# CHEAT SHEET — Information Security Assurance + Operating Systems

## INFORMATION SECURITY ASSURANCE

### Cryptography Basics

- Confidentiality, Integrity, Availability
- Plaintext → Algorithm → Ciphertext
- Key determines encryption output

### Symmetric vs Asymmetric Encryption

Symmetric:

- Same key, fast (AES, DES)

Asymmetric:

- Public/Private key, slower (RSA, ECC)

### Hashing & Digital Signatures

Hashing:

- One■way, fixed output (SHA■256)

Digital Signatures:

- Hash → Sign with private key → Verify with public key

### PKI Overview

- CA issues certificates, RA verifies
- X.509 certificates

### Access Control Models

- DAC – Owner controls access
- MAC – Based on security labels
- RBAC – Based on roles

### Authentication Methods

- Passwords
- Biometrics
- MFA – 2+ factors (know/have/are)

IAM Principles

- Least Privilege, SoD, Accountability
- Provisioning/Deprovisioning

OPERATING SYSTEMS

Process

- Program in execution; has memory, registers

Kernel

- Core OS component

Program

- Passive file on disk

Process Life Cycle

- New → Ready → Running → Waiting → Terminated

Process Control Block (PCB)

- PID, PC, registers, memory, scheduling info

Schedulers

- Long■term, Medium■term, Short■term

Dispatcher

- Performs context switching

Context Switch

- Saves/restores process state

Program Counter (PC)

- Next instruction address

Scheduling Algorithms

- FCFS, SJF, SRTF, Priority, Round Robin

Inter■Process Communication (IPC)

- Pipes, Message Queues, Shared Memory, Semaphores, Sockets

### Multithreading Basics

- Threads share code/data; lightweight

### Process vs Thread

- Process: heavyweight, own memory
- Thread: lightweight, shared memory

### Thread Building Blocks

- PC, registers, stack

### Fibers

- User■managed lightweight threads

### Preemptive vs Cooperative

- Preemptive: OS interrupts
- Cooperative: threads yield

### Single vs Multiprocessor Scheduling

- Single CPU vs load balancing

### Thread Pools

- Pre■created thread workers

### Synchronization Tools

- Mutex, Semaphore, Monitors, Locks

### Types of Threads

- User■Level Threads (ULT)
- Kernel■Level Threads (KLT)

### ULT vs KLT

- ULT: fast, user■managed, no multi■CPU
- KLT: slower, OS■managed, supports multi■CPU