Lehrstuhl für Angewandte Informatik in den
Kultur-, Geschichts- und Geowissenschaften

Otto-Friedrich-Universität Bamberg

Informatik für die Kulturwissenschaften

Übung: Kommandozeile

Christian Kremitzl
Prof. Dr. Christoph Schlieder

Altes Übungsblatt

- Aufgabe 1
  - Euer Rechner hat einen 2-Kern-Prozessor. 5 Prozesse sind gestartet und laufen.

  a) Wie viele Prozesse können sich im aktiven Zustand befinden?

  b) Welche beiden Zustände können die übrigen Prozesse einnehmen?

  c) Wie entsteht der Eindruck, dass alle Prozesse gleichzeitig laufen?

Altes Übungsblatt

- Aufgabe 1
  - Euer Rechner hat einen 2-Kern-Prozessor. 5 Prozesse sind gestartet und laufen.

  a) Wie viele Prozesse können sich im aktiven Zustand befinden?

  b) Welche beiden Zustände können die übrigen Prozesse einnehmen?

  c) Wie entsteht der Eindruck, dass alle Prozesse gleichzeitig laufen?

- Lösung

  a) 2 Kerne
  → max. 2 Prozesse gleichzeitig aktiv

  b) bereit, blockiert

  c) durch schnelles Wechseln zwischen den Prozessen

Altes Übungsblatt

- Aufgabe 2
  - Ihr bearbeitet eine größere Menge an Bildern und euer Rechner läuft sehr langsam.
  - Ihr vermutet eine hohe Auslastung der CPU. Aus der Prozessverwaltung geht aber hervor:
    - CPU: 30% ausgelastet
    - Arbeitsspeicher: 97% belegt
    - Festplatte 95% der maximalen Übertragungsrate genutzt
  - Warum läuft der Rechner wahrscheinlich so langsam?

Altes Übungsblatt

- Aufgabe 2
  - Ihr bearbeitet eine größere Menge an Bildern und euer Rechner läuft sehr langsam.
  - Ihr vermutet eine hohe Auslastung der CPU. Aus der Prozessverwaltung geht aber hervor:
    - CPU: 30% ausgelastet
    - Arbeitsspeicher: 97% belegt
    - Festplatte 95% der maximalen Übertragungsrate genutzt
  - Warum läuft der Rechner wahrscheinlich so langsam?
- Lösung
  - Die CPU ist nicht ausgelastet, dürfte also nicht der Flaschenhals sein.
  - Der Arbeitsspeicher ist voll, also werden wahrscheinlich Inhalte auf die Festplatte ausgelagert.
  - Die Festplatte ist ausgelastet und bremst als langsamste Komponente alles andere aus.

Altes Übungsblatt

- Aufgabe 3
  - Das OSI-Referenzmodell enthält sieben Schichten, wobei wir zwischen den obersten dreien keinen Unterschied machen.

5.–7. Anwendung
4. Transport
3. Vermittlung
2. Sicherung
1. Bitübertragung

  - Ordnet die folgenden Begriffe jeweils der richtigen Schicht zu:

Ethernet	POP3	TCP
HTTP	Repeater	TLS
IMAP	Router	UDP
IP-Adresse	SMTP	Web Browser
MAC-Adresse	Switch	WLAN

  - Schlagt ggf. nach, wofür die Begriffe stehen.

Altes Übungsblatt

- Aufgabe 3
  - Das OSI-Referenzmodell enthält sieben Schichten, wobei wir zwischen den obersten dreien keinen Unterschied machen.

Ethernet	POP3	TCP
HTTP	Repeater	TLS
IMAP	Router	UDP
IP-Adresse	SMTP	Web Browser
MAC-Adresse	Switch	WLAN

Schicht	Adressen	Hardware	Protokolle	Software
5.–7. Anwendung				
4. Transport				
3. Vermittlung				
2. Sicherung				
1. Bitübertragung				

Altes Übungsblatt

- Aufgabe 3
  - Das OSI-Referenzmodell enthält sieben Schichten, wobei wir zwischen den obersten dreien keinen Unterschied machen.

Schicht	Adressen	Hardware	Protokolle	Software
5.–7. Anwendung			HTTP, SMTP, POP3, IMAP	Web Browser
4. Transport			TCP, UDP, TLS	
3. Vermittlung	IP-Adr.	Router		
2. Sicherung	MAC-Adr.	Switch	Ethernet, WLAN	
1. Bitübertragung		Repeater	Ethernet, WLAN	

Altes Übungsblatt

- Aufgabe 4
  - Ihr entwickelt ein System zur Videotelefonie mit integrierten Textnachrichten.
  - Für die Transportschicht stehen euch TCP und UDP zur Verfügung.
  - Wie setzt ihr die beiden ein und warum?

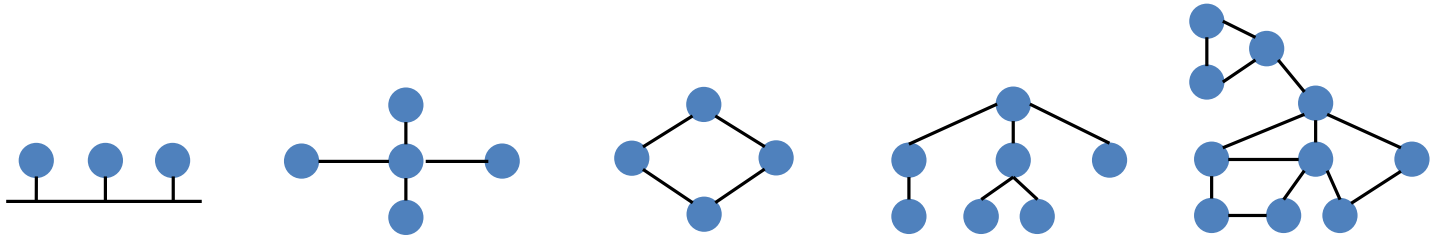
Altes Übungsblatt

- Aufgabe 4
  - Ihr entwickelt ein System zur Videotelefonie mit integrierten Textnachrichten.
  - Für die Transportschicht stehen euch TCP und UDP zur Verfügung.
  - Wie setzt ihr die beiden ein und warum?
- Lösungsvorschlag
  - UDP hat höheren Durchsatz, dafür können aber Pakete verloren gehen.
  - TCP stellt Zustellung sicher, braucht dafür aber mehr Bandbreite.
  - Bei Videotelefonie ist eine geringe Latenz wichtiger als evtl. Paketverlust → UDP
  - Bei Textnachrichten ist die sichere Zustellung wichtiger als die Latenz → TCP

Altes Übungsblatt

- Aufgabe 5
  - Vergleicht die folgenden Netzwerk-Topologien:
  a) Bus
  b) Stern
  c) Ring
  d) Baum
  e) vermascht
  - Gebt in Abhängigkeit von der Zahl der Rechner *n* jeweils den bestmöglichen, den schlechtestmöglichen und einen mittleren Wert für die folgenden Parameter an:
    - Zahl der Hops zwischen zwei beliebigen Rechnern
    - Zahl der nötigen Verbindungen
    - Nötige Änderungen zum Einfügen eines weiteren Rechners
    - Zahl der Rechner, die ausfallen können, bevor das Netz zerfällt

Altes Übungsblatt



	Bus	Stern	Ring	Baum	Vermascht
Hops	1	2	$1 \,..\, \frac{n}{2}, \quad \sim\frac{n}{4}$	$1 \,..\, 2d$	$1 \,..\, n-1$
Verbindungen	Bus $+ n$	n	n	$n-1$	$n-1 \,..\, \sum_{k=1}^{n-1} k$
Einfügen eines Rechners	Zusätzliches Kabel einstecken, ggf. weiteren Hub aufstellen	Zusätzliches Kabel einstecken, ggf. Switch ausbauen	Eine Verbindung trennen, neuen Rechner einfügen, Ring wieder schließen	Rechner an geeigneter Stelle einstecken	Mindestens ein Kabel einstecken, beliebig viele weitere
Ausfälle ohne Fragmentierung	n (ohne den Hub)	n (ohne den Switch)	1 (oder mehrere direkte Nachbarn)	0 innere Knoten, aber beliebig viele Blattknoten	n voll vermascht, sonst $1 \,..\, n$, je nach Position

Altes Übungsblatt

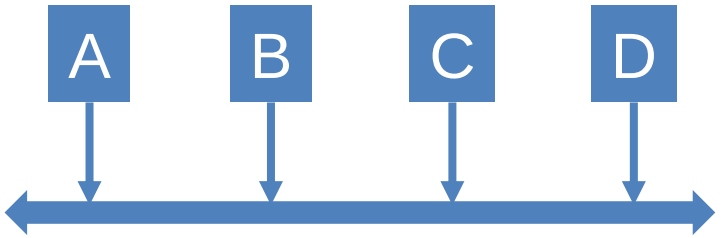
- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.
  - a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?
  - b) Rechner A möchte gleichzeitig Daten an C senden. Ist das möglich?
  - c) Was ist in diesem Beispiel die Kollisionsdomäne?

Altes Übungsblatt

- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.

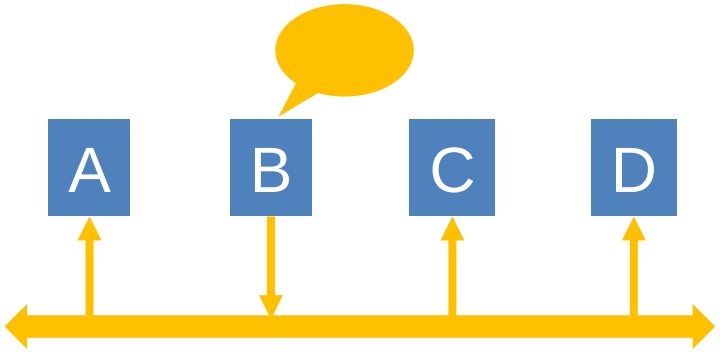
Altes Übungsblatt

- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.



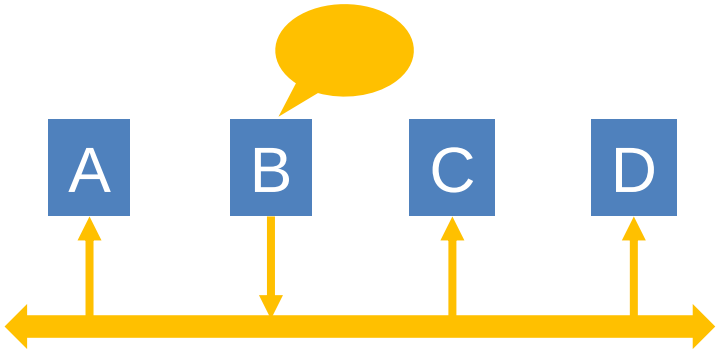
Altes Übungsblatt

- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.
  - a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?



Altes Übungsblatt

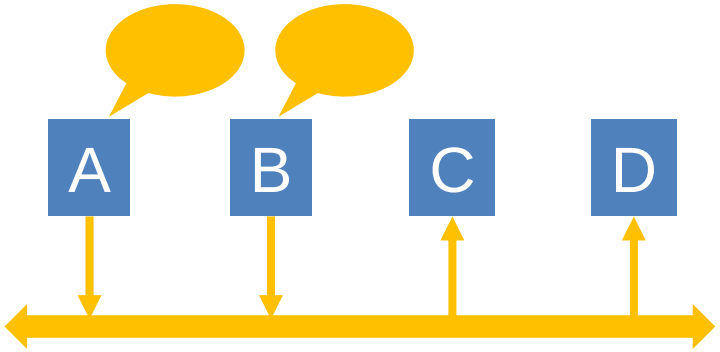
- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.
  - a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?



→ alle

Altes Übungsblatt

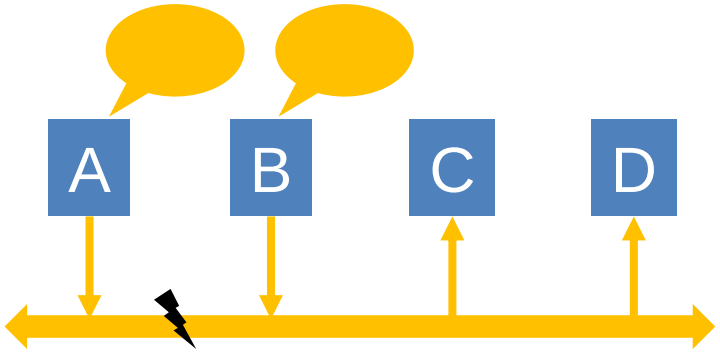
- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.
  - a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?
  - b) Rechner A möchte gleichzeitig Daten an C senden. Ist das möglich?



→ alle

Altes Übungsblatt

- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.



  a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?

→ alle

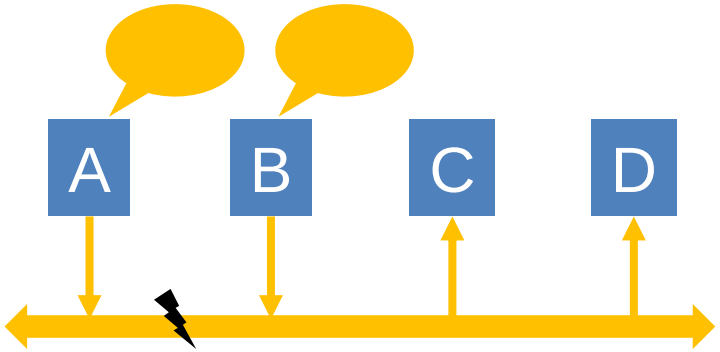
  b) Rechner A möchte gleichzeitig Daten an C senden. Ist das möglich?

→ nein, Kollision

  c) Was ist in diesem Beispiel die Kollisionsdomäne?

Altes Übungsblatt

- Aufgabe 6
  - Stellt eine Bustopologie mit den Rechnern A, B, C und D grafisch dar.



  a) Rechner B und D kommunizieren miteinander. Welche Rechner haben Zugriff auf die Daten?

→ alle

  b) Rechner A möchte gleichzeitig Daten an C senden. Ist das möglich?

→ nein, Kollision

  c) Was ist in diesem Beispiel die Kollisionsdomäne?

→ {A, B, C, D}

Altes Übungsblatt

- Aufgabe 7
  - IPv4-Adressen werden zur Zeit von IPv6-Adressen abgelöst, weil es nicht mehr genügend IPv4-Adressen gibt.
  - Wie viele IPv4- und IPv6-Adressen lassen sich jeweils maximal bilden?

Altes Übungsblatt

- Aufgabe 7
  - IPv4-Adressen werden zur Zeit von IPv6-Adressen abgelöst, weil es nicht mehr genügend IPv4-Adressen gibt.
  - Wie viele IPv4- und IPv6-Adressen lassen sich jeweils maximal bilden?
- Lösung
  - IPv4-Adressen bestehen aus $4 \cdot 8 = 32$ Bit.
  - Anzahl der IPv4-Adressen: $2^{32} = 4.294.967.296 \approx 4{,}3 \cdot 10^9$
  - IPv6-Adressen bestehen aus $8 \cdot 16 = 128$ Bit.
  - Anzahl der IPv6-Adressen: $2^{128} \approx 3{,}4 \cdot 10^{38}$

Informatik für die Kulturwissenschaften

- Vier Böcke:
  - 1. Digitale Informations-verarbeitung ✓
  - **2. Rechnersysteme und Softwareentwicklung**
  - 3. Datenbanken und Datenmodellierung
  - 4. Fachinformationssysteme
- Block 2:
  - LE 04: Rechneraufbau und -funktion
  - **LE 05: Betriebssystem und Rechnernetze**
  - LE 06: Software-entwicklung
  - LE 07: Algorithmisches Denken

Informatik für die Kulturwissenschaften

- Vier Böcke:
  - 1. Digitale Informations-verarbeitung ✓
  - **2. Rechnersysteme und Softwareentwicklung**
  - 3. Datenbanken und Datenmodellierung
  - 4. Fachinformationssysteme

- Block 2:
  - LE 04: Rechneraufbau und -funktion
  - **LE 05: Betriebssystem und Rechnernetze**
  - **Praxisübung Kommandozeile**
  - LE 06: Software-entwicklung
  - LE 07: Algorithmisches Denken

Teil 1
Linux und Terminal

Teil 2
Live-Übungen

Teil 3
Übungsblatt

Linux

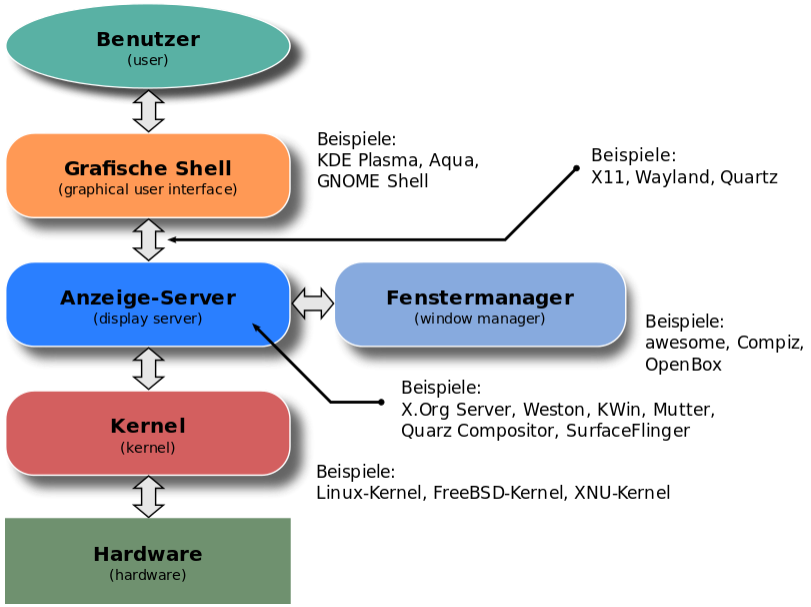
- GNU/Linux
  - Linux: Freier Betriebssystem-Kernel
  - GNU: Umgebung
- Entwicklung
  - Linus Torvalds: “I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386(486) AT clones”
  - heute meist bezahlte Entwickler

Warum Linux?

- Wirtschaftlich
  - freie Software (kostenfrei)
  - ressourcensparend
  - theoretisch unabhängig von Zulieferern
  - anpassbar
- Verbreitet in Web und Unterhaltungselektronik
  - Webspaces, (V-)Server
  - Netzwerkhardware
  - Smartphones
  - Set-top-Boxen
- Ideologisch
  - freie Software (Freiheit)
  - communitygetrieben
  - anpassbar, vielfältige Auswahlmöglichkeiten
- Visionen umsetzen
  - Tails (Privatsphäre und Anonymität)
  - Ubuntu („Linux for human beings“)
  - Hot Dog Linux (Retro-GUI)
  - Red Star OS (Überwachung)

Linux – Modularität

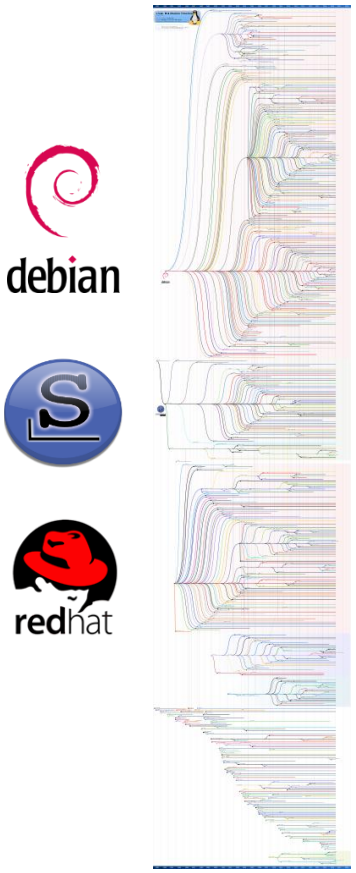
- Modulare Schichten
  - zwischen Hardware und Benutzer
- Distributionen
  - erleichtern Management durch Vorgaben und Paketierung



https://commons.wikimedia.org/wiki/File:Schema_der_Schichten_der_grafischen_Benutzeroberfläche.svg

Linux – Distributionen

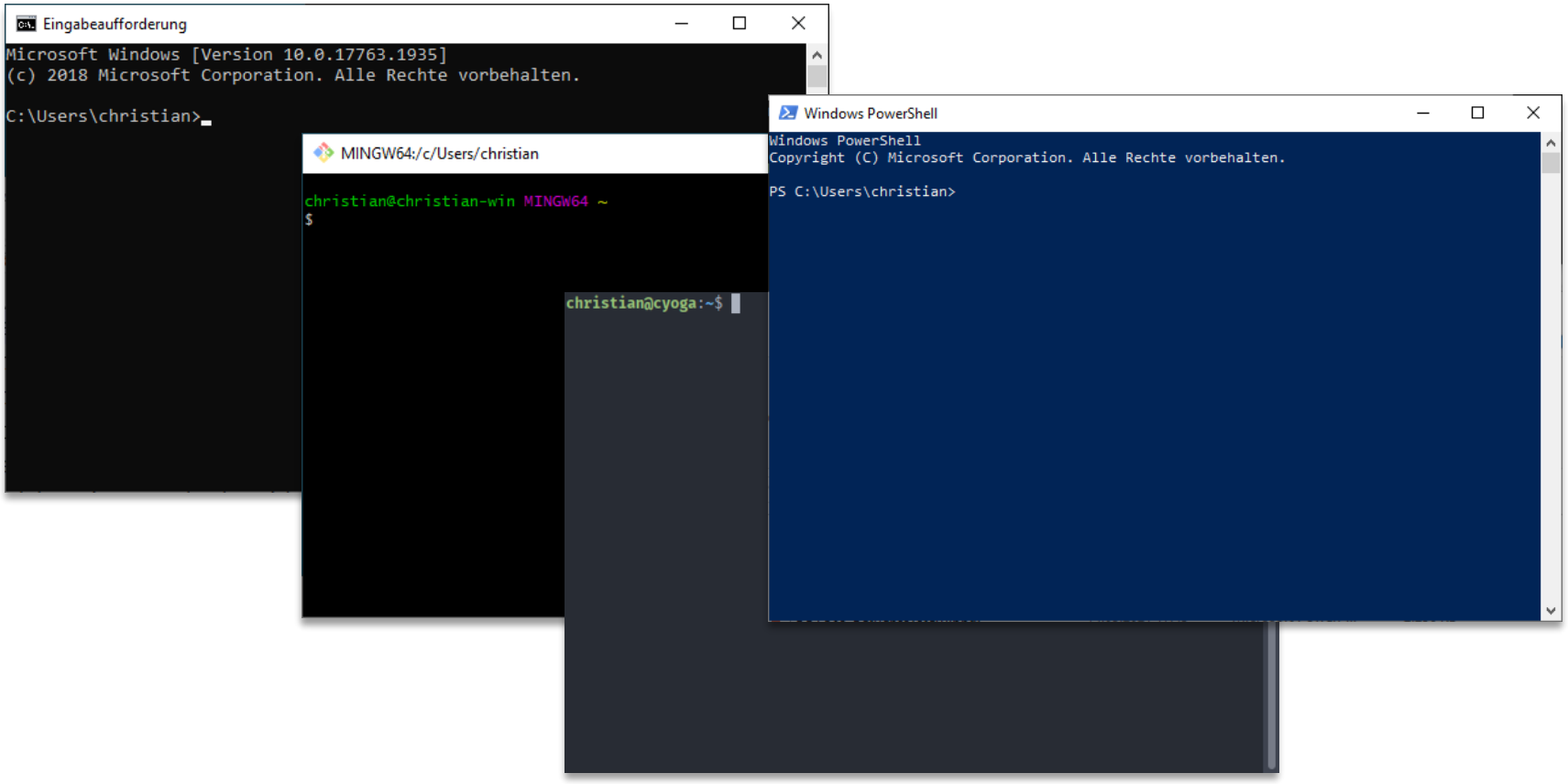
- Vorgefertigte Distributionen
  - meist mit Paketverwaltung inkl. Updates
  - verschiedene Ideen/Konzepte
- 3 große Familien
  - Debian (mit Ubuntu)
  - Slackware (mit SUSE)
  - Redhat (mit Fedora)
- 3 kleinere Familien
  - Gentoo, Arch, Android
- Unzählige Einzelprojekte



Linux auf dem Desktop

- Anzeigeserver
  - X-Server (1984-heute)
  - Wayland (2008-heute)
  - SurfaceFlinger (Android)
- Fenstermanager
  - Compiz, e17, i3, Mutter, …
- Desktopumgebung
  - Gnome, KDE, Mate, Cinnamon, Xfce, LXDE
- GUI-Toolkit
  - Gtk, Qt

https://upload.wikimedia.org/wikipedia/commons/d/dc/Gnome_Shell_40.1.0.png
https://upload.wikimedia.org/wikipedia/commons/2/24/XFCE-4.12-Desktop-standard.png
https://upload.wikimedia.org/wikipedia/commons/c/cd/Cinnamon_4.2.3_screenshot.png



Kommandozeile

Unix-Philosophie

- One thing well
  - Programme sollen nur eine Aufgabe erledigen, die aber gut
- Work together
  - Programme sollen zusammenarbeiten
- Text streams
  - Programme sollen Text als universelle Schnittstelle verwenden

- Beispielprogramme
  - `cd`: wechselt Verzeichnisse
  - `ls`: listet Verzeichnisinhalte
  - `echo`: gibt Text aus
  - `cat`: zeigt Inhalte an
  - `sort`: sortiert Text
  - `uniq`: löscht doppelte Zeilen
  - `diff`: zeigt Unterschiede
  - `curl`: HTTP-Client
  - `grep`: findet Text
  - `wc`: zählt Wörter oder Zeilen

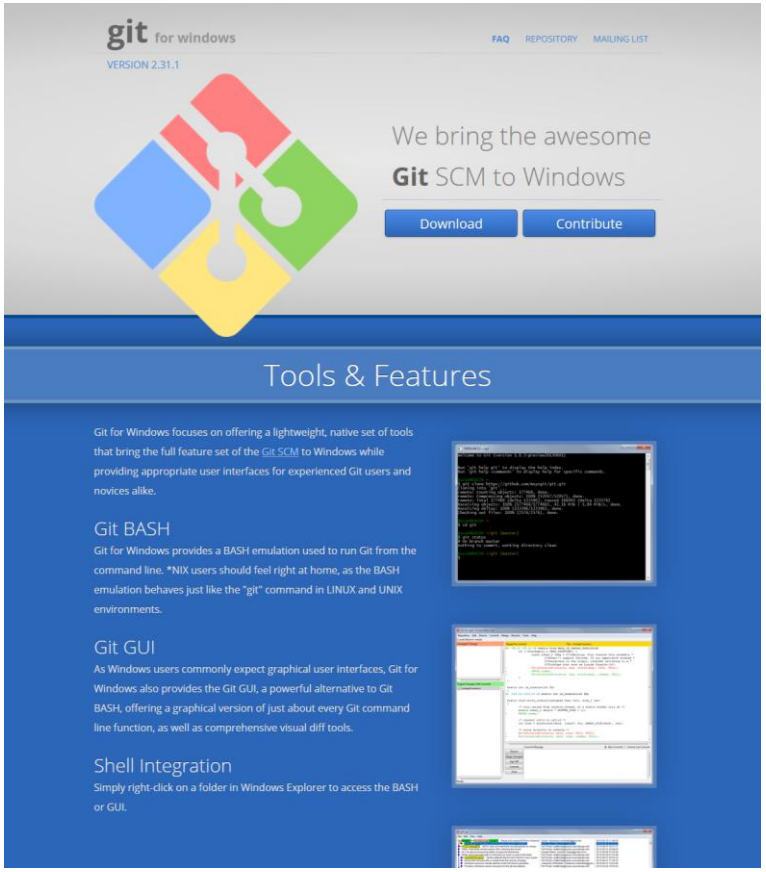
Teil 1
Linux und Terminal

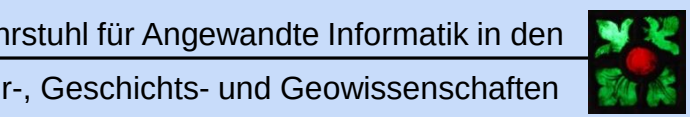
Teil 2
Bash-Praxis

Teil 3
Übungsblatt

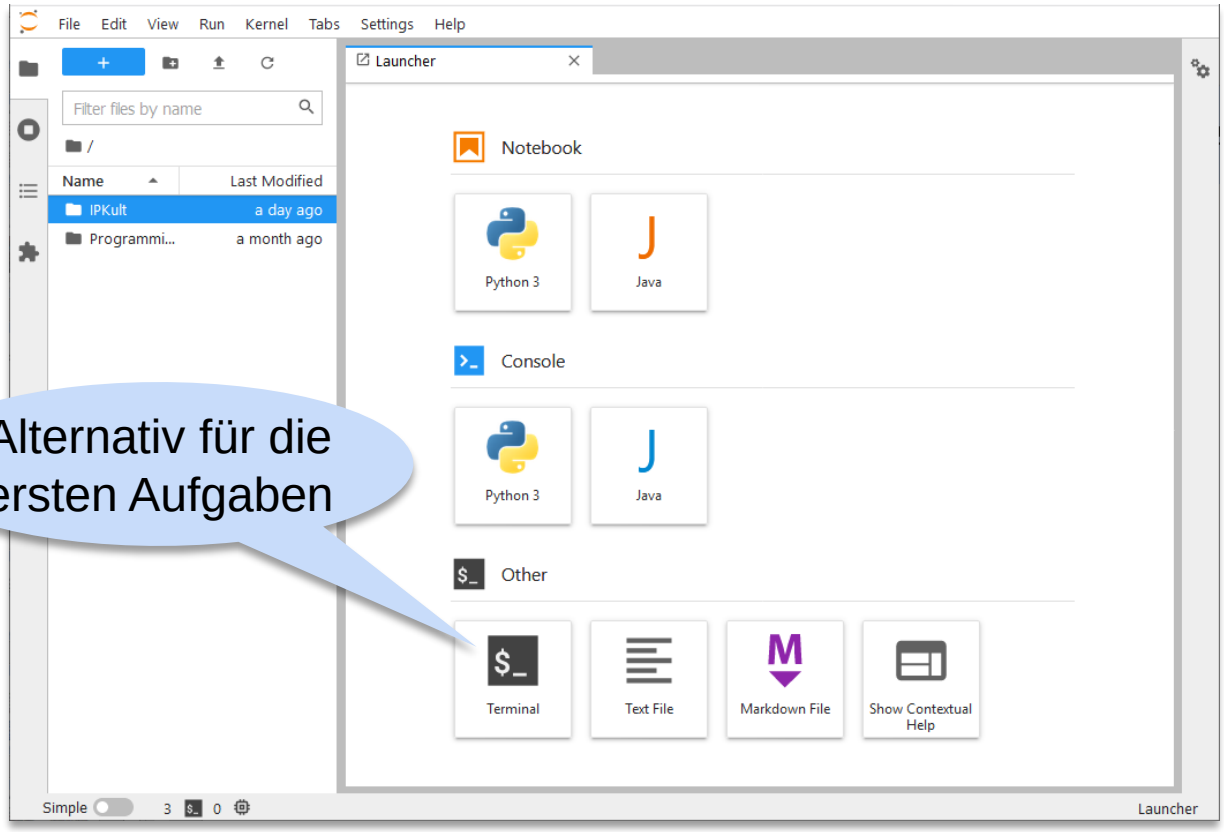
Git Bash

- Für Windows-Nutzer:
  - Ihr könnt unser Jupyter verwenden, und dort ein Terminal öffnen: https://jupyter.kinf.wiai.uni-bamberg.de
  - Zusätzlich solltet ihr bitte auch lokal die Git Bash installieren: https://gitforwindows.org/
- Für Mac-/Linux-Nutzer:
  - Ihr könnt einfach euer vorhandenes Terminal verwenden.





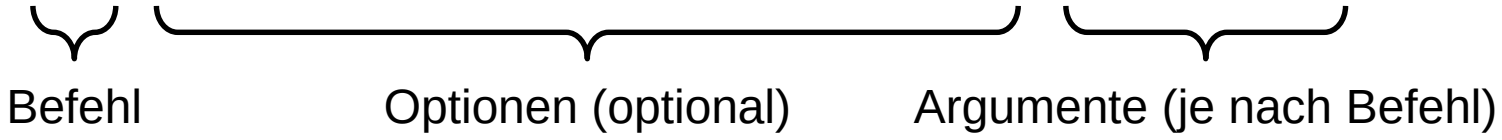
Jupyter



https://jupyter.kinf.wiai.uni-bamberg.de

Bash-Syntax

```
$ ls -l --all --human-readable /usr/bin
```



- Zu beachten
  - case sensitive
  - Leerzeichen relevant
- Befehl
  - `ls` listet die Einträge im gegebenen Verzeichnis auf
- Optionen
  - Langformen: `--all --human-readable`
  - Kurzformen: `-a -h`
  - Kurzformen zusammen: `-ah`

Piping

```
$ cat foo.txt | wc -w
```

Erster Befehl Pipe Zweiter Befehl

- Pipe
  - Zeichen: senkrechter Strich
  - Übergibt den Output des ersten Befehls als Input an den zweiten Befehl
  - Lässt sich auch mehrfach anwenden
- Befehle
  - `cat <datei>`
    gibt den Inhalt einer Datei aus
  - `wc -w`
    zählt die Wörter des übergebenen Texts

Pfade

- Pfade
  - Trennzeichen: `/`
  - Absolute Pfade beginnen mit `/` und starten immer im Wurzelverzeichnis
    - `/usr/bin/ls`
  - Relative Pfade beginnen nicht mit `/` und starten im aktuellen Arbeitsverzeichnis
  - Das aktuelle Verzeichnis heißt auch `.`
  - Das übergeordnete Verzeichnis heißt immer `..`
- Wildcards
  - Wenn Pfade ein Sternchen (`*`) enthalten, wird der Befehl mehrfach ausgeführt, wobei das Sternchen nacheinander durch alle möglichen Namen ersetzt wird.
  - Etwas präziser sind `{a,b,c}` und `{a..c}`, wobei nur die aufgeführten Ausdrücke eingesetzt werden (hier jeweils `a`, `b` und `c`).

Echo

```
$ echo foo
```

- Echo
  - gibt seinen Input als Output aus
  - Dabei werden aber zum Beispiel Wildcards angewendet:
    - `echo {5..55}`
    - `echo h{a..z}llo`
    - `echo h{a,e}llo`

Hilfe

- Manual pages
  - `man ls`
  - ausführliche Dokumentation
  - verlassen mit `q`
- Help
  - `ls --help`
  - kurze Übersicht
  - sofort neuer Prompt

```
LS(1)                          User Commands                          LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
Manual page ls(1) line 1 (press h for help or q to quit)
```

```
christian@cyoga:~$ ls --help
Aufruf: ls [OPTION]... [DATEI]...
Auflistung von Informationen über die DATEIen (Vorgabe ist das aktuelle
Verzeichnis). Alphabetisches Sortieren der Einträge, falls weder -cftuvSUX noch
--sort angegeben wurden.

Erforderliche Argumente für lange Optionen sind auch für kurze erforderlich.
  -a, --all                  Einträge, die mit . beginnen, nicht verstecken
  -A, --almost-all           implizierte . und .. nicht anzeigen
      --author               mit -l, den Urheber jeder Datei ausgeben
  -b, --escape               nicht-druckbare Zeichen im C-Stil ausgeben
      --block-size=GRÖßE     mit -l werden Größenangaben bei der Ausgabe mit
                               GRÖßE skaliert; Beispiel „--block-size=M“;
                               siehe GRÖßE-Format weiter unten
  -B, --ignore-backups       implizite Einträge, die mit ~ enden, nicht ausgeben
  -c                         mit -lt: Sortieren nach und Anzeige von ctime
                               (Zeit der letzten Veränderung der Datei-Status-
                               informationen); mit -l: ctime anzeigen, neueste
                               zuerst
  -C                         Einträge mehrspaltig ausgeben
      --color[=WANN]         Steuerung, wann Farbe zum Unterscheiden der
                             Dateitypen eingesetzt wird; WANN kann „always“
                               (immer; Voreinstellung wenn weggelassen),
                               „auto“ oder „never“ (nie) sein; mehr dazu weiter
```

Dateiverwaltung

- Ordner anlegen, Ordner wechseln
  - `mkdir <name>; cd <name>`
- Datei anlegen (oder Änderungsdatum setzen)
  - `touch <name>`
- Datei umbenennen / verschieben
  - `mv <quelle> <ziel>`
- Datei löschen
  - `rm <pfad>`

Spitze Klammern sind Platzhalter und müssen ersetzt werden.

Aufgabe: Dateiverwaltung

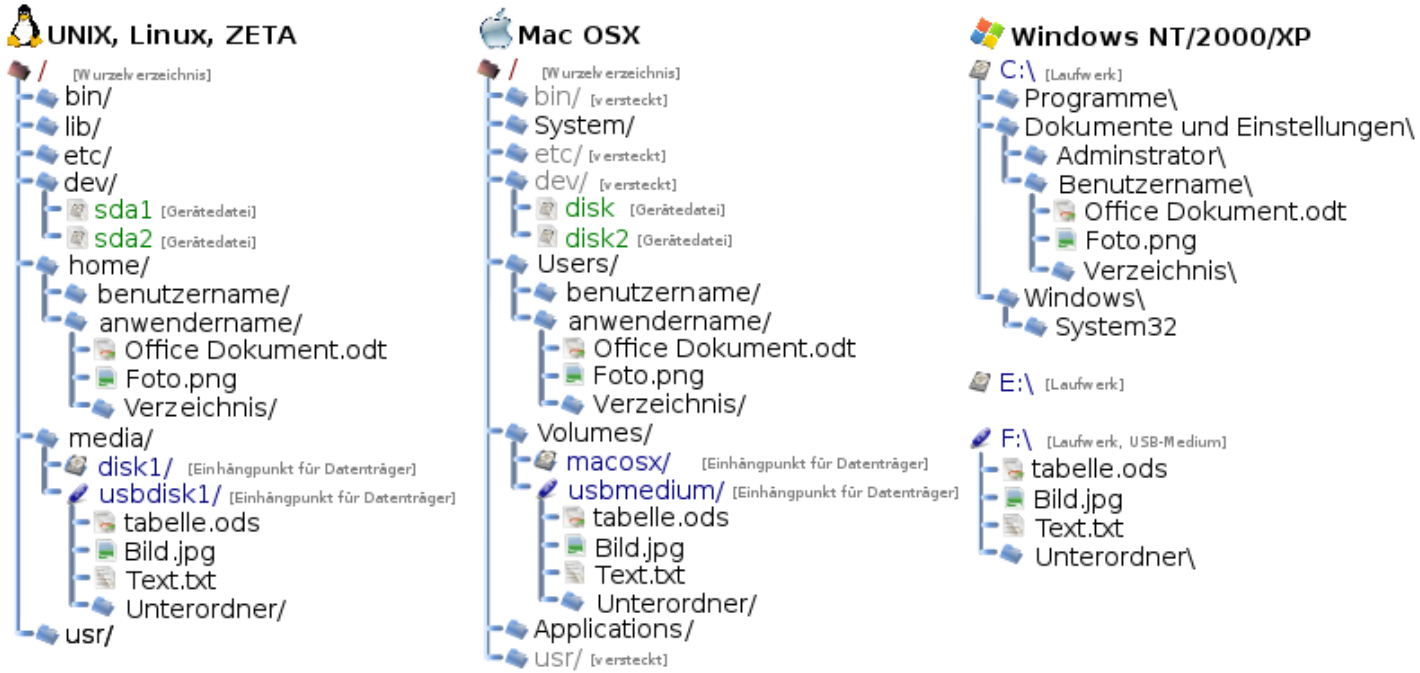
- Legt per Befehl eine Datei `test.txt` an. Lasst euch den aktuellen Ordnerinhalt auflisten, um das zu prüfen.
- Legt einen Ordner `temp` an.
- Verschiebt die Datei in den Ordner. Wechselt in den Ordner und lasst euch den Inhalt anzeigen.
- Löscht die Datei wieder.
- Führt im Ordner `temp` die folgenden Befehle aus. Was passiert? (Betrachtet das Ergebnis auch im Dateimanager)

```
mkdir -p IPKult/LE{1..14}
touch IPKult/LE{1..14}/{VL,Ü}.txt
```

Spitze Klammern sind Platzhalter und müssen ersetzt werden.

- Ordner anlegen, wechseln
  - `mkdir <name>; cd <name>`
- Datei anlegen (oder Änderungsdatum setzen)
  - `touch <name>`
- Datei umbenennen / verschieben
  - `mv <quelle> <ziel>`
- Datei löschen
  - `rm <pfad>`

Hierarchien im Vergleich



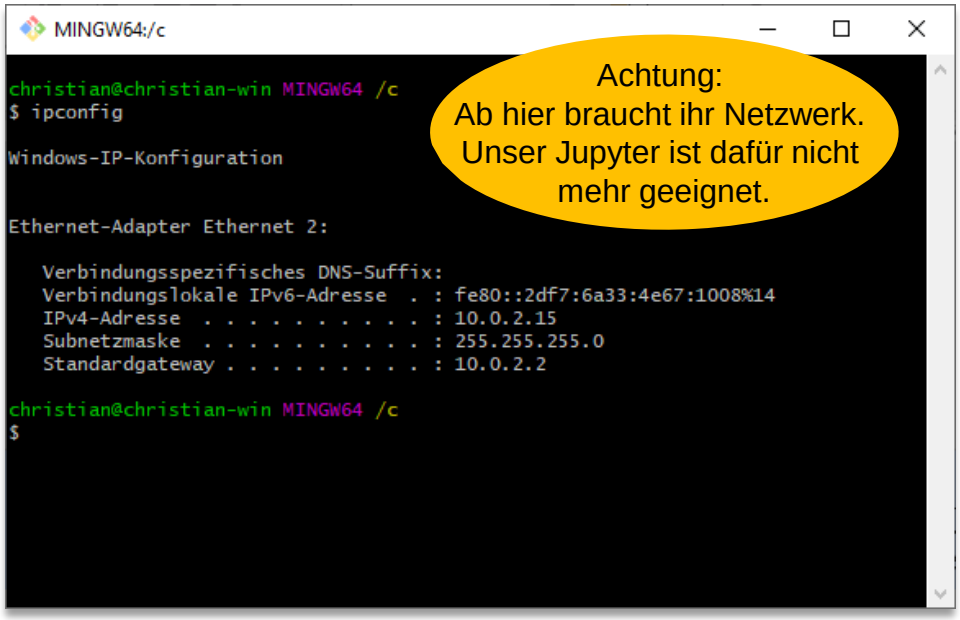
http://de.wikipedia.org/w/index.php?title=Datei:Filesystem.svg&filetimestamp=20100727221414&

Aufgabe: Textverarbeitung

- Führt der Reihe nach die folgenden Befehle aus und findet heraus, was sie tun:
  - `wget https://www.kinf.uni-bamberg.de/files/words.txt`
    - Falls `wget` nicht installiert ist, könnt ihr es mit `curl -O` (Großbuchstabe O) versuchen.
    - Falls ihr gerade keine Bash mit Internetzugriff habt, könnt ihr die Datei auch per Browser herunterladen und mit dem nächsten Befehl weitermachen.
  - `sort words.txt | uniq -c | sort -n`
  - `sort words.txt | uniq -c | sort -n | wc -l`
    - Wenn ihr den letzten Befehl mit der Cursor-hoch-Taste wieder anzeigt, braucht ihr nicht alles nochmal abtippen.
  - `sort words.txt | uniq -c | sort -nr > frequencies.txt`

IP-Adresse des eigenen Rechners

- Befehl
  - Windows: `ipconfig`
  - Linux: `ip a`
- Ausgabe
  - Zeigt für alle Netzwerkadapter die aktuellen Konfigurationswerte an
  - Darunter auch die IP-Adressen

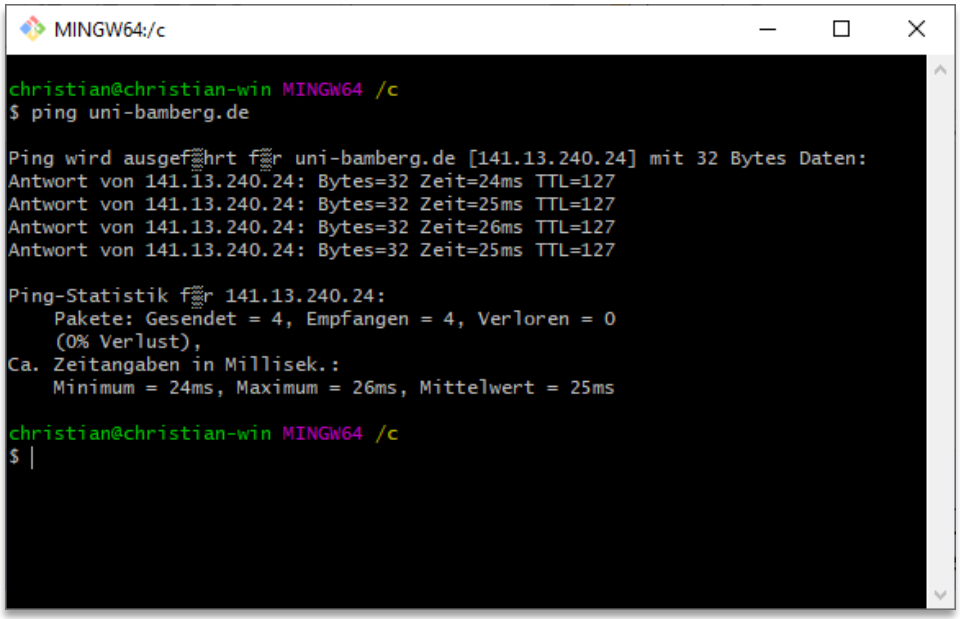


Kurz gefragt

- Einen Ping senden
  - Wie kann man über die Kommandozeile bei einem anderen Computer „anklopfen“?
  - Wozu könnte das gut sein?

Ping

- Befehl: `ping <ziel>`
  - Ein einzelnes Datenpaket (32 Byte) wird an den Zielrechner gesendet
  - Unter Windows viermal, unter Linux bis Abbruch (Strg+C)
  - Bei Zeitüberschreitung tritt ein Fehler bei der Datenübertragung auf
  - Zeigt die IP-Adresse des Zielrechners an



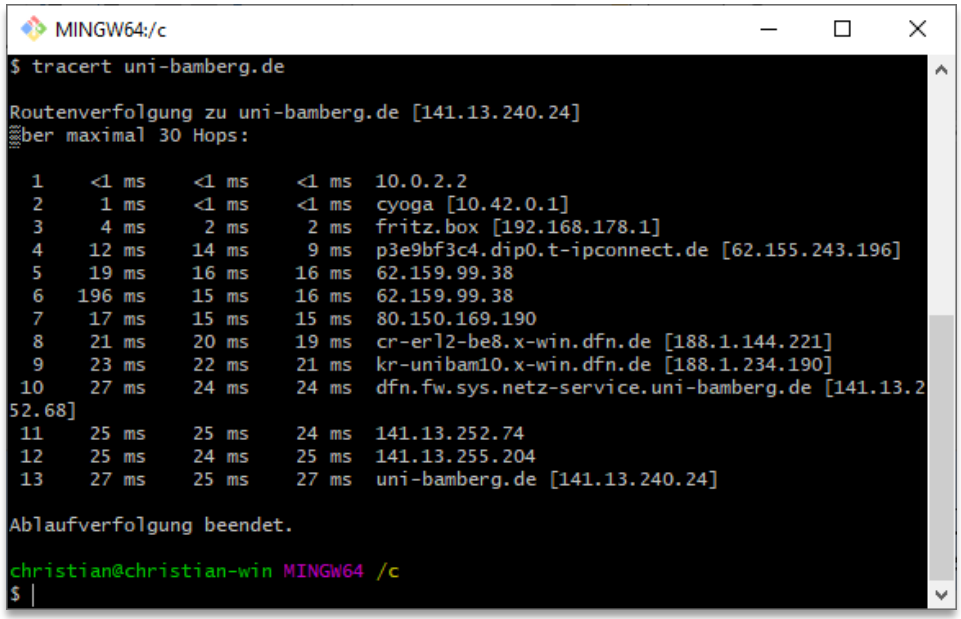
Aufgabe: Traceroute

- Aufgabe
  - Findet heraus, was der Befehl `tracert` (Windows) bzw. `traceroute` (Linux) macht.

- Aufgabe
  - Findet die IP-Adressen der folgenden Rechner heraus und analysiert die Netzwerkstruktur:
    - euer eigener Rechner
    - euer Router
    - www.uni-bamberg.de
    - kinf.uni-bamberg.de
    - www.fau.de
    - www.ckre.de

Traceroute

- Traceroute
  - Zeigt alle Hops zwischen dem eigenen Rechner und dem gegebenen Zielrechner an
  - Jeweils mit Latenzen, Namen und IP-Adressen



Linux auf dem Server

- Headless
  - Ohne direkte Ein-/Ausgabe, nur Strom und Netzwerk
- Wartung
  - erfolgt in der Regel automatisiert
- Zugang
  - per SSH übers Netzwerk

SSH

- SSH
  - Ein Netzwerkprotokoll bzw. die entsprechenden Programme, mit denen man sich über eine verschlüsselte Verbindung auf entfernten Geräten anmelden kann.
  - Anmeldebefehl:
    `ssh <user>@<host>`
- Aufgabe
  - Meldet euch (im VPN!) per SSH auf folgendem Host an:
    `praktomat.kinf.wiai.uni-bamberg.de`
  - Name: `ipkult{100..200}`
  - Pw: `ipkult Uebung %10`
  - Legt eine nach eurem Namen benannte Datei an.
  - Findet mit `ps aux` heraus, wie viele Prozesse gerade laufen (Bitte nicht selbst zählen!)

Teil 1
Linux und Terminal

Teil 2
Live-Übungen

Teil 3
Übungsblatt

Übungsblatt

- Aufgabe 1
  - Falls ihr die Datei words.txt noch nicht heruntergeladen habt, holt das von Folie 42* nach.
  - Der Befehl `grep <muster>` filtert aus einem übergebenen Textstream die Zeilen heraus, die `<muster>` enthalten.
  - Nutzt eure Bash-Kenntnisse, um alle Wörter aus `words.txt`, die ein „`a`“ enthalten, in eine neue Datei `words_with_a.txt` zu schreiben.
  - Was tut anschließend der folgende Befehl?

```
diff words.txt \
words_with_a.txt
```

* Falls die Folienreferenz mal wieder kaputt sein sollte, sucht bitte nach der Überschrift „Aufgabe: Textverarbeitung“.

Der Backslash am Zeilenende ermöglicht einen Zeilenumbruch im Befehl. Wenn ihr den ganzen Befehl einzeilig tippt, lasst den Backslash einfach weg.

Übungsblatt

- Aufgabe 2
  - Der Befehl `ping www.uni-bamberg.de` liefert folgende Ausgabe:

```
PING baurz24.urz.uni-bamberg.de (141.13.240.24) 56(84) Bytes Daten.
64 Bytes von baurz24.urz.uni-bamberg.de (141.13.240.24):
     icmp_seq=1 ttl=60 Zeit=24.8 ms
64 Bytes von baurz24.urz.uni-bamberg.de (141.13.240.24):
     icmp_seq=2 ttl=60 Zeit=47.1 ms
64 Bytes von baurz24.urz.uni-bamberg.de (141.13.240.24):
     icmp_seq=3 ttl=60 Zeit=47.0 ms
```

  - Welche Informationen könnt ihr daraus ableiten?

Übungsblatt

- Aufgabe 3
  - Verwendet `wget` (oder `curl -O`), um die Datei `https://www.kinf.uni-bamberg.de/files/archive.zip` herunterzuladen.
  - Nutzt dann `unzip`, um das Paket zu entpacken, lasst euch die darin enthaltene Datei anzeigen und folgt den Anweisungen.

Übungsblatt

- Zusatzaufgabe (schwierig!)
  - Habt ihr am Ende von Aufgabe 3 ein Passwort vermisst?
  - Die Datei `https://www.kinf.uni-bamberg.de/files/passwords.txt` hat die Antwort – aber leider nicht nur die richtige.
  - Befehl #1 unten führt den Befehl `echo` für jede Zeile der Datei `passwords.txt` aus.
  - Befehl #2 umgeht die interaktive Passwortabfrage beim Auspacken.
  - Kombiniert die Befehle so, dass der Rechner alle Passwörter für euch durchprobiert.

```
while read PASS; do echo "$PASS"; done < passwords.txt    # 1
unzip -P meinPasswort secret.zip                          # 2
```