

# Vorlesung Betriebssysteme

## Sommersemester 2017

Prof. Dr. Lars-Olof Burchard  
Hochschule Darmstadt

### 2. Praktikumsaufgabe (Shell)

1. Programmieren Sie in C / C++ eine auf der virtuellen Linux Maschine im Labor lauffähige Shell, die analog der realen UNIX Shell folgende Funktionalität aufweisen soll:
  - 1.1 Mittels Tastatur können Befehle eingegeben werden. <enter> bzw. <return> beenden die Eingabe eines Befehls und danach wird ein neuer Prozess erzeugt, der den eingegebenen Befehl (ein beliebiges existierendes Programm) ausführt.
  - 1.2 Die Shell wartet auf Beendigung jedes eingegebenen Befehls bevor weitere Befehle akzeptiert werden.
  - 1.3 Parameter sollen korrekt extrahiert und dem aufzurufenden Programm übergeben werden.
  - 1.4 Der Befehl **logout** soll Ihre Shell beenden. Dabei soll zuerst abgefragt werden, ob wirklich ein logout erfolgen soll und erst dann die Shell beendet werden.
2. Erweitern Sie die Shell so, dass Befehle, deren letztes Zeichen **&** ist, im Hintergrund ausgeführt werden. Das bedeutet, dass danach sofort weitere Befehle in der Shell eingegeben werden können während der mit **&** beendete Befehl ausgeführt wird. Zu jedem solchen Hintergrundprozess soll die Shell die entsprechende PID mit ausgeben.

Geben Sie zum Testen die u.a. folgenden Befehle (in dieser Reihenfolge) in Ihrer Shell ein:

```
> ./myshell
myshell> echo 1 2 3
myshell> date
myshell> date &
myshell> ps ax
myshell> trasdasd
myshell> ps ax
myshell> logout
Wollen Sie die Shell wirklich beenden (J/N)? J
>
```

Was passiert, wenn der „Befehl“ in Zeile 5 von Ihrer Shell abgearbeitet wird? Können Sie eine Veränderung der Ausgabe von „**ps ax**“ in Zeile 4 bzw. Zeile 6 feststellen?

### Hinweise:

1. Die Shell soll generell beliebige Befehle (z.B. auch **firefox**, **gedit**, **ls**, ...) ausführen können. Diese Befehle sollen in der Shell **nicht** hardcodiert sein.
2. Informieren Sie sich u.a. über die Systemaufrufe **fork**, **waitpid**, und **execvp** oder eine Variante davon (nicht **system!**) und verwenden Sie diese.
3. Die genaue Syntax und weitere Erläuterungen zu diesen Systemaufrufen finden Sie u.a. in den man pages, d.h. zum Beispiel mittels „**man execvp**“.
4. Achtung: Es ist nicht notwendig, in der Shell Befehle wie „**cd**“ zum Verzeichniswechsel zu implementieren.