CS736 Quiz #2

- What are the differences among internal procedures, entry procedures and external procedures of a Mesa monitor? Which ones are executed with the monitor lock held?
- Why is it ok to notify a conditional variable at arbitrary times in Mesa, whereas it is not ok to do so in Hoare's monitor definitions?
- What are the verification rules for Mesa monitors regarding the monitor invariants?
- Name the solution to the priority inversion problem.

Virtual Memory

- Study of Representative Implementations
  - VAX/VMS (1978)
  - UNIX (1981)
  - Multics (1969)
  - Mach (1984-1987)
- Policy Studies
- Distributed Shared Memory

Virtual Memory in VAX/VMS

- Goal: support time-sharing, batch, real-time
  - predictable performance when necessary
  - 32-bit virtual address space, 512-byte pages
- Process address space
  - high-end address space is kernel address space (system space)
  - low end is user address space; two segments, P0 and P1, that grow in opposite directions

Page Table Organization

- One page table per region
- PTE (32 bits):
  - a valid bit (the highest bit)
  - a protection field (the next five bits)
  - a modify bit (bit 26)
  - a field used by OS (5 bits)
  - the physical page frame number (20 bit)
  - if valid bit is 0, 26-0 can be used for disk block number for on-disk copies;

Accessing Page Tables

- Base and length registers
  - system page table
  - user region page tables (themselves pagable)
- Steps in translation
- Translation look-aside buffer (TLB)
  - two sections: one for kernel, one for user
  - why?
- What happens to TLB at context switch?

Use of Regions in VAX/VMS

- System space: kernel ("executive")
- P0: text, starts at second page
  - catch reference to address 0
- P1: command interpreter, user stack and kernel stack;

Memory Management

- Constraints: slow CPUs, small and slow disks, and small memories
- Concerns:
  - heavily paging program stall others
  - high cost of program startup and restart time
  - increased disk load due to paging
  - CPU cost of searching page lists in memory
- Solution: paging + swapping

Page Replacement

- process-local page replacement policy
  - limit on the resident size of a process
  - resident-set list
  - FIFO within the resident set
  - free list and modified (dirty) list
  - approximation of LRU via FIFO+free/dirty lists

Clustering on Page In/Out

- clustering:
  - relative small page size, and cost of disk operation
  - reading and writing several pages at one I/O operation
    - how does kernel choose cluster size and perform cluster I/O?
  - clustering at process startup time
  - delayed write-backs and write-back clustering

Swapper

- Swapping:
  - swapping as load control
  - one I/O operation for swap in/out