Clocks
(CSE 422S)

Ken Wong
Washington University

kenw@wustl.edu
www.arl.wustl.edu/~kenw

Clock Software (1)
- Hard Clock routine is called after each 10 msec tick
- Keep time of day
  » Record time since Jan 1, 1970 during system boot
  » Count ticks since system boot
  » Compute actual time/date when user asks for time/date
- Prevent processes from getting too much CPU time
  » Decrement a process' quantum counter after every tick
  » Clock driver calls CPU scheduler when quantum reaches 0
- Account for CPU usage
  » Update process' usage after each tick (inaccurate)

Clock Software (2)
- Handle alarm and other "timer" system calls
  » Call softclock() routine if current interrupt-priority level is low enough
  » softclock() handles lower-priority timer processing
    » Run at software-interrupt level
    » e.g., TCP timer tick is 200 msec or 500 msec
- Watchdog timers
  » e.g., resume spinning floppy disk
- Profiling, monitoring, statistics gathering
  » Process region usage histogram

Simple Clock
» Interrupts on every voltage cycle (50 or 60 Hz)

More Typical Clock (Programmable)
» Frequencies above 1 GHz (1 nsec per pulse)
» Every pulse causes counter to decrement
» Interrupt when counter reaches 0
» Program to interrupt every 10 msec tick

Crystal Oscillator
Load
16-bit Countdown Counter
Holding Register

1-kenw, 12/8/2005
4-kenw, 12/8/2005
Soft Clock

- Callout queue records waiting events
- Delta
  - Number of ticks from preceding event record
- Callback function
  - Function that should be called when time expires

<table>
<thead>
<tr>
<th>Queue</th>
<th>Delta (ticks)</th>
<th>Callback function</th>
<th>Argument</th>
<th>Expiration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>f()</td>
<td></td>
<td>10 msec</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>g()</td>
<td></td>
<td>40 msec</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>h()</td>
<td></td>
<td>40 msec</td>
</tr>
</tbody>
</table>