

# Vorlesung Betriebssysteme

## Wintersemester 2021/2022

Prof. Dr. Lars-Olof Burchard  
Hochschule Darmstadt

### 3. Praktikumsaufgabe (Shell, Teil 2)

Erweitern Sie Ihre in Aufgabe 2 programmierte Shell so, dass Sie einzelne laufende Prozesse (egal ob Vorder- oder Hintergrundprozesse) beliebig anhalten bzw. weiterlaufen lassen können. Dazu müssen Sie Ihre Shell aus Aufgabe 2 so erweitern, dass:

1. ein laufender Vordergrundprozess bei Eingabe von STRG-Z (bzw. CTRL-Z) mithilfe des Signals SIGTSTP angehalten wird und die Shell die Kontrolle wiedererlangt. Die PID des angehaltenen Prozesses soll dabei ausgegeben werden. Die Wirkungsweise von SIGTSTP können Sie auf einer realen Shell wie z.B. bash beobachten.
2. ein laufender Hintergrundprozess mit Prozess ID *pid* durch Eingabe von **stop** *<pid>* mithilfe des Signals SIGTSTP angehalten wird. Prüfen Sie hier mittels 'ps ax' auf einer anderen Shell, ob Ihr Prozess auch tatsächlich angehalten wurde!
3. ein angehaltener Prozess mit ID *pid* bei Eingabe von **cont** *<pid>* weiterläuft. Wenn der Prozess ein Vordergrundprozess war, soll er ebenfalls im Vordergrund weiterlaufen, andernfalls soll der Prozess im Hintergrund weiterlaufen.

Ihre Shell soll eine beliebige Anzahl von Prozessen auf diese Art verwalten können. Außerdem soll die Shell noch folgende Funktionalität erfüllen:

1. Ihre Shell soll beendete Hintergrundprozesse korrekt erkennen und diese aus der in Aufgabe 2 jeweils ausgegebenen Liste entfernen. Das bedeutet, dass auch keine Zombieprozesse hinterlassen werden, wenn Hintergrundprozesse beendet werden. Stellen Sie dazu sicher, dass die Shell geeignet auf das Beenden eines Prozesses reagiert. Hinweis: Sie sollen dafür die Information abfangen, dass ein Prozess beendet wurde (unter Zuhilfenahme von SIGCHLD).
2. Wenn die Shell mittels "logout" beendet werden soll, soll zuerst geprüft werden, ob noch ein Prozess im Hintergrund läuft und in diesem Fall eine geeignete Ausgabe (z.B. von der Art „Es laufen noch Prozesse im Hintergrund mit den PIDs: 52360, 49231“) angezeigt werden, anstatt die Shell zu beenden.

Starten und testen Sie Ihre Shell **gründlich**. Entwerfen Sie dazu eigene Testfälle, die alle oben genannten Fälle korrekt abdecken und dokumentieren Sie diese. Quellcode und Dokumentation der Tests (ASCII oder PDF) sollen in Ihr gitlab Repository hochgeladen werden.

Beachten Sie bei der Implementierung folgende Hinweise:

- Installieren Sie in Ihrer Shell den notwendigen Signalhandler über die Aufrufe **signal** oder alternativ **sigaction**. Informieren Sie sich über diese Funktionen, die einzelnen Signale und ihre Bedeutung, z.B. mittels „**man 7 signal**“!
- Zum Verschicken eines Signals an einen anderen Prozess nutzen Sie den Systemaufruf **kill** („**man 2 kill**“).
- Die folgenden Signale sind für diese Aufgabe relevant: SIGTSTP, SIGCHLD, SIGCONT. Es sollte nicht notwendig sein, mit anderen Signalen zu arbeiten. Recherchieren Sie, welche Bedeutung diese Signale jeweils haben und wann diese ausgeliefert werden.

- Die Signalhandler müssen nur im Elternprozess implementiert und installiert werden.
- Damit das Signal SIGTSTP nur an Ihre Shell gesendet wird und nicht an mglw. laufende Hintergrundprozesse, muss jeder Kindprozess nach der Erzeugung einer anderen Prozessgruppe zugewiesen werden. Als ID der Prozessgruppe können Sie die PID des Prozesses nutzen. Den Wechsel der Prozessgruppe erreicht man mit dem Systemaufruf **setpgid**. Beispiel: Wenn PID des Prozesses 1234 ist, dann wird dem Prozess auch die Prozessgruppe (PGID) mit der ID 1234 zugewiesen

Beispiel:

```
> ./myshell
Welcome to myshell!
$ firefox
^Z [SIGTSTP: pid 5613]
$ cont 5613
^Z [SIGTSTP: pid 5613]
$ cont 5613
    ... hier den Vordergrundprozess 5613 beenden ...
$ firefox &
[Hintergrund: 5622]
$ stop 5622
$ cont 5622
$ logout
logout nicht möglich. Es laufen noch Prozesse
im Hintergrund mit den PIDs: 5622
    ... hier den Hintergrundprozess 5622 beenden ...
$ logout
Wollen Sie die Shell wirklich beenden(J/N)? J
>
```