Computer System Overview

Chapter 1 of [OS4e], Chapter 2 of [OSC]:

- Basic Elements
- Processor Registers
- Instruction Execution
- Interrupts
- The Memory Hierarchy
- Cache Memory

Basic Elements

Computer = Processor + Memory + I/O modules
- Processor (CPU) executes instructions
- Instructions and data are stored at addresses in main memory
- I/O modules (device controllers) handle data transfers between computer and environment
Processor Registers

Set of registers provides local storage area for processor
- smaller and faster than main memory

Registers are of two types:
- user-visible registers
- control and status registers

Instructions reference user-visible registers
- processor reads from or writes to those registers during instruction execution
- registers contain data being processed, pointers and indices to addresses; condition codes are sometimes visible
- compilers and assembly-language programmers try to optimise register use
Control and status registers affect how processor executes instructions

- Examples:
  - PC contains memory address of instruction to be fetched
  - IR contains instruction most recently fetched
  - PSW contains condition codes, interrupt enable bit, supervisor mode bit
- Some registers may be referenced by instructions executed in supervisor mode; others are accessible only to hardware

Instruction Execution

Execute cycle involves data transfer between processor and memory (or an I/O module), data processing or changing control flow

Most processors pipeline instruction execution and allow Direct Memory Access
Interrupts

*interrupt* (or *exception*)

- signal sent to processor
  - e.g. attempt to divide by zero
  - e.g. illegal attempt to access address
  - e.g. execution of *trap* instruction (to make “system call”)
  - e.g. I/O transfer has completed
- source and priority of interrupt are recorded

A more realistic model of instruction execution makes provision for interrupts and, hence, Operating Systems!
**Interrupt handling**

- Contents of PC and PSW are stored automatically.
- Interrupt service routine (ISR) is executed in supervisor mode.
- ISR may store contents of other registers.
- ISR may call other operating system routines.
- Eventually, contents of registers may be restored and execution continued in user mode from point of interruption.

N.B. Interrupt handling may itself be interrupted!

→ Store contents of registers in system stack.

(a) Interrupt occurs after instruction at location $N$. 

(b) Diagram illustrating the flow of execution.
The Memory Hierarchy

Storage devices can be put in order of increasing capacity, namely,
- registers, cache memory, main memory, hard disk, tape
  - access time also increases
  - cost per bit decreases
  - first three are *volatile*

*caching*
- copying information into faster device

N.B. “cache memory” is not the same as “disk cache” (space allocated in main memory)
Cache Memory

- Processors can execute instructions faster than instructions (and data) can be fetched from main memory!
- Cache memory provides a solution which relies on *locality of reference* and is invisible to OS.