

# OPERATING SYSTEMS BEISPIEL 3

## Aufgabenstellung – hangman

In dieser Aufgabe implementieren Sie das Spiel „Hangman“ in C. Die Implementierung soll aus zwei Programmen bestehen: einem Server, der die Geheimwörter angibt sowie den Spielstand verwaltet, und einem Client, mit welchem der Benutzer versuchen kann, Geheimwörter zu ermitteln. Die Interprozesskommunikation soll mittels Shared Memory realisiert werden und die Synchronisation mittels Semaphoren erfolgen.

Der Client soll dem Spieler ein Interface bieten, das den aktuellen Spielstand darstellt und dem Benutzer die Möglichkeit bietet, Buchstaben des Geheimwortes zu erraten. Der Server soll angeben, wie oft und an welcher Stelle des Geheimwortes der geratene Buchstabe vorkommt. Falls der Buchstabe im Geheimwort nicht enthalten ist, soll der Server einen weiteren Teil eines Galgens mit einem Gehängten zeichnen.

Es soll nicht zwischen Groß- und Kleinbuchstaben unterschieden werden. Wenn der Spieler (Client) nach 9 Fehlern das Geheimwort noch nicht erraten hat, wird der Galgen mit dem Gehängten komplett gezeichnet und der Spieler verliert.

## Anleitung

Der Server muss eine beliebige Anzahl von Clients, die gleichzeitig und unabhängig voneinander spielen können, unterstützen. Die Kommunikation soll mittels einem einzigen Shared Memory Objekt erfolgen (**nicht** einem pro Client).

## Server

USAGE: `hangman-server [input_file]`

Der Server legt zu Beginn die benötigten Ressourcen an. Falls eine Eingabedatei als Argument übergeben wird, liest der Server die Geheimwörter aus der Datei `input_file` ein. Beim Einlesen wird jede Zeile als ein Geheimwort interpretiert, wobei alle Zeichen außer ASCII Buchstaben und Leerzeichen ignoriert werden. Falls keine Eingabedatei angegeben wird, soll stattdessen von der Standardeingabe `stdin` eingelesen werden, bis EOF (`Ctrl+D`) auftritt.

Danach verwaltet der Server die Spiele der Clients und bedient auch deren Anfragen. Der Server muss dazu alle notwendigen Informationen für jeden Client verwalten. Insbesondere muss der Server alle Vorkommnisse der bisher korrekt geratenen Buchstaben und die Anzahl an Fehlern den Clients in jeder Runde bekannt geben.

Wenn das Spiel endet, soll der Server dem Client das Ergebnis übermitteln. Wenn ein Client ein neues Spiel starten will, soll der Server das Spiel mit einem bisher (für diesen Client) nicht verwendeten Geheimwort starten. Falls der Client ein Spiel mit allen Geheimwörtern gespielt hat, muss der Server den entsprechenden Client darüber informieren.

Wenn ein registrierter Client aus irgendwelchen Gründen (inklusive Signale) terminiert, soll der Server die entsprechenden Ressourcen freigeben. Zusätzlich muss der Server vor seiner eigenen Terminierung alle verwendeten Ressourcen (z.B., dynamisch reservierter Speicher, Semaphore, Shared Memory, usw.) freigeben. Der Server selbst soll durch die Signale `SIGINT` und `SIGTERM` „sauber“ terminierbar sein.

## Client

USAGE: `hangman-client`

Der Client soll beim Starten zunächst versuchen sich zum Server zu verbinden („verbinden“ heißt hier zu überprüfen, ob die Semaphore, die der Server anlegt, tatsächlich existieren). Nach einem erfolgreichen Verbindungsaufbau registriert sich der Client beim Server und kann ein neues Spiel beginnen.

Der Client erlaubt nun dem Spieler in jedem Spielzug, einen bisher nicht gewählten Buchstaben auswählen. Falls das eingegebene Zeichen kein Buchstabe ist oder schon einmal geraten wurde, muss eine Fehlermeldung ausgegeben werden. Der Client schickt den ausgewählten Buchstaben an den Server, der den Spielzug durchführt und den Spielstand als Antwort bereit stellt.

Der nach jedem Spielzug ausgegebene Spielstand muss mindestens die folgenden Elemente enthalten:

- Das Geheimwort, wobei erratene Buchstaben auf ihren entsprechenden Positionen im Wort großgeschrieben dargestellt werden, während die restlichen Buchstaben durch ‘\_’ ersetzt werden. Leerzeichen werden von Anfang an dargestellt.
- Eine ASCII Zeichnung die ein anfaenglich unvollständiges Bild von einem Galgen mit Gehängten darstellt. Die Zeichnung muss nach jedem Fehler des Spielers erweitert und mit dem 9. Fehler vollständig werden.
- Eine Liste aller Buchstaben, die der Spieler bisher in diesem Spiel geraten hat.

Nach Ende eines Spiels wird zuerst eine entsprechende Nachricht und der aktuelle Win/Loss Stand (Anzahl der gewonnen und verlorenen Spiele) ausgegeben. Dann wird der Spieler gefragt, ob er mit dem nächsten Spiel weitermachen will, worauf der Spieler mit ja/nein (y/n) antworten soll. Falls er mit ‘y’ antwortet, wird auf ein neues Spiel vom Server gewartet, sonst soll der Client terminieren.

Falls der Server nach einem Spiel keine weiteren Spiele erstellen kann, also der Client ein Spiel mit jedem Geheimwort des Servers gespielt hat, wird der endgültige Win/Loss Stand ausgegeben und der Client beendet. Der Client kann jederzeit mit den Signalen `SIGINT` oder `SIGTERM` beendet werden. Außerdem soll der Client terminieren, wenn der Server beendet wurde.

Achten sie auf eine saubere Terminierung. Auch bei fehler- bzw. signalbezogener Terminierung sollen die Clients und den Server sauber terminieren und alle Ressourcen freigegeben werden.

## Richtlinien

Bitte beachten Sie auch die *Richtlinien für die Erstellung von C-Programmen* sowie die *Allgemeinen Hinweise zur Beispielgruppe 3* auf der Übungswebsite.