

Übungsblatt 3 - Betriebssysteme WS 2018

Aufgabe 1 (Dateien)

Lesen Sie im System-Manual die Einträge zu den Systemaufrufen *open*, *read* und *write*. Schreiben Sie ein C-Programm *copy*, das eine Datei kopiert.

Aufruf-Syntax:

```
copy <INPUT> <OUTPUT>
```

- <INPUT> ist der Pfad der zu kopierenden Datei. Wenn hier ein Minuszeichen steht, soll die Standardeingabe verwendet werden.
- <OUTPUT> ist der Pfad der Kopie oder ein Minuszeichen. Wenn hier ein Minuszeichen steht, soll die Standardausgabe verwendet werden.
- Wenn die Datei nicht vorhanden ist, soll sie neu erzeugt werden. Wenn sie schon vorhanden ist, soll der alte Inhalt gelöscht werden.

Aufgabe 2 (Ein-/ Ausgabeumlenkung)

Lesen Sie im System-Manual den Eintrag zum Systemaufruf *dup2*. Schreiben Sie ein C-Programm *redirect*, das ein anderes Programm aufruft, so dass dessen Standardeingabe, Standardausgabe und Standard-Fehlerausgabe mit *dup2* in Dateien umgelenkt werden.

Aufruf-Syntax:

```
redirect <INPUT> <OUTPUT> <ERROR> <PROGRAM> [<PARAM>...]
```

- <PROGRAM> ist das aufzurufende Programm
- <INPUT>, <OUTPUT> und <ERROR> sind die Pfade der Dateien, auf den die Eingabe, Ausgabe bzw. Fehlerausgabe umgelenkt werden sollen. Wenn bei einem dieser Argumente statt eines Dateipfads ein Minuszeichen steht, soll die entsprechend Umlenkung nicht gemacht werden.
- Hinter <PROGRAM> können beliebige weitere Argumente kommen, die an das aufzurufende Programm als Parameter durchgereicht werden sollen.

Aufgabe 3 (Pipeline)

Lesen Sie im System-Manual die Einträge zu den Systemaufrufen *pipe* und *waitpid*. Schreiben ein C-Programm *pipeline*, das eine Pipeline mit zwei Subprozessen realisiert.

Aufruf-Syntax:

```
pipeline <PROGRAM> [<PARAM>...] | <PROGRAM> [<PARAM>...]
```

Der Hauptprozess soll warten, bis beide Subprozesse fertig sind und dann selbst terminieren.

Beispiel:

```
pipeline ls -l /tmp | wc -l
```

Aufgabe 4 (Subprozessstatus)

Erweitern Sie das Pipeline-Programm der letzten Aufgabe wie folgt:

Sobald ein Subprozess beendet ist, soll der Hauptprozess dessen Terminierungsstatus (exit-Wert oder Signal-Information) ausgeben.

Aufgabe 5 (Signalbehandlung)

Lesen Sie im System-Manual die Einträge zu *sigaction* und *alarm*.

Schreiben ein C-Programm, *signaltest* das seine Standardeingabe auf seine Standardausgabe kopiert. Bei der Eingabe *STRG-C* (Signalnummer: SIGINT) soll jedes Mal eine Meldung ausgegeben werden. Wenn nach 60 Sekunden die Eingabe noch nicht vollständig gelesen bzw. kopiert wurde, soll das Programm mit einer Meldung „Zeit abgelaufen“ terminieren. Benutzen Sie dazu *alarm*.