GK435 Betriebssysteme 7.Kapitel

Autor: Dejan Rajic

Datum: 30.05.2022

Benutzermodus und Kernelmodus

Es gibt vier Privilegienstufen die auch Ringe heißen. Die Nummerierung geht von 0 -3. Der Ring 0 ist der sogenannte Kernelmodus. Prozesse die in ihm laufen haben uneingeschränkten zugriff auf die Hardware, es können auch physische Speicher im Real Mode adressiert werden. Der dritte Ring ist der Benutzermodus, über ihn laufen alle übrigen Prozesse, er arbeitet nur mit virtuellem Speicher. In den heutigen Betriebssystemen werden nur zwei Privilegienstufen verwendet.

Systemaufrufe und Bibliotheken

Ein Systemaufruf ist ein Funktionsaufruf im Betriebssystemkern, der einen Sprung vom Benutzermodus in den Kernelmodus auslöst (Moduswechsel). Beim Moduswechsel gibt ein Prozess die Kontrolle über den Hauptprozessor an den Betriebssystemkern ab und ist solange unterbrochen, bis die Anfrage bearbeitet ist. Wenn der Systemaufruf abgeschlossen ist, gibt der kern den Prozessor an den Prozess im Benutzermodus ab. Ein Beispiel wäre "ioct1" welches beim Auswerfen einer CD, Auslesen von Status und Verbindungsinformationen oder Zugriff auf Sensoren über einen Bus. Eine neue Version von Betriebssystemen kann auch Änderungen an einzelnen Systemaufrufen enthalten, deshalb sollte man beim Programmieren Funktionen aus Bibliotheken verwenden. Die Bibliothek ist zuständig für die Vermittlung der Kommunikation zwischen den Benutzerprozessen mit dem Betriebssystemkern und für das Anweisen der Moduswechsel.

Ablauf eines Systemaufrufs

Der Systemaufruf read besteht aus 11 Schritten

In den Schritten 1-3 legt der Benutzerprozess die Parameter auf den Stack. Im 4.ten Schritt ruft er Die Bibliotheksfunktionen für read auf.

5ter Schritt => speichert die Nummer des Systemaufrufs im Akkumulator Register (EAX)

6ter Schritt => die Exception 0x80 wird ausgelöst, um die Modi zu wechseln

7ter Schritt => die Exception Handler-Funktion ruft die entsprechenden Funktionen im Kern aus der System Call Table mit den Register EBX, ECX und EDX gespeicherten Argumenten auf.

8ter Schritt => Der Systemaufruf wird gestartet

9ter Schritt => Der Exception-Handler gibt die Kontrolle an die Bibliothek zurück, die den Softwareinterrupt ausgelöst hat.

10ter Schritt =>Funktion kehrt zum Benutzerprozess zurück wie eine normale Funktion.

11ter Schritt => Der Stack wird aufgeräumt.