

# Testat zur Bescheinigung aktiver Teilnahme am Praktikum

Für die Bestätigung aktiver Teilnahme am praktischen Teil der Veranstaltung soll eine minimalistische Shell in der Programmiersprache C implementiert werden.

Wie Sie in den bisherigen Übungen bereits feststellen konnten, ist eine Shell nichts anderes als eine Schnittstelle, die es den Nutzern erlaubt, Programme zu starten oder kleine Skripte zu schreiben. Den Skriptinterpreter werden wir uns an dieser Stelle natürlich sparen und auch Annehmlichkeiten wie die Benutzung der Cursortasten sollen nicht implementiert werden.

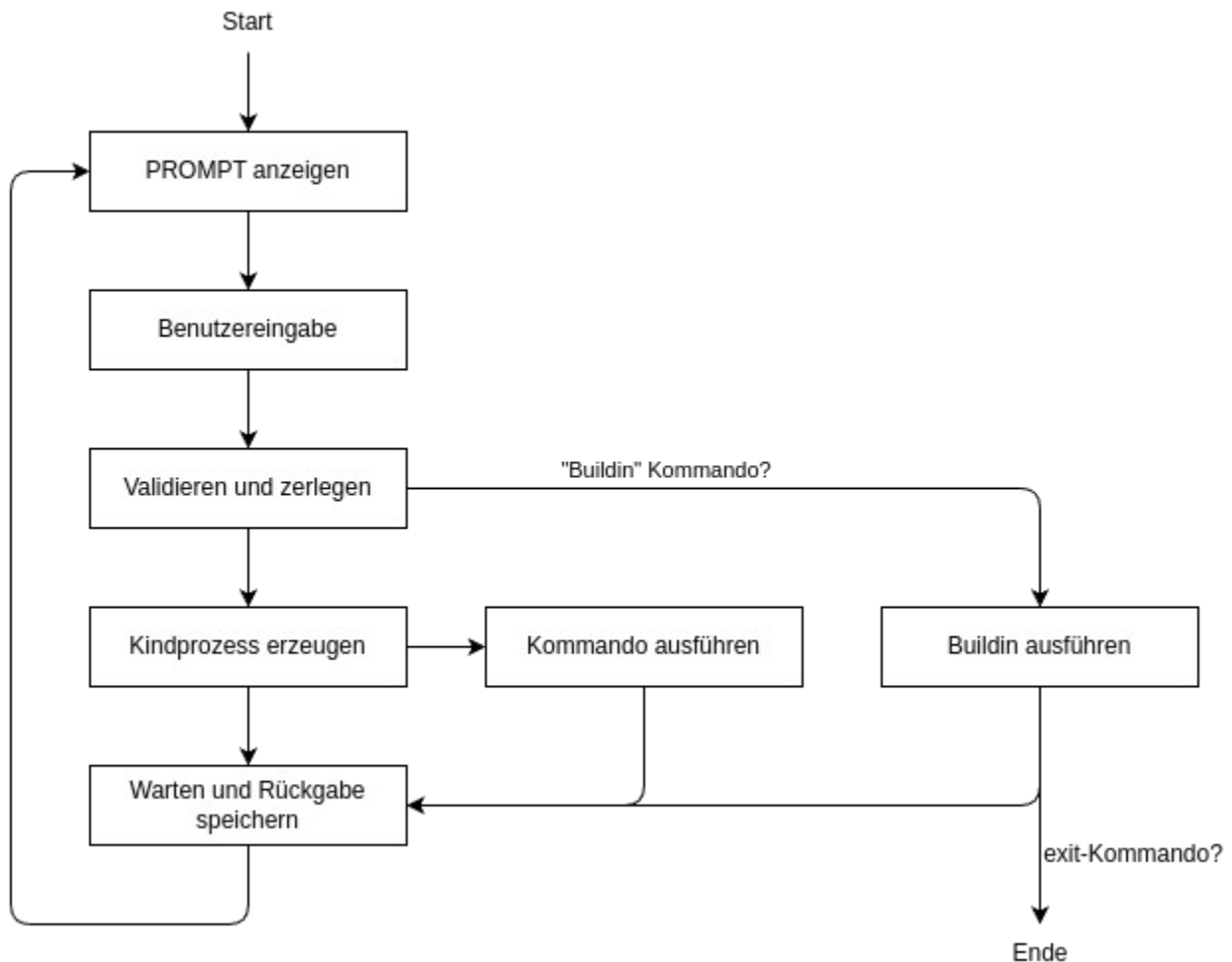
**Späteste Abgabe in Moodle: 31. Juli 2025, 23:59 Uhr**

## Zu implementierende Funktionen

- Zu Beginn und nach jedem ausgeführten Kommando wird ein Prompt ausgegeben, der (mindestens) das aktuelle Verzeichnis und ein Trennzeichen anzeigt (z.B. `/home/fd3430>` ).
- Es werden Eingaben beliebiger im System vorhandener Kommandos entgegen genommen (implementieren Sie keine Kommandos selbst, sondern erlauben Sie lediglich die Ausführung von Kommandos wie z.B. `ls`, `ps`, `dir`, usw.). Dabei soll natürlich die PATH-Variable ausgewertet werden, so dass Sie einfach `ls` statt `/usr/bin/ls` eingeben können.
- Bei der Ausführung von Programmen soll die Nutzung von Parametern möglich sein (also nicht nur die Ausführung des Kommandos `ps` bzw. `ls`, sondern auch `ps ax` oder `ls -a -l`).
- Die Shell soll blockieren (warten) bis das jeweils aufgerufene Programm terminiert. Anschließend kann ein weiteres Programm gestartet werden.
- Der Aufruf mehrere Programme in nur einer Kommandoeingabe soll möglich sein. Die Programme und deren Parameter werden dabei durch `;` getrennt (z.B. `ls -al ; ps ax`).
- Zwei Programme sollen durch eine Pipe (Trennzeichen `|`) verbunden werden können. (Mehr als zwei Programme sind nicht nötig, also nur eine einzige Pipe pro Eingabe).
- Das Wechseln von Verzeichnissen soll möglich sein. Dies muss von Ihnen implementiert werden, da `cd` kein Programm im System ist, sondern eine Funktion der Shell.
- Die Shell soll durch die Eingabe von `exit` beendet werden können. Auch dies ist kein Systemprogramm, sondern eine Funktion der Shell.
- Der Rückgabewert des letzten Kommandos soll durch die Eingabe des Kommandos `ret` angezeigt werden. Implementieren Sie dazu das Kommando `ret`.
- Das Signal `SIGHUP` soll abgefangen werden. Ein eigener Signalhandler soll ebenfalls den Rückgabewert des letzten Kommandos anzeigen.

Achten Sie auf eine ordentliche Formatierung (z.B. gleichmäßige Einrückung), behandeln Sie mögliche Fehler und kommentieren Sie den Programmcode wo nötig so, dass sie ihn erklären könnten. Fügen Sie außerdem ganz oben einen Kommentar ein, der mindestens Ihren Namen und idealerweise eine Kurzbeschreibung des Programms enthält.

## Beispielhafter Programmablauf



## Hinweis (nur Informativ)

Die Funktionsweise einer Shell ist im POSIX-Standard (POSIX.1-2024) definiert:

<https://pubs.opengroup.org/onlinepubs/9799919799/>

(im Index auf „Shell & Utilities“ klicken)