

Vorlesung Betriebssysteme

Wintersemester 2017/2018

Prof. Dr. Lars-Olof Burchard
Hochschule Darmstadt

3. Praktikumsaufgabe (Shell, Teil 2)

Erweitern Sie Ihre in Aufgabe 2 programmierte Shell um eine **Jobkontrolle**, die es erlaubt **alle** von der Shell gestarteten und noch laufenden Prozesse wie folgt zu verwalten:

1. Erweitern Sie die Shell so, dass bei Eingabe von CTRL-C der aktuell laufende Vordergrundprozess (kein Hintergrundprozess!) das Signal SIGINT zugestellt bekommt, nicht Ihre Shell.
2. Erweitern Sie die Shell so, dass bei Eingabe von CTRL-Z der aktuell laufende Vordergrundprozess (kein Hintergrundprozess!) das Signal SIGTSTP zugestellt bekommt.
3. Implementieren Sie einen Shell Befehl **fg**, der einen mittels CTRL-Z (SIGTSTP) angehaltenen Prozess wieder in den Vordergrund bringt.
4. Implementieren Sie einen Shell Befehl **bg**, der einen mittels CTRL-Z (SIGTSTP) angehaltenen Prozess im Hintergrund weiterlaufen lässt.
5. Stellen Sie sicher, dass die Shell nur dann beendet werden kann, wenn sämtliche im Hintergrund laufenden Prozesse

- ebenfalls beendet sind, andernfalls soll eine Fehlermeldung ausgegeben werden.
6. Stellen Sie sicher, dass die Shell keine Zombie Prozesse hinterlässt, indem Sie geeignet auf das Beenden eines Prozesses reagieren.
 7. Wenn Sie das Signal SIGINT oder SIGTSTP an Ihre Shell senden, sollen Hintergrundprozesse dieses Signal **nicht** erhalten. Um zu verhindern das Signale, die an Elternprozesse (d.h. die Shell) geschickt werden, auch an Kindprozesse (d.h. die Hintergrundprozesse) gesendet werden, ändern Sie die Prozessgruppe des Kindprozesses auf die PID des Kindprozesses. Sie können dafür den Systemaufruf **setpgid** nutzen.
 8. Wenn kein Vordergrundprozess läuft, soll mittels CTRL-C die Shell beendet werden können. Dazu soll allerdings vorher eine Abfrage („Wollen Sie die Shell wirklich beenden? (J/N)“ angezeigt und die Antwort entsprechend ausgewertet werden.

Hinweise:

- Verwenden Sie Signalhandler und die Aufrufe **signal** oder alternativ **sigaction**. Informieren Sie sich über diese Funktionen, die einzelnen Signale und ihre Bedeutung, z.B. mittels „**man 7 signal**“!
- Die Signalhandler müssen nur im Elternprozess implementiert und installiert werden.
- Starten Sie Ihre Shell von der Kommandozeile (auch wenn Sie mit Netbeans entwickeln), da Netbeans häufig für „merkwürdige“ Effekte sorgt.

Beispiel

```
> ./myshell
Welcome to myshell. Exit with "logout".
$ firefox
^Z [caught SIGTSTP]
$ bg
[4132]
$ logout
Unfinished jobs: [4126] Cannot log out.
$ firefox
^C [caught SIGINT]
$ firefox
^Z [caught SIGTSTP]
$ fg
^C [caught SIGINT]
$ logout
All done. Good bye!
>
```

Testen Sie Ihre Shell mit folgenden Befehlen/Eingaben in dieser Reihenfolge (Achtung: am Schluss sollte es notwendig sein, 2x **logout** einzugeben):

```
$ firefox
^Z [caught SIGTSTP]
$ ps ax
$ fg
^C [caught SIGINT]
$ firefox
^Z [caught SIGTSTP]
$ bg
^C [caught SIGINT]
$ ps ax
^C [caught SIGINT]
$ logout
$ logout
All done. Good bye!
```