A deadlock occurs when two or more processes are waiting for an event that can only be generated by one of the waiting processes.

**DL prevention:** A sign which says "No right turn into HKU" for East-bound traffic

**DL avoidance:** Look and proceed (or not proceed)

**DL detection/recovery:** Let it happen! ... then recover
Deadlock Characterization

- Resource allocation
  - Request => Use => Release

- Conditions
  - Mutual exclusion: resources cannot be shared
  - Hold and wait: a process holds some resources but needs a resource that is held by another process
  - No preemption: resources cannot be preempted
  - Circular wait: there must be a circular chain of processes, each of which is waiting for a resource held by the next in the chain
Deadlock Handling

- **Prevention**
  - Ensure that at least one of the conditions necessary to characterize a deadlock will never hold

- **Detection and recovery**
  - Allows deadlocks to occur
  - Detection algorithm is run periodically
    - Allocated resources X waiting processes
  - Recovery algorithm is run whenever a deadlock is detected
    - Process termination
    - Resource preemption (rollback)

- **Practice**
  - Too expensive, seldom used!