Teknik Informatika
Institut Teknologi Sumatera

Sistem Operasi (Judul mata Kuliah)

Judul Pertemuan

Martin Clinton Manullang

Program Studi Teknik Informatika

February 16, 2025

Contents

This is an example of **bold text**, *italic text*, and `monospaced text`.

Block Title

This is a normal block with some dummy text.

Alert Block

This is an **alert** block showing important information.
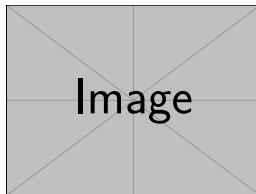
Two Column Layout Example

Left Column: Lists and Text

- First item with *italics*.
- Second item with **bold**.
- Third item with `typewriter font`.

Right Column: Table and Graphic

ID	Name	Value
1	Alpha	10
2	Beta	20
3	Gamma	30

Elaborated Table Example

Table: Dummy Data Table

Category	Count	Percentage
Category A	15	37.50%
Category B	20	50.00%
Category C	5	12.50%

Contents

Python Code Examples

Sample Implementation

```python
    # This is a sample Python function
def calculate_factorial(n):
    """Calculate the factorial of a number"""
    if n == 0 or n == 1:
        return 1
    else:
        return n * calculate_factorial(n - 1)

# Example usage
number = 5
result = calculate_factorial(number)
print(f"Factorial of {number} is {result}")
```

- Here's a sorting algorithm implementation:

```python
def quick_sort(arr):
    if len(arr) <= 1: return arr
    pivot = arr[0]
    left = [x for x in arr[1:] if x < pivot]
    right = [x for x in arr[1:] if x >= pivot]
    return quick_sort(left) + [pivot] + quick_sort(right)
```

When working with Python, you can create a list using square brackets like `my_list = [1, 2, 3]` or define a dictionary with `my_dict = {"key": "value"}`.

Function definitions are straightforward: `def greet(name): return f"Hello {name}"` can be used to create simple greeting functions.

For loops are common in Python: `for i in range(5): print(i)` will print numbers from 0 to 4.

CPU Central Processing Unit - The primary processor that executes instructions

RAM Random Access Memory - Temporary storage for running programs

GPU Graphics Processing Unit - Specialized processor for rendering graphics

SSD Solid State Drive - Fast storage device with no moving parts

Contents

Introduction

Complex systems can be understood through careful analysis and decomposition into smaller, manageable components.

- **Primary Concept:** The fundamental principle builds upon the interaction between multiple interconnected elements.

Introduction

Complex systems can be understood through careful analysis and decomposition into smaller, manageable components.

- **Primary Concept:** The fundamental principle builds upon the interaction between multiple interconnected elements.
- **Detailed Analysis:** Each component serves a specific purpose within the larger framework, contributing to the overall functionality of the system.

Long Descriptive Text Example

Understanding Complex Concepts

Introduction

Complex systems can be understood through careful analysis and decomposition into smaller, manageable components.

- **Primary Concept:** The fundamental principle builds upon the interaction between multiple interconnected elements.
- **Detailed Analysis:** Each component serves a specific purpose within the larger framework, contributing to the overall functionality of the system.
- **Implementation:** The practical application involves careful consideration of various factors including performance, scalability, and maintainability.

Key Considerations

When implementing complex systems, it's crucial to maintain a balance between theoretical principles and practical constraints.

Future Directions

Ongoing research continues to explore new methodologies and approaches for optimizing system performance and reliability.