

## How Does Hardware Execute Software?

1. What is the Instruction Set Architecture (ISA) and what is the primary component of the ISA? [3]

**Solution:**

The ISA is the interface between the hardware and software. The primary component of the ISA is its assembly language

2. Give three instructions of the MIPS (Microprocessor without Interlocked Pipeline Stages) ISA and explain carefully what they do. [6]

**Solution:**

- `lw $s1, ($s2)` - Register \$s1 takes the value of the memory location held in register \$s2
- `addi $s1, $s2, x` - register \$s1 takes the value of register \$s2 plus the number x
- `beq $s1, $s2, ,` - if the value of register \$s1 is equal to the value of the register \$s2 then go to memory location m

3. What do we mean when we say that a MIPS (Microprocessor without Interlocked Pipeline Stages) instruction is 32 bits wide? [2]

**Solution:**

There are 32 bits in the MIPS instruction

4. What is a process control block (PCB) and what is one used for by the operating system? Name 3 items in a PCB. [6]

**Solution:**

- **Process control block(PCB)** - a data structure that the kernel uses in order to manage a process
- It is used so that the current situation of any process is fully understood, this means that the CPU can better handle processes
- Components
  - the process's unique ID
  - The current state of the process
  - CPU scheduling information such as the process priority

5. What are the three types that MIPS instructions are partitioned into? Briefly describe these types.

[6]

**Solution:**

- I type instructions
  - Involve data transfer
  - 6 bit op code
  - 5 bit Rs (Source Register)
  - 5 bit Rd (Destination Register)
  - 16 bit address
- R type instructions
  - Work on registers
  - 6 bit op code
  - 5 bit Rs1 (first source register)
  - 5 bit Rs2 (Second source register)
  - 5 bit Rd (Destination Register)
  - 5 bit shift (The shift amount for shift instructions)
  - 6 bit funct (A supplement for opcode for some instructions)
- J type instructions
  - Involve jumps
    - \* 6 bit opcode
    - \* 26 bit address

6. Detail four different functions of the operating system.

[4]

**Solution:**

- Virtualises a machine
  - Provides abstractions that present clean interfaces to make the computer easier to use.
- Starts and stops programs
  - Ensures that when a program is stopped, it's memory is freed up
- Manages memory
  - Ensures programs can still run, even if the memory they request is in use.
- Handles input/output
  - Handles interrupts from inputs and outputs so that they don't have to wait for the processor to finish what it is doing.

7. What is data security (in the context of operating systems)?

[1]

**Solution:**

Ensuring that the memory allocated to each program is kept separate and secure from other programs

8. What is virtualization (in the context of operating systems)?

[2]

**Solution:**

Providing abstractions that present clean interfaces to make the computer easier to use

9. Explain what an interrupt is and how the operating system uses interrupts to deal with peripheral devices that operate much more slowly than the CPU.

[4]

**Solution:**

- An interrupt is an input from the user which should be handled immediately by the cpu
- The operating system ensures the actions are coordinated
- Ensures no programs end up being closed because a mouse is clicked

10. What is process within an operating system and what are the two essential elements of a process? Is it the case that an operating system for a CPU with one processor can only have one process?

[5]

**Solution:**

- Process - Each executing program
- A process consists of
  - **Thread** - A sequence of instructions in the context of a sequential execution
  - **Address space** - Consists of some memory locations that the corresponding process can read from and write to
- Multiple processes can be run on one processor, providing there is no mutual exclusion

11. Explain the principle of mutual exclusion and give an illustration to show that ensuring mutual exclusion is important.

[6]

**Solution:**

- **Mutual Exclusion** - Ensuring that two threads are not in the critical selection at the same time
- **Critical Selection** - Exclusive access to some shared resource such as memory location
- Two threads have access to the same counter, and each want to increment the counter, depending on how the read, increment, store is handled, the result will be different

12. Explain how the principle of virtual memory works. Why is virtual memory required?

[3]

**Solution:**

- Moving memory backwards and forwards between RAM and the hard disk as and when required
- Virtual memory is required when there is insufficient RAM to store what all the running programs wish to store in RAM
- Lower priority processes are moved to hard disk first

13. Describe the life-cycle of processes within the operating system (be sure to explain each component of the life-cycle).

[10]

**Solution:**

- **new:** the process being created
- **ready:** the process not executing on the CPU but is ready to execute
- **running:** the process is executing on the CPU
- **blocked:** the process is waiting for an event (and so not executable)
- **exit:** the process has finished

There are the following state transitions

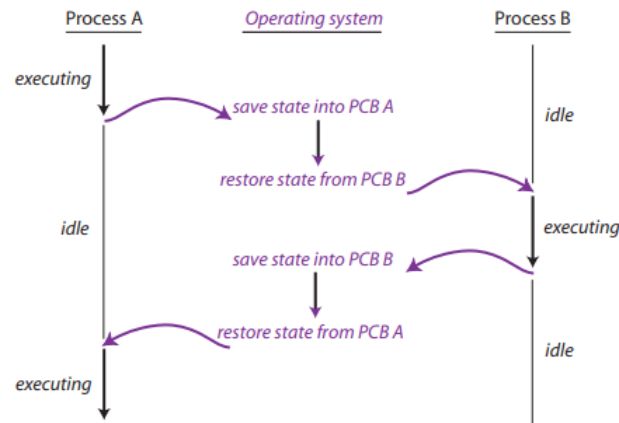
- **admit:** the control overheads as regards a process have been set up and the process is moved to the run queue
- **dispatch:** the scheduler allocates the CPU to an executable process
- **timeout/yield:** the executing process is forced to/volunteers to release access to cpu
- **event-wait:** a process is waiting for an input/output event, for example, and gives up access to the cpu
- **event:** an event occurs and wakes up a process
- **release:** a process terminates and releases access to the CPU and other resources

14. Explain the principle of context switching with regard to process control blocks.

[5]

**Solution:**

**Context switching** - Where the operating system pauses one process and resumes another process



15. Give two illustrations of principles of Computational Thinking in the context of instruction set architectures and operating systems.

[2]

**Solution:**

- Tessellation - Changing the model of processes to "cells", each with their own guarantees of resources
- Technological advances such as multi-core, cloud computing and embedded systems