VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

B.Tech. Winter Semester 2024-25
School Of Computer Science and Engineering
(SCOPE)

Notes
Information Security

Apurva Mishra: 22BCE2791
Date: CAT - II

Contents

1 Module:3 Operating Systems Security 3

1.1 Security in Operating System 3

1.2 Security in the design of OS 6

1.3 Trusted Operating System Design 8

1.4 Rootkit. 10

2 Module:4 Security Countermeasures 7 hours 10

2.1 Design of Firewalls 10

2.2 Types 10

2.3 Personal Firewalls 10

2.4 Configurations 10

2.5 Network Address Translation 10

2.6 Data Loss Prevention 10

2.7 Intrusion Detection and Prevention Systems: 10

3 Module:5 Database Security 10

3.1 Database Security 10

3.2 Database Security Requirements 10

3.3 Reliability and Integrity 10

3.4 Sensitive Data 10

3.5 Types of Disclosures 10

3.6 Preventing Disclosures 10

3.7 Inference 10

3.8 Multilevel Databases 10

3.9 Multilevel Security 10

3.10 Database Attacks 10

3.11 SQL Injection Attacks. 10

1 Module:3 Operating Systems Security

1.1 Security in Operating System

1.1.1 Functions of OS

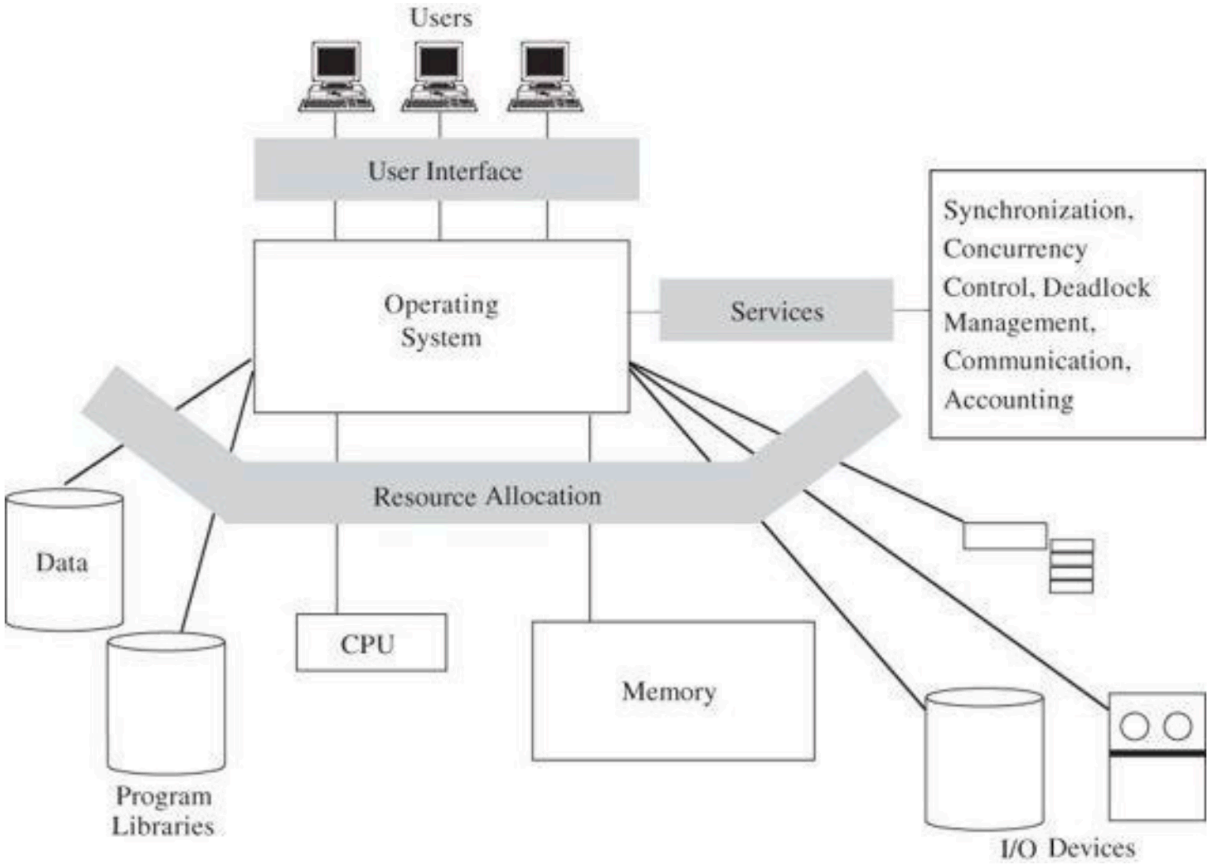

FIGURE 5-1 Operating System Functions

1. **Enforce Sharing**: Integrity control such as monitors and transaction processes.
2. **IPC**
3. **Protection of Critical Sections**

4 **Interface to Hardware**
5. **User Authentication**
6. **Separate Memory**: Each user program runs in protected portions of memory

1.1.2 Multi-Programming and Shared Use

Realizing that two users could interleave access to the resources of a single computing system, researchers developed concepts such as scheduling, sharing, and concurrent use. Multiprogrammed operating systems, also known as monitors, oversaw each program’s execution. Monitors took an active role, whereas executives were passive. That is, an executive stayed in the background, waiting to be called into service by a requesting user.

1.1.3 Multitasking

1. **Process**: Program run by user
2. **Domain**: Resources allocated to process
3. **Threads**: Process consist of one of more threads
4. **Tasks**: Threads can spawn task which are smmaller executable units of code

OS switches between process allocating and deallocating resources as needed.

1.1.4 Layered Design

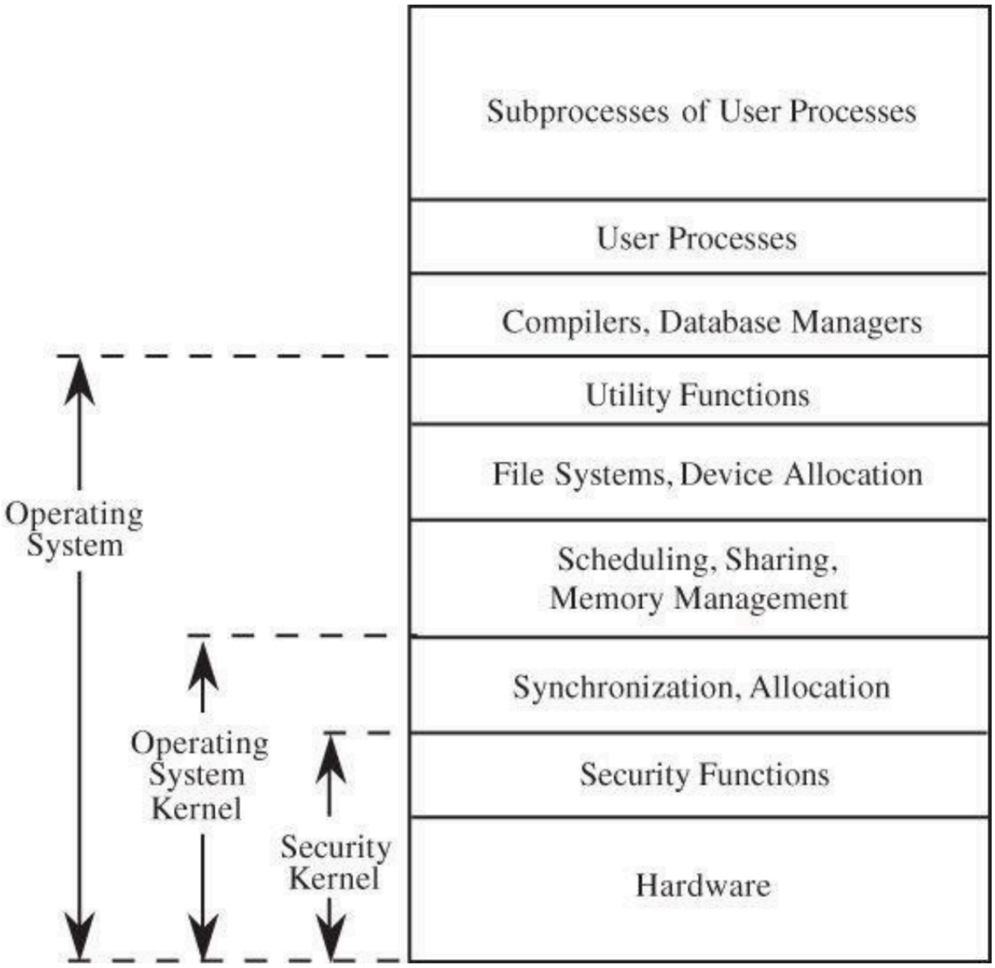
Subprocesses of User Processes
User Processes
Compilers, Database Managers
Utility Functions
File Systems, Device Allocation
Scheduling, Sharing, Memory Management
Synchronization, Allocation
Security Functions
Hardware
Operating System
Operating System Kernel
Security Kernel

FIGURE 5-2 Layered Operating System

OS also expose certain **hooks** to which programs such as antivirus can attach to so that they load before user executables.

1.1.5 Self Protection by OS

The default expectation is one level of hardware-enforced separation (two states). This situation means that an attacker is only one step away from complete system compromise through a **get-root** exploit.

This is partly mitigated by the layerd nature of the OS.

1.1.6 Tools to implement Security Functions

1. **Audit Logs**: However this is not possible, on scale due to sheer volume of data.
2. **Virtualization**: presenting a user the appearance of a system with only the resources the user is entitled to use.
3. **Honeypot**: system to lure an attacker into an environment that can be both controlled and monitored
4. **Separation**: Using
   - Space: Physical
   - Time: Physical
   - Access Control: Logical
   - Cryptography

IMPORTANT

5. **Fence**: There are two ways:
   1. Hardware Fence which is hardcoded
   2. **Fence Register** which contains the address of the end of OS and can be reprogrammed.
6. **Base and Bound Register**:
   1. **Base Register**: Starting address of the program memory
   2. **Bound Register**: Upper address limit for the program.
7. **Tagging Architecture**: Every word of machine memory has one or more word to determine access rights.
8. **Virtual Memory**:
   1. **Segmentation**: A logical contiguous sequence of memory that is can be mapped to actual physical memory. A table must map logical segment to physical address.
      Advantages:
      - Can be easily moved in physical memory without affecting logiacl mapping.
      - A segment can be moved from main memory.
      - Every reference passes through the table, so it can be checked.
      - Table can also be used to maintain access control.
   2. **Paging**: Program and memory are divided in equal size pieces called pages. Program pages are mapped to pages in physical memory and may not be contiguous. More memory efficient management than segments.

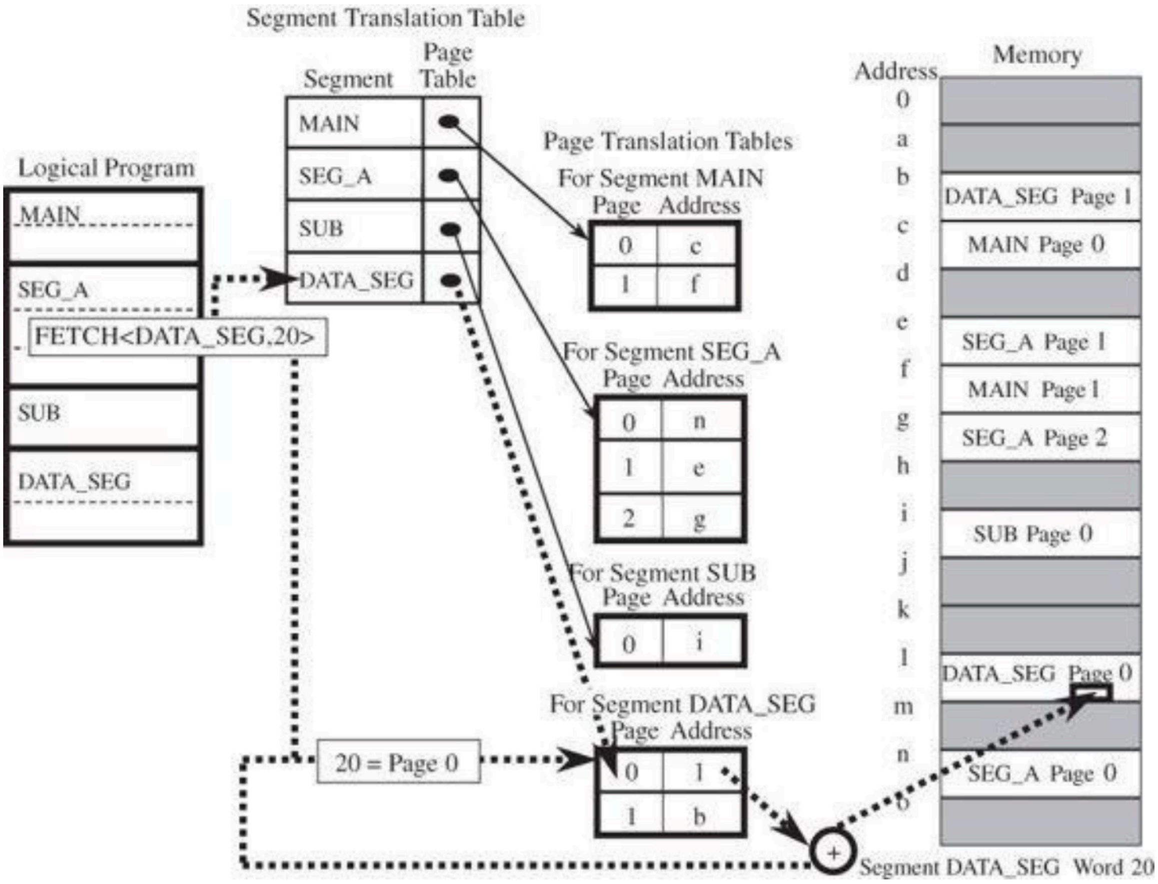

FIGURE 5-14 Address Translation with Paged Segmentation

Figure 1: Segmentation + Paging Combination

1.2 Security in the design of OS

Secure Design principles:

1. Least privilege
2. Open Design
3. Permission Based
4. Separation of Privileges

1.2.1 Simplified Design,

Security should be incroporated from the start. Keep the design simple to keep it maintaineable and auditable.

1.2.2 Layered Design,

1. Hardware
2. Kernel
3. OS
4. User

With sublayers. This allows system to **evolve** over time.

Advantages:

1. **Layered Trust**:
   Most sensitive tasks are assigned to lower levels, with increasing level of authorization required to access.
2. **Encapsulation**
3. **Damage Control/Isolation**
4. **Auditability**: Lower layers can be analyzed more critically.

1.2.3 Kernelized design,

Security functions are contained in Security Kernel.

Advantages:
1. **Access Control**
2. **Easier Tracing**
3. **Modifiability**
4. **Compactness**
5. **Verifiability**

1.2.4 Reference Monitor,

This is like a wall. It separates subjects and objects, enforcing that a subject can access only those objects allowed by security policy.

Reference Monitor **must me:**
1. **Tamperproof**
2. **Unbypassable**
3. **Verifiable**

1.2.5 Trusted Systems

One with evidence to substantiate the claim it implements some function or policy. We can say the code is trused if its **rigorously tested and anlysed.**

Key Characteristics:
1. *Functional Correctness*
2. *Integrity*
3. *Limited Privileges*

1.2.6 Trusted Systems Functions

Trusted Computing Base (TCS): Everything necessary in computer to enforce security policy.

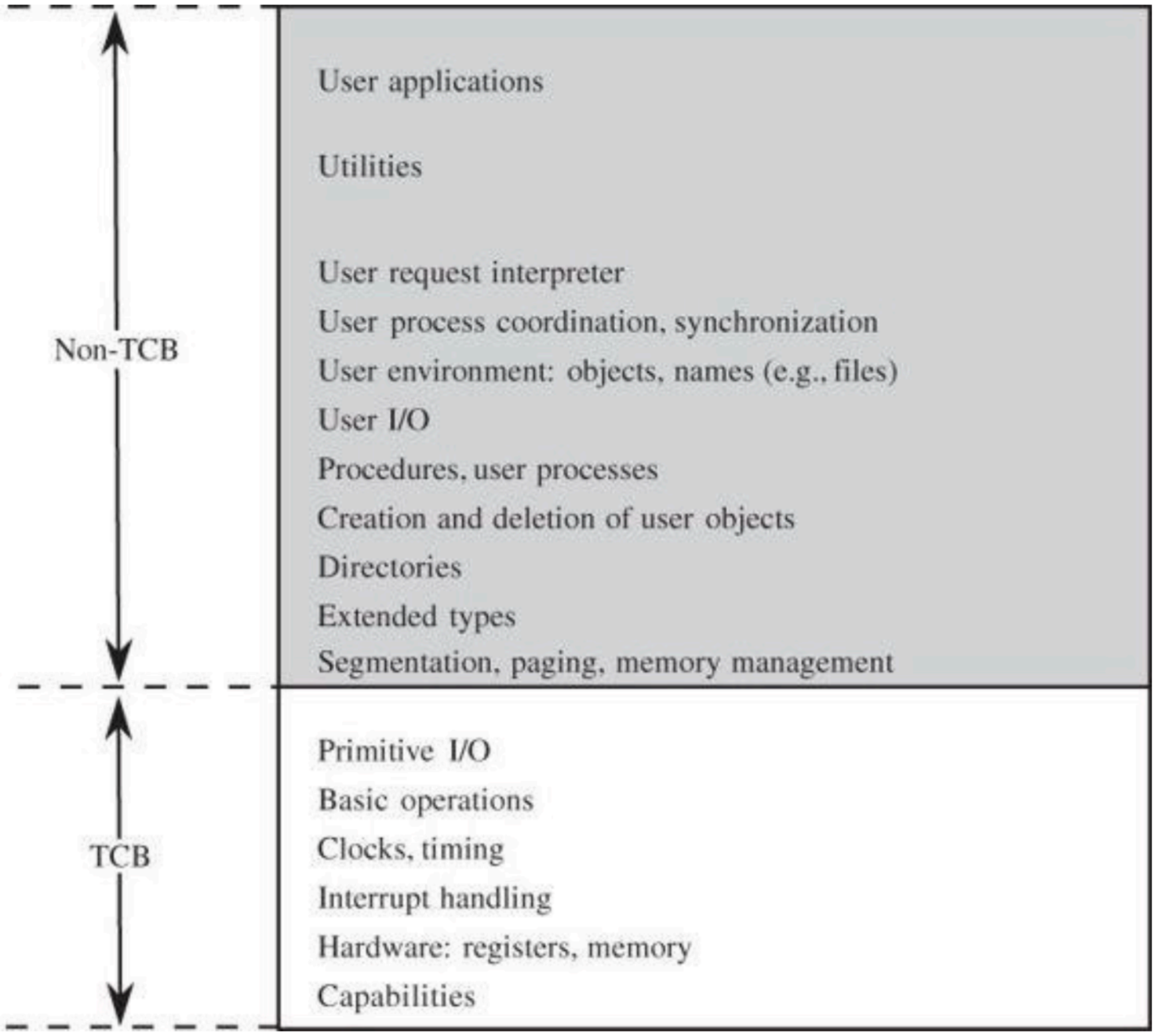
Non-TCB
TCB
User applications
Utilities
User request interpreter
User process coordination, synchronization
User environment: objects, names (e.g., files)
User I/O
Procedures, user processes
Creation and deletion of user objects
Directories
Extended types
Segmentation, paging, memory management
Primitive I/O
Basic operations
Clocks, timing
Interrupt handling
Hardware: registers, memory
Capabilities

Figure 2: Parts on OS mananged by TCB

TCB Monitors:

1. **Process Activation**: This involves complete change of register allocation, file access control, (**domains**) etc which are security sensitive.
2. **Execution Domain Switching**: Process running under one **domain** invokes another domain for services.
3. **Memory Protection**: The **domain** (*memory*) is monitored my TCB.
4. **I/O**: Information flowing in and out of **domain**.

Thus *TCB* manages **domain**:

- Invocation
- Execution
- Access
- Information flowing in and out of it

1.3 Trusted Operating System Design

1. **Layered Design with Security Kernel**

2. **Secure Startup**: When system starts, all security functions must be working properly and no effect remain from previous session. Thus not malicious code can block security enforcement.
3. **Trusted Path**: is an unforgeable connection by which the user can be confident of communicating directly with the operating system, not with any fraudulent intermediate application.
4. **Object Reuse**: All space previously allocated is cleared before it is reassigned.
5. **Audit**: All security related changes can be tracked.

1.4 Rootkit.

2 Module:4 Security Countermeasures 7 hours

2.1 Design of Firewalls

2.2 Types

2.3 Personal Firewalls

2.4 Configurations

2.5 Network Address Translation

2.6 Data Loss Prevention

2.7 Intrusion Detection and Prevention Systems:

2.7.1 Types of IDSs

2.7.2 Intrusion Prevention system

2.7.3 Intrusion Response

2.7.4 Goals of IDSs

2.7.5 Strength and Limitations

3 Module:5 Database Security

3.1 Database Security

3.2 Database Security Requirements

3.3 Reliability and Integrity

3.4 Sensitive Data

3.5 Types of Disclosures

3.6 Preventing Disclosures

3.7 Inference

3.8 Multilevel Databases

3.9 Multilevel Security

3.10 Database Attacks

3.11 SQL Injection Attacks.