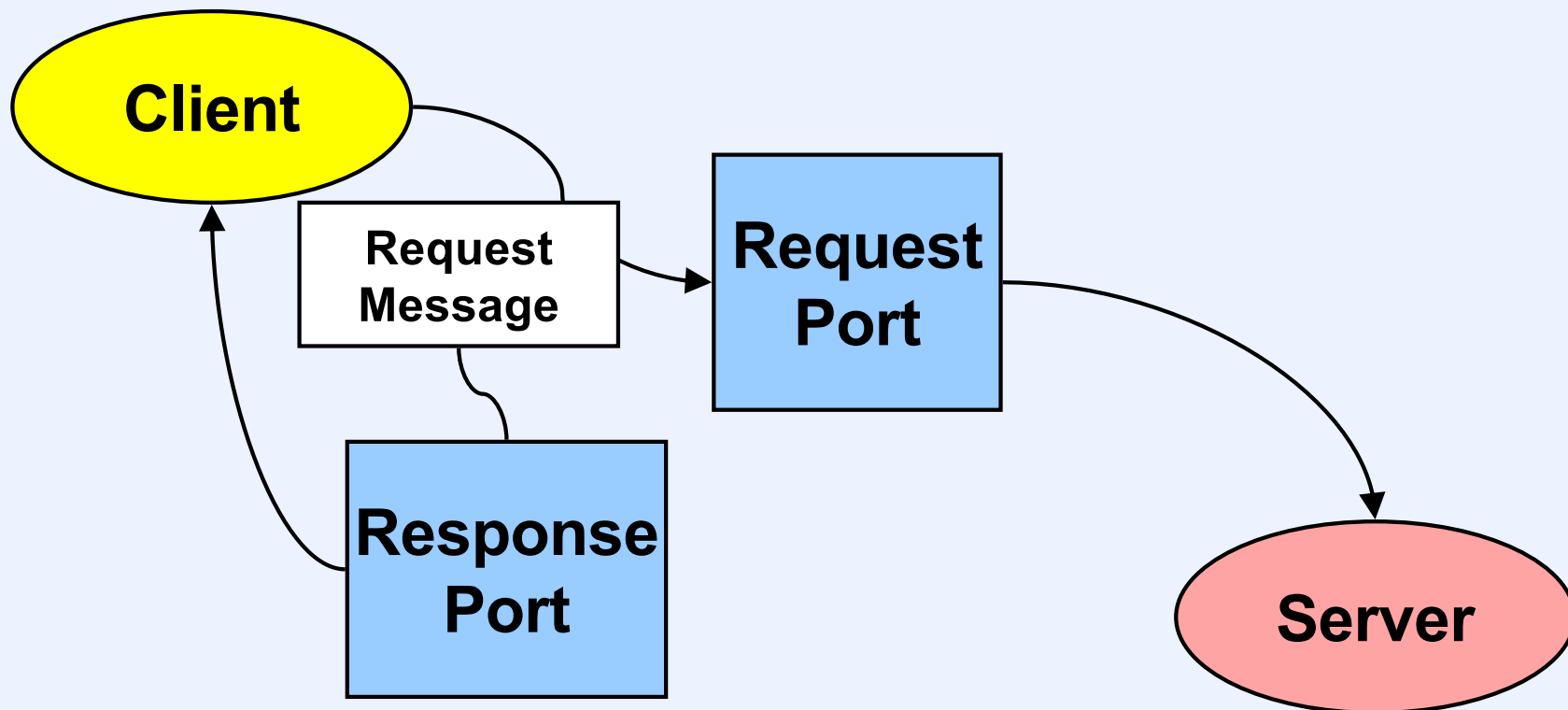
Mach Ports (1)

- Access rights



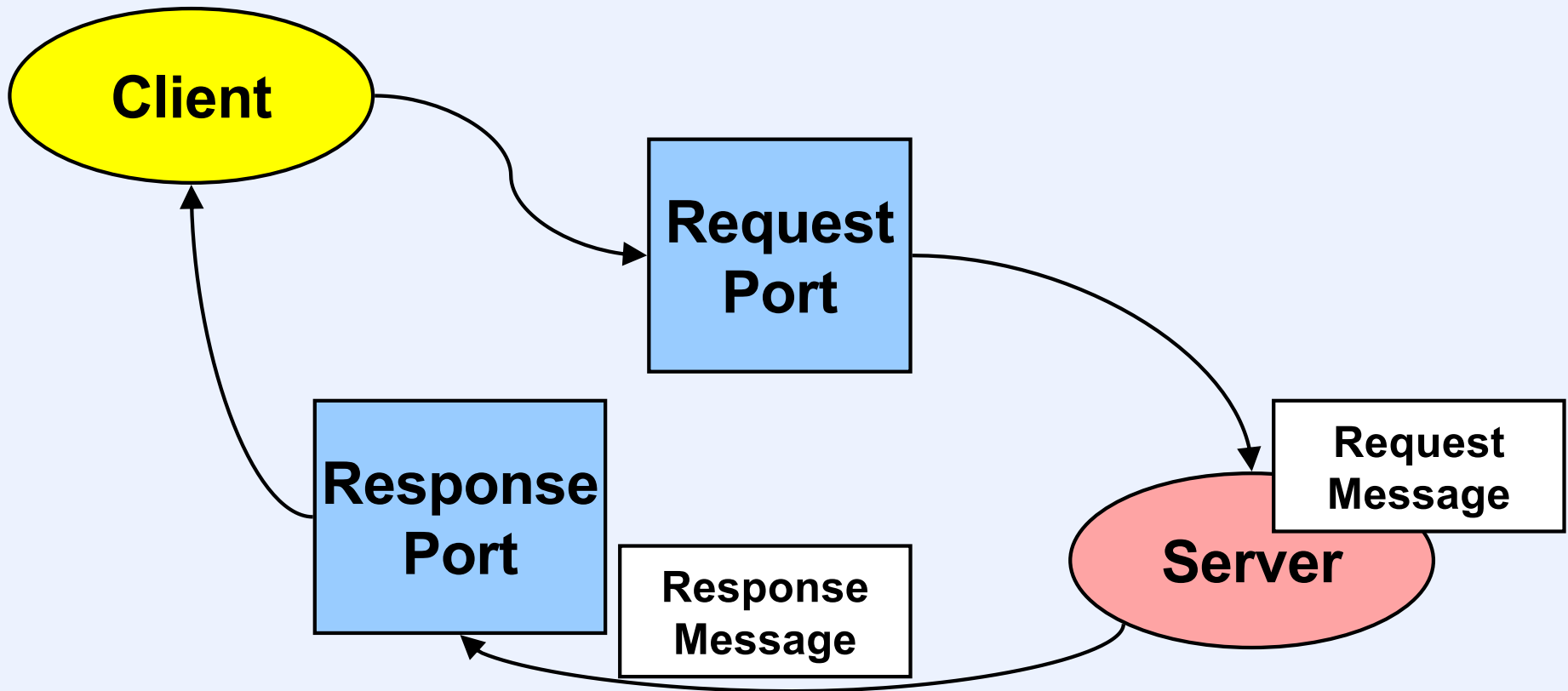
Mach Ports (2)

- Communication construct



Mach Ports (3)

- Communication construct



Method Invocation

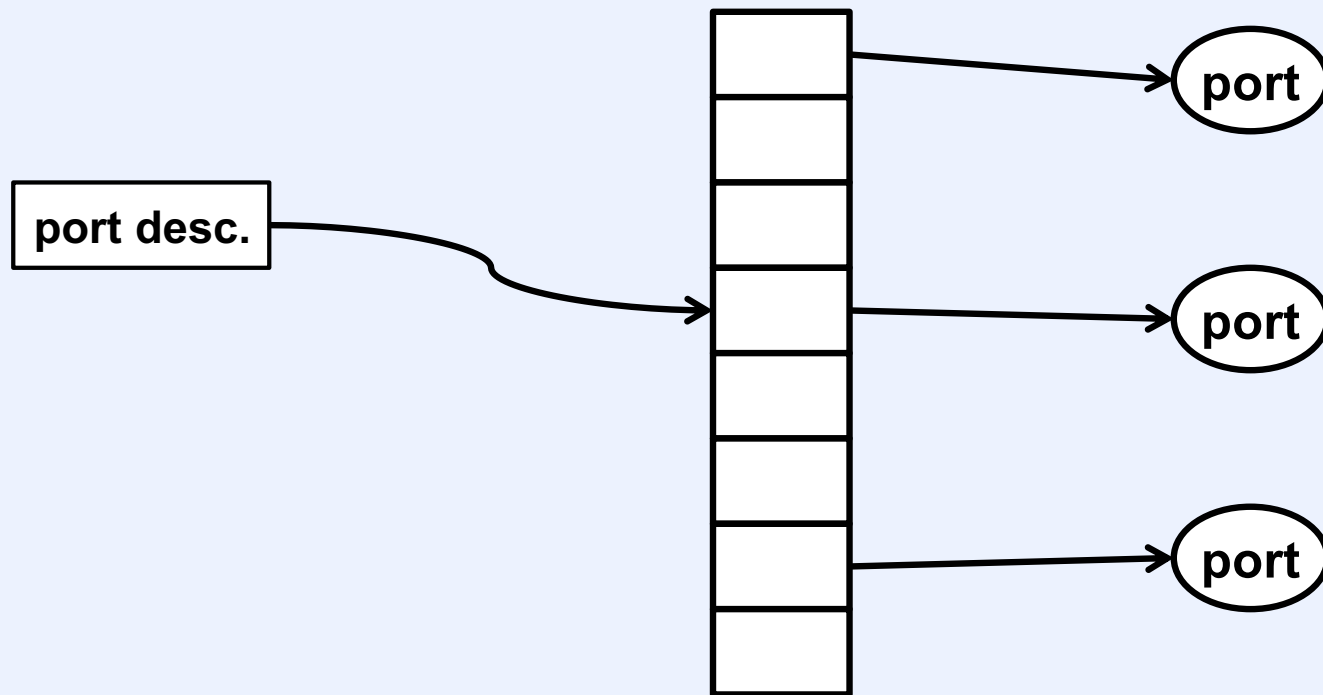
- ***Tasks*** implement objects
- **Access rights to *ports*** are secure object references
- ***Messages*** are method invocations and responses

Messages

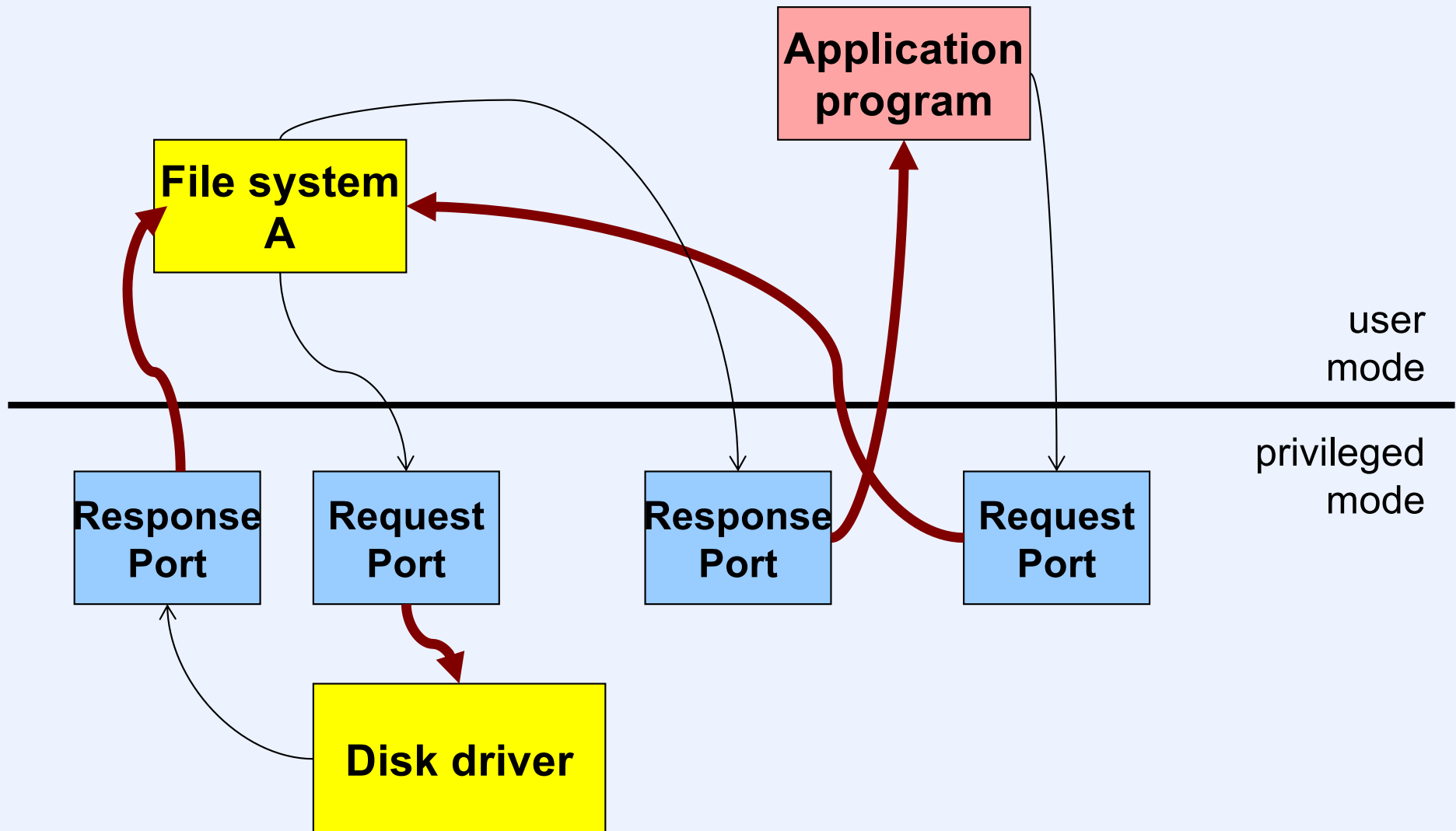
- **Cost of passing messages is critical factor**
- **Small messages are copied**
- **Large messages within single address space passed by reference**
- **What about messages across address spaces?**

Implementing Port Rights

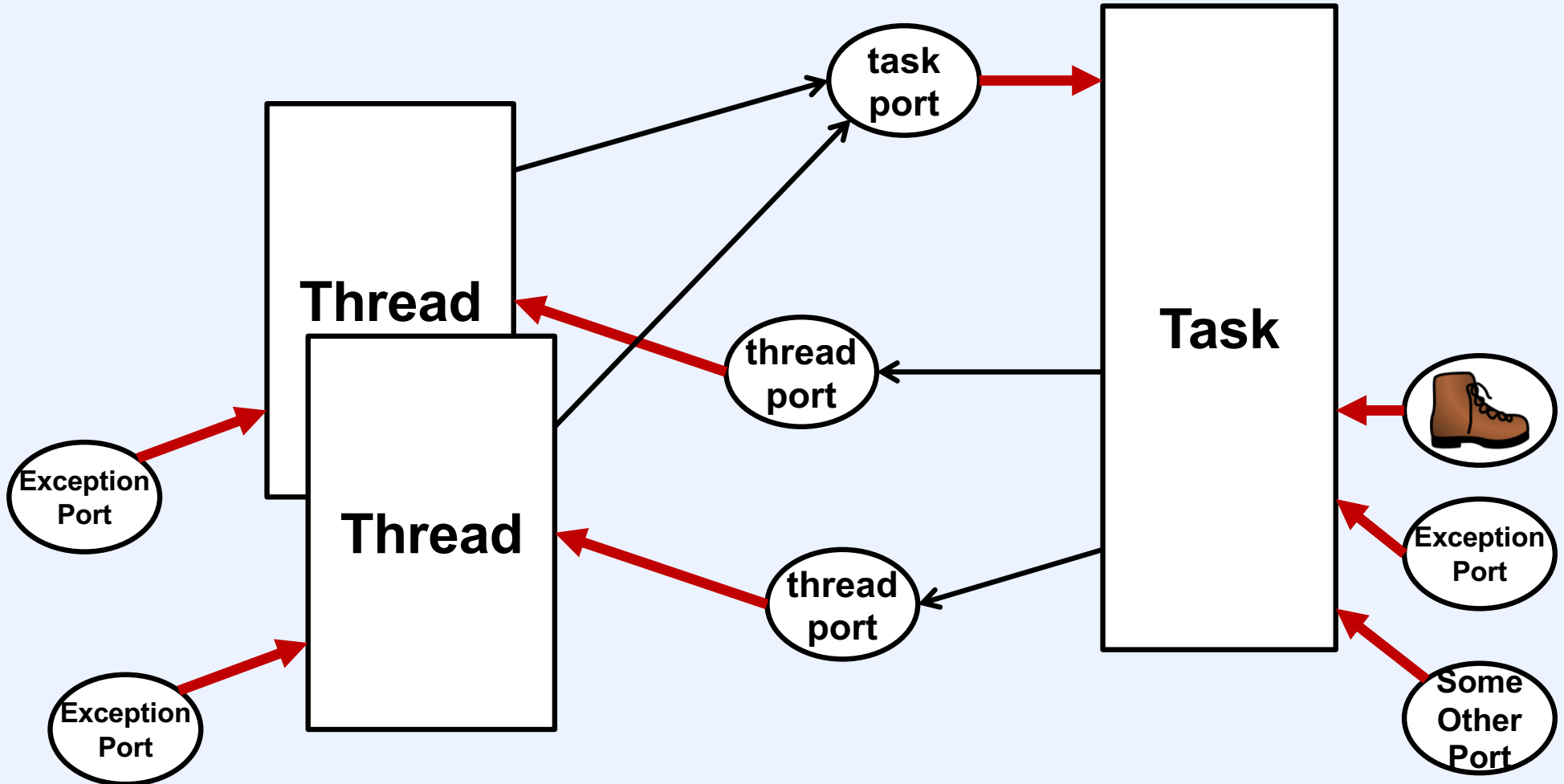
- References to ports must be secure and unforgeable
 - how are they done?



Example



Task and Thread Objects



Virtual Memory

- **Memory cache objects**
 - implemented in kernel
 - represent what's in real memory
- **Memory objects**
 - implemented in kernel or as user tasks
 - represent what's mapped into real memory

Devices

- **Device master port exported by kernel**
- **Tasks holding send rights may request access to any device**
 - **send rights given for device port**

Successful Microkernel Systems

-
-
- ...

Attempts

- **Windows NT 3.1**
 - graphics subsystem ran as user-level process
 - moved to kernel in 4.0 for performance reasons
- **Apple 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 ...