### Operating System Course Outline

### Course Overview:

The Operating System course provides a comprehensive exploration of the foundational principles and theoretical aspects governing modern computer operating systems. The course is designed for students pursuing a deeper understanding of the critical role operating systems play in managing computer resources and facilitating efficient computation.

### Key Topics:

1. **Introduction to Operating Systems:**
   * Overview of OS functions and historical context.
   * Types of operating systems and their characteristics.
2. **System Structures:**
   * Kernel architecture and different kernel types.
   * User mode vs. Kernel mode, system calls, and APIs.
3. **Process Management:**
   * Understanding processes and threads.
   * Process synchronization and inter-process communication.
4. **Memory Management:**
   * Memory hierarchy and virtual memory concepts.
   * Page replacement algorithms and memory allocation strategies.
5. **File Systems:**
   * File concepts, attributes, and organization.
   * File operations, access control, and file system implementation.
6. **Security and Protection:**
   * Identifying security threats and vulnerabilities.
   * Authentication, authorization, encryption, and access control mechanisms.
7. **Review and Advanced Topics:**
   * Recap of key concepts.
   * Exploring emerging trends in operating systems.
   * Case studies of modern OS architectures.

### Assignments:

1. **Research Paper Analysis:**
   * Students analyze a recent research paper related to an advanced operating system topic. Emphasis on critical evaluation and presentation of findings.
2. **Case Study on OS Security:**
   * Investigate a real-world case study involving a security breach or successful protection measures in an operating system context.
3. **Memory Management Algorithm Implementation:**
   * Students implement a selected memory management algorithm in Python to gain practical insights.

### Learning Outcomes:

By the end of the course, students will:

1. **Understand Operating System Fundamentals:**
   * Grasp the core functions and goals of operating systems, and differentiate between various types.
2. **Comprehend System Structures:**
   * Analyze different kernel architectures, understand user mode vs. kernel mode, and articulate the role of system calls.
3. **Master Process and Memory Management:**
   * Demonstrate a deep understanding of processes, threads, synchronization, virtual memory, and memory management algorithms.
4. **Proficient File System Knowledge:**
   * Acquire knowledge of file systems, including file concepts, organization, and implementation.
5. **Apply Security Measures:**
   * Identify and address security threats, employing authentication, authorization, encryption, and access control mechanisms.
6. **Explore Advanced Topics:**
   * Investigate emerging trends and case studies, fostering an ability to critically evaluate modern operating system architectures.

**Meeting 1-2: Introduction to Operating Systems**

* Overview of Operating Systems
* Functions and goals of an Operating System
* Historical perspective
* Types of Operating Systems (Batch, Time-sharing, Real-time, etc.)

**Meeting 3-4: System Structures**

* Kernel architecture
* Microkernel vs. Monolithic kernel
* User mode vs. Kernel mode
* System calls and APIs

**Meeting 5-6: Process Management**

* Processes and threads
* Process states and life cycle
* Process synchronization
* Inter-process communication

**Meeting 7: Midterm Review**

**Meeting 8: Midterm Examination**

**Meeting 9-10: Memory Management**

* Memory hierarchy
* Virtual memory concepts
* Page replacement algorithms
* Memory allocation strategies

**Meeting 11-12: File Systems**

* File concepts and attributes
* File system organization
* File operations and access control
* File system implementation

**Meeting 13: Security and Protection**

* Security threats and vulnerabilities
* Authentication and authorization
* Encryption techniques
* Access control mechanisms

**Meeting 14-15: Review and Advanced Topics**

* Recap of key concepts
* Emerging trends in Operating Systems
* Case studies of modern OS architectures

**Meeting 16: Final Examination**

### Notes:

* Assign relevant readings and research papers for each topic.
* Encourage discussions on real-world examples and case studies.
* Integrate practical examples or demonstrations even though there are no lab sessions.
* Provide additional resources for further exploration.

For the examination meetings, ensure the questions cover a broad range of topics discussed during the course, including theoretical and practical aspects where applicable. This outline aims to balance theoretical understanding with real-world applications, fostering a comprehensive grasp of Operating Systems.