## I/O Notification (cont) Communicating with the CPU Polling I/O Interrupt I/O device places information in a status register - Whenever an I/O device needs attention from the processor, it - The OS periodically checks the status register interrupts the processor - Whether polling is used is often dependent upon whether the - Interrupt must tell OS about the event and which device device can initiate I/O independently (accomplished using a Cause register or vectored interrupts) For instance, a mouse works well since it has a fixed I/O rate and Example: Intel 80x86 has 256 vectored interrupts initiates its own data - I/O interrupts are therefore asynchronous events • For others, such as disk access, I/O only occurs under the control of the OS, so we poll such devices only when the OS knows it is · Processor waits until current instruction is completed active - Interrupts may have different priorities - Advantages Advantage: Execution is only halted during actual transfer • Simple to implement Disadvantage: Software overhead of interrupt processing · Processor is in control and does the work - Disadvantage · Polling overhead and data transfer consume CPU time EE108b Lecture 17 C. Kozyrakis EE108b Lecture 17 29 C. Kozyrakis 30 Polling and Programmed I/O I/O Interrupts add read data sub User (1) I/O interrupt from and program memory or nop (2) Save state Busy wait loop Is the device (3) Interrupt ready? unless the device service addr is very fast! read yes Interrupt no store service store Processor may be routine (4) Restore rti data to inefficient way to state and return device transfer data Memory done? no yes EE108b Lecture 17 EE108b Lecture 17 C. Kozyrakis 31 C. Kozyrakis 32