

# Advanced Unix System Administration

Lecture 3  
February 20, 2008

Steven Luo  
<sluo+decal@OCF.Berkeley.EDU>

# Processes

- Signals
  - Allow processes to communicate with each other and the kernel
  - Provide primitive mechanism for implementing callbacks – signals can be trapped and a “signal handler” called
  - If not handled, signals perform a default action (usually exit)
  - Signal programming is tricky because of synchronization and syscall restarting issues
  - Try ``man kill`` or ``kill -L`` for more information

# Processes

- Threads
  - Recall that the kernel keeps lots of state for each process
  - But if the processes are related, we might be able to get away with less of that
  - Threads = “lightweight processes”
  - When threads have kernel support, they're much faster to create and switch
  - Shared resources means programming is more difficult

# Memory Management

- Usually, an  $n$ -bit processor can address  $n$  bits of memory
- Especially on 64-bit systems, this tends to be much more memory than actually exists on the system
- Besides, the physical address of a particular byte may not be a particularly convenient way to work with it

# Memory Management

- Solution: paged memory, virtual memory
  - Divide up physical memory into pages (usually 4K or 8K) and keep track of pages of memory
  - Keep a page table of pages and the memory addresses used to access them
  - Creates more flexibility: per-process virtual address space, non-contiguous allocations, shared memory, etc.

# Memory Management

- Virtual memory
  - As long as the information stored at the address can be retrieved somehow, there's nothing wrong
  - We can map pieces of disk storage (whether swap or memory-mapped files) to an address
  - This is slow, so we can also (and where possible, usually do) keep a copy in physical memory
  - When demand spikes, we can quickly drop pages backed by non-volatile storage

# Memory Management

- Caching VMs
  - RAM is much faster than disk, so keeping info in RAM will speed up many tasks
  - Some kernels (i.e. Linux) will cache file accesses in “free” memory
  - Again, pages can be dropped quickly if memory pressure arises – though this may not always be profitable
  - Mantra: (truly) free memory is wasted memory