Notes for 10/12/09

Exception versus interrupt

- Exceptions come from inside CPU
- Interrupts come from outside CPU

Context switch—when operating system switches which process is running on the CPU

Currently running process will run until an interrupt:

- For example, a hard drive interrupt to tell the CPU that the data is ready in RAM
- And I/O in general, e.g. network cards
- Or a keyboard interrupt when a key was pressed
- Timer interrupt, for example, generating interrupts 100 times a second so that the operating system can schedule processes

Or until an exception:

- For example, a virtual memory exception, such as when you are accessing an address that there is no page for, in which case the operating system can fix this, or such as when you are accessing invalid addresses, in which case UNIX will send a SIGSEGV signal to your process
- Arithmetic exceptions—such as divide by zero, or overflow on a signed add, in which case UNIX will send a SIGFPE signal
- Illegal instruction—when you jump to memory that cannot be decoded as an instruction, which sends a SIGILL signal on UNIX
- System call—used to request service from operating system

Significance of precise interrupts:

- You want to resume on the exact instruction you left off from
- In case of exception, you want to rerun the instruction that generated the exception

How an exception goes down, for example, for a virtual memory exception:

- OS is trapped
- OS looks at cause register for cause of exception
- In this case, it will see that it is a virtual memory exception
- It will then look at the EPC for the location of the program counter to get the instruction that generated the exception
- It will look at the address that the instruction was attempting to access
- It can then, for example, load data into physical memory and rerun the exception

On x86, the operating system will populate an interrupt vector table with interrupt handlers, which is a table of function pointers that the operating system fills to handle the different interrupt numbers. Note that, with MIPS, you have only one interrupt handler, but it checks the cause register to decide what happened.

Virtual memory—makes it look like each process has the address space to itself. For example, the assembler tends to put e.g. the .text and .data sections at the same address. This abstraction also provides security, since by not providing pages to other processes' pages, you cannot read or write to that them.

IBM System/38 → AS/400 → iSeries
Capabilities Architecture—different from the architecture made popular by the IBM 360 that we are familiar with
With these, everything is a pointer, e.g. even to files
A tag bit is used to say if anything is a pointer or not
If you try to do arithmetic on a pointer, it will unset the tag bit
The tag bit signifies that you have the capability to deference the pointer
On this architecture, everything, including files, is in the same address space