

## Arbeiten mit einer Unix-Shell

Eine *Shell*, auch *Terminal* oder *Konsole* genannt, ist ein Kommandozeileninterpreter, das eine Zeile Text einliest und als Befehl ausführt. Wie der *Finder* (auf Mac OS X) ist die Shell eine Schnittstelle zwischen Benutzer und Computersystem, bietet allerdings keine graphische Ansicht. Die Verschachtelung der Verzeichnisse, d.h. an welchem Ort ein Ordner zu finden ist, ist dieselbe.

Öffnen eines Terminals: Im Dock ganz unten auf dem Desktop eines Macs befindet sich ein Icon *X11* sowie ein Icon *Terminal*. In Windows könnt ihr nach *Shell* suchen und findet *Powershell* oder ein ähnliches Programm. Alle öffnen durch Klicken ein neues Terminal Fenster, in dem ein *Prompt* steht, d.h. ein kurzer Text, der zeigt, dass man nun was eintippen kann. Dieser besteht aus dem Computernamen und dem Verzeichnis, in dem man sich befindet, also z.B.:

```
[multiple-man:~] cluser %
```

Wie beim Öffnen eines neuen *Finder* Fensters befindet man sich zunächst immer im Home Verzeichnis des Benutzers (hier *cluser*). Dies wird auch durch die Tilde (~) ausgedrückt.

Um ein Perl Skript auszuführen, muss man dem System nun zunächst den Weg/Pfad zu dem Ordner erklären, in dem er das Skript findet.

Mit dem Befehl **ls** (=list) werden alle Verzeichnisse und Dateien angezeigt, die der Computer an dem jetzigen Ort direkt ansteuern kann. Meist sind das im Home Verzeichnis Ordner wie *Desktop*, *Documents*, *Downloads*, etc.

Angenommen, das auszuführende Perl Skript liegt in einem Ordner, welcher wiederum auf dem Desktop liegt. Um dorthin zu kommen, muss man das Verzeichnis wechseln, indem man den Befehl **cd** (= change directory) und den Zielordner, hier also zunächst *Desktop*, eingibt:

```
cd Desktop
```

Wenn man sich nun im Ordner *Desktop* befindet (wird auch im *Prompt* angezeigt), kann man sich, wenn man möchte, zur Übersicht erneut durch **ls** die verfügbaren Verzeichnisse anzeigen lassen. Daraufhin wechselt man wieder mithilfe des Befehls **cd** bis man an dem Ort seines Skripts angekommen ist.

Nun kann man es einfach ausführen mit: `perl skriptname.pl`

Achtung: Wenn der Name eines Ordners oder ein Skriptname Leerzeichen enthält (Bsp. *SoSe 2012*), wird das in der Kommandozeile nicht erkannt. Deshalb immer Unterstriche oder Bindestriche in der Namensgebung verwenden oder alles zusammen schreiben.

Übersicht von Befehlen:

**cd Verzeichnisname:** Wechselt in ein direkt untergeordnetes Verzeichnis

**cd .. :** Wechselt in das direkt übergeordnete Verzeichnis. Wichtig ist das Leerzeichen zwischen cd und Punkten.

**cd ~ :** Wechselt in das Home Verzeichnis. Alternative: cd home.

**ls:** Liefert eine Liste an Verzeichnissen und Dateien, die am jeweiligen Ort ansteuerbar sind

**pwd:** Gibt den Pfadnamen an, in dem man sich befindet. (Print working directory)

**perl skriptname.pl:** Führt ein Perlskript aus

**Ctrl+c:** Stoppt die Befehlsausführung, wenn etwas zu lange dauert oder nicht tut, was es soll

**Ctrl+d** Wenn Ctrl-c versagen sollte (passiert manchmal), dann kann man es noch mit diesem Kommando probieren. Da wird dann definitiv alles abgeschossen, manchmal auch das Terminal gleich mit.

Praktische Tipps:

**Page up/down:** Mit den Pfeiltasten rauf und runter kann man durch die *History* der bereits angewendeten Befehle gehen. Das erspart einem lästiges Tippen eines gerade schon mal verwendeten Befehls, besonders wenn mehrere Skripte ausgeführt werden. So oft die Pfeiltasten drücken, bis der gewünschte Befehl erreicht ist, diesen gegebenenfalls verändern, *enter* drücken.

**Tab:** Hat man die ersten Buchstaben eines Befehls oder einer Datei oder eines Verzeichnisses getippt und drückt dann die Tabulator-Taste, wird eine automatische Vervollständigung durchgeführt. Gibt es mehrere Möglichkeiten der Vervollständigung, bekommt man diese beim zweiten Mal Drücken der Tabulator-Taste angezeigt und muss noch so viele Buchstaben zufügen, bis nur noch eine Möglichkeit bleibt. Tippfehler werden so vermieden und Zeit gespart.

Erstellt von Anja Leiderer SoSe 2012, Teile übernommen von Christine Rieger „Tutorium zu Perl für Linguisten 2010“.