

Einführung in die Kommandozeile

Was ist die Kommandozeile?

- Werkzeug zum textuellen Arbeiten mit dem Computer
- Klassische Aufgaben wie etwa..
 - Dateien: öffnen, schließen, anlegen, löschen, umbenennen ..
 - Durch Verzeichnisse navigieren
- ..werden über (Text)befehle umgesetzt (Beispiel: pwd)

Ein kleines Beispiel..

```
In [3]: %%bash
cd ../datasets/inaugural
ls -C
```

1789-Washington.txt	1849-Taylor.txt	1909-Taft.txt	1969-Nixon.txt
1793-Washington.txt	1853-Pierce.txt	1913-Wilson.txt	1973-Nixon.txt
1797-Adams.txt	1857-Buchanan.txt	1917-Wilson.txt	1977-Carter.txt
1801-Jefferson.txt	1861-Lincoln.txt	1921-Harding.txt	1981-Reagan.txt
1805-Jefferson.txt	1865-Lincoln.txt	1925-Coolidge.txt	1985-Reagan.txt
1809-Madison.txt	1869-Grant.txt	1929-Hoover.txt	1989-Bush.txt
1813-Madison.txt	1873-Grant.txt	1933-Roosevelt.txt	1993-Clinton.txt
1817-Monroe.txt	1877-Hayes.txt	1937-Roosevelt.txt	1997-Clinton.txt
1821-Monroe.txt	1881-Garfield.txt	1941-Roosevelt.txt	2001-Bush.txt
1825-Adams.txt	1885-Cleveland.txt	1945-Roosevelt.txt	2005-Bush.txt
1829-Jackson.txt	1889-Harrison.txt	1949-Truman.txt	2009-Obama.txt
1833-Jackson.txt	1893-Cleveland.txt	1953-Eisenhower.txt	README
1837-VanBuren.txt	1897-McKinley.txt	1957-Eisenhower.txt	summary.txt
1841-Harrison.txt	1901-McKinley.txt	1961-Kennedy.txt	
1845-Polk.txt	1905-Roosevelt.txt	1965-Johnson.txt	

Vor- /Nachteile Kommandozeile

Vorteile:

- Kann ohne grafische Oberfläche genutzt werden..
 - SSH: Wartung/Nutzung von entfernten Rechnern zB. Webserver
- "Mächtigere Befehle"
- Möglichkeit zur Automatisierung (Bash-Scripts)

Nachteile:

- Anfangs: Höhere Lernkurve
- Für kleine Aufgabe (doch) zu umständlich (zB. einzelne Dateien löschen geht grafisch schneller)

Navigation

- Beim Start befindet sich Kommandozeile in einem "Startordner"
 - `pwd` public working directory
- Wie bei Windows und co. können sie Ordner wechseln..
 - `cd Pfadname`, etwa: "`C:\Users\Desktop`" oder "`\home\pi`"
 - `cd ~` wechselt ins HOME-Verzeichnis (je nach Betriebssystem)
- `ls` zeigt den Inhalt des aktuellen Ordners an, mit `ls -a` auch Verstecktes

```
In [ ]: %%bash
        pwd
        #cd ~
        #ls -a
        #
```

Mit (Text)dateien arbeiten

Unser Trainingsbeispiel heute: *inaugural adresses*

- Textdateien aus NLTK, einer Bibliothek zur Sprachverarbeitung in Python
- 56 Textdateien , Antrittsrede jeweils eines Präsidenten

```
In [6]: %%bash
cd ../datasets/inaugural
ls -C
```

1789-Washington.txt	1849-Taylor.txt	1909-Taft.txt	1969-Nixon.txt
1793-Washington.txt	1853-Pierce.txt	1913-Wilson.txt	1973-Nixon.txt
1797-Adams.txt	1857-Buchanan.txt	1917-Wilson.txt	1977-Carter.txt
1801-Jefferson.txt	1861-Lincoln.txt	1921-Harding.txt	1981-Reagan.txt
1805-Jefferson.txt	1865-Lincoln.txt	1925-Coolidge.txt	1985-Reagan.txt
1809-Madison.txt	1869-Grant.txt	1929-Hoover.txt	1989-Bush.txt
1813-Madison.txt	1873-Grant.txt	1933-Roosevelt.txt	1993-Clinton.txt
1817-Monroe.txt	1877-Hayes.txt	1937-Roosevelt.txt	1997-Clinton.txt
1821-Monroe.txt	1881-Garfield.txt	1941-Roosevelt.txt	2001-Bush.txt
1825-Adams.txt	1885-Cleveland.txt	1945-Roosevelt.txt	2005-Bush.txt
1829-Jackson.txt	1889-Harrison.txt	1949-Truman.txt	2009-Obama.txt
1833-Jackson.txt	1893-Cleveland.txt	1953-Eisenhower.txt	README
1837-VanBuren.txt	1897-McKinley.txt	1957-Eisenhower.txt	summary.txt
1841-Harrison.txt	1901-McKinley.txt	1961-Kennedy.txt	
1845-Polk.txt	1905-Roosevelt.txt	1965-Johnson.txt	

head - Einen Überblick verschaffen

- head Dateiname ermöglicht eine Dateivorschau
 - konkrete Datei: head 1789-Washington.txt oder Dateimuster: head *.txt oder head 17**.*txt
- -c beschränkt Übersicht auf n Zeichen
 - head -c140 1789-Washington.txt zeigt ersten 140 Zeichen

```
In [7]: %%bash
cd ../datasets/inaugural
head -c100 "1789-Washington.txt"
```

Fellow-Citizens of the Senate and of the House of Representatives:

Among the vicissitudes incident

```
In [10]: %%bash
cd ../datasets/inaugural
head -c100 200*.txt
```

==> 2001-Bush.txt <==

President Clinton, distinguished guests and my fellow citizens, the peaceful transfer of authority i

==> 2005-Bush.txt <==

Vice President Cheney, Mr. Chief Justice, President Carter, President Bush, President Clinton, membe

==> 2009-Obama.txt <==

My fellow citizens:

I stand here today humbled by the task before us, grateful for the trust you ha

> - Ausgaben in Dateien umleiten

- Mit > leiten sie Textausgabe in eine Datei um
 - Beispiel: `head -c140 1789-Washington.txt > zusammenfassung_washington.txt`
 - Befehl leitet die ersten 140 Zeichen der Washington-Rede in neue Datei

```
In [12]: %%bash
cd ../datasets/inaugural
head -c140 "1789-Washington.txt" > "zusammenfassung_washington.txt"
```


Zwischenaufgabe:

1. Lassen sie sich eine Zusammenfassung (~100 Zeichen) aller Reden in eine Datei schreiben
2. Öffnen sie die erzeugte Datei mit dem Befehl `nano name-der-datei`

```
In [4]: %%bash
cd "../datasets/inaugural"
head -c140 *.txt > summary.txt
ls -C
#nano name-der-datei
#----Alternativ----
#cat name-der-datei
```

1789-Washington.txt	1869-Grant.txt	1949-Truman.txt
1793-Washington.txt	1873-Grant.txt	1953-Eisenhower.txt
1797-Adams.txt	1877-Hayes.txt	1957-Eisenhower.txt
1801-Jefferson.txt	1881-Garfield.txt	1961-Kennedy.txt
1805-Jefferson.txt	1885-Cleveland.txt	1965-Johnson.txt
1809-Madison.txt	1889-Harrison.txt	1969-Nixon.txt
1813-Madison.txt	1893-Cleveland.txt	1973-Nixon.txt
1817-Monroe.txt	1897-McKinley.txt	1977-Carter.txt
1821-Monroe.txt	1901-McKinley.txt	1981-Reagan.txt
1825-Adams.txt	1905-Roosevelt.txt	1985-Reagan.txt
1829-Jackson.txt	1909-Taft.txt	1989-Bush.txt
1833-Jackson.txt	1913-Wilson.txt	1993-Clinton.txt
1837-VanBuren.txt	1917-Wilson.txt	1997-Clinton.txt
1841-Harrison.txt	1921-Harding.txt	2001-Bush.txt
1845-Polk.txt	1925-Coolidge.txt	2005-Bush.txt
1849-Taylor.txt	1929-Hoover.txt	2009-Obama.txt
1853-Pierce.txt	1933-Roosevelt.txt	all_seperated.txt
1857-Buchanan.txt	1937-Roosevelt.txt	README
1861-Lincoln.txt	1941-Roosevelt.txt	summary.txt
1865-Lincoln.txt	1945-Roosevelt.txt	zusammenfassung_washington.txt

(Text)daten verarbeiten

Wir wollen nun einmal tiefer in die Daten schauen...

**Was könnte
interessant sein?**

wc - Wörter zählen

- wc gibt #Zeilen, #Zeichen, #Wörter und den Dateinamen aus
 - Beispiel: `wc 1789-Washington.txt` erzeugt `13 1431 8619 1789-Washington.txt`
- `wc -c/wc -m` (characters) `wc -l` (lines) `wc -w` (words) um Ausgabe zu filtern

```
In [5]: %%bash
cd "../datasets/inaugural"
# wc 1789-Washington.txt
# wc -c 1789-Washington.txt
# wc -l 1789-Washington.txt
# wc -w 1789-Washington.txt
wc --help
```

Usage: wc [OPTION]... [FILE]...

or: wc [OPTION]... --files0-from=F

Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. A word is a non-zero-length sequence of characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in the following order: newline, word, character, byte, maximum line length.

-c, --bytes	print the byte counts
-m, --chars	print the character counts
-l, --lines	print the newline counts
--files0-from=F	read input from the files specified by NUL-terminated names in file F; If F is - then read names from standard input
-L, --max-line-length	print the maximum display width
-w, --words	print the word counts
--help	display this help and exit
--version	output version information and exit

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>

Report wc translation bugs to <<http://translationproject.org/team/>>

Full documentation at: <<http://www.gnu.org/software/coreutils/wc>>

or available locally via: info '(coreutils) wc invocation'

Zwischenaufgabe:

1. Zählen sie die Wörter aller Reden
 - Welcher Präsident war am "gesprächigsten"?
 - Gibt es Ausreißer? Wie lassen sich diese Ausreißer erklären?

sort - Dateien/Ausgaben sortieren

- `sort dateiname` sortiert Datei zeilenweise
- standardmäßig wird aufsteigend lexikographisch sortiert
 - $a < b < \dots z < A < B < \dots Z$
 - $0 < 1 \dots !$ Fehler bei mehrziffrigen Zahlen, `-n` erzwingt numerische Sortierung
- `man sort` bzw. `sort --help` für weitere Einstellungen

sort - Dateien/Ausgaben sortieren


```
In [7]: %%bash
cd ../datasets/inaugural
wc -w *.txt | sort
```

```
79 summary.txt
135 1793-Washington.txt
571 1945-Roosevelt.txt
698 1865-Lincoln.txt
984 1905-Roosevelt.txt
1089 1849-Taylor.txt
1129 1869-Grant.txt
1130 1829-Jackson.txt
1177 1809-Madison.txt
1178 1833-Jackson.txt
1210 1813-Madison.txt
1233 1977-Carter.txt
1341 1873-Grant.txt
1360 1941-Roosevelt.txt
1390 1961-Kennedy.txt
1431 1789-Washington.txt
1500 1965-Johnson.txt
1533 1917-Wilson.txt
1582 2001-Bush.txt
1598 1993-Clinton.txt
1683 1957-Eisenhower.txt
1685 1885-Cleveland.txt
1699 1913-Wilson.txt
1730 1801-Jefferson.txt
1816 1937-Roosevelt.txt
1819 1973-Nixon.txt
1885 1933-Roosevelt.txt
2013 1893-Cleveland.txt
2056 2005-Bush.txt
2141 1969-Nixon.txt
2167 1805-Jefferson.txt
2170 1997-Clinton.txt
2212 1901-McKinley.txt
```

grep - Worte / Wortmuster suchen

- grep ermöglicht die Suche von Wort bzw. Wortmustern in Dateien
 - Ausgabe: Wort + umliegender Text
 - Ähnlich zu: "Strg+F" im Browser
- grep -n grep -b geben an wo sich Worte im Text befinden
- grep -c zählt wie oft das Wort im Text vorkommt
- grep --help

```
In [16]: %%bash
cd "../datasets/inaugural"
#grep "war" *.txt
#grep -n "war" *.txt
grep -c "war" 200*.txt
#grep --help
```

```
2001-Bush.txt:4
2005-Bush.txt:6
2009-Obama.txt:7
```

Zwischenaufgabe

1.Schauen sie, wie oft bestimmte Keywords in den einzelnen Reden vorkommen

- Interessante Keywords: terror, war, freedom, citizen ..

2.Sortieren sie die Aufgabe - verwenden sie hierfür `sort -t: -k2nr`

```
In [ ]: %%bash
        cd ../datasets/inaugural
        grep -c " terror " *.txt | sort -t: -k2nr
```

Ein komplexeres Beispiel - Worthäufigkeiten

Als abschließendes Beispiel wollen wir einmal die Worthäufigkeiten in allen Reden erfassen

```
In [ ]: %%bash
cd ../datasets/inaugural
export LC_ALL='C'
cat *.txt | tr " " "\n" | sort -b | uniq -c | sort -nr
```

Funktionsweise

1. Fügt alle Texte zu einem großen Text zusammen - `cat *.txt |`
2. Schreibe so um, dass jedes Wort in einer neuen Zeile steht - `tr " " "\n" |`
3. Sortiere diese Ausgabe - `sort |`
4. Fasse gleiche Zeilen zusammen und zähle `uniq -c |`
5. Sortiere diese Ausgabe wieder , numerisch und absteigend `sort -nr`

Mit PDFs arbeiten

Das Kommandozeilentool `pdftotext` ermöglicht es, PDF-Dateien in Text umzuwandeln

- Funktioniert nur falls "wirklicher" Text im PDF vorhanden ist
- Funktioniert nicht bei sog. Faksimile , d.h. zB. Bilder auf denen Text steht
- Befehl: `pdftotext Dateiname`

```
In [14]: %%bash
          cd ../pdf
          #ls
          #pdftotext bitcoin-whitepaper.pdf
```

Aufgabe

1. Wandeln sie das im Ordner pdf hinterlegte Bitcoin-Whitepaper in eine Textdatei um
2. Nutzen sie head um eine Übersicht über die Datei zu bekommen
3. Nutzen sie grep und schauen sie
 - wie oft typische Buzzwords wie *Blockchain* , *Crypto* oder *Bitcoin* im Text auftauchen
 - welche Worte für die technische Beschreibung genutzt werden

Fazit

- Kommandozeile bringt mächtige Tools mit - Nutzen sie sie!
- textuelles Arbeiten: anfangs sperrig, sobald gelernt: angenehm/nützlich
- Auch Alltags-Software bringt Textkommandos mit
 - zB. Google: define: Automat oder site:wikipedia.org

define: Automat (https://www.google.de/search?q=define:+Automat&ie=utf-8&oeab&gws_rd=cr&dcr=0&ei=7qmqWunLDcewkW4mICYBw).

site:wikipedia.org (https://www.google.de/search?safe=off&client=firefox-b-ab&dcr=0&ei=GaqgWvKJlo_VkwWpzZroDg&q=Commerzbank+site%3Awikipedia.ab.3...15794.20526.0.20718.31.30.0.0.0.180.2690.19j10.30.0..2..0...1.1.64.psy-ab..1.17.1768.6..0j35i39k1j0i131k1j0i67k1j0i22i30k1.102.9gWM-cBRdLY).
