

# Betriebssysteme Praktikum 1

Florina Nehmer

Jakob Ledig

18. April 2017

## 1. Bash

### 1.a. „Erste Erfahrungen mit der Bash“

- strg-C: Läuft kein Programm, geschieht nichts.
- cat: Gibt den Inhalt der Datei aus, auch Binärdateien
- cd: Der Name des Verzeichnisses, in das gewechselt wird, steht nach dem Wechsel auch vor dem Cursor.
- chmod: kann Rechte auch oktal (z.B. „0866“) entgegen nehmen, potentiell extrem gefährlich wenn man nicht weiß was man tut (etwa beim Aufruf „chmod 000 /“).
- cp: Identische, aber vom Original unabhängige Kopie entsteht am Ort des zweiten Parameters.
- date: Das ausgegebene Datum wird automatisch in seiner Darstellung der gewählten Uhrzeitsnotation angepasst (z.B. WD, DD, Mon, HH:MM:SS, Zeitzone, Jahr)
- df: Die Information kann per Parameter auch auf einzelne Laufwerke oder gar Ordner beschränkt werden.
- echo: Schreibt auf die Kommandozeile was immer als Parameter übergeben wird, z.B. auch Variablen.
- env: Ausgabe kann u.U. recht lang ausfallen, wenn viel Software Anpassungen der Umgebungsvariablen erforderlich gemacht oder selbst vorgenommen hat.
- exit: Es gibt wohl wenig mehr darüber zu sagen als „das Fenster/Skript wird geschlossen/beendet“.
- export: Jeder danach erzeugte Prozess hat Kenntnis von und Zugriff auf die genannte Variable.

- **find:** Findet Dateien im Dateisystem, kann per `-iname` auch case-insensitiv aufgerufen werden.
- **grep:** Mächtiges Durchsuchen mithilfe Ausdrücken auch für „gepipete“ Programmausgaben. Markiert Auffindungen rot.
- **ls:**  $\Rightarrow$  „list“et Dateien und Verzeichnisse, zeigt mit `-a` z.B. auch versteckte Dateien an und mit `-l` auch die Rechtevergabe.
- **man:** Zeigt die sog. Manpage an, falls eine unter `/usr/share/man` existiert.
- **mkdir:** Verzeichnis mit dem als Parameter übergebenen Namen entsteht, Leerzeichen müssen wie immer escaped werden.
- **more:** Anders als `less` liest `more` die anzuzeigende (nur Text-)Datei nicht erst vollständig ein, bevor sie ausgegeben wird.
- **mv:** Statt wie bei `cp` eine Kopie anzulegen, wird die angegebene Datei nicht kopiert, sondern an anderer Stelle im Verzeichnisbaum eingehängt.
- **PROG:** Startet das angegebene Programm (falls in `$PATH` verfügbar) und zeigt dessen Ausgaben in der Kommandozeile an.
- **ps:** Zeigt alle Prozesse, die aus diesem Terminal gestartet wurden, mit ihrer PID.
- **ptree:** Bei aufwändigen Window-Managern wie Unity ist der ausgegebene Baum nicht selten recht unübersichtlich wegen der vielen Kindprozesse.
- **VAR:** Der hinterlegte Wert ist bis zur Neuvergabe oder Schließen der Sitzung unter dem Namen der Variable verfügbar.

## 1.b. Fragen beantworten

### Variablen

**\$HOME** : Link aufs Homeverzeichnis des aktuell eingelogten Users.

**\$PATH** : In den hier vorgehaltenen Pfaden abgelegte (Binär-)dateien werden ausgeführt, wenn ihre Name in der Bash aufgerufen wird.

**\$UID** : Gibt die ID des aufrufenden Users aus. Wird beginnend mit 1000 numerisch fortlaufend vergeben.

**\$USER** : Name des aufrufenden Users.

## Eingabe von \$HOME

Der Befehl wechselt ins Homeverzeichnis des Users. Kürzer ist:

```
cd ~
```

## Pfeilasten und STRG + D

↑ geht in der Befehlshistorie rückwärts, ↓ vorwärts. STRG + D schließt das Terminal und beendet die Sitzung.

## .bashrc

bashrc ist ein Shellsript, dessen Inhalt beim Öffnen eines Terminalfensters ausgeführt wird.

## 2. Bash-Script

### 2.a. frename.sh

Siehe Quellcode

### 2.b. try\_host.sh

Siehe Quellcode

### 2.c. ausführbar machen

chmod<sup>1</sup> +x \$Dateiname, danach ist das Skript ausführbar.

### 2.d. \$PATH erweitern

Der folgende Befehl hängt den aktuellen Pfad an PATH an, der zweite exportiert die neue Variable an alle Kindprozesse.

```
PATH=$PATH:$(pwd)
export PATH
```

Der Vorgang ist jedoch nicht persistent.

---

<sup>1</sup> ggf. als root auszuführen, d.h. je nach Distribution entweder als Benutzer root oder mit vorangestelltem „sudo“.