Process management in the Mach System

lecture-28

Process Management

- Page fault- Performs better than Unix processes.

  - Each thread runs on a processor.
- Mach IPC is used for thread synchronization.
- Cthreads package.
- CPU scheduler - Global run queues with appropriate locks and local run queues. – Heavy weight context switching.
- Fixed Time quantum – What if the threads are lesser than processors?? Mach uses a variable time quantum inversely proportional to the no. of threads.
- Exception Handling – RPC message passing for handling.

IPC

- IPC -> ports and messages.
- Memory management is used in its implementation.
- Conversely, IPC is used in memory management.
- **Ports:**

  - Enable a thread to send data to another thread in a controlled manner.

  - Send & receive *rights – Port name + capability.*

  - Only one task with receive rights

  - Can be multiple with send rights

  - Sending receive rights to another task causes ownership of receive rights to change.

IPC Contd..

- Ports are location independent.

  - Ensures portability through this one communication mechanism.

Messages – Fixed-length header + variable number of typed data objects.

- Header - Destination port + reply port +length of the message.
- Data (inline data – versions vary).
- Port rights (only way to send port rights is in messages)
- Pointers to “out of line” data (Large messages).
  - Two cases: receiver on same vs. different.
- Used to implement remote procedure calls (RPC).

IPC Contd..

- Receiver on same computer
  - No need to necessarily copy message from sender to receiver
  - Takes time to copy messages.
  - Instead, when message contents unchanged, use virtual memory-based technique for improving efficiency
    - A kind of shared memory solution.
    - “copy-on-write”

IPC contd…

- Receiver on different computers.

  - In comparison with UNIX which uses low-level network protocols.

  - Mach provides an optimized solution.

  - Provided by NetMsgServer.

“User-Level” Message Server: NetMsgServer

- Enables location-transparent naming of ports
  - Does not matter which computer a port is on.
  - NetMsgServer dynamically resolves the addresses.
- Services:
  - Data Independence.
  - Network wide name service
  - Allows ports to be registered
  - Distributed database of port rights
  - Forwarding of messages by acting as proxy ports.
  - Data conversions (different computer architectures).

Memory Management

- A memory object represents a secondary object that is mapped into the address space of a task.

  - Treated just like any other object.
- Unlike traditional UNIX, which implied a contiguous virtual memory space Mach allowed for *sparse* address spaces, where regions of memory could be allocated from anywhere in the address space.
- No regular page table.
- User-level memory managers can be used instead for memory to be paged.

Contd..

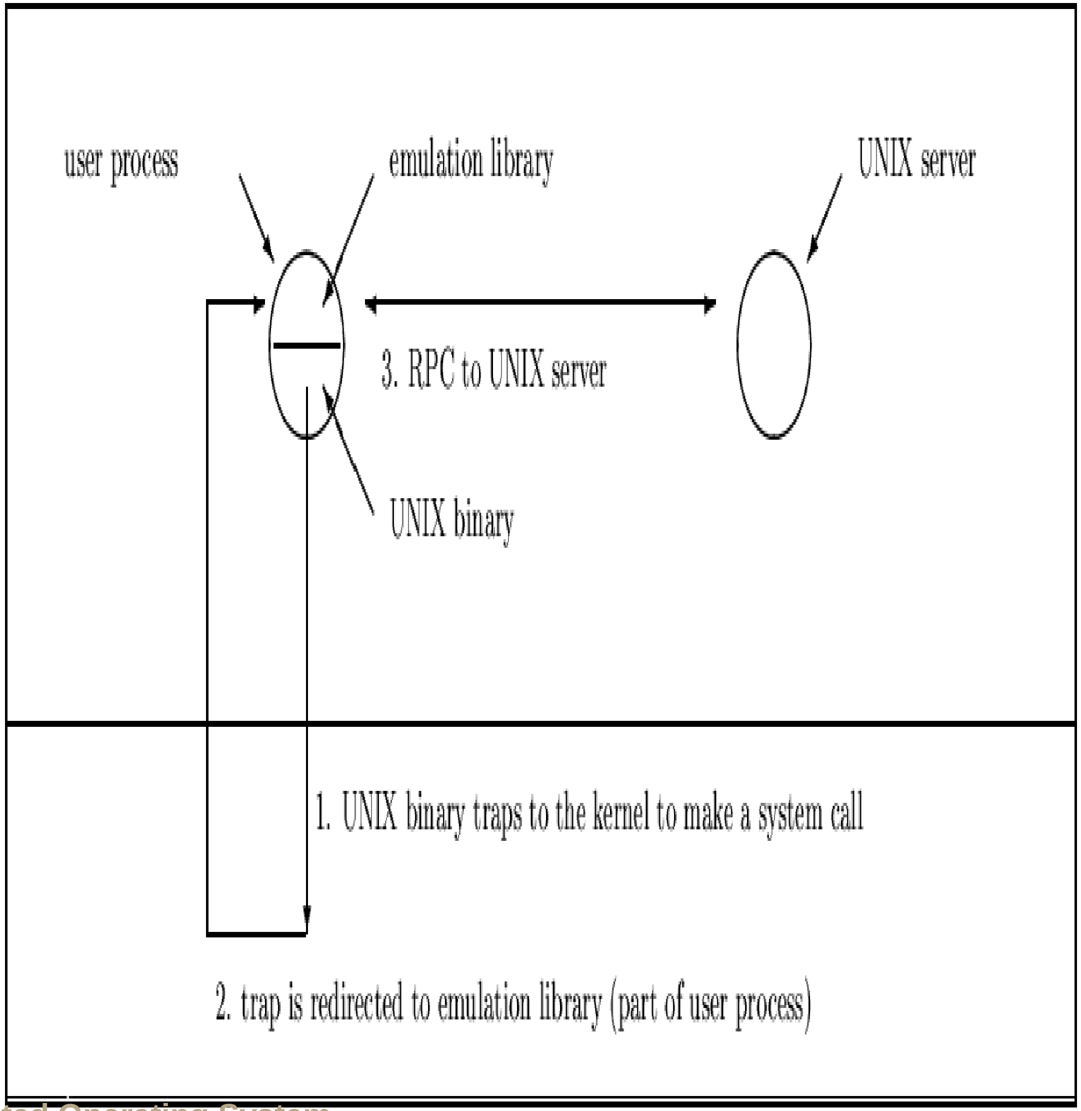
- Mach takes care of basics only
  - Acts as interface between hardware and user-level
    - e.g. receives page faults from hardware
    - Notifies relevant task (via port) of page fault
  - Implements pageout policy (FIFO).
  - Supplies default Memory Manager in some cases where user level fails.
- User-Level Memory Managers
  - Handle majority of memory management - can page memory.
  - System calls used to communicate with kernel for memory mapping / page-in / page-out / provide page-level locking
  - Responsible for consistency of the contents of a memory object mapped by tasks on different machines.

Shared Memory

- Mach approaches the shared memory in a different way.
- Consistent shared memory is supported only for shared processors.
- Tasks running on processors sharing memory
  - – Standard FORK system call , Parent declares regions to be inherited by the child task.
  - – Doesn't use copy-write strategy. But readable-writable technique.
  - – shared page is readable: may be replicated on multiple machines.
  - – shared page is writable: only one copy is changed.
- External memory manager – NetMemServer – Handles shared read-write from different machines willing to share memory.

System Calls

- Traps to the kernel.
- Upcalls into emulation library(USER LEVEL).
- Switch to any thread waiting on a port for operations like disk writes.
- Returns to emulation library.
- Returns from trap.
- System call is slow compared to traditional systems.


user process
emulation library
UNIX server
3. RPC to UNIX server
UNIX binary
1. UNIX binary traps to the kernel to make a system call
2. trap is redirected to emulation library (part of user process)

Summary

- Unix code – evicted from the kernel – can be replaced by another code at the user level.
- Successful in implementing multiprocessing and distributed computing.
- Extensibility at the expense of speed.
- Integrated memory management and IPC.
- IPC (message passing, system calls are very SLOW.

ASSIGNMENT

- Q: Explain process management in Mach system.