

Betriebssysteme

Verfasser Kacper Bohaczyk

Datum 30.05.2022

Arbeitsschritt

x86-kompatible Prozessoren --> vier Privilegienstufen (Ringe) --> verbessern die Stabilität und Sicherheit. Jeder Prozess --> ausgeführt in einem Ring --> kann sich nicht selbständig aus diesem befreien --> Registers Current Privilege Level(CPL) --> speichert die aktuellen Ringnummer. Ring-0 --> Kernmodus --> läuft der Betriebssystemkern --> Prozesse im Kernelmodus haben vollen Zugriff auf die Hardware --> kann physischen Speicher direkt im Real Mode adressieren. Ring-3 --> Benutzermodus --> üblichen Prozesse --> arbeiten ausschließlich mit virtuellem Speicher. Moderne Betriebssysteme verwenden ausschließlich zwei Privilegienstufen --> Grund: populäre Hardware-Architekturen wie Alpha-Prozessor verwenden nur zwei Stufen.

Systemaufrufe und Bibliotheken

Systemaufruf --> direkt oder über Umweg einer Bibliothek aufgerufen werden --> Funktionsaufruf im Betriebssystemkern --> Sprung vom Benutzermodus in den Kernelmodus --> Moduswechsel --> ein Prozess gibt die Kontrolle über den Hauptprozessor an den Betriebssystemkern ab --> unterbrochen bis die Anfrage bearbeitet ist. Nach dem Systemaufruf gibt der Kern den Prozess wieder im Benutzermodus ab --> der Prozess führt daraufhin seine Abarbeitung fort an der Stelle an der der Moduswechsel zuvor angefordert wurde --> Leistung eines Systemaufrufs --> Kern (außerhalb des Adressraums) des aufgerufenen Prozesses erbracht.

Beispiel für Systemaufruf --> ioctl --> Prozesse unter Linux realisieren gerätespezifische Anweisungen --> ermöglicht Kommunikation und Steuerung von zeichenorientierten und blockorientierten Geräten --> Einsatzszenarien: Formatieren einer Diskettenspur, Initialisieren eines Modems oder einer Soundkarte, das Auswerfen einer CD aus dem Laufwerk, das Auslesen von Statis- und Verbindungsinformationen der WLAN-Schnittstelle oder der Zugriff auf Sensoren und Aktoren über einen Bus. Kerne moderner Betriebssysteme --> mehrere hundert Systemaufrufe --> Prozess-, Datei- und Verzeichnissverwaltung enthält Tab. Eine Liste mit den Namen der Systemaufrufe --> Quellen des Linux.Kerns. Aus eigenen Programmen Systemaufrufe aufzurufen --> in Praxis nicht empfehlenswert --> schlecht portabel --> nicht alle Systemaufrufe bei verschiedenen Betriebssystemfamilien identisch --> nicht Garantiert das eine neue Version des Betriebssystemkerns nicht Veränderungen an einzelnen Systemaufrufen enthält --> sollte bei der Entwicklung eigener Software lieber auf die Funktionen einer Bibliothek zurückgreifen --> Wrapper-Funktionen --> Vermittlung der Kommunikation zwischen den Benutzerprozessen mit dem Betriebskern und anweisen der Moduswechsel zwischen Benutzer. und Kernelmodus --> UNIX C Standart Library.

Ablauf eines Systemaufrufs

Jeder Abschnitt enthält eine Beschreibung der einzelnen Arbeitsschritte. read --> liest eine bestimmte Menge von Bytes aus der Datei(fd) und schreibt sie in einem Puffer(buffer). Schritte 1-3 --> der Parameter wird auf den Stack gelegt von Benutzerprozess. Schritt 4 --> ruft die Bibliotheksfunktion für read auf. Schritt 5 --> speichert die EAX im RAX. Im Schritt 6 --> wird eine Exception ausgelöst --> wechselt in Kernelmodus. Betriebskern --> System Call Table --> jedem Kernel wird eine eindeutige Nummer und eine Kernel-interne Funktion zugeordnet. Exception-Handler --> Funktion im Kern --> EAX ausliest. Schritt 7 --> registern in EBX,

ECX und EDX. Im Schritt 8 und 9 gibt der Exception-Handler die Kontrolle an die Bibliothek zurück. Schritt 10 -> kehrt zum Benutzerprozess so zurück wie es eine normale Funktion getan hätte. Schritt 11 --> Abschluss --> Stack wird aufgeräumt. Benutzerprozess kann dann weiterarbeiten.

Quellen

[1] Buch 2022 "Betriebssysteme kompakt_Springer Buch"