

---

# Übungsblatt 2: Weitere Aufgaben

---

## 1 Was machst Du, wenn Dir die genaue Semantik eines UNIX-Kommandos entfallen ist?

`man Kommandoname, -h, --help`, seltener: `--?`

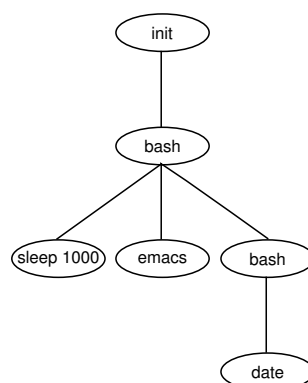
## 2 `bla` sei ein im Pfad ausführbares Programm. Was ist der Unterschied zwischen dem Aufruf `bla` und dem Aufruf `bla &` in der Shell? Welche Auswirkungen hat dies, wenn `bla` von Standard Input liest bzw. auf Standard Output schreibt?

`bla` = Vordergrund (Tastatur, Fenster, ...)

`bla &` = Hintergrund (Wird bei Input gestoppt, Output erfolgen auf `stdout/stderr`)

## 3 Gegeben sei der folgende Prozessbaum: (`init`) $\rightarrow$ (`bash`). Wie ändert er sich nach Eingeben der folgenden Kommandofolge im Shell?

```
sleep 1000 &  
emacs &  
bash  
date
```



#### **4 Nenne drei Beispiele für Informationen, die der Betriebssystemkern über einen Prozess wissen muss**

- geöffnete Dateien
- Prozesszustand (Running, Ready-to-run, Blocked)
- Prozess-ID
- Vater-Prozess

#### **5 Was ist eine Pipe?**

Mittel zur Interprozesskommunikation, undirektionaler, sequentieller Bytestrom, Zugriff wie auf Dateien (`read()`/`write()`)

#### **6 Wie macht man ein soeben editiertes Shell-File ausführbar?**

Setzen des x-Bits: `chmod +x file`

#### **7 In welche Bereiche (Segmente) ist der (virtuelle) Adressraum eines Programms in Ausführung in UNIX unterteilt, und welche Eigenschaften kennzeichnen sie?**

- Textsegment (Programmcode, lesbar und ausführbar)
- Datensegment (globale Daten, lesbar und schreibbar)
  - `.data` → initialisierte Variablen `!= 0`
  - `.bss` → nicht initialisierte Variablen, `== 0`
- Stacksegment („temporäre Daten“, lesbar und schreibbar, nach dem LIFO-Verfahren sortiert)

#### **8 Wozu wird der Stack verwendet?**

Retten von Bearbeitungszustand bei Funktionsaufrufen (Programmzähler, Parameter, u. U. Registerwerte) und Ablegen von lokalen Variablen

#### **9 Welchem Zweck dienen Bibliotheken (Libraries)?**

Bereitstellung allgemein verfügbarer Routinen/Funktionen

#### **10 Welche Aufgabe erfüllt ein Linker?**

Bindet mehrere Objektdateien zu einem gemeinsamen virtuellen Adressraum zusammen. Repositionierung und Auflösung von Querverweisen

## **11 Wozu wird beim Assemblieren eine Symboltabelle angelegt?**

Eine Symboltabelle listet die im Programm benutzten Symbole (z. B. Funktionsbezeichner, Variablennamen) zusammen mit der virtuellen Adresse, an der sie definiert wurden, auf und liefert damit wichtige Informationen für die Auflösung von Querverweisen beim Zusammenbinden (Linken). (Informationen u. U. getrennt in Symboltabelle, Stringtabelle und Relocationtabelle).

## **12 Welchen Vorteil hat es, Bibliotheken mit Position Independent Code zu versehen?**

Derselbe Code kann zu verschiedenen Programmen hinzugebunden werden (auch zwischen verschiedenen Programmen geteilt werden, muss also nur einmal im Hauptspeicher abgelegt werden).