Tugas Sistem Oprasi

Sebuah gambar berisi Grafis, papan klip, logo, simbol

Konten yang dihasilkan AI mungkin salah.

Dosen : Dr Ferry Astika Saputra ST, M.Sc

Nama : Naf’an Nashohu Zayyan

NRP : 3124521008

Kelas : D3 IT A

Politeknik Elektronika Negeri

Surabaya (PENS) La

Soal

1.1

What are the three main purposes of an operating system?

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?

1.3

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

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 answers.

1.5

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

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. Access I/O device.

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 job or the operating system itself. Describe two difficulties that you think could arise with such a scheme.

1.8

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

1.9

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

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?

**1.11**

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

1.12

How do clustered systems differ from multiprocessor systems? What is required for twomachines belonging to a cluster to cooperate to provide a highly available service?

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.

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?

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.

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.

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 and that it is not possible.

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?

1.19

Rank the following storage systems from slowest to fastest:

a. Hard-disk drives b. Registers

c. Optical disk d. Main memory

e. Nonvolatile memory f. Magnetic tapes

g. Cache

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.

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

1.22

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

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

1.24

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

1.25

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

1.26

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

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.

Jawaban :

### 1.1

1. Resource Management – Manages hardware resources like CPU, memory, and I/O devices.
2. Process Management – Provides execution environment and manages running applications.
3. User Interface – Provides an interface between users and the computer (CLI or GUI).

1.2

It is appropriate when user convenience and ease of use are prioritized over raw efficiency, such as in personal computers, graphical user interfaces, and modern cloud computing. Such a system is not truly wasteful because it improves usability, multitasking, and system reliability, making the overall computing experience better for users.

1.3

The main challenge is ensuring deterministic and predictable response times. A real-time OS must meet strict deadlines and guarantee that high-priority tasks execute within the required time constraints. Any delay or failure to meet deadlines can lead to system failure in critical applications (e.g., medical devices, industrial automation, and avionics).

1.4

* Yes: Including them provides an integrated user experience, ensuring compatibility and security. Examples are macOS and Windows, which include default browsers and email clients.
* No: An OS should focus on core functionalities, leaving applications as optional installations. This approach keeps the system lightweight, flexible, and reduces security vulnerabilities.

1.5

The distinction ensures that critical system operations (like memory and process management) can only be performed in kernel mode, preventing user applications from executing privileged operations that could compromise system integrity. This protects against accidental or malicious modifications.

1.6

* Privileged Instructions:
  + (a) Set value of timer
  + (c) Clear memory
  + (e) Turn off interrupts
  + (f) Modify entries in the device-status table
  + (g) Switch from user to kernel mode
* Non-Privileged Instructions:
  + (b) Read the clock
  + (d) Issue a trap instruction (used to switch to kernel mode)
  + (h) Access I/O device (some I/O operations can be non-privileged)

1.7

1. Lack of Updates or Modifications – The OS cannot be updated, patched, or fixed without complete system replacement.
2. Inflexibility – Dynamic system processes, like device drivers or kernel extensions, cannot be loaded or modified, leading to compatibility issues with new hardware.

1.8

1. Virtualization Mode – A separate mode for running virtual machines securely without interfering with the host OS.
2. Secure Execution Mode – A trusted mode that allows executing sensitive operations, such as encryption, without interference from normal applications.

1.9

A timer can be programmed to generate periodic interrupts at a known frequency. The OS keeps track of these interrupts and calculates the current time by counting them since a reference point (such as system startup).

1.10

* Benefits:
  1. Faster Access to Data – Reduces latency by storing frequently used data closer to the CPU.
  2. Reduced Load on Main Memory – Decreases the need to access slower RAM or disk storage.
* Drawbacks:
  1. Consistency Issues – Cached data may become outdated, leading to stale reads.
  2. Higher Cost – Cache memory (SRAM) is more expensive than regular RAM.
* Why not make cache as large as RAM?  
  If cache were as large as RAM, it would become slow and expensive, reducing its efficiency. Cache works best when it is small and stores only frequently accessed data.

1.11

1. **Client-Server Model**
   * A centralized system where **clients** request services, and a **server** provides them.
   * The server manages resources and processes client requests.
   * Examples: Web services, email servers, database servers.
2. **Peer-to-Peer (P2P) Model**
   * A decentralized system where all nodes (peers) act as both **clients and servers**.
   * Each peer can request and provide services, distributing workload among nodes.
   * Examples: BitTorrent, blockchain networks, Skype (older versions).

1.12

Clustered systems consist of multiple independent computers working together, while multiprocessor systems have multiple CPUs within a single system. To cooperate, two machines in a cluster must use shared storage and a cluster management system to ensure high availability through failover mechanisms.

1.13

Two ways a cluster can manage disk access:

1. **Shared-disk approach**: Both nodes access the same storage system, ensuring data consistency but requiring a locking mechanism to prevent conflicts.
2. **Replication-based approach**: Each node has a local copy of the data, improving read performance but requiring synchronization to maintain consistency.

1.14

Interrupts allow the CPU to respond to events asynchronously. A trap is a specific type of interrupt triggered by the program itself, often for system calls or error handling. Traps can be generated intentionally for debugging or requesting OS services.

1.15

The **HZ** variable defines clock ticks per second, and **jiffies** counts the total ticks since boot. The system uptime can be calculated as jiffies / HZ.

1.16

a. The CPU initiates DMA transfer by configuring the DMA controller with memory addresses and data size.  
b. The DMA controller sends an interrupt when the transfer is complete.  
c. DMA allows concurrent CPU execution but may cause cache coherence issues and memory bus contention.

1.17

A secure OS is possible without privileged mode using software-based enforcement, but it is less efficient and vulnerable to exploits compared to hardware-enforced protection.

1.18

Caching is hierarchical to balance speed and cost. Local caches reduce latency, while shared caches optimize inter-core communication.

1.19

1. **Magnetic tapes**
2. **Optical disk**
3. **Hard-disk drives**
4. **Nonvolatile memory**
5. **Main memory**
6. **Cache**
7. **Registers**

**1.20**

In an SMP system, different local caches may hold different values for the same memory location due to the lack of proper cache coherence. For example, if one processor updates a variable stored in memory, another processor might still have the old value in its local cache, leading to inconsistencies.

**1.21**

a. In single-processor systems, cached data coherence is not a major issue because only one processor accesses memory.  
b. In multiprocessor systems, multiple processors with local caches may lead to cache coherence problems, requiring cache coherence protocols.  
c. In distributed systems, maintaining consistency across different machines over a network is a challenge, and consistency protocols like eventual consistency or strong consistency mechanisms are needed.

**1.22**

Memory protection can be enforced using mechanisms such as memory segmentation and paging, which restrict a program’s access to only its allocated memory space. The operating system sets up privileged access levels and uses hardware-based protection such as MMU (Memory Management Unit) to enforce these rules.

**1.23**

a. A campus student union would be best suited for a LAN, as it covers a small geographical area.  
b. Several campus locations across a statewide university system would require a WAN to connect distant locations.  
c. A neighborhood might use either a LAN (for small areas) or a WAN (if covering a broader area with multiple networks).

**1.24**

Challenges in designing OS for mobile devices include:

* Limited battery life requiring power-efficient OS scheduling.
* Smaller and varying hardware specifications requiring OS optimization.
* Touchscreen interfaces needing specialized input handling.
* Security concerns due to always-on connectivity.

**1.25**

Advantages of peer-to-peer (P2P) systems:

* No central point of failure, making them more fault-tolerant.
* More scalable since resources are distributed among peers.
* Reduced dependency on centralized infrastructure, lowering costs.

**1.26**

Examples of distributed applications suitable for P2P:

* File-sharing applications (e.g., BitTorrent).
* Decentralized cryptocurrencies (e.g., Bitcoin).
* P2P messaging and VoIP services (e.g., Skype in early versions).
* Distributed computing projects (e.g., SETI@home, Folding@home).

**1.27**

Advantages and disadvantages of open-source operating systems:

* **Advantages:** Free to use, high security due to public code review, customizable, supported by a large community. Beneficial for developers, researchers, and companies looking for flexible solutions.
* **Disadvantages:** May lack vendor support, compatibility issues with some proprietary software, steep learning curve. Can be a disadvantage for non-technical users or businesses needing guaranteed support.