

Linux_Bash

October 8, 2019

1 Herzlich Willkommen zum

2 Zusatztutorial in der Computerlinguistik

7.10.-11.10.2019

Tutoren: Yannick Kaiser, Anna Steinberg

Alle Unterlagen findet ihr auf <https://github.com/anbestCL/zusatztutorialCL>

Bei Fragen meldet Euch gerne bei uns: zusatztutorial@cis.lmu.de

3 Wochenplan

4 Einführung in Linux und Bash

5 Warum Unix?

- frei und open source -> wird in der Wissenschaft gerne gesehen
- vorinstallierte Programme, die besonders für Computerlinguisten interessant sind
- individuell anpassbar
- universeller
- unterstützt nativ UTF-8

6 Was sind Shell, Terminal und Konsole?

Shell: The shell is the program which actually processes commands and returns output. Most shells also manage foreground and background processes, command history and command line editing. These features (and many more) are standard in bash, the most common shell in modern linux systems. There are also other types of shells, e.g. zsh.

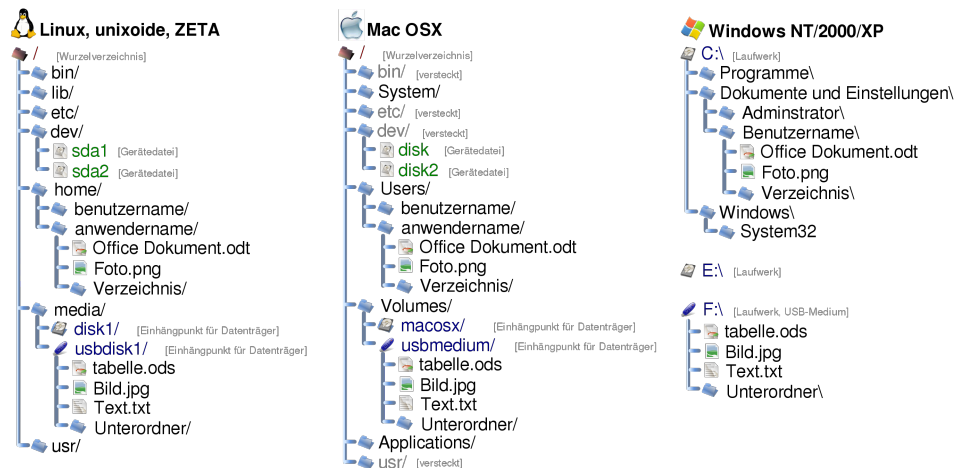
Terminal: A terminal refers to a wrapper program which runs a shell.

Konsole: The console is a special sort of terminal. Modern linux systems provide virtual consoles that don't need to be installed or configured to run. These are accessed through key combinations. Interacting with the console is also done using a shell program.

[Quelle](#)

Zeit	Montag	Dienstag	Mittwoch	Donnerstag	Freitag
09:00-10:00	Hilfe für: <ul style="list-style-type: none"> • CIP-Kennung • Linux + Eduroam einrichten 				
10:00-12:00	<ul style="list-style-type: none"> • Einführung in Linux und Bash • Einführung in Git(Hub) und Nano 	NLP-Pipeline: <ul style="list-style-type: none"> • Einlesen von Korpora • Parsing, Chunking • Tokenisierung, POS-Tagging, NER 	Mathematische Grundlagen: <ul style="list-style-type: none"> • Statistik • Stochastik 	Mathematische Grundlagen: <ul style="list-style-type: none"> • Lineare Algebra • Embeddings 	Machine Learning Grundlagen: <ul style="list-style-type: none"> • Klassifikation mit Perceptron • Non-Linearities • Neuronale Netze
12:00-13:00	Mittagspause	Mittagspause	Mittagspause	Mittagspause	Mittagspause
13:00-15:00	<ul style="list-style-type: none"> • Einführung in Python und NLTK • Set-Up von Python, Jupyter Notebooks und Entwicklungsumgebung (PyCharm) 	NLP-Pipeline: <ul style="list-style-type: none"> • Relation Extraction • Knowledge Bases 	Mathematische Grundlagen: <ul style="list-style-type: none"> • Automaten 	Klassifikation und Clustering: <ul style="list-style-type: none"> • Naive Bayes • K-Means • K-Nearest Neighbor 	Ausblick und Tipps: <ul style="list-style-type: none"> • Beispiel-aufgaben • Hinweise auf Online-Kurse

schedule.png



Mac_Linux_Windows.png

7 Navigation

Quelle

8 Wichtigste Befehle

- `man` : manual page of commands
- `pwd` : aktueller Dateipfad
- `ls` : list all files of directory
- `cd` : change directory
 - `.`
 - `..`
 - `~`
 - `*`
- `touch` : erstellt Datei, ändert Time Stamp
- `cat` : gibt Inhalt einer Datei zurück
- `grep` : innerhalb der Datei nach regulären Ausdrücken suchen
- `head/tail` : einen Teil der Datei ausgeben
- `less` : durch Datei scrollen
- `cp`: copy file or directory
- `mv` : move
- `rm` : remove
- `find` : sucht Datei
- `file` : gibt Informationen zur Datei zurück
- `mkdir` : make directory
- `rmdir` : remove directory

Notes: datei erstellen datei bearbeiten datei verschieben datei umbenennen datei löschen

9 Prozesse

- `ps` : aktuell laufende Programme
- `top` : aktuell laufende Prozesse
- `kill [PID]` : beendet den Prozess mit der PID

10 Programminstallation

sudo: führt Programm mit root-Zugriff (Admin, höhere Rechte) aus
Achtung bei graphischen Programmen!

Linux	Mac
<code>sudo apt-get htop</code>	<code>brew install htop</code>

- Zur Installation von Programmen, die Apple nicht mitliefert, empfiehlt sich den Paketmanager [Homebrew](#)

`which [program]` - gibt Dateipfad eines Programms zurück

11 Umgebungsvariablen

Terminal verwaltet Umgebungsvariablen, z.B. `PATH`,

`HOME`, und erleichtert damit dem Nutzer die Eingabe von Programmbefehlen:

`/Library/Frameworks/Python/3.7 -> python3`

in einer `.bashrc` (Linux) bzw. einer `.bash_profile` (Mac) Datei ist mithilfe von Umgebungsvariablen die Information gespeichert, wo das Terminal nach dem Programm mit dem Namen "python3" suchen muss

```
export PATH="/Library/Frameworks/Python/3.7:$PATH"
```

12 Pip-Installation

pip: Paketverwaltungsprogramm für Python-Pakete aus dem Python Package Index [PyPI](#)

pip für Python 2.7, pip3 für Python 3.6 and higher

auch beliebt: Paketmanager [anaconda](#)

```
pip3 install --user numpy scipy matplotlib ipython jupyter pandas
```

Pip installs Python packages whereas conda installs packages which may contain software written in any language.

For example, before using pip, a Python interpreter must be installed via a system package manager or by downloading and running an installer. Conda on the other hand can install Python packages as well as the Python interpreter directly.

Occasionally a package is needed which is not available as a conda package but is available on PyPI and can be installed with pip. In these cases, it makes sense to try to use both conda and pip.

13 Einführung in Git

In []:

A Typical Workflow

- ❶ Get current project state from remote
 - ▶ Initially: Clone project.
`git clone git@github.com:username/projectname.git`
 - ▶ Later: Fetch and merge changes from remote.
→ Possibly resolve conflicts.
`git pull`
- ❷ Make changes
 - ▶ Add a File.
`git add CHANGES.txt`
 - ▶ Edit a File.
`vi README.txt`
- ❸ Add and merge the changes locally.
`git commit -a -m "Summary of changes."`
- ❹ Fetch and merge changes from remote.
→ Possibly resolve conflicts.
`git pull`
- ❺ Push changes to remote.
`git push`

git_summary.png