Verteilte Systeme

3. Prozesse

Sommersemester 2011

Institut für Betriebssysteme und
Rechnerverbund

TU Braunschweig

Dr. Christian Werner
– Bundesamt für Strahlenschutz –

Überblick

- Prozesse
- Threads
- Client-Prozesse
- Server-Prozesse

Prozesse

- „Programm im Zustand der Ausführung“
- Verwaltung liegt beim Betriebssystem
- Woraus besteht ein Prozess?
  - Ausführbarer Maschinencode
  - Exklusiv nutzbarer Speicherbereich
  - Ressourcendescriptoren zum Zugriff auf Dateien und Sockets
  - Sicherheitsattribute („process owner“ in Verbindung mit der vom BS verwalteten Rechtestruktur)
  - Zustand (Dieser wird normalerweise in den Registern des Prozessors gehalten, wenn der Prozess läuft. Anderenfalls im RAM)
- Vergleichsweise hoher Aufwand, um mehrere Prozesse voneinander unabhängig zu halten
- Lösung: Threads

Threads

- Unterteilung (d. h. Parallelität) innerhalb eines Prozesses
- „light-weight process“
- Die Threads innerhalb eines Prozesses teilen sich alle Ressourcen!

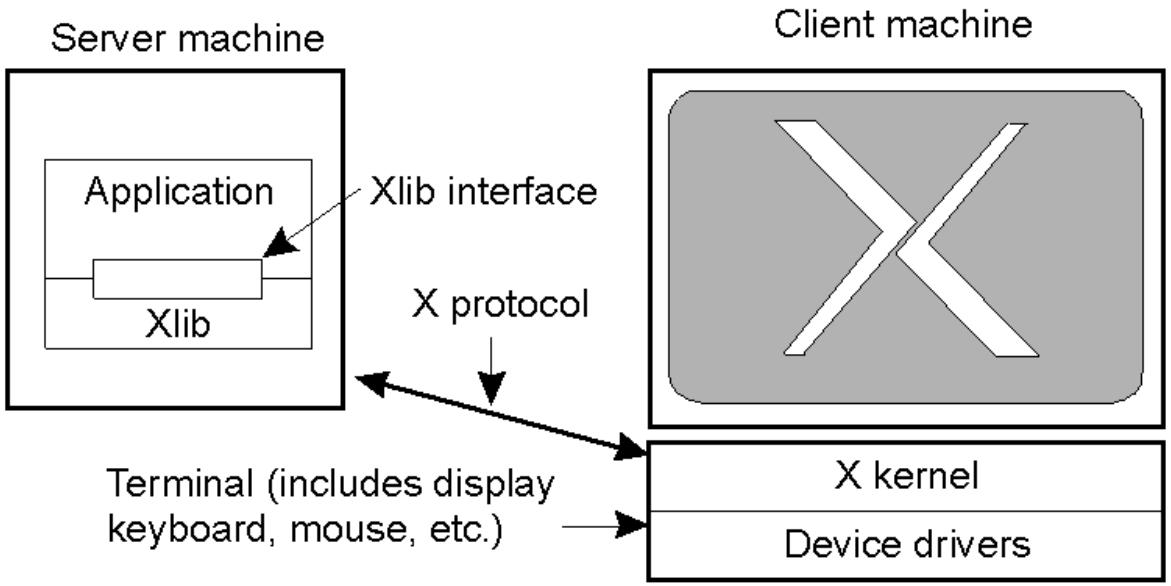
Vorteil:

- In vielen Fällen Leistungsgewinn schon in nicht verteilten Systemen
- Einfache Verwendung von blockierenden Systemaufrufen in verteilten Systemen (I/O-Operationen)

- Nachteil:
  - Threads sind nicht gegeneinander geschützt
  - Ein fehlerhafter Thread führt zum Abbruch des gesamten Prozesses!

Client-Prozesse

- Schnittstelle zu menschlichen Benutzer
  - X Window-System
- Schnittstelle zu entferntem Server (Verteilungstransparenz)



Multithreaded-Clients

- In Weitverkehrsnetzen kann es zu langen Latenzzeiten zwischen Prozessen kommen

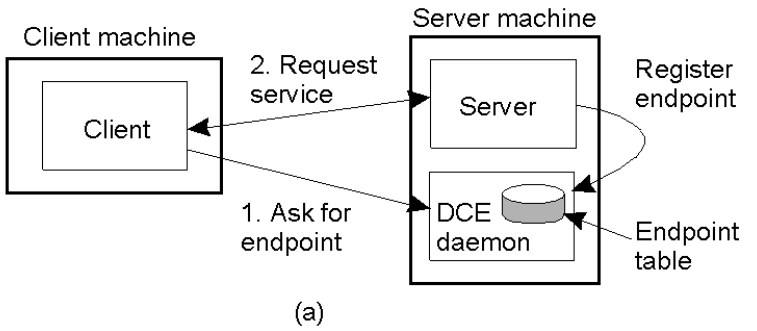
- Beispiel: Webbrowser
  - Verbirgt Latenzzeiten durch Anzeigen von Daten, wenn die Datenübertragung noch nicht vollständig abgeschlossen ist
  - Einfache Entwicklung mit Hilfe von Multithreading

Server (I)

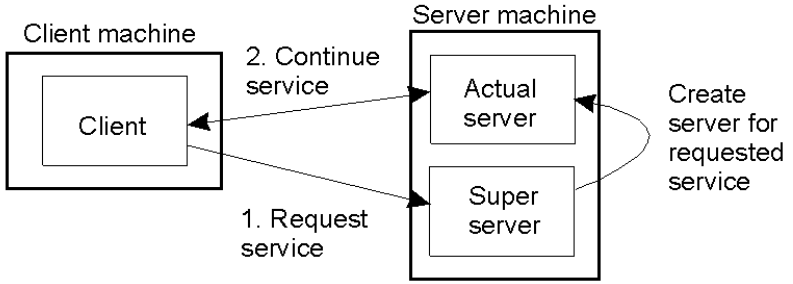
- Ein Server ist ein Prozess, der einen Dienst für mehrere Clients implementiert
- Ein Server wartet auf Anforderung vom Client und stellt sicher, dass diese verarbeitet wird
- **Iterativer Server**: verarbeitet Anforderung selbst
- **Nebenläufiger Server**: gibt Anforderung an einen anderen Prozess weiter und wartet auf nächste eingehende Anforderung

Server (II)

- Clients senden Anforderungen immer an einen Endpunkt (**Port**)
- Jedem Endpunkt ist ein spezifischer Dienst zugeordnet (HTTP-Server: TCP-Port 80)
- Server kann Endpunkte dynamisch zugewiesen bekommen (Bsp.: DCE Dämon)
- Bei vielen passiven Servern wird ein **Super-Server** eingesetzt



(a)



(b)

Server (III)

- **Statusloser Server**:
  - Speichert keine Informationen über den Status seiner Clients
  - Beispiel: Web-Server
- **Statusbehafteter Server**:
  - Hält einen Systemzustand.
  - Beispiel: FTP-Server

Zusammenfassung

- Prozesse sind eine basale Grundlage für Kommunikation
- → Interprozesskommunikation
- Zwei Möglichkeiten:
  - Prozesse laufen auf einem Host
  - Prozesse laufen auf mehreren Hosts
- Multithreading in verteilten Systemen
  - Effizientsteigerung
  - Verwendung von blockierenden I/O-Operationen
- Server-Entwurf
  - Entwurfsaspekte (statuslos, statusbehaftet,…)

Diskussion