Prof. Dr. M. Jäger
FB MNI

16. November 2011

Übungsblatt 3 - Betriebssysteme

Aufgabe 1

Lesen Sie im System-Manual (*man*-Kommando) die Einträge zum *open*-Systemaufruf und zu *read*, *write* und *lseek*.

Achtung: Oft gibt es zu einem Stichwort mehrere Einträge in unterschiedlichen Sektionen des Manuals, z.B. gibt es ein Kommando namens *read* mit einem Manualeintrag in Sektion 1, während der gleichnamige read-Systemaufruf in Sektion 2 beschrieben ist.

Das Kommando „`man -s 2 read`“ liefert in diesem Fall den richtigen Eintrag zum read-Systemaufruf, während „`man read`“ die Kommandobeschreibung aus Sektion 1 des Manuals anzeigt.

Beantworten Sie folgende Fragen:

1. Was genau bedeuten die Parameter im Aufruf

   ```
   int fd=open("begruessung", O_RDWR | O_CREAT | O_TRUNC, 0755)
   ```

2. Welche Gründe kann es für das Scheitern eines *open*-Aufrufs geben.
3. Welchen *return*-Wert liefert *open* beim Scheitern?
4. Woran erkennt man beim Lesen mit *read* das Dateiende?
5. Woran erkennt man beim Schreiben mit *write*, dass kein Platz mehr auf dem Datenträger vorhanden ist?

Aufgabe 2

Schreiben Sie ein Programm „erzeuge-datei“, das eine Datei zum Schreiben öffnet, dort einen Text hineinschreibt und die Datei wieder schließt. Der Dateipfad wird als erster Parameter an das Programm übergeben, der Text als zweiter. Falls die Datei noch nicht existiert, soll sie erzeugt werden, andernfalls soll der alte Inhalt beim Öffnen gelöscht werden.

Anschließend soll das Programm die selbe Datei zum Lesen öffnen und den Inhalt am Bildschirm anzeigen. Verwenden Sie zum Lesen und Schreiben der Dateien **nur** die Systemaufrufe „open“, „read“, „write“, „close“.

Aufgabe 3

Lesen Sie sich den Manualeintrag von `dup2` durch.

Schreiben Sie ein Programm, das in einem Subprozess ein anderes Programm aufruft, nachdem es dessen Standardausgabe auf eine Datei umgelenkt hat. Der Programmpfad wird als erster, der Dateipfad wird als zweiter Parameter übergeben.

Aufgabe 4

Lesen Sie sich den Manualeintrag von `pipe` durch.

Schreiben Sie ein Programm, in dem ein Subprozess erzeugt wird, der mit seinem Elternprozess über eine Pipe Daten austauscht. Der Elternprozess schreibt einen Text in die Pipe, der Kindprozess liest den Text und zeigt ihn auf dem Bildschirm an. Der Text wird als erster Parameter an das Programm übergeben. Verwenden Sie zum Lesen und Schreiben der Daten **nur** die Systemaufrufe „read“ und „write“.