

Praktikum 3

1. Aufgabe

Folgende Fragen dienen neben der Vorlesung als Vorbereitung für den Ilias-Test. Nutzen Sie zur Lösung auch die Unix-Dokumentation von der Laborhomepage.

1. Erklären Sie, wofür man eine Pipe benötigt!
2. Was ist die Voraussetzung für eine namenlose (unnamed) Pipe?
3. In welche Richtungen und welcher Reihenfolge werden die Daten in einer Pipe üblicherweise transferiert?
4. Wann setzt man eine benannte (named) Pipe ein?
5. Fehlermeldungen eines selbstgeschriebenen C-Programms sollen immer auf dem Bildschirm erscheinen, auch wenn die Standardausgabe in eine Datei oder eine Pipe gelenkt wird. Wie erreiche ich das?

2. Aufgabe: Prozess-Synchronisation mit pipe

Schreiben Sie ein C-Programm-System, dem beim Programmaufruf die beiden zu verbindenden Programme als Parameter ohne oder mit einer Option übergeben werden, die grundsätzlich mit einem Minuszeichen beginnt. Geben Sie Fehlermeldungen aus, wenn z.B. die Anzahl der Aufrufparameter nicht gültig ist, Aufrufe fehlgeschlagen sind usw. (**Hinweis:** Das Programm baut den einfachen Shellaufruf von `<PROG1> [-OPT] | <PROG2> [-OPT]` nach.)

Der Programmstarts könnten z.B. lauten:

```
./pipetest <prog1> <prog2>
./pipetest <prog1> <option1> <prog2>
./pipetest <prog1> <prog2> <option2>
./pipetest <prog1> <option1> <prog2> <option2>
```

3. Aufgabe: Prozess-Synchronisation mit named pipe

Der Verschiebechiffre (auch Caesar-Chiffre genannt), ist eine Methode, mit der auf einfache Weise Texte verschlüsselt werden können. Dies geschieht durch Ersetzung von Buchstaben – bei ROT13 im Speziellen wird jeder Buchstabe des lateinischen Alphabets durch den im Alphabet um 13 Stellen davor bzw. dahinter liegenden Buchstaben ersetzt. (Anm. Es handelt sich um eine keinesfalls sichere Verschlüsselung nach heutigen Maßstäben).

Schreiben Sie zwei C-Programme, die eine Kommunikation zwischen Caesar (Sender) und einem Boten (Empfänger) implementiert, der eine verschlüsselte Nachricht mittels einer named pipe erhält. Lesen Sie dazu im Sender Eingaben (Wörter und Sätze) ein, die Sie mittels einer Funktion `rot13(char a[], char b[])` „verschlüsseln“ und in eine named pipe schreiben. Der Empfänger soll die verschlüsselten Texte aus der named pipe auslesen und auf dem Bildschirm darstellen.

Wichtig sind hier die Funktionen `mkfifo()` und `unlink()`. Beachten Sie, falls die named pipe schon existiert, dass Sie dann vorher im Programm gelöscht werden muß.

Hinweis: Achten Sie bei der Implementierung auf die Verwendung von sicheren Funktionen zum Einlesen der Zeichenketten. Für die Umsetzung der `rot13()`-Funktion informieren Sie sich über die `ascii`-Codetabelle und beachten Sie Groß-Kleinschreibung.

Hinweise

Informieren Sie sich zu den Befehlen `pipe`, `dup2`, `fork`, `close`, `wait`, `execlp`, `fopen`, `fprintf`, `fgets`, `fflush`, `fclose`, `unlink` und `mkfifo`.

Lernziele

Kommandozeilenargumente, Prozesse, Pipes, Streams