

Vernetzung und Betrieb

Netzwerkkommunikation

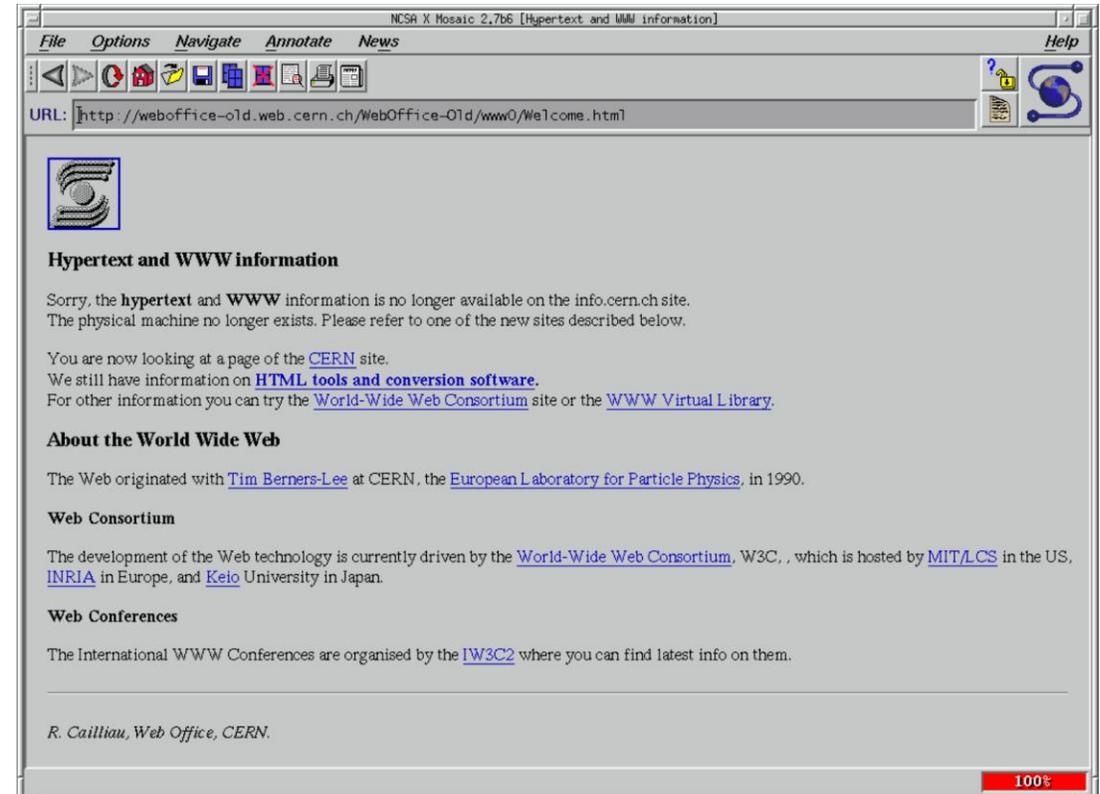
- Fehlererkennende und –korrigierende Codes helfen bei der korrekten Übertragung von Bitfolgen zwischen einem Sender und einem Empfänger.
- Aber was hat das eigentlich mit dem zu tun, woran viele Endnutzer bei dem Begriff „Computer-Netzwerk“ zuerst denken – dem Internet?
- Wir haben bereits gesehen, dass man Medien wie Texte, Bilder, Audio, etc. als Bitfolgen kodieren kann.
- Wenn man also Bitfolgen übertragen kann, kann man auch andere Inhalte übertragen, wenn diese entsprechend kodiert sind.

Das Internet

- Das Internet ist ein globales Netzwerk von Computern.
- Auf Grundlage der technischen Möglichkeit, Daten zwischen zwei Computern auszutauschen, existieren eine Reihe von Anwendungen.
 - Email
 - Dateitransfer (FTP)
 - Secure Shell (SSH) für verschlüsselte Verbindungen
 - etc.
- Im normalen Sprachgebrauch wird der Ausdruck „Internet“ aber oft synonym mit einer bestimmten Anwendung verwendet: dem World-Wide-Web (WWW).

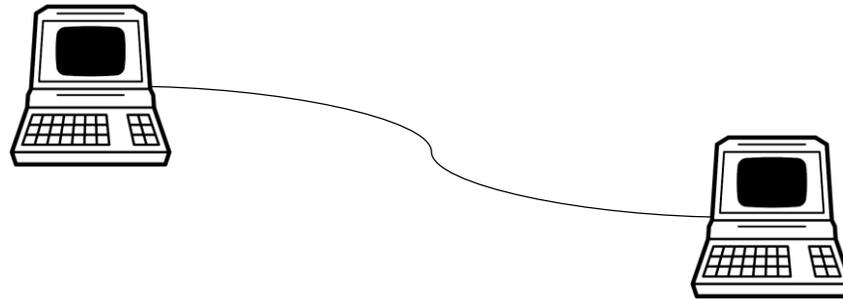
Informationsdarstellung im WWW

- Der Zugang eines Benutzers zum WWW erfolgt über einen Webbrowser (z.B. Google Chrome, Apple Safari, Mozilla Firefox,...)
- Die grundlegende Interaktion ist das Anklicken von *Links* auf *Webseiten*.
- Doch wie kommen diese Webseiten in den Browser? Und wie sind die dann dargestellten Inhalte repräsentiert?



Verbindungen zwischen Rechnern

- Im einfachsten Fall ist ein einziger Rechner mit einem weiteren verbunden.



- Damit ein Datenaustausch zwischen diesen erfolgen kann, müssen beide „die selbe Sprache“ sprechen.

Protokolle

- Eine (Kommunikations-)Protokoll ist eine gemeinsame Vereinbarung darüber, wie Information repräsentiert wird, sodass eine Kommunikation zwischen Sender und Empfänger stattfinden kann.
- Im Internet sind immer mehrere standardisierte Protokolle auf unterschiedlichen Ebenen im Einsatz, z.B.:
 - Spezifikation von Kabeln, Volt-Werten, Taktfrequenzen, etc. auf der untersten (physikalischen Ebene)
 - Spezifikation von fehlererkennenden und –korrigierenden Maßnahmen auf der nächsten Ebene etc.

Hypertext

- Das WWW war ursprünglich für den Austausch einfacher Hypertext-Dokumente konzipiert.
- Hypertexte zeichnen sich gegenüber anderen Texten dadurch aus, dass sie speziell für das Lesen mittels IT-Technologie gedacht sind. Sie enthalten Verweise (Hyperlinks) auf andere Dokumente, die durch die Anzeigesoftware unmittelbar aufgerufen werden können.
- Für diesen Austausch wurde ebenfalls ein spezielles Protokoll entworfen, das HyperText Transfer Protocol (HTTP).

HTTP

- Die zugrundeliegende Annahme bei HTTP ist, dass eine Kommunikationsseite Hypertext-Dokumente vorhält, die von der anderen Seite angefragt werden können.
- Dazu stellt die anfragende Seite (genannt Client) zunächst eine Verbindung zur anbietenden Seite (genannt Server) her und schickt über diese dann eine Anfrage.
- Der Server schickt daraufhin ein Dokument an den Client oder – falls dies nicht möglich ist – antwortet mit einem vordefinierten Fehlercode.
 - Beispiel: 404 not found, falls das angefragte Dokument nicht existiert.

Anfragemethoden

- Der Client teilt dem Server sein Anliegen in Form einer bestimmten Nachricht mit, die im Englischen “request” genannt wird.
- Die beiden wichtigsten sind *get* (Aufforderung an den Server, ein bestimmtes Dokument zu senden) und *post* (Senden von Daten an den Server).

```
GET /index.html HTTP/1.1
```

```
POST /index.html HTTP/1.1
```

```
Host: example.com
```

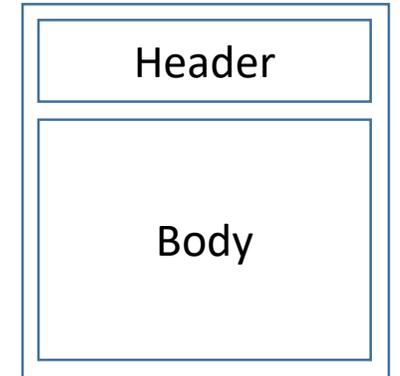
```
Content-Length: 16
```

```
Hi, how are you?
```

- Das Format der Anfrage und der Antworten ist im HTTP-Protokoll definiert.

Hypertext Markup Language (HTML)

- Hypertext-Dokumente im WWW sind in dem HTML-Standard verfasst und bestehen aus zwei Teilen.
 - Header: Metadaten wie Titel, Character-Encoding, etc.
 - Body: Das eigentliche Dokument.
- Neben dem reinen Textinhalt erlaubt HTML auch die Angabe weiterer Metadaten direkt im Text durch sog. Markup-Tags.
 - Beispiel Fettdruck: *Tags stehen in `spitzen` Klammern.*
- Manche Tags werden als Beginn-/Ende-Paare spezifiziert, andere nicht.



Beispiel für ein HTML-Dokument.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Beispiel-Dokument</title>
  </head>
  <body>
    <h1>Beispiel</h1>
    Dies ist ein <b>kurzes</b> Dokument. <p>
    Auf Wiedersehen.
  </body>
</html>
```

Darstellung im Webbrowser:

Beispiel

Dies ist ein **kurzes** Dokument.

Auf Wiedersehen.

Standardisierte Elemente

- Der HTML-Standard spezifiziert alle erlaubten Tags und wo sie erlaubt sind.
- Semantisches Markup: spezifiziert die Struktur des Dokuments (Überschriften, Abschnitte, etc.) und die Elemente, aus denen es besteht (Tabellen, Bilder, Texte, etc.).
- Layout-Markup: bezieht sich rein auf die visuelle Darstellung und steuert diese (Fettdruck, Kursivdruck, Zeilenumbrüche, etc.)
- Manche Tags erlauben die Spezifikation gewisser Attribute, z.B. das `<meta>`-Element im Header: `<meta charset="utf-8">`

Attributname Wert

HTML + CSS + JavaScript = modernes WWW

- Die Ansprüche und Erwartungen an Webseiten sind seit den Anfangstagen immer weiter gewachsen.
- Der HTML-Standard wurde und wird kontinuierlich erweitert.
 - Mit der Programmiersprache JavaScript können Inhalte und Darstellung dynamisch manipuliert werden.
 - Mit der StyleSheet-Sprache CSS erfolgt eine bessere Trennung zwischen Inhalt- und Layoutbeschreibungen.

Cascading Style Sheets (CSS)

- HTML-Dokumente beziehen oftmals eine externe Datei mit Layout-Informationen der verschiedenen Elemente des Dokuments mit ein.
- Aus Layoutsicht werden die verschiedenen Elemente dabei grundlegend als (teils geschachtelte) Rechtecke angesehen, die mit Attributen bzgl. ihrer Darstellung versehen sind.
- CSS erlaubt es, die Werte dieser Attribute zu spezifizieren.

```
h1 {  
    font-family: TimesNewRoman;  
    color: black;  
    margin-bottom: 20px;  
}
```

Beispiel für ein HTML+CSS

index.html

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Beispiel-Dokument</title>
    <link rel="stylesheet" href="mystyle.css">
  </head>
  <body>
    <h1>Beispiel</h1>
    Dies ist ein <b>kurzes</b> Dokument. <p>
    Auf Wiedersehen.
  </body>
</html>
```

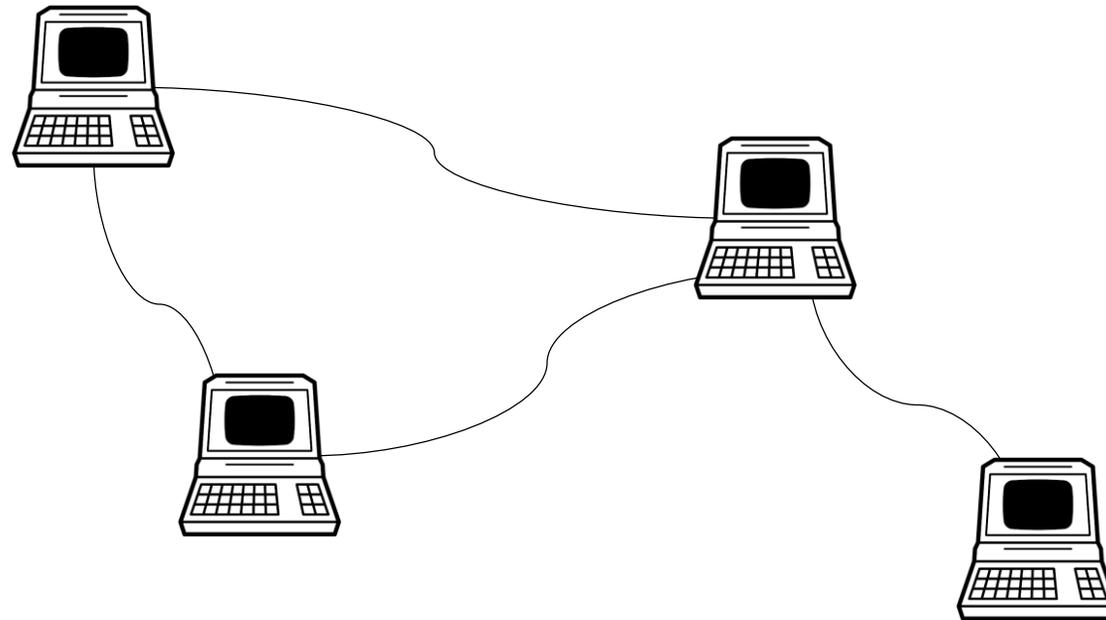
mystyle.css

```
body {
  padding: 10px;
}

h1 {
  font-family: TimesNewRoman;
  color: black;
  margin-bottom: 20px;
}
```

Verbindungen zwischen Rechnern

- Im einfachsten Fall ist ein einziger Rechner mit einem weiteren verbunden
- Wie kann ein Client einen Server anfragen, mit dem er nicht direkt verbunden ist?



Paket-basierte Kommunikation

- Der Datenaustausch im Internet ist Paket-basiert: Bitfolgen werden in bestimmte Einheiten aufgeteilt (Datenpakete) und diese unabhängig voneinander versendet. Der Empfänger fügt sie wieder zusammen.
- Damit ein Paket den Weg vom Client zum Server finden kann, muss es an den Zwischenstationen geeignet weitergeleitet werden.
- Das übernehmen spezielle Systeme, die Router genannt werden.
- Sie sind oftmals mit mehreren anderen Routern verbunden.

Routing

- Die Datenpakete enthalten Zielinformationen in Form einer IP-Adresse. Dies ist eine 32-Bit-Zahl, die typischerweise als vier separate Bytes angegeben wird, z.B. 192.176.0.2.
- Der Router entscheidet anhand einer Routing-Tabelle, wohin er das Paket weiter schickt.
- Die Einträge in der Routing-Tabelle geben dabei an, wohin ein Paket abhängig von der enthaltenen Zieladresse weitergeleitet werden soll.
- So „hüpft“ das Paket von Router zu Router bis zum Ziel.

Routingtabelle

1) Network Destination	2) Netmask	3) Gateway	4) Interface	5) Metric
101.25.67.0	255.255.255.0	192.168.2.1	eth0	1
default	0.0.0.0	127.0.0.1	eth0	0
192.25.67.0	255.255.255.0	192.168.2.0	eth0	10

- Zunächst berechnet der Router das bitweise AND zwischen Zieladresse und Spalte 2).
- Das Ergebnis wird mit Spalte 1) verglichen: ist es identisch, so erfolgt die Weiterleitung über den Datenausgang in Spalte 3), welcher durch die Hardware aus Spalte 4) realisiert ist.
- Spalte 5) gibt eine Kostenmetrik für die Weiterleitung an, der Router versucht, die günstigste Route zu finden.

• Beispiel: $101.25.67.24 \wedge 255.255.255.0 = 101.25.67.0 \rightarrow 192.168.2.1$

Zieladresse des Datenpakets Netmask aus Zeile 1 entspricht Spalte 1, Zeile 1 daher Weiterleitung via Gateway Zeile 1

Domain Name System (DNS)

- IP-Adressen wie 192.168.2.1 sind zwar für Rechner hübsch, für Menschen aber nicht besonders nutzerfreundlich.
- Um dieses Problem zu lösen, wurde ein hierarchisch organisiertes Benamungsschema entwickelt.
- Ein Domänen-Name ist ein nach einem bestimmten Schema aufgebauter Name für eine IP-Adresse.
 - z.B. www.uni-saarland.de
- Um die zu einem solchen Domänen-Namen zugehörige IP-Adresse zu finden, kann man spezielle Internetserver anfragen (Domain Name Server)

Aufbau von Domännennamen

- 1. Top-Level Domänen: die Endung eines Domänennamens
 - Generisch: .com, .edu, .org, .info, etc.
 - Länderspezifisch: .de, .fr, .ie, etc.
- 2. Ebene: der eigentliche Name
- 3. Ebene: Unterdomänen
- Die Anordnung erfolgt durch Punkte getrennt von rechts nach links:
www.google.com

Zusammenfassung Informationsdarstellung im Internet

- Das World-Wide-Web wurde zum Austausch von Hypertext-Dokumenten zwischen einem Client (in Form eines Webbrowsers) und einem Web-Server konzipiert.
- Als Kommunikationsgrundlage dient das HTTP-Protokoll.
- Informationen werden mit den HTML und CSS-Standards dargestellt.
- Der Informationsfluss in dem global verteilten Internet wird mittels Routing realisiert.

Auffinden von Informationen im WWW

- Bei Billionen von Webseiten könnte das Auffinden gewünschter Informationen sprichwörtlich zur Suche nach der Nadel im Heuhaufen werden.
- In den Anfangszeiten des gab es Bestrebungen, die verschiedenen Webseiten systematisch zu kategorisieren, dies war aber schnell nicht mehr praktikabel.
- Stattdessen etablierten sich „Suchmaschinen“, also spezielle Webseiten, die eine Schlagwortsuche zum Finden relevanter Webseiten erlaubten.

Information Retrieval

- Bereits vor dem WWW war die Suche in großen Dokumentensammlungen eine wichtige Herausforderungen, dem sich das Gebiet des „Information Retrieval“ widmet.
- Die klassische Aufgabe besteht darin, zu einer gegebenen Menge von Schlagwörtern diejenigen Dokumente der Sammlung zu finden, die diese enthalten.
- Sämtliche Dokumente bei jeder neuen Anfrage nach den angefragten Wörtern zu durchsuchen, dauert zu lange. Stattdessen wird in einem Vorverarbeitungsschritt ein sog. Index aufgebaut.
- Dazu ist ein wichtiger Schritt, die eine Suchmaschine leisten muss, das sog. *Crawlen* von Internetseiten, also das Auffinden und Abspeichern derselben.

Dokumentenindizierung

- **Definition.** Sei D eine Sammlung von Textdokumenten und V die Menge aller darin enthaltene Term. Ein *Index* ist eine Abbildung $I: V \rightarrow 2^D$, die jedem Term $w \in V$ die Teilmenge derjenigen Dokumente von D zuordnet, die w enthalten.
- Dabei entspricht der Begriff Term im Wesentlichen Wörtern, kann aber auch andere Folgen von Buchstaben, Ziffern oder Sonderzeichen umfassen.
- Um einen Dokumentenindex aufzubauen, benötigt man ein Verfahren, ein Dokument in die in ihm enthaltenen Terme aufzusplitten. Dies nennt man Term-Extraktion oder auch Tokenisierung.
- Zudem findet oft noch eine Term-Normalisierung statt, z.B. Reduktion auf den Wortstamm oder eine Konvertierung nach Kleinbuchstaben.

Beispiel

- D bestehe aus folgenden Dokumenten:
 - $d_1 =$ Ich bin Sam.
 - $d_2 =$ Sam und ich sind weg.
 - $d_3 =$ Der Weg zu mir und dir ist lang.
- Daraus könnte z.B. der folgende Index I erstellt werden:

der	$\mapsto \{d_3\}$	lang	$\mapsto \{d_3\}$	und	$\mapsto \{d_2, d_3\}$
du	$\mapsto \{d_3\}$	sein	$\mapsto \{d_1, d_2, d_3\}$	weg	$\mapsto \{d_2, d_3\}$
ich	$\mapsto \{d_1, d_2, d_3\}$	sam	$\mapsto \{d_1, d_2\}$	zu	$\mapsto \{d_3\}$

Suchanfragen

- Eine Suchanfrage (engl. *query*) besteht im einfachsten Fall aus einer Menge von Termen, die in den Dokumenten enthalten sein sollten.
- Auf die Query wird dieselbe Term-Extraktion (und ggf. Term-Normalisierung) angewandt wie bei der Erstellung des Index.
- Das Ergebnis besteht aus der Schnittmenge der im Index für jeden Suchterm abgebildeten Teilmenge von D .
- Im obigen Beispiel würde die Suchanfrage “Ich weg” dann zu folgendem Ergebnis führen:

$$I(\text{ich}) \cap I(\text{weg}) = \{d_1, d_2, d_3\} \cap \{d_2, d_3\} = \{d_2, d_3\}$$

Boole'sche Anfragen

- Die Boole'schen Operatoren können dazu verwendet werden, komplexere Anfragen zu formulieren.
- Beispiel: $(\text{jaguar} \wedge \sim\text{tier}) \vee \text{bentley}$
- Bei heutigen Internet-Suchmaschinen ist dieses Feature aber zur Vereinfachung der Benutzerschnittstelle weitestgehend abgeschafft worden.
- Stattdessen sind meist nur Positiv- und Negativlisten von Termen, die in den gefundenen Dokumenten enthalten bzw. nicht enthalten sein sollten, erlaubt.

Ranking der Suchergebnisse

- Nicht alle Dokumente, die die Suchterme enthalten, sind als Ergebnis gleich gut. Statt einer Menge werden die Dokumente daher typischerweise nach Relevanz sortiert (*Ranking*).
- Ein weit verbreiteterer Ansatz basiert dabei auf folgender Überlegung:
 - Kommt ein Term t in einem Dokument d besonders häufig vor, ist d in Bezug auf t relevanter als ein anderes Dokument d' , in dem t seltener vorkommt.
 - Kommt ein Term t in sehr vielen Dokumenten vor, hat er weniger Aussagekraft als ein Term t' , der nur in wenigen Dokument vorkommt.

Formalisierung der Überlegung

- Term-Frequency (tf)
 - Besteht ein Dokument d aus n Termen, wovon der Term t k -mal vorkommt, so ist $tf(t, d) = \frac{k}{n}$
- Inverse Document Frequency (idf)
 - Besteht eine Dokumentsammlung D aus N Dokumenten, wobei der Term t in genau m Dokumenten in D vorkommt, so ist $idf(t, D) = \log\left(\frac{N}{m}\right)$
- Die Relevanz eines Dokuments $d \in D$ für einen Suchterm t ist dann:
$$tfidf(t, d) = tf(t, d) \cdot idf(t, D)$$

Beispiel

- Index:

der $\mapsto \{d_3\}$
du $\mapsto \{d_3\}$
ich $\mapsto \{d_1, d_2, d_3\}$
lang $\mapsto \{d_3\}$
sein $\mapsto \{d_1, d_2, d_3\}$
sam $\mapsto \{d_1, d_2\}$
und $\mapsto \{d_2, d_3\}$
weg $\mapsto \{d_2, d_3\}$
zu $\mapsto \{d_3\}$

t	$tf(t, d_1)$	$tf(t, d_2)$	$tf(t, d_3)$	$idf(t, D)$
der	0	0	1	$\log\left(\frac{3}{1}\right) = \log(3)$
du	0	0	1	$\log(3)$
ich	1	1	1	$\log\left(\frac{3}{3}\right) = 0$
lang	0	0	1	$\log(3)$
sein	1	1	1	0
sam	1	1	0	$\log\left(\frac{3}{2}\right)$
und	0	1	1	$\log\left(\frac{3}{2}\right)$
weg	0	1	1	$\log\left(\frac{3}{2}\right)$
zu	0	0	1	$\log(3)$

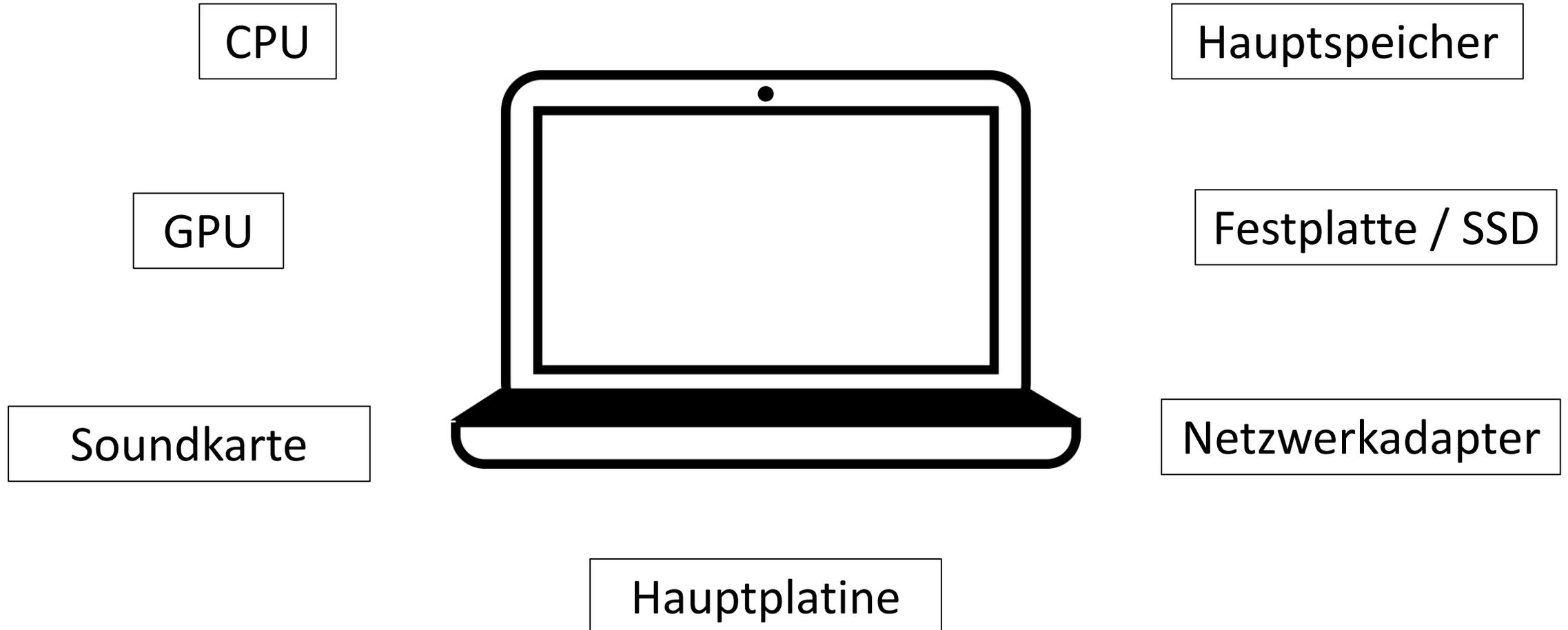
PageRank

- Ende der 1990er-Jahre revolutionierte ein kleines 2-Mann Startup im Silicon Valley die Branche der Internet-Suchmaschinen.
- Die grundlegende Erkenntnis, die eine rein Term-basierte Suche ergänzte, war das Ausnutzen der Hyperlink-Struktur im WWW.
- Idee:
 - Verweisen viele Dokumente per Link auf ein bestimmtes Dokument d , so kann d als besonders wertig angesehen werden.
 - Verweist ein besonders wertiges Dokument auf andere Dokumente, so werden diese dadurch ebenfalls aufgewertet.
- Google's PageRank Algorithmus bringt diese beiden Aspekte in Einklang und berücksichtigt die Wertigkeit eines Dokuments beim Ranking.

Zusammenfassung Suchmaschinen

- Webseiten müssen von Suchmaschinen zunächst „gecrawlet“, also gefunden und zur Weiterverarbeitung in eine Dokumentensammlung heruntergeladen werden.
- Offline kann ein Index erstellt werden, der Terme t auf Dokumente abbildet, die t enthalten.
- Wird eine Suchanfrage gestellt, können mittels Index relevante Dokumente gefunden werden.
- Das Ranking ist die Sortierung der Suchergebnisse nach Relevanz, womöglich unter Berücksichtigung der Hyperlinkstruktur (PageRank).

Hauptkomponenten eines Informatiksystems



Aufgaben der Hauptkomponenten

- CPU – Verarbeitung von Daten
- RAM – Speichern der Daten für die CPU
- Festplatte / SSD – langfristige Datenspeicherung auch ohne Strom
- GPU – Ansteuern des Bildschirm, Graphikberechnung
- Soundkarte – Erzeugen von Klängen
- Netzwerkadapter – Verbindung mit anderen Systemen zum Datenaustausch
- Hauptplatine – Integration der anderen Komponenten, Externe Anschlüsse

Von low-level Komponenten zum benutzbaren Gesamtsystem

- In den CPU-Beispielen bestand die einzige Möglichkeit, Einfluss auf die Verarbeitung von Daten zu nehmen, darin, bestimmte Werte in ein Register zu schreiben.
 - Wie dies in der Praxis vor sich geht, wurde nicht thematisiert.
- Unsere alltägliche Erfahrung in der Benutzung von Informatiksystemen sieht ohnehin ganz anders aus.
- Das liegt daran, dass zwischen Hardware und Benutzer noch weitere technische und Abstraktionsebenen liegen.

Betriebssysteme und Applikationen

- Der Benutzer manipuliert in der Regel nicht direkt Hardware-Register oder andere Speicherzellen.
- Stattdessen führt er bestimmte Applikationen aus.
 - Webbrowser als Beispiel
- Doch selbst diese Programme greifen nicht direkt auf die Hardware zu; stattdessen wird der Zugriff durch das *Betriebssystem* realisiert und verwaltet.

Benutzer

Applikationen

Betriebssystem

Hardware

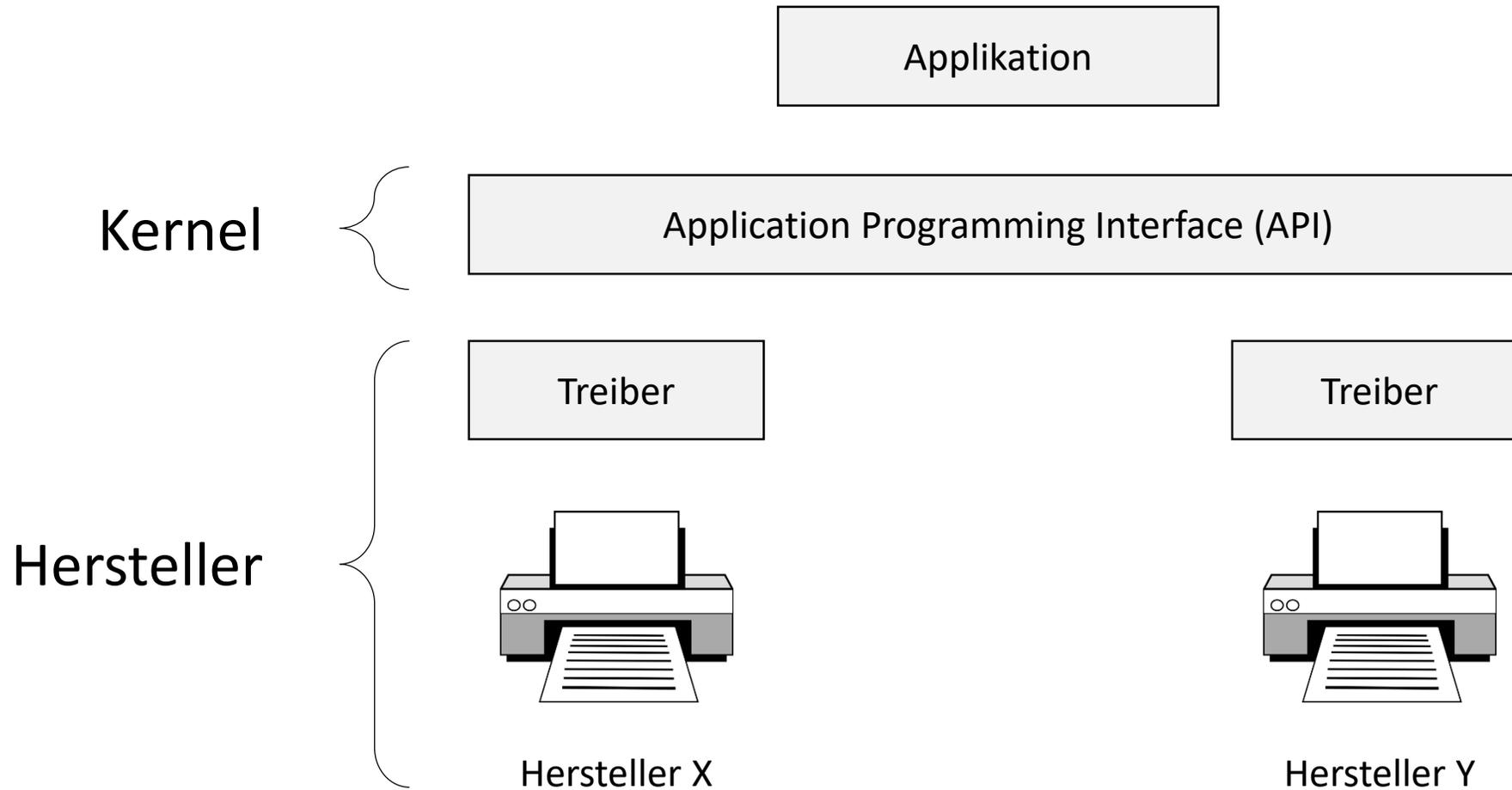
Betriebssysteme (OS)

- Modernen Informatiksysteme erlauben in der Regel das gleichzeitige Ausführen mehrerer Programme (Multi-Tasking)
- Dadurch konkurrieren diese um die verfügbaren Hardware-Ressourcen
 - CPU-Zeit, Speicherzugriffe, externe Hardware
- Eine Aufgabe des Betriebssystems besteht daher darin, diese Zugriffe zu gewährleisten und zu verwalten. Dies geschieht im sog. *Kernel*.
- Gerade die faire Zuteilung von Rechenzeit an Programme (engl. *scheduling*) ist ein wichtiges und schwieriges Unterfangen.

Hardware-Treiber

- Viele Rechnerarchitekturen verfolgen einen modularen Ansatz, der es erlaubt, Geräte und Hardwarekomponenten unterschiedlicher Hersteller zu kombinieren.
- Geräte und Komponenten desselben Typs unterscheiden sich dabei in der Regel durch unterschiedliche technische Implementierungen.
- Es wäre ungünstig, wenn alle Applikationen auf diese Unterschiede eingehen müssten.
- Eine weitere Aufgabe des OS besteht daher darin, für Applikationen eine einheitliche Zugriffsschnittstelle zu bieten.

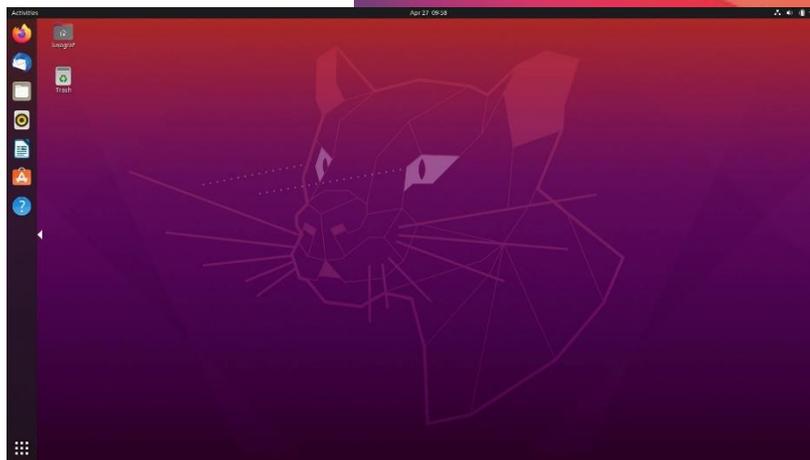
Beispiel



Benutzerschnittstellen

- Zwei grundlegende Paradigmen sind graphische vs. textbasierte Benutzerschnittstellen.
- Bei graphischen Benutzerschnittstellen erfolgt die Interaktion in der Regel mit der Maus durch „direkte Manipulation“.
- Bei textuellen Schnittstellen werden Kommandos getippt. Sämtliche Ein- und Ausgaben erfolgen in Textform.

Beispiele für OS-Benutzerschnittstellen



```
**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.

Red Hat Enterprise Linux 8.5 (Ootpa)
Kernel 4.18.0-348.20.1.el8_5.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

rhel8 login: Gangrif
Password:
Login incorrect

rhel8 login: gangrif
Password:
Last login: Mon Mar 28 15:17:07 on tty2
[gangrif@rhel8 ~]$ _
```

Grundlegende Funktionalität

- Über die Benutzerschnittstelle stellt das Betriebssystem dem Anwender grundlegende Funktionalitäten zur Verfügung.
- Im Wesentlichen sind dies:
 - Interaktion mit dem Dateisystem (Verschieben, Löschen, Anlegen, etc. von Dateien)
 - Starten, Beenden, Hinzufügen und Löschen von Programmen.
 - Inspektion und Konfiguration variabler Aspekte des Systems, wie z.B. Bildschirmauflösung, Netzwerk-Adresse, etc.

Gängige Betriebssysteme

- Personal Computers
 - Microsoft Windows
 - Apple macOS
 - GNU/Linux
- Smartphones
 - iOS
 - Android
- Viele spezialisierte Rechner bieten ein eigenes OS an.

Zusammenfassung Betriebssysteme

- Der Zugriff auf die Hardware eines Informationssystem wird mediiert durch das Betriebssystem.
- Die Kernel-Komponente ist für die Verwaltung der Ressourcen und deren Zurverfügungstellung an unterschiedliche Programme verantwortlich.
- Eine Treiberschnittstelle erlaubt die Einbindung unterschiedlicher Hardware; eine Programmierschnittstelle (API) versteckt die Unterschiede für den Zugriff durch Applikationen.
- Zur Interaktion mit dem Betriebssystem gibt es eine (graphische oder text-basierte) Nutzerschnittstelle.