

- Sie können fünf wesentliche Komponenten der Ein-/Ausgabe(E/A)-Architektur eines Betriebssystemsystems benennen
- Sie können je einen Vor- und Nachteil einer standardisierten E/A-Schnittstelle aufzeigen
- Sie können erklären welche Aufgaben die Treiber in einem Betriebssystem erfüllen und wie sie sich in das Betriebssystem einfügen
- Sie können drei wesentliche Synchronisationsprobleme beim Einsatz von Treibern und ihre mögliche Behebung erklären
- Sie können die Begriffe Treiberschnittstelle, Treiberhierarchie, Treiberinstanz und Prozess-/Unterbrechungskontext erklären

Lernziele

- Sie können zwei wesentliche Unterschiede zwischen Kernmodus- und Benutzermodustreiber erklären
- Sie unterscheiden zeichenorientierte und blockorientierte Kernmodustreiber
- Sie verstehen die Ein-/Ausgabeschnittstelle und können sie für eigene Programme nutzen
- Sie können einen einfachen interrupt-gesteuerten zeichenorientierten Kernmodustreiber entwickeln
- Sie können die zwei wichtigsten Schritte bei der Installation eines Kenmodustreibers aufzählen
- Sie können fünf typische Treiberaufgaben beschreiben

- Sie können sieben Komponenten des Windows Ein-Ausgabesystems unterscheiden
- Sie können die Unterschiede zwischen dem WDM- und dem WDF-Treibermodell beschreiben
- Sie kennen fünf WDM-Objekttypen und ihre Anwendung
- Sie können das WDF-Objektmodell erklären
- Sie können einen einfachen WDF-Kernmodustreiber entwickeln

- Sie können drei Entwurfsprobleme beschreiben
- Sie können *Architekturmodelle* und *Architekturansätze* miteinander vergleichen
- Sie können die Architekturen bestehender (Windows, Unix, Linux) und zukünftiger Betriebssysteme (MSR Singularity) anhand von Architekturmodellen und Architekturansätzen beurteilen
- Sie können die Idee der Systemobjekte auf konkrete Betriebssysteme anwenden

- Sie können vier verschiedenen Rechnerstrukturen geeignete Betriebssystemarten zuordnen und ihre Wahl begründen
- Sie unterscheiden drei verschiedene Implementierungen von Cluster-Systemen
- Sie können vier Programmiermodelle für die parallele Programmausführung auf mehreren Prozessoren kurz beschreiben
- Sie können erklären wie das Experimental-Betriebssystem „*fos (factored operating system)*“ die Anforderungen an hoch skalierbare verteilte Betriebssysteme löst

Lernziele

- Sie können die Relokation und Adressbindung von Programmbibliotheken erklären
- Sie kennen drei Unterscheidungsmerkmale für die Klassifikation von Bibliotheken
- Sie können drei Bibliotheksarten unterscheiden
- Sie können die Effizienz von Programmbibliotheken anhand von fünf bestimmten Merkmalen beurteilen

- Sie können einfache Programmbibliotheken unter Unix als *static library*, *shared library* oder *dynamic loadable library* selbst erstellen
- Sie sind in der Lage die Rolle der GOT und PLT bei der Verwendung gemeinsamer Bibliotheken zu erklären

- Sie können einfache Programmbibliotheken unter Windows selbst entwickeln
- Sie sind in der Lage das Laden von Windows-Programmen und Bibliotheken zu analysieren
- Sie können die Unterschiede zwischen implizitem, explizitem und verzögertem Laden von Programmbibliotheken erklären

- Sie können die Funktionsweise von fünf verschiedenen Scheduling-Implementierungen beschreiben
- Sie kennen zwei unterschiedliche Verfahren zur dynamischen Prioritätsmodifikation
- Sie können das MS Windows Thread-Zustandsmodell und das Unix System V Rel. 3 Prozess-Zustandsmodell erklären
- Sie kennen drei Unterschiede zwischen dem Multiprozessor-Scheduling des Windows und des Linux