Graduate Operating Systems
(Memory Management)

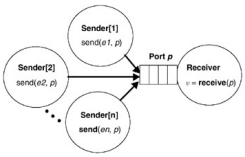
Fall 2020

Paper "Duality"

- Brief history of Mach:
  - 1975 U-Rochester: "Rochester Intelligent Gateway" (modular OS + message passing)
  - 1981 CMU: "Accent" (message passing OS)
  - 1984 CMU: "Mach" (compatible with Unix, threads, IPC, multiprocessor, VM)
  - Convoluted rest of history; one of more interesting tidbits is 4.3BSD with core parts of kernel replaced with Mach yielding NeXTSTEP, which later led to OS X

Paper “Duality”

- Message interface: app can “control” OS
- Mach: object-oriented interface, distributed computing, multiprocessing, portability, UNIX compatibility, UNIX performance
- **Ports:** kernel-provided IPC mechanism; unidirectional, kernel-protected **channel** (n send endpoints, 1 receive endpoint).

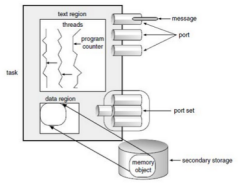


Send rights & receive rights

Paper “Duality”

- Ports kind of like file descriptors (integers)
- Mach IPC messages:
  - Data objects exchanged between threads (and user-level & kernel-level)
  - Either inline data or out-of-line (OOL) data; latter: kernel reserves region in receiver’s virtual address space by mapping pages (copy-on-write)

Paper “Duality”



- Memory objects:
  - Virtual memory: map process address onto memory objects; accessed via messages
  - Portability: separate machine-dependent from machine-independent components:
    - Dependent: pmap (kernel; manages MMU)
    - Independent: kernel code for page management
    - Independent: external pager (user-level, management of backing store)
  - Memory map: ordered, linked list of references to memory objects (“files”)

Paper “Duality”

- Message structure
- msg_send, msg_receive, msg_rpc
- Default ports and custom ports