Permissions:

* Every process has from 6 to 21 permission related IDs
  + Real UID & GID
  + Effective UID & GID
  + “saved setuid” and “saved setgid”
  + 0-15 supplementary GIDs
* Every file/directory traditionally has 2 IDs and 11 access bits, plus sticky bit:
  + UID & GID
  + 3 bits specifying read/write/execute for UID
  + 3 bits specifying read/write/execute for GID
  + 3 bits specifying read/write/execute for “other”
  + “setuid” bit
  + “setgid” bit
  + “sticky” bit
* Nowadays, most file systems support advanced file permissions, known as Access control Lists or ACLs

Read/Write Access

* To determine whether process may read (write) a file:
  + Is effective UID == file’s UID?
    - Does UID have read (write) permission?
  + Is effective GID == file’s GID?
    - Does GID have read (write) permission?
  + Repeat above for supplementary GIDs
  + Does file permit read(write) access to “other”?

Execute Access

* Using same algorithm, execute access is permitted/denied
* In addition, if file has “setuid” (setgid) bid, then exec call changes process’s UID (GID) to that of file’s owner
* Both effective UID (GID) and saved-setuid (gid) are changed to file’s UID (GID)