OS Concepts

- Process management
- Scheduling
- Memory management
- I/O, file systems
- Networking
- Security

Process Management

- Process
  - Is a programme in execution
  - Consists of:
    - Programme code
    - Data
    - Program counter
      - points to next instruction
      - also know as instruction pointer (IP)
    - Stack pointer
      - Points to the top of the stack (where temporary data is stored)
    - Registers
    - Other info

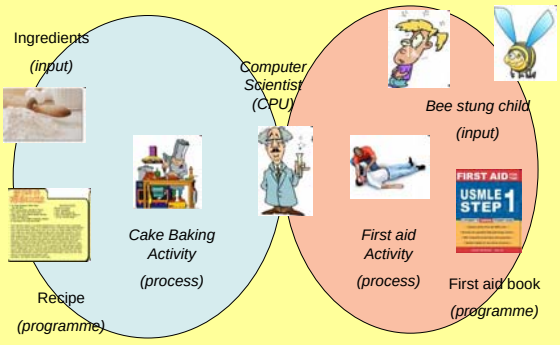
Process Management

- OS stores process information in the **process table**
  - One entry for each process
- a process consists of:
  - its address space (core image)
  - process table entry
- a process can
  - create 1 or more child processes
  - request or release memory
  - communicate with other processes
- a process usually has a user id (*uid*) and a group id (*gid*)
- Interrupts can occur

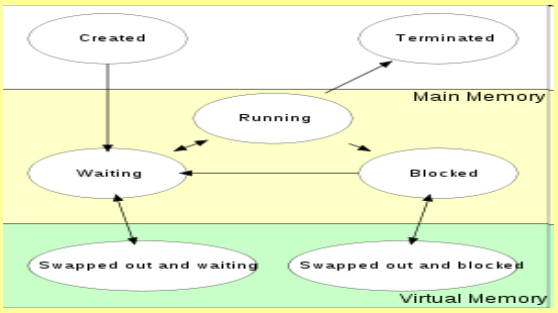
Process Management

- Difference between process and programme
  - Programme
    - is a list of instructions
  - Process
    - is an activity,
    - it has a programme, input, output, state
- Process hierarchies
  - Higher-priority process
  - Switching between processes

Process and Programme analogy



Process States



Process States

- Created
  - The process is set up
- Waiting
  - The process is ready and runnable, waiting for some CPU time
- Running
  - The process is using the CPU
- Swapped out and waiting
  - The process is not in main memory and is waiting to get access to the CPU
- Blocked
  - The process is unable to run until some external event happens
- Swapped out and blocked
  - The process is not in main memory and is waiting for some external event o happen
- Terminated
  - The process is finished

Process Management – Interprocess Communication

- Process can communicate with other processes
- 4 conditions required for successful cooperation between processes

  1, No two processes may be simultaneously inside their **critical sections**
  - (critical section that part of the programme where the shared memory is accessed is called the critical section)

  2. No assumptions may be made about speeds or the number of CPUs

  3. No process running outside its critical section may block other processes

  4. No process should have to wait forever to enter its critical section

Process Management – Interprocess Communication

- Mutual exclusion is important to prevent problems
- Mutual exclusion:
  - When one process is updating shared memory in its critical region, no other process will enter *its* critical region and cause problems
- Many ways of implementing mutual exclusion
  - Disabling interrupts
    - OK for internal OS, not for general processes
  - Semaphores (Dijkstra, 1965)
  - Beware of **deadlocks** (when processes are blocked and remain so forever)
  - Dining Philosopher Problem (Dijkstra, 1965)
  - Sleeping Barber Problem

Process management - Summary

- Understand the difference between a **programme** and a **process**
- Understand the different process states and how a process changes states
- Know the conditions for successful interprocess communication
- Understand mutual exclusion and the problems associated with it