

Interprozesskommunikation

Bei der **Interprozesskommunikation** geht es um den Austausch von Informationen zwischen zwei (oder mehr) Prozessen bzw. Threads. Damit alle Beteiligten die ausgetauschten Informationen in gleicher Weise verstehen können, sind bestimmte Regeln der Kommunikation einzuhalten, das sogenannte *Protokoll*.

Zwei Threads kommunizieren über gemeinsame Variablen

→ Synchronisation Threads beim Zugriff auf gemeinsamen Datenbereiche wird erforderlich, da es zu kritischen Abläufen kommt

Zwei Prozesse kommunizieren über gemeinsame Speicherobjekte (Dateien)

→ Prozess A erzeugt während Laufzeit Daten und speichert diese in einer Datei ab.

→ zweiter Prozess B liest Datei zu einem **späteren** Zeitpunkt ein und kann so die enthaltenen Informationen weiterverarbeiten.

→ sorgt Betriebssystem nur für eine Synchronisation beim *gleichzeitigen Zugriff* der beiden Prozesse auf die (z.B. über Semaphore).

→ Falls Prozess B Dateiinhalt ausliest, **bevor** Prozess A die Informationen hineingeschrieben hat, so ist nicht zu erwarten, dass Prozess B das erwartete Ergebnis berechnen kann

Zwei Prozesse kommunizieren über Shared Memory

→ Man spricht von **Shared Memory**, wenn Teil des Hauptspeichers gemeinsam mehreren Prozessen zur Verfügung gestellt wird

Zwei Prozesse kommunizieren über Pipes

→ *Einweg-Kommunikationskanäle*, die Prozess ermöglichen, Daten über das BS als Datenstrom an anderen Prozess zu übertragen

Zwei Prozesse kommunizieren über Sockets

→ Sockets sind eine vom Betriebssystem bereitgestellte Kommunikationsmöglichkeit über TCP

Revision #2

Created 24 September 2022 16:24:24 by Merith Holtmann

Updated 2 October 2022 20:59:29 by Merith Holtmann