



Betriebssysteme und nebenläufige Programmierung

SoSe 2026

Roland Wismüller
Betriebssysteme / verteilte Systeme
roland.wismueller@uni-siegen.de
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 20. März 2026



Betriebssysteme und nebenläufige Programmierung

SoSe 2026

10 Schutz



Inhalt:

- ➔ Einführung
- ➔ Schutzmatrix
- ➔ Zugriffskontrolllisten und *Capabilities*

- ➔ Tanenbaum 9.6
- ➔ Stallings 15.2.3
- ➔ Nehmer/Sturm 11

10.1 Einführung



Sicherheitsdienste: AAA („triple A“)

- ➔ **Authentifizierung**
 - ➔ Feststellung der Identität eines Benutzers
 - ➔ typisch: Paßwort-Abfrage bei der Anmeldung
- ➔ **Autorisierung**
 - ➔ Vergabe von Zugriffsrechten an Benutzer
 - ➔ Wer darf was im System tun?
- ➔ **Accounting**
 - ➔ Protokollierung von Aktivitäten, Abrechnung

Speicherung der Schutzmatrix

- ➔ Schutzmatrix sehr groß und sehr dünn besetzt
- ➔ Daher: zeilen- oder spaltenweise Speicherung in Listen

	Datei 1	Datei 2	Datei 3	Datei 4	Datei 5	Drucker	Plotter
Benutzer 1	Lesen	Lesen Schreiben					
Benutzer 2		Lesen	Lesen	Lesen Schreiben Ausführen		Schreiben	
Benutzer 3		Lesen		Lesen Ausführen	Lesen Schreiben Ausführen	Schreiben	Schreiben

Capability

Zugriffskontrollliste
Access Control List (ACL)

10.3 Zugriffskontrolllisten und *Capabilities* ...



Zugriffskontrollliste (*Access Control List, ACL*)

- ➔ Spalte der Schutzmatrix
- ➔ Gibt für ein Objekt an, welche Subjekte welche Rechte an dem Objekt haben
- ➔ Wird zusammen mit dem betroffenen Objekt gespeichert
 - ➔ z.B. bei Datei im zugehörigen *I-Node*
- ➔ Listenelemente: Paare (Subjekt, Rechte)
 - ➔ Subjekt: Benutzer und/oder Benutzergruppe
 - ➔ für Subjekt oft auch Platzhalter (*Wildcard*) erlaubt
 - ➔ erster passender Eintrag wird verwendet



Capability

- ➔ Zeile der Schutzmatrix
- ➔ Wird vom BS-Kern an Subjekte (Prozesse) übergeben, berechtigt zur Ausführung von Operationen auf Objekten
- ➔ *Capability* muß vor Manipulation geschützt werden!
 - Speicherung im BS-Kern, Prozeß erhält nur Verweis (*Handle*)
 - kryptographischer Schutz (analog zu digitaler Signatur)
 - geeignet für verteilte Systeme: *Capability* kann als Nachricht weitergegeben werden
- ➔ Problem: (selektiver) Widerruf von Rechten schwierig



Beispiel: Zugriffsschutz in UNIX und Windows (2000/NT)

- ➔ Mischform aus ACLs und *Capabilities*
 - Rechte an Objekten werden über ACL spezifiziert
 - Prüfung der ACL aber nur beim Öffnen
 - Datei-/Geräte-*Handle* hat die Funktion einer *Capability*
 - bei den eigentlichen Zugriffen ist keine Prüfung der ACL mehr notwendig
- ➔ UNIX: ACL unterscheidet bei Subjekten nur zwischen Eigentümer, Mitgliedern derselben Gruppe und allen anderen
 - Speicherung in 9 Bits im *I-Node*: rwx r-x ---
user group others
- ➔ Windows: ACL mit Einträgen für beliebige Benutzer / Gruppen

- ➔ Sicherheitsdienste: Authentifizierung, Autorisierung, Accounting
- ➔ Schutzmatrix:
 - ➔ wer darf welche Operationen auf welchen Objekten ausführen?
- ➔ Zugriffskontrollliste (*Access Control List*, ACL)
 - ➔ Spalte der Schutzmatrix, beim Objekt gespeichert
- ➔ *Capability*
 - ➔ Zeile der Schutzmatrix, an Subjekt übergeben