Sistem Operasi

Tugas Latihan Sistem Bilangan

Dosen Pengampu :
Dr Ferry Astika Saputra ST, M.Sc

Disusun Oleh :
Bagus Insan Pradana D3 IT A (**3124521007**)

PROGRAM STUDI D3 TEKNIK INFORMATIKA PSDKU LAMONGAN
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2025

Pratice Exercises.

1.1 What are the three main purposes of an operating system?

Answer :

The three main purposes of an operating system are:

1. Resource Management: The OS manages hardware resources such as the CPU, memory, storage, and I/O devices, ensuring efficient allocation and utilization.
2. Process and Task Management: The OS schedules and manages processes, allowing multiple applications to run simultaneously while ensuring fair use of CPU time.
3. User Interface and System Services: The OS provides an interface (CLI or GUI) for users to interact with the system and offers essential services like file management, security, and networking.

1.2 We have stressed the need for an operating system to make efficient use of the computing hardware. When is it appropriate for the operating system to forsake this principle and to "waste" resources? Why is such a system not really wasteful?

Answer :

It is appropriate for an operating system to "Waste" resources when prioritizing convenience, responsiveness, and system security over raw efficiency because the comfort of the user is number 1.

Why is it not truly wasteful? Although some resources appear underutilized, they contribute to system stability, usability, and security. This trade-off ensures that the system is able to meet the latest modern computing needs by making it more effective for users rather than just being optimized for pure efficiency.

1.3 What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

Answer :

The main difficulty in writing an operating system for a real-time environment is ensuring that the operating system is within the fixed time constraints of a real-time system. In other words, the computation must be completed within a defined deadline. Otherwise, system failure can occur.

Overall, writing a real-time OS required efficient, predictable, and low latency execution, making it more complex than general-purpose OS development.

1.4 Keeping in mind the various definitions of ***operating system,*** consider whether the operating system should include applications such as web browsers and mail programs. Argue both that it should and that it should not, and support your answer.

Answer :

Yes, it should. Because :

- User Convenience – Pre-installed apps provide immediate access without extra setup.
- Better Integration – Optimized for performance and security within the OS
- Standardization & Security – Ensures a consistent experience with managed updated

No, it shouldn't. Because :

- Bloatware – Unnecessary software & apps consume memory and resources.
- Limited User Choice – Reduces competition and customization.
- Security Risks – vulnerabilities in bundled apps that can affect the entire system and possibly can make malware go through the system.

1.5 How does the distinction between kernel mode and user mode function as a rudimentary form of protection (security)?

Answer :

It is able to prevent user applications from directly accessing critical system resources, also reducing the risk of crashes or malicious actions.

By enforcing these access levels, the OS maintains security, stability, and process isolation.

1.6 Which of the following instructions should be privileged?
a. Set value of timer.
b. Read the clock.
c. Clear memory.
d. Issue a trap instruction.
e. Turn off interrupts.
f. Modify entries in device-status table.
g. Switch from user to kernel mode.
h. Acces I/O device.

Answer :

Privileged :

a. Set value of timer: Prevents user programs from disabling time-sharing.
c. Clear memory: Prevents unauthorized memory access or deletion.
e. Turn off interrupts: Prevents user programs from blocking OS control.
f. Modify entries in device-status table: Controls hardware access and must be restricted.

g. Switch from user to kernel mode: Only the OS should control mode switching.
h. Access I/O device: Direct I/O access could lead to security risk or resource conflicts.

Non-Priviliged :

b. Read the clock: Simply retrieves time, no security risk.
d. Issue a trap instruction: Used by programs to request OS services

Conclusion: Privileged instructions control hardware and critical resources, ensuring security and system stability by restricting direct access to the OS.

1.7 Some early computers protected the operating system by placing it in a memory partition that could not be modified by either the user's job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.

Answer :

Placing the operating system in a read-only memory partition has its own challenges :

1. Lack of OS Updates and Fixes: If the OS is unmodifiable, bug fixes, security patches, and feature upgrades become impossible without replacing hardware or using inefficient workarounds.
2. Inflexibility in Resource Management: The OS cannot dynamically adjust memory allocation or modify kernel data structures, limiting its ability to manage system resources efficiently.

This approach increases stability but sacrifices adaptability, making it impractical for modern systems requiring frequent updates and optimizations.

1.8 Some CPUs provide for more than two modes of operation. What are two possible uses of these multiple modes?

Answer :

Two possible uses for these additional modes are:

1. Enhanced Security & Virtualization:
   Extra modes, such as hypervisor mode (ring -1), allow the OS to run virtual machines efficiently, isolating guest operating systems from the host for better security and performance.
2. Granular Privilege Levels: Intermediate modes (ring 1 and 2 in x86) allow finer control over access to system resources.

These additional modes improve system security, stability, and virtualization capabilities.

1.9 Timers could be used to compute the current time. Provide a short description of how this could be accomplished.

Answer :
A timer can be used to compute the current time by counting lock ticks from a known reference point. This method allows the system to keep track of time accurately, even if there is no real-time clock hardware available.

1.10 Give two reasons why caches are useful. What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

Answer :
Give two reasons why caches are useful!

1. Speed Improvement (Caches store frequently accessed data closer to the CPU, reducing access time compared to slower main memory or storage.)
2. Reduced Bottlenecks (By storing frequently used data, caches reduce the load on primary storage.

What problems do they solve?

- Remove slow memory access (Helps bridge the speed gap between the CPU and slower memory/storage).
- Fix Repeated Data Access (Able to avoid refundant fetches from slower storage by keeping frequently used data readily available.)

What problems do they cause?

- Consistency Issues (Cache data may become outdated if the main storage is modified).
- Higher Cost per Byte (Cache memory is much more expensive than primary storage (DRAM) or secondary storage (HDD/SSD).).

If a cache can be made as large as the device for which it is caching (for instance, a cache as large as a disk), why not make it that large and eliminate the device?

- Cost Inefficiency (A cache as large as a disk would be extremely expensive).
- Diminishing Return (Caches work best with a small, frequently accessed subset of data; caching an entire device provides little extra benefit).
- Higher Power & Space Constraints (Large caches consume more power and physical space, making them impractical for many systems).

1.11 Distinguish between the client-server and peer-to-peer models of distributed systems.
Answer :

The Client-Server model is a centralized network model or a server that provides services and multiple clients who request them. The server's also able to manage resources, authentication, and data access. (For example: Websites, Email servers.)

- **Pros** : Centralized control, easier management/maintenance, & better security.
- **Cons** : Single point of failure, server overload.

The Peer-to-Peer model is a decentralized network model where devices called **"peers"** communicate directly with each other without relying on a central server. Peers also act as both clients and servers, sharing resources directly. (For example: BitTorrent, Blockchain.)

- **Pros** : No point of single failure, scalable, fault-tolerant, efficient resource use.
- **Cons** : Harder to secure and manage, security risk.

1.12 How do clustered systems differ from multiprocessor systems? What is required for two machines belonging to a cluster to cooperate to provide a highly available service?
Answer :

The difference between Clustered and Multiprocessor Systems :

1. Structure :
    - Clustered Systems: Multiple independent machines connected via a network
    - Multiprocessor Systems: Multiple CPUs in a single system sharing memory & resources
2. Resource Sharing :
    - Clustered Systems: Each machine has its own memory & OS instance
    - Multiprocessor Systems: CPUs share memory, OS, and resources
3. Scalability :
    - Clustered Systems: Easily expanded by adding more machines
    - Multiprocessor Systems: Limited by hardware constraints
4. Failure Handling :
    - Clustered Systems: Can continue running if one node fails
    - Multiprocessor Systems: Failure of a CPU may crash the system

The Requirements for High Availability in a Clustered System
For two machines in a cluster to be able to work together efficiently & ensure high availability, they need:

1. **Shared Storage**: Both machines should access the same data via **NAS** or **SAN**.
2. **Failover Mechanism**: If one machine fails, the other must take over automatically.
3. **Load Balancing**: Requests should be distributed between machines to prevent overloading.
4. **Cluster Management Software**: Software like "Pacemake, Kubernetes, or Windows Failover Clustering" coordinates system operations and failure recovery.

1.13 Consider a computing cluster consisting of two nodes running a database. Describe two ways in which the cluster software can manage access to the data on the disk. Discuss the benefits and disadvantages of each.
Answer :

Two Ways to Manage Access to Data in a Clustered Database System:

1. Shared Disk

- **How does it work? :** Both nodes access a shared disk (SAN or NAS), and the cluster software ensures data consistency.
- **Benefits :**
  - No need for data replication
  - Easier to maintain and update because all data is already in one place.
- **Disadvantages :**
  - Risk of contention (Both nodes must coordinate to prevent conflicts)
  - Performance bottlenecks if multiple nodes frequently access the same data

2. Shared Nothing
   - **How does it work? :** Each node has its own database copy, and synchronization.
   - **Benefits :**
     - Better scalability
     - No single point of failure (If one node fails, other still have data)
   - **Disadvantages :**
     - More complex to manage
     - There is a risk of data conflicts if updates occur on multiple nodes before synchronization.

1.14 What is the purpose of interrupts? How does an interrupt differ from a trap? Can traps be generated intentionally by a user program? If so, for what purpose?

Answer :

What is the purpose of interrupts? :

Interrupts allow the CPU to temporarily stop its current task to handle important/urgents events efficiently, improve system responsiveness and multitasking.

How does an interrupt differ from a trap? :

Interrupts	Trap
External (Hardware-Generated)	Internal (Software-Generated)
Asynchronous	Synchronous
Handles external events	Handles exceptions/system request

Can traps be generated intentionally by a user program? If so, for what purpose? :

Yes, it can! A trap can be intentionally triggered by a user program using the system call to request OS service, like :

- File operations
- Process control
- Memory allocation

1.15 Explain how the Linux kernel variables HZ and jiffies can be used to determine the number of seconds the system has been running since it was booted.

Answer :

In the Linux kerenel, HZ and jiffies are used to measure system uptime:

- HZ (Defines the number of clock ticks (jiffies) per second.
- Jiffies (A counter that increments with every clock tick since system booted).

Calculating System Uptime

The number of seconds since boot can be determine using :

Uptime (seconds) = jiffies/HZ

For example (example calculations) :

Assume **HZ = 1000** & **jiffies = 5.000.000** , then the calculation should be like this :

Uptime = 5.000.000 / 1000 = 5000 seconds = 1.39 hours

1.16 Direct memory access is used for high-speed I/O devices in order to avoid increasing the CPU's execution load.
a. How does the CPU interface with the device to coordinate the transfer?
b. How does the CPU know when the memory operations are complete?
c. The CPU is allowed to execute other programs while the DMA controller is transferring data. Does this process interfere with the execution of the user programs? If so, describe what forms of interference are caused.

Answer :

a. How does the CPU interface with the device to coordinate the transfer?
The CPU set up the DMA controller (DMAC) by specifying :
1. Source and destination addresses in memory.
2. Amount of data to transfer.
3. Direction of transfer
4. Control signals

Once the setup is complete, the DMA controller takes over the data transfer without CPU interference.

b. How does the CPU know when the memory operations are complete?
The DMA controller will notify the CPU with two ways :
1. Interrupts (The DMA controller sends an interrupt when the transfer is finished)
2. Polling (The CPU checks a status register to determine if the transfer has finished)

c. Does the DMA controller interfere with user programs? If so, describe what forms of interference are caused :
Yes, it is. DMA can interfere with user programs, usually in these two methods :
1. Memory Access Contention (Since DMA & CPU share the same memory, the cpu CPU might experience delays when trying to access RAM while a DMA transfer in process).

2. Bus Contention (The system bus is occupied during DMA transfer, slowing down CPU process operation that need the bus)

However, modern DMA controller use this techniques called burst mode & cycle stealing so it will be able to minimize interference and ensure smooth multitasking.

1.17 Some computer systems do not provide a privileged mode of operation in hardware. Is it possible to construct a secure operating system for these computer systems? Give arguments both that it is possible and that it is not possible!

Answer :

Is it possible to construct a secure operating system for these computer systems? :

The common answer is **no, it's not** possible because a truly secure OS is nearly impossible. The main reason is that some computer systems didn't provide a privileged mode. In other words, the OS will not be able to restrict access to hardware or critical system operations.

Give arguments both that it's possible & it's not possible :

Possible	It's Not Possible
Software-Based Protection: Security mechanisms can enforce rules.	**Lack of Hardware Enforcement** Without a privileged mode, any program can execute any instruction, making it impossible to prevent any malware or unknown access.
User-Level Emulation The OS can run in user mode with software-controlled access to resources, preventing direct hardware control.	**No Memory Isolation** Programs can access or modify each other's memory, leading to crashes or security breaches.
Capabilities-Based Security System can use capability-based access control (reducing risk)	**Impossible to Enforce System Calls** The OS can't control access to critical operations.
Language-Based Security Enforcing strict coding practices and sandboxing that are able to prevent unauthorized access	**Difficult to Prevent Rootkits** Since all applications have full control, malware can easily compromise the systems.

1.18 Many SMP systems have different levels of caches; one level is local to each processing core, and another level is shared among all processing cores. Why are caching systems designed this way?

Answer :

Caching systems are designed this way with local and shared caches to strike the perfect balance between speed, efficiency, & scalability in multi-core (SMP) systems. This will have a great impact, for example :

1. Faster data access
2. Reduced memory bottleneck
3. Efficient data sharing

4. Better scalability

This design will also be able to optimize performance, reduce delays, and enhance the CPU efficiency in SMP systems.

1.19 Rank the following storage systems from slowest to fastest:
a. Hard-disk drives (HDD)
b. Registers
c. Optical disk
d. Main memory
e. Nonvolatile memory
f. Magnetic tapes
g. Cache
Answer :

Ranking the storage systems from slowest (highest latency) to fastest (lowest latency) :

1. Magnetic Tapes – (Extremely slow)
2. Optical Disk, for example: CD/DVD/Blu-ray – (Slow)
3. Hard-Disk Drives (HDD) – (Moderate speed)
4. Nonvolatile Memory, for example: SSD, Flash Storage – (Fast)
5. Main Memory (RAM) – (Very fast)
6. Cache – (Faster than RAM)
7. Registers – (Fastest)

1.20 Consider an SMP system similar to the one shown in Figure 1.8. illustrate with an example how data residing in memory could, in fact, have a different value in each of the local caches.
Answer :

Cache Inconsistency in SMP Systems :

In an SMP system, each CPU core has its local cache, & they all share the same main memory. This will cause cache inconsistency, the situation where different cores hold random values of the same memory in their caches.

Example how data residing in memory could have different value in each of the local caches (Cache Inconsistency) :

Step 1: Initial State (Shared Memory)

- Assume two CPU cores both read a variable X = 10 from main memory.
- Each core stores X = 10 in its L1 cache.

Step 2: Core 1 Modifies X

- Core 1 update X = 20 in its local cache
- However, core 2 still sees X = 10 in its cache because the update still hasn't reached main memory/core 2's cache.

Step 3: Cache Inconsistency

- Now, if Core 2 reads X, it still sees it as the old value "X = 10", while Core 1 sees it as X = 20.
- This data inconsistency will lead to unexpected behavior in programs.

That's how the Cache Inconsistency will act as a problem in SMP system, but there's a way to solve this issue, by using SMP systems cache coherence protocols, such as:

- MESI (Modified, Exclusive, Shared, Invalid)
- Write-through policy
- Snooping & Bus monitoring

1.21 Discuss, with examples, how the problem of maintaining coherence of cached data manifests itself in the following processing environments:
a. Single-processor systems
b. Multiprocessor systems
c. Distributed systems
Answer :

a. System-Processor Systems

In single-core systems, the issue is less severe because only one processor can modify memory. However, I/O devices and DMA can also modify memory directly, leading to stale cache values.

For example :

- The CPU caches variable X = 10
- A DMA transfers update X = 20 in memory
- The CPU still sees X = 10 because cache is outdated
- Solution: Use write-through caching/cache flushing while DMA writes data.

b. Multiprocessor Systems (SMP)

Each processor has its local cache, causing it to create inconsistent data copies. Cache coherence protocols are required to maintain consistency.

For example :

- Core 1 & Core 2 cache X = 10
- Core 1 updates X = 20 in its cache, but Core 2 still sees X = 10
- The system must invalidate Core 2's cache/update it to ensure consistency.
- Solution: Cache coherence mechanism like MESI.

c. Distributed Systems

Multiple machines have independent memory & caches, leading to data synchronization issues over a network. Maintaining consistency is more complex due to network delays.

For example :

- A banking system with multiple servers handling transactions.
- Server 1 is updating an account balance, but Server 2 still sees it as the old balance.
- Solution: Use distributed caching techniques like Write-invalidate, Write-update, Eventual Consistency

1.22 Describe a mechanism for enforcing memory protection in order to prevent a program from modifying the memory associated with other programs.

Answer :

To prevent this, operating systems use a memory protection mechanism enforced by the hardware. One such mechanism is using a combination of base and limit registers.

Base & Limit register mechanism:

1. Base Register: Stores the starting address of a program's memory space
2. Limit Register: Defines the size of the program's memory space

How does it work?

When a program tries to access memory, the CPU checks if the memory address falls within the allowed range. But if the address is outside this range, a trap occurs, preventing form an unknown access.

1.23 Which network configuration – LAN or WAN – would best suit the following environments?

a. A campus student union

b. Several campus locations across a statewide university system

c. A neighborhood

Answer :

LAN : Covers a small geographical area
WAN : Covers a large geographical area and connects multiple LANs

a. A campus student union.

Best network configuration: **LAN**

Reason? The students' union is a single building or a small campus area, making **LAN** the most suitable choice.

b. Several campus location across a statewide university system.

Best network configuration: **WAN**

Reason? WAN is needed to connect multiple campuses across a large geographic area; it requires long-distance communication. They use technologies like VPNs or MPLS for secure, interconnected networks, making **WAN** the most suitable choice because it can cover a wide/large area.

c. A neighborhood

Best network configuration: **LAN** or **WAN** (Depends on the size of the area)

For small neighborhoods (Local coverage area), **LAN** may be the most suitable choice. But if it's for multiple neighborhoods (City wide area), **WAN** for sure will fulfill your network configuration needs.

1.24 Describe some of the challenges of designing operating systems for mobile devices compared with designing operating systems for traditional PCs.

Answer :

What makes designing operating systems for mobile devices challenging is that because they have less storage, the OS must manage memory carefully, and we also must manage power consumption carefully.

Mobile devices are not as powerful as PC, so there will be a need to optimize the OS much more, compared to the OS from most PC

1.25 What are some advantages of peer-to-peer systems over client-server systems?

Answer :

Advantages of peer-to-peer systems over client-server systems:

1. Scalability (As new peers add resources)
2. Cost Efficiency (No need for a dedicated server)
3. Fault Tolerance & Reliability (As failure of one peer, will has no effect on the system)
4. Load Balancing (Distributed among peers)
5. Privacy & Anonymity (More private & no central authority)
6. Direct Communication
7. Decentralization

1.26 Describe some distributed applications that would be appropriate for a peer-to-peer system.

Answer :

1. File Sharing/P2P Network (For example: BitTorent, eMule)

The reason? : Users are able to upload & download files directly from other peers, with no central server bottleneck.

2. Cryptocurrency & Blockchain (For example: Bitcoin, Ethereum)

The reason? : Transactions are validated by multiple nodes (decentralized), and no central authority controls the network. Also, ensures fault tolerance and security.

3. Decentralized messaging & communication (For example: Session, Briar, Tox)

The reason? : Messages are relayed directly between peers, avoiding central servers, and are more private and censorship-resistant.

4. Online Gaming

The reason? : Players directly connect to each other instead of the central server. Reduced server cost & much faster local connection.

5. Edge Computing & IoT Networks (For example: Smart home devices)

The reason? : Devices communicate without needing a central cloud server. Increases efficiency and reduces latency.

6. Collaborative Distributed Storage (For example: IPFS, Storj)

The reason? : Data is split and stored across multiple nodes, improving redundancy.

1.27 Identify several advantages and several disadvantages of open-source operating systems. Identify the types of people who would find each aspect to be an advantage or a disadvantage.

Answer :

An open-source OS, such as Linux, allows its user to view, modify, & distribute its source code. While it may offer many benefits, it also has some disadvantages depending on the user's needs. Here are some advantages & disadvantages of using open-source operating systems:

Advantages	Disadvantages
It's Free! No licensing fees: There's no need to pay for licenses, unlike Windows or macOS.	Steeper learning curve: Requires command-line knowledge for advanced tasks.
Customization & Flexibility: Users can modify the OS to fit their needs & liking	Software & Hardware compatibility issues: For a gamer, it may be a big downside. Some proprietary software may have limited or no support for the current OS. For some, certain hardware may lack proper drivers.
Security & Transparency: The code is publicly reviewed, reducing hidden vulnerabilities and spyware.	Limited official support: No official customer support, but some paid open-source OS most likely has its own customer support.
Community support & Rapid updates: Strong community support provides free troubleshooting and bug fixes	Fragmentation: Many Linux distributions differ in package management and system architecture, making software development harder.
Lightweight & Efficient: It can run much smoother on low-end devices, unlike the latest version of Windows.	Complex to use: Not a good option for casual user. A lot of new things need to learn, unlike using other OS like Windows or macOS.

Overall, these open-source OS are not for everyone, but they may suit you if you are a:

Developers, Tech-savvy users, security experts, & organizations needing flexibility

And not ideal for:

Casual users, gamers, or businesses company that requires official support & full software compatibility.
