

Übungsblatt 5 – Prozesskommunikation

Aufgabe 1: Schauen Sie sich noch einmal alle Folien aus der Vorlesung an und stellen Sie sicher, dass Sie sie verstanden haben. Falls in der Vorlesung nicht alle Folien besprochen wurden, dann arbeiten Sie die nicht besprochenen Folien selbstständig durch. Benutzen Sie dabei auch die Literatur. Falls Fragen offen bleiben sollten, besprechen Sie diese während der Übungsstunde mit dem Betreuer.

Aufgabe 2:

- Lesen Sie das Kapitel über Prozesskommunikation, insb. pipes und sockets in dem Buch, das Sie sich besorgt haben. Die Abschnitte, in denen es in deutlich tiefere Details geht als in der Vorlesung angesprochen, können Sie überspringen.
- Erstellen Sie Ihre persönliche handschriftliche Zusammenfassung der wichtigsten Inhalte der Lehrveranstaltung.
- Überlegen Sie sich drei Verständnisfragen zur Lehrveranstaltung und halten diese schriftlich fest.

Wichtige Hinweise:

- Schreiben Sie alle Übungsprogramme **vernünftig formatiert** und **sinnvoll mit Kommentaren dokumentiert**.
- Bevor Sie einen Systemaufruf programmieren, **lesen Sie** in der **Dokumentation** der C-Library die Beschreibung dieses Systemaufrufs nach.
- Schreiben Sie alle folgenden C-Programme so, dass bei jedem Systemaufruf **überprüft** wird, ob ein **Fehler** aufgetreten ist. Falls Fehler auftreten, dann geben Sie eine aussagekräftige Fehlermeldung aus.
- Schalten Sie** beim übersetzen immer alle **Compilerwarnungen an!** Benutzen Sie außerdem den Debug-Modus, so dass Sie ggf. mit einem Debugger nachvollziehen können, was ihr Programm macht.
- Automatisieren Sie alle Übersetzungsvorgänge mit Hilfe eines **Makefile** Fügen Sie ein target **clean** ein, das beim Aufruf alle Zwischen-, Hilfs- und erzeugte Programmdateien wieder löscht.
- Wenn Sie beim programmieren Fehler gemacht haben, überlegen Sie, wie Sie diesen Fehler zukünftig vermeiden, bzw. schneller entdecken können.

Aufgabe 3: Bidirektionale Kommunikation durch pipes. Finden Sie die für diese Aufgabe relevanten Abschnitte in der Dokumentation der C-Library und lesen diese durch.

a) Schreiben Sie ein C-Programm, das

- einen Kindprozess erzeugt.
- Das Kind soll vom Elternprozess Textzeilen (über eine pipe) einlesen und diese in Großbuchstaben umwandeln und dem Elternprozess (über eine pipe) zurück schicken.
- Bekommt das Kind eine Zeile mit weniger als 10 Zeichen, dann beendet es sich.
- Der Elternprozess liest vom Benutzer so lange Textzeilen ein, wie der Benutzer mehr als 10 Zeichen in die Zeile tippt.
- Der Elternprozess sendet die Zeilen (über eine pipe) an das Kind, liest die Antwort des Kindes (aus einer pipe) und zeigt sie auf dem Bildschirm an.

b) Compilieren und starten Sie das Programm.

Aufgabe 4: Vorbereitung auf die Socket-Programmierung. Finden Sie die für diese Aufgabe relevanten Abschnitte in der Dokumentation der C-Library und lesen diese durch.

- a) Kopieren Sie den auf dem Moodle-Server zur Verfügung gestellten Quellcode `SocketExample.c` auf ihren lokalen PC.
- b) Schauen Sie sich den Quellcode an und vollziehen Sie die Vorgehensweise der socket-Programmierung an diesem Beispiel nach.
- c) Öffnen Sie ein zweites Terminal-Fenster und geben darin das Kommando `nc -l 3000` ein. Machen Sie sich klar, was dieses Kommando macht.
- d) Compilieren Sie `SocketExample.c` und lassen es laufen. Was sehen Sie im zweiten Terminal-Fenster? Warum?

Aufgabe 5: Freiwillige Zusatzaufgabe: Schreiben eines Serverprogramms, das über einen Socket angesprochen wird. Finden Sie die für diese Aufgabe relevanten Abschnitte in der Dokumentation der C-Library und lesen diese durch. Ein Grundgerüst für das Programm findet man in der Dokumentation.

a) Schreiben Sie ein C-Programm, das

- auf dem lokalen Rechner ein TCP-Socket auf Portnummer 3000 einrichtet und dort auf Verbindungsaufbau wartet.
- Wenn eine Verbindung aufgebaut wird, dann gibt der Server-Prozess die übertragenen Bytes nach `stderr` aus.

b) Compilieren und starten Sie das Programm.

c) Öffnen Sie ein zweites Terminal-Fenster und starten dort das Kommando `nc` so, dass es sich mit dem Server verbindet. Senden Sie einige Textzeilen an den Server und bauen die Verbindung wieder ab.