

Pre Course, 12.03 09:00 – 12:00

Start eines Computers

- lokale Dateisystem mit Windows Dateien ist auf der Festplatte gespeichert.
- Beim starten lädt ein festes kleines Programm auf einem extra Speicher (bootloader) Teile des Betriebssystems von der Festplatte in den Arbeitsspeicher, damit der Prozessor damit rechnen kann und das eigentliche Betriebssystem arbeiten kann
- Die graphische Oberfläche kommt ganz zum Schluss und ist eigentlich nur Beiwerk
- Ein Betriebssystem kann auch komplette ohne grafische Oberfläche gesteuert werden (Kommandozeile)

Aufbau Computer

- Prozessor: Führt alle Berechnungen aus, besitzt meist mehrere Prozessorkerne, die parallel rechnen können.
 - *Wie viele Kerne hat mein Rechner?*
- Arbeitsspeicher/Ram: Flüchtig aber sehr schneller Zwischenspeicher für gerade benötigte Programme und Daten.
 - *Wie viel Megabyte Arbeitsspeicher hat mein Rechner? Wie viel davon sind zur Zeit schon belegt?*
- Festplatte: Langsamer aber dauerhafter Speicher für alle Benutzerdaten/ Dateisystem/ Betriebssystem.
 - *Wie viele lokale Festplatten hat mein Computer. Wie viele Gigabyte sind darauf?*

Dateisystem

- *Neuen Ordner "introduction-to-r" auf dem Desktop erstellen*
- Dateisystem verwaltet und strukturiert Daten auf der Festplatte für uns: Finden, speichern, umbenennen, löschen, verändern
- Dateien und Ordner haben immer einen festen logischen Dateipfad ausgehend von der Festplatte, auf der sie gespeichert sind
- In der Realität sind die Daten auf der Festplatte überall verstreut, aber das Dateisystem merkt sich zu jedem Dateipfad den realen Ort auf der Festplatten
- Der Desktop ist auch nur ein Ordner, den Windows aber besonders anzeigt
- *Wie lautet der Dateipfad zu dem neuen Ordner auf dem Desktop?*

Dateien

- Persistente Ansammlung an Information in Bits (0 und 1) gespeichert.
- Dateien haben einen Namen und eine Endung/ ein Dateiformat.
- *Wie kann ich die Dateieindung in Windows anzeigen lassen?*
- Mit der Dateieindung weiß Windows, mit welchem Programm es die Datei öffnen soll.
- *Was passiert, wenn ich die Dateieindung ändere?*
- Erst die Anwendung kann aus einer Datei für Menschen interpretierbare Informationen machen.
- Typische Klassen von Dateien: Ausführbare Dateien, Text, Audio, Bilder, proprietäre Dateien
- *Ladet den Titanic Datensatz von der Website herunter, erstellt einen neuen Ordner „data“ im Projekt und speichert den Datensatz dort ab. Welches Dateiformat besitzt die Datei? Handelte es sich um eine Text, ausführbare oder proprietäre Datei?*

Was ist eine Programmiersprache?

- Abstrakte Rechenvorschriften, die von einem Computer ausgeführt werden können. Liefern immer das gleiche Ergebnis (in der Regel).
- Sprache definiert eine bestimmte Syntax (Grammatik) und Semantik (Bedeutung): Anweisung nach vorgegebenem Muster
- Maschinensprache: Grundlegende sehr schwer lesbare und komplizierte Sprache, die direkt von einem bestimmten Prozessor verstanden wird. Letzendlich müssen alle Befehle in Maschinensprache übersetzt werden.
- Betriebssysteme werden auch in einer bestimmten Sprache programmiert.
- Höhere Sprachen: Für den Menschen einfach zu lesende Sprachen. Werden in Quelltexten (Textdateien) festgehalten.
- Höhere Sprachen können nicht direkt von einem Computer ausgeführt werden, da sie viel zu abstrakt sind. Ein Compiler ist ein bestimmtes Programm, welches die Programmiersprache in Maschinensprache übersetzen kann.

Wie funktioniert programmieren?

- Festlegen auf eine Programmiersprache und einen Compiler (zB Java, Python oder R)
- Schreiben von Quellcode in eine Textdatei. Regeln der Programmiersprache müssen befolgt werden.
- Compilervorgang starten und Programm ausführen.
- In R wird zur Laufzeit compiliert (Interpreter): der Code kann direkt in eine Konsole geschrieben werden, wird direkt in Maschinensprache übersetzt und das Ergebnis wird angezeigt.

Programmieren in R

- R ist eine Programmiersprache mit bestimmten Regeln
- Wenn man R auf einem Computer installiert, dann installiert man den *Interpreter*, d.h. das Programm, welches R Code als Text entgegennimmt, übersetzt und das Ergebnis wieder zurückliefert
- *Öffne die Kommandozeile in Windows und starte den Interpreter für die Programmiersprache R. Führe einfache Berechnungen durch.*
- Zwei Arten der Bedienung:
- Interaktiv: R Code wird direkt in die Konsole geschrieben und übersetzt
- Skript: R Code wird in Textdateien festgehalten und dann mit einem extra Befehl übersetzt
- Quellcode in R: Textdateien mit der Endung `.R`
- *Erstelle einen Ordner "source" im Projektordner und erstelle darin eine neue Datei "helloWorld.R". Öffne die Datei mit einem Texteditor*

RStudio, eine IDE für R

- RStudio ist ein zusätzliches Programm, welches die Arbeit mit der Programmiersprache und dem Interpreter vereinfacht
- RStudio wird nicht für das Ausführen von R Code benötigt! Es erleichtert nur die Bedienung.
- Verwaltung von Dateien und Quellcode in Projekten
- Texteditor mit Syntaxvervollständigung und Tooltips
- Integrierte Konsole
- Integriertes Anzeigen von Plots (graphische Ausgabe)

- *Öffne RStudio und erstelle ein neues Projekt im bestehenden Projektordner*
- Einzelne Projekte in RStudio sind in einem zentralen Ordner organisiert. Dieser Ordner enthält eine Text-Datei mit der Endung `.RProj`
- Projekte können auch direkt mit einem Klick auf die Projektdatei geöffnet werden.

Relative Dateipfade

- Anstatt volle Dateipfade anzugeben, können auch relative Dateipfade verwendet werden.
- `./`, `../`, `~/`
- *Vervollständige Aufgaben 1.1 und 1.2*

Netzlaufwerke des RRZK

- Jeder Nutzer (Student/Mitarbeiter) besitzt ein persönliches privates Laufwerk an der Uni.
- Wie eine Dropbox. Man kann von überall, auch von zu Hause, darauf zugreifen.
- Das Projekt ist zur Zeit nur lokal auf dem Rechner gespeichert. Wenn wir es im Netzlaufwerk abspeichern, können wir morgen wieder darauf zugreifen.
- Arbeitsablauf: Projektordner vom Netzlaufwerk auf den Desktop ziehen. Arbeiten im Projekt. Projektordner wieder zurückziehen. Vorteil von relativen Dateipfaden!