



Úvod od počítačových sítí

Mgr. Klára Pešková, Ph.D., Klara.Peskova@mff.cuni.cz
Katedra softwaru a výuky informatiky

Ráda zodpovím vaše dotazy:

- Přijďte na Zoomovou konzultaci v době, kdy je předmět rozvržen
- Napište mi e-mail

Domácí úloha:

V ReCodExu máte zadanou úlohu “Simple Static Web”– cílem je vytvořit jednoduchý statický web, požadavky jsou v zadání úlohy poměrně přesně specifikovány.

Za splnění povinných prvků dostanete od ReCodExu jeden bod.

Kromě ReCodExu budu web kontrolovat i já. Web musí alespoň připomínat stránku, která by mohla na Internetu reálně existovat, pouze poskládané povinné prvky nestačí. Pokud web odsouhlasím, dostanete navíc jeden bonusový bod (body se zobrazí jako 1+1), pokud budu mít k vašemu řešení připomínky, napíšu vám je jako komentář k úloze přímo v ReCodExu.

Abyste se mohli přihlásit ke zkoušce z Úvodu to počítačových sítí, je nutné mít v ReCodExu 1+1 bod.

WWW – World Wide Web

- Nejrozšířenější služba Internetu
- Vznikl jako experiment v CERNu
- Vloni
 - sdílení informací, dat
 - prostředí pro aplikace, které jsou přístupné odkudkoliv
- Letos
 - sociální kontakty
 - nakupování
 - kultura
 - výuka
 - zábava ...

facebook

rohlik
.CZ

zoom

DRAMOX

NETFLIX

Úvod do počítačových sítí (2020)

2 / 32

World Wide Web znamená “Pavučina rozšířená přes celý svět”.

Pojmy World Wide Web a Internet jsou někdy chybně zaměňovány. Web je pouze jedna ze služeb internetu.

Historie I (antika)

- 1945 – Vannevar Bush
 - Mozek pracuje s asociacemi
 - Koncept foto-elektricko-mechanického zařízení Memex – rozšíření paměti, propojení (odkazy) mezi dokumenty
- 60. léta – Theodore Nelson poprvé použil slovo hyper-text – text propojený asociacemi
 - Projekt Xanadu
 - Systém pro sdílení informací
 - Implementován jako prototyp

Vannevar Bush byl americký vědec, vizionář, který působil na MIT. Během svého života si nechal patentovat řadu vynálezů. Je známý především díky analogovému počítači určenému k řešení diferenciálních rovnic, který Bush sestrojil a na němž pracoval od r. 1927. Memex byl Bushův koncept (hypotetického) “chytrého” prohlížeče mikrofilmů, se strukturou podobnou hypertextu. Svým textem “As We May Think” (1945), ve kterém Memex představil světu, ovlivnil následující generace informatiků. Memex vycházel z myšlenky, že mozek pracuje s asociacemi, a měl být jakýmsi mechanickým rozšířením paměti: “[A] device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory.”

Ted Nelson, který zavedl pojmy “hypertext” a “hypermedia”, byl silně ovlivněn Bushovými texty. Od roku 1960 se věnoval projektu Xanadu, jehož cílem bylo vytvořit počítačovou síť s jednoduchým uživatelským rozhraním. Nelson věnoval práci na projektu Xanadu většinu svého produktivního života. Projekt Xanadu stále existuje, podle jeho tvůrců: “The World Wide Web trivialises our original hypertext model with one-way ever-breaking links and no management of version or contents.”

Historie II (středověk)

- U vzniku WWW stojí Tim Berners-Lee
 - 1980 – ve švýcarském CERNu navrhl hypertextový systém pro sdílení dat mezi fyziky
 - 1989 – propojení hypertextu a internetu – implementace komunikace mezi klientem a serverem
 - zveřejnil úplně první webovou stránku
<http://info.cern.ch/hypertext/WWW/TheProject.html>
- NCSA Mosaic
 - 1993 – webový prohlížeč (Marc Andreessen a Eric Bina)
 - 1995 => Internet Explorer



Úvod do počítačových sítí (2020)

4 / 32

Tim Berners-Lee navrhl a implementoval první prohlížeč webu, který fungoval zároveň jako editor webu (software se jmenoval WorldWideWeb), protokol HTTP a první webový server – CERN HTTPd (Hypertext Transfer Protocol daemon).

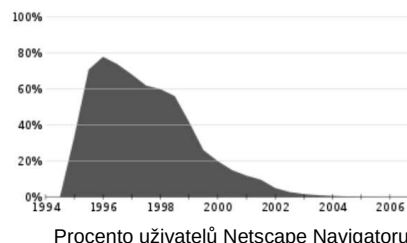
Zajímavosti

- Britský vědec sir Tim Berners-Lee je nyní ředitelem konsorcia W3C, které dohlíží na další vývoj webu.
- První webová stránka popisovala projekt WorldWideWeb. Od zveřejnění této stránky uplyne letos 20. prosince 30 let.
- První verze WWW softwaru obsahovala WYSIWYG editor, znalost HTML tak nebyla nutná.
- Tim Berners-Lee řekl: “Creating the web was really an act of desperation, because the situation without it was very difficult...”

Jeden z prvních prohlížečů Mosaic významně přispěl k popularizaci internetu. Umožňoval zobrazit text i grafiku. Od roku 1995 však ztrácel na popularitě, vedení v oblíbě mezi prohlížeči převzal Netscape Navigator. V r. 1995 koupil licenci Mosaicu Microsoft, což bylo základem pro vznik Internet Exploreru.

Historie III (novověk)

- 1996 – válka prohlížečů
 - Internet Explorer vs. Netscape Navigator
- 1997 – HTML 4 přijato jako standard W3C
- 2002 – Koncept “Web 2.0”
 - Obsah webu tvoří sami uživatelé
- 2004–2006 – první AJAX aplikace
- 2010 – HTML5
 - Začátek konce Flashe (podpora od firmy Adobe oficiálně končí 31. prosince 2020)



Netscape Navigator byl v polovině 90. let synonymem pro internetový prohlížeč. Umožňoval mimo jiné začít číst text stránky před stažením obrázků a dalších multimediálních prvků, což bylo v době dial-up internetu velké plus. Na konci 90. let byl převálcován Internet Explorerem, se kterým Netscape Navigator nedokázal udržet krok v marketingové strategii ani ve vývoji.

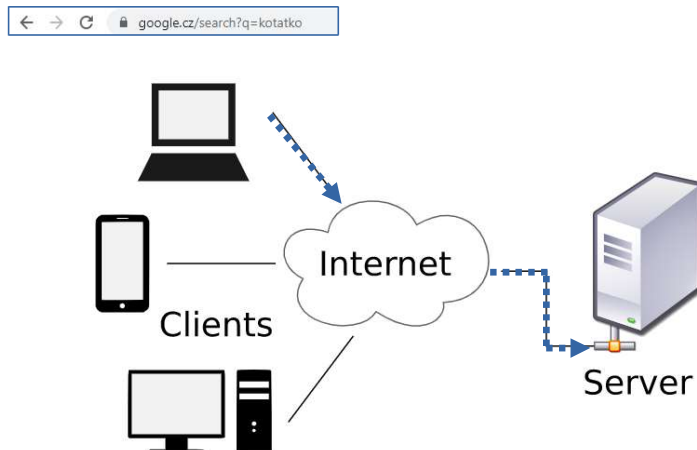
Web 2.0 – etapa vývoje webu, ve které byl neměnný obsah stránek nahrazen prostorem pro sdílení a společnou tvorbu obsahu. Tento pojem se nevztahuje k žádným technickým specifikacím, ale ke změnám ve způsobu, jakým jsou webové stránky navrženy a používány. [Wiki]

AJAX aplikace – web je čím dál tím víc interaktivní (pojem AJAX bude vysvětlen dále v těchto slidech).

Adobe Flash je multimediální softwarová platforma pro tvorbu animací, interaktivních webových aplikací, mobilních aplikací, her, přehrávačů videí ve webových prohlížečích, ... Flash využívá programovací jazyk ActionScript. Od používání Flashe se postupně upouštělo, zejména kvůli závislosti na firmě Adobe, nepodporování v telefonech s Androidem ani na iPhonech, špatné přístupnosti webu pro zrakově postižené osoby, a v neposlední řadě kvůli otázkám bezpečnosti.

Funkcionalita Flashe je nahrazena hlavně HTML5.

Zobrazení webové stránky - **klient-server**



Úvod do počítačových sítí (2020)

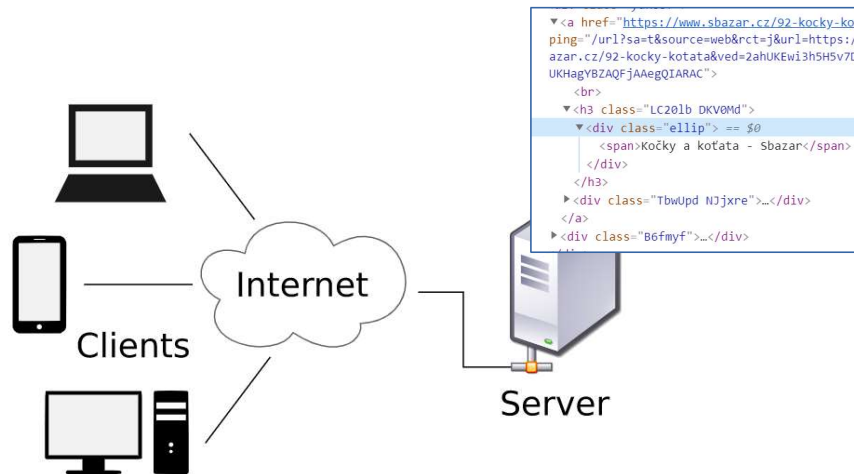
6 / 32

Schéma ilustruje, jak probíhá zobrazení webové stránky, a popisuje klient-server architekturu:

- Webové stránky jsou umístěny na serverech (serverem je zde myšlen počítač, který je připojen k internetu na stálé adrese).
- V prvním kroce uživatel zadá adresu (přesněji URL) požadované stránky do adresové řádky prohlížeče (prohlížeč v této architektuře figuruje jako klient; může být na různých typech zařízení).
- Požadavek je potom propagován internetem k příslušnému serveru.

... (pokračujte na další slide)

Zobrazení webové stránky - klient-server

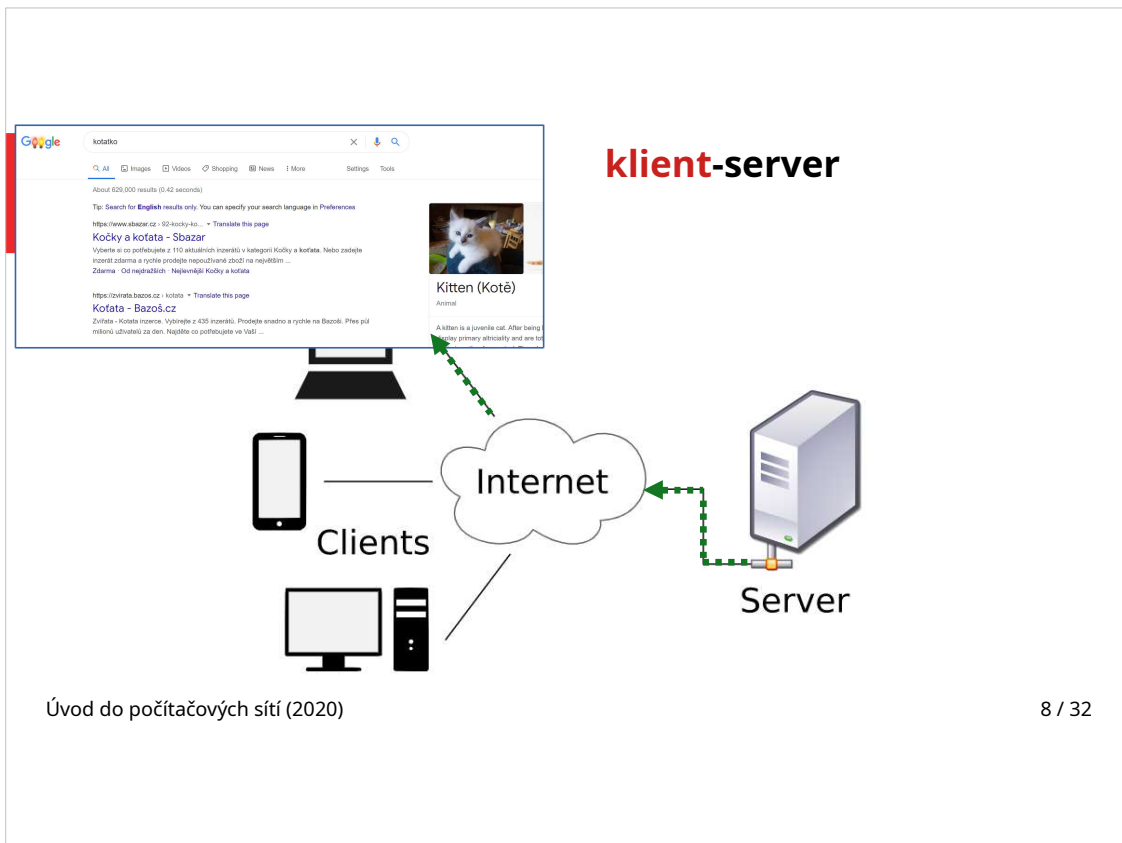


Úvod do počítačových sítí (2020)

7 / 32

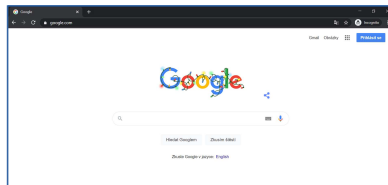
- Požadavek dorazí na server, kde se najde příslušný HTML soubor (případně požadovaný multimediální obsah), nebo se HTML na serveru vygeneruje (podrobněji později).

... (pokračujte na další slide)



- HTML kód (případně další vyžádané soubory) přes internet dorazí zpátky ke klientovi, kde je HTML kód prohlížečem interpretován a zobrazen v podobě, která je pro uživatele dobře čitelná.

Webová stránka



```
<!DOCTYPE HTML>
<html>
<head>
  <title>Google</title>
  <link rel="stylesheet"
        type="text/css"
        href="styles.css">
</head>

<body>
Text contents...
...

<source src="movie.mp4"
        type="video/mp4">
</video>
```

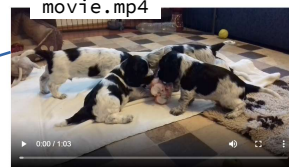
styles.css

```
body {
  Font: 12pt Calibri;
}
p {
  Margin: 10px;
}
```

image.png



movie.mp4



Webová stránka je tvořena HTML kódem – značkovacím jazykem, který definuje význam jednotlivých prvků na stránce .

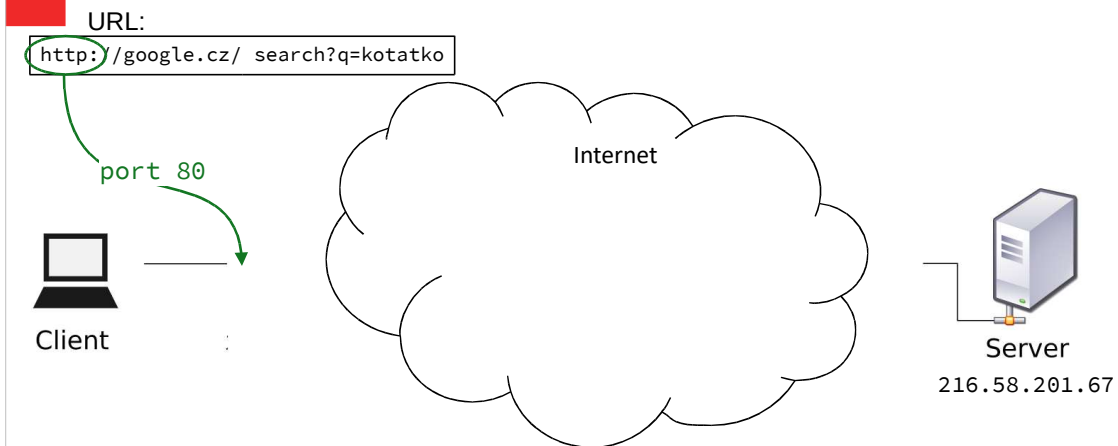
CSS (Cascading Style Sheets) styly potom říkají, jak bude který prvek na stránce vypadat (velikost písma, barva, umístění).

Webová stránka může obsahovat odkazy na obrázky a další multimediální prvky (ty server pošle klientovi spolu s HTML kódem stránky).

HTML může obsahovat také odkazy na jiné webové stránky, či multimediální prvky umístěné i na jiných serverech.

- Poznámka:
 - Prohlížeče většinou “nějakým způsobem” zobrazí i kód, který není úplně správný.
 - Každý klient může HTML interpretovat trochu jinak, zvláště pokud kód obsahuje chyby.

Zobrazení webové stránky podrobněji



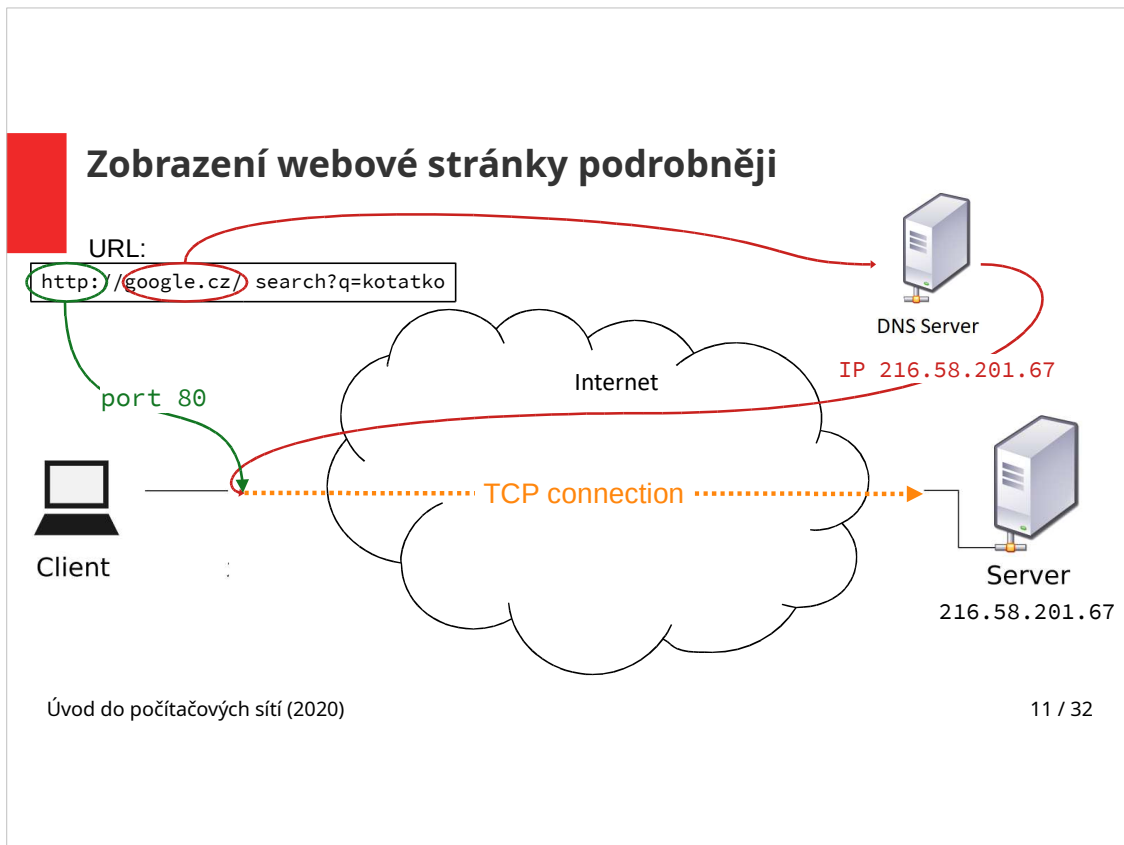
Úvod do počítačových sítí (2020)

10 / 32

Celý průběh zobrazení webové stránky si popíšeme ještě jednou, tentokrát s několika technickými detaily navíc:

- URL, které klient zadá do adresového řádku prohlížeče (klienta) se nejprve “přeloží”.
- Pokud není explicitně zadáno číslo portu, určí se podle použitého protokolu; např. protokol HTTP funguje na portu 80.

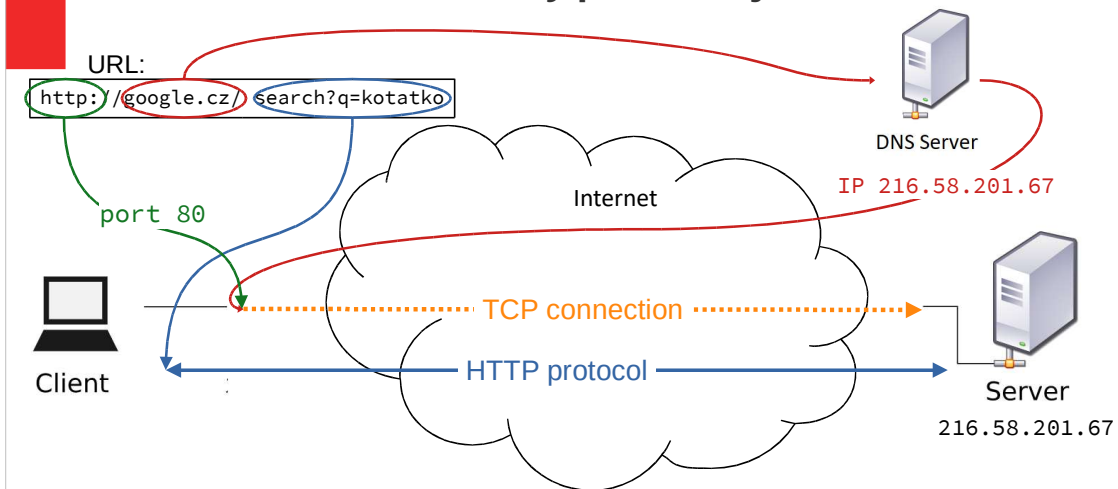
... (pokračujte na další slide)



- Doména se přeloží na IP adresu:
 - DNS (Domain Name System) klient (který je součástí operačního systému) umožní prohlížeči požádat DNS server o překlad domény
 - DNS server vrátí IP adresu serveru
- Mezi klientem a serverem s příslušnou IP adresou se vytvoří TCP spojení

... (pokračujte na další slide)

Zobrazení webové stránky podrobněji



- Následně se už ke komunikaci mezi klientem a serverem používá protokol HTTP
- zbývající část URL (za doménou), použije ve své hlavičce (Headers) HTTP Request. Tato část určuje, kde se požadovaný dokument vyskytuje na serveru, případně slouží pro předání parametrů skriptu na serveru.

URL (Adresa)

- Uniform Resource Identifier (URI)
 - Znakový řetězec, který má následující formát:
`<schema>:<hierarchical_part>?<query>#fragment`
 - `query` a `fragment` jsou nepovinné
- Uniform Resource Locator
 - URI které popisuje umístění zdroje (resource)
`protocol://username:password@domain:port/path
?parameter=value¶meter2=value2#element_id`
 - Příklad:
`https://is.cuni.cz/studium/rozvrh/roz_ucitel.php?ucitel=13270`

URL je speciální případ URI.

Některé části URL jsou povinné, některé nepovinné a některé z nich se ve většině případů nepoužívají:

- **Protocol:** používá se vždy, i když dnešní prohlížeče ho většinou před uživatelem skrývají
- **username:password@:** tato část se většinou nepoužívá, slouží pro HTTP autentikaci
- **Domain:** používá se vždy
- **Port:** většinou není potřeba explicitně uvádět, odvodí se z použitého protokolu
- **Path:** použije se téměř vždy; odpovídá adresářové struktuře na serveru (pokud není použito maskování URL)
- **Dvojice parametr-hodnota:** používá se pro předání parametrů skriptu v případě, že stránka je dynamicky generovaná na serveru
- **Element_id:** říká, na který HTML element se má v rámci HTML dokumentu při načtení stránky přejít (stránka může být při zobrazení rovnou "posunuta" dolů na HTML prvek se požadovaným id)

HTTP – Hyper-text transfer protocol

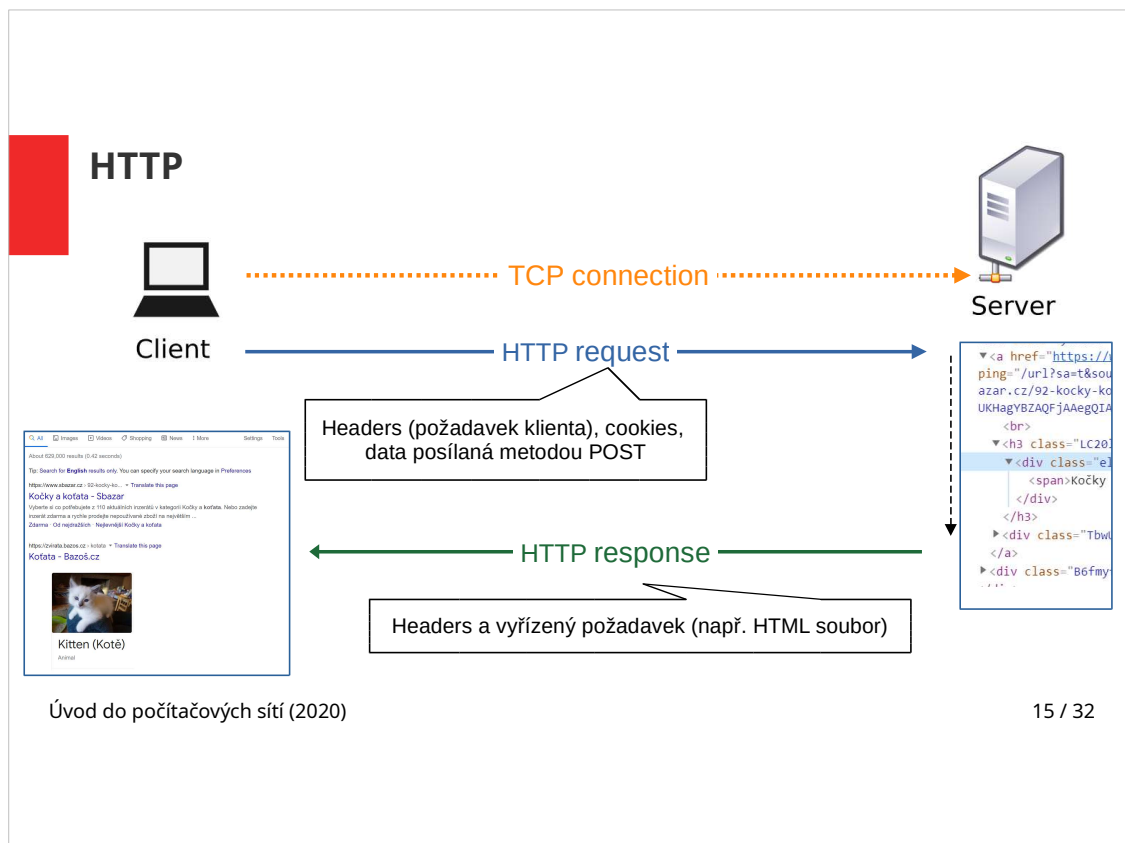
- Jednoduchý textový protokol
- Funguje přes TCP
- Navržený pro získávání dat
 - Původně pouze text
 - Rozšíření podporuje různé typy dat a kódování (MIME)
- Průběh komunikace
 - Uživatel (klient) pošle HTTP Request
 - obsahuje detaily o požadovaném obsahu
 - Server odpoví pomocí HTTP Response
 - (většinou) obsahuje požadovaná data

Úvod do počítačových sítí (2020)

14 / 32

MIME (Multipurpose Internet Mail Extensions) – “víceúčelová rozšíření internetové pošty” – standard, který umožňuje přenášet texty v různých kódováních či binární data pomocí kanálů, které byly původně navrženy pouze pro přenos textu v kódování ASCII. Standard byl vyvinut pro elektronickou poštu, ale používá ho např. právě i protokol HTTP.

HTTP Response – obsahuje požadovaná data, pokud nedojde k chybě.



- Mezi klientem a serverem se vytvoří TCP spojení
- Klient pošle HTTP Request (požadavek)
 - Hlavička (Headers) požadavku říká, co klient od serveru požaduje, případně se v requestu posílají cookies a data předávaná metodou POST
- Server načte nebo vygeneruje požadovaná data
- Výsledek odešle klientovi jako HTTP Response (odpověď)
 - obsahem Response může být např. HTML soubor
- V rámci jednoho TCP spojení může proběhnout více HTTP požadavků/odpovědí

Statické webové stránky vs webové aplikace

- Statické stránky – server vrátí požadovaný dokument
 - WWW jako “Sít’ dokumentů”
 - Každý dokument má své URL
 - Dokumenty jsou propojené odkazy
 - HTML v tomto kontextu chápeme jako formát dokumentu
- Webové aplikace – zobrazený dokument se vygeneruje na základě požadavků klienta
 - HTML a CSS tvoří výstup programu

Webové stránky mohou být buď statické nebo dynamické.

Obsah statických stránek je stále stejný – autor je samozřejmě může upravovat a aktualizovat, důležité ale je, že v daný okamžik je obsah stránky stejný pro každého uživatele, který si stránku zobrazí. Statickou stránku typicky tvoří HTML dokument, který je umístěn na serveru spolu se styly, obrázky a dalšími prvky, které stránka obsahuje. Statická stránka má stálé URL, které vychází z adresářové struktury na serveru a které je unikátním globálním identifikátorem tohoto dokumentu.

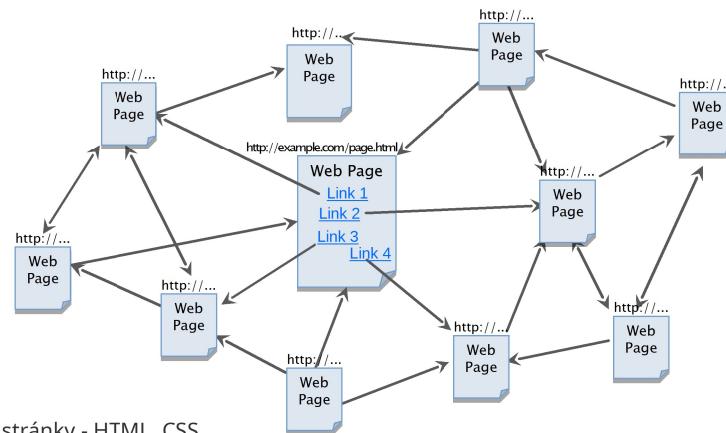
Webové aplikace reagují na požadavky klienta, WWW využívají jako prostředí, které je snadno dostupné odkudkoliv.

Webové aplikace používají HTML a CSS pro komunikaci s uživatelem – pro přijetí jeho požadavků a následně pro zobrazení výstupu. Zobrazené HTML tedy není předem pevně dané, generuje se dynamicky na základě požadavků uživatele.

Webové aplikace mohou být jednoduché – např. stránky, které pouze dynamicky generují obsah jednotlivých podstránek na základě toho, na kterou položku v menu uživatel klikne, nebo také velmi složité, používající množství komplexních formulářů, jako např. elektronické bankovníctví, redakční systémy atp. Tyto aplikace mohou mít stejnou funkcionalitu, jako by nabízely jejich desktopové ekvivalenty, navíc díky zobrazení v prohlížeči jsou přirozeně multiplatformní a dostupné odkudkoliv.

Webové aplikace často využívají k ukládání informací i svého obsahu databázi běžící na serveru.

Síť dokumentů (Web of documents)



- Statické stránky - HTML, CSS
- Pro jejich zobrazení (teoreticky) stačí prohlížeč, server je ale potřeba pro zveřejnění na internetu

Představa statických stránek jako “Web of documents”, tj. stránek s neměnným obsahem, které mohou být navzájem propojeny odkazy. Každá stránka má svou stálou webovou adresu (URL).

Odkazy:

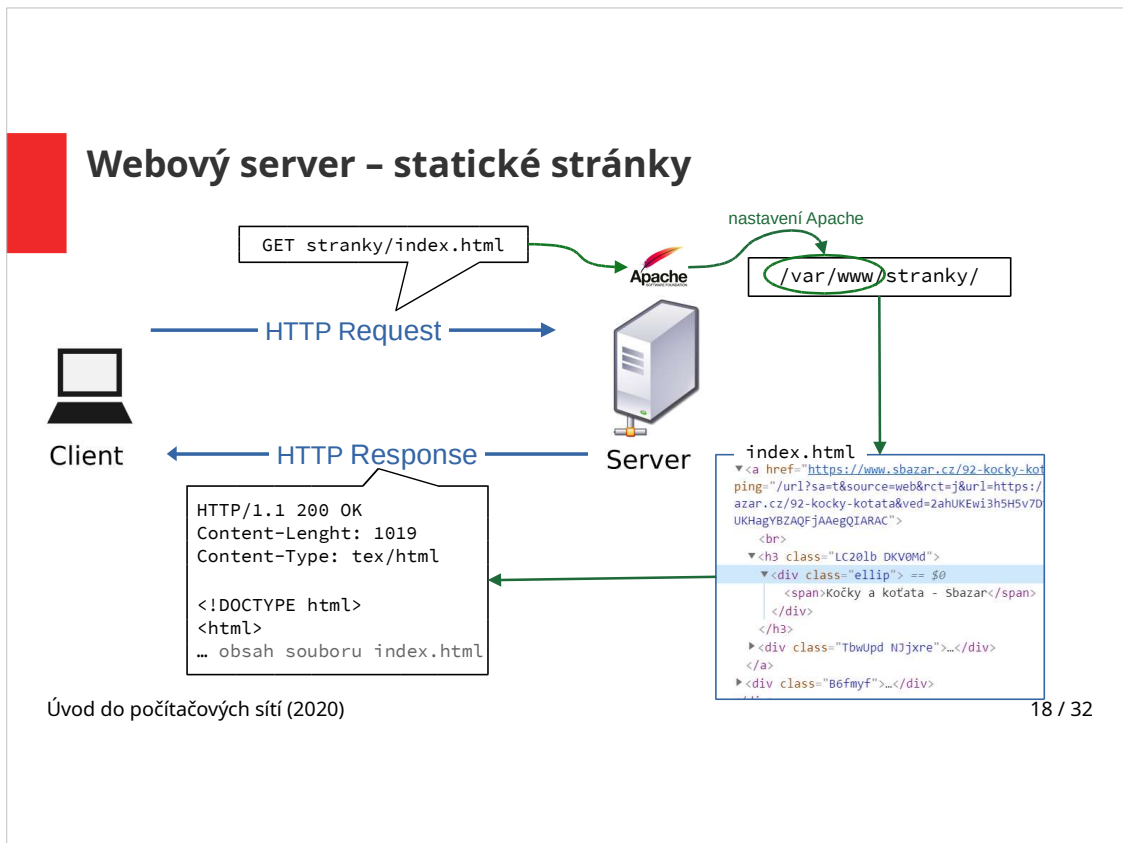
Odkaz (link) je HTML prvek, který obsahuje URL jiné webové stránky, stránka je tímto způsobem propojená s jinou stránkou (toto propojení je pouze jednosměrné, druhá stránka nemusí odkazovat na tu původní).

Při kliknutí na odkaz prohlížeč přejde na novou stránku, tj. vyžádá si novou stránku pomocí HTTP Requestu typu GET. Původní stránka se uloží do historie prohlížeče, prohlížeč nám potom umožní vrátit se “zpět”.

Poznámka:

Protože statické stránky používají pouze HTML a CSS, stačí pro jejich zobrazení internetový prohlížeč (může se hodit při vývoji stránek, případně při vytváření domácího úkolu).

Pro zveřejnění webových stránek už samozřejmě server potřeba je, nicméně nemusí na něm běžet např. PHP ani databáze.



V případě statické webové stránky existuje stránka, kterou uživatel požaduje, na serveru přímo jako HTML dokument (soubor).

- Klient vyšle HTTP Request typu GET, jehož obsahem je adresa stránky na serveru – cesta k tomuto souboru odpovídající serverové adresářové struktuře
- Na serveru (hardwaru, počítači) běží softwarový webový server (např. open-source webový HTTP server Apache), který přijme HTTP Request
- Cesta k souboru v Requestu je relativní vůči adresáři, který je zpřístupněn pro softwarový webový server (absolutní adresa tohoto adresáře závisí na nastavení webového serveru)
- Na serveru je nalezen požadovaný dokument, a HTML soubor je vrácen jako tělo HTTP Response
- Pokud není zadáno konkrétní jméno souboru (.html), tedy pokud URL končí / za názvem adresáře, zobrazí se automaticky soubor s názvem index.html

Poznámka:

Termín **server** je používá jak pro hardware (server=počítač), tak pro software, který komunikuje HTTP protokolem (webový server).

Webové aplikace

- Interaktivní prvky
 - Hypertextové odkazy -> změna vzhledu stránky
 - Formuláře – vstup od uživatele
 - Po odeslání mohou být zadaná data zpracována na serveru
- Dynamické webové stránky
 - Server-side
 - Stránka je vygenerovaná na serveru na základě požadavků klienta
 - Client-side
 - Skript, který běží v prohlížeči, reaguje na akce uživatele, a modifikuje podle toho zobrazovaný HTML dokument

Webová aplikace obsahuje prvky, které umožňují interakci s uživatelem. Typickými příklady interaktivních prvků jsou hypertextové odkazy a HTML formuláře.

Na základě vstupu od uživatele – tj. toho, na co uživatel kliká, nebo co vyplní do formuláře, se potom dynamicky vygeneruje stránka, která je “ušitá na míru” pro konkrétního uživatele.

Stránky se mohou dynamicky generovat buď na serveru (tomuto způsobu se říká server-side) nebo přímo v prohlížeči (client-side). V prvním případě server zareaguje na požadavek, vygeneruje podle něj HTML a pošle ho klientovi jako novou stránku. V případě client-side v prohlížeči běží skript, který modifikuje stávající HTML stránku podle akcí uživatele přímo v prohlížeči, bez komunikace se serverem.

Webové formuláře

- Skládají se z různých ovládacích HTML prvků
 - Políčka pro zadávání textu (input)
 - Rozbalovací výběr (select)
 - Výběr jedné z možností (radio button)
 - Zaškrtnutí políčka (checkbox)
 - Možnost nahrát soubory
 - Tlačítka
- Formulářové prvky jsou chytřejší (více user-friendly) verze odkazů
 - Uživatel odešle data, výsledkem je nová stránka
 - Formulář může být odeslán pomocí HTTP metod GET nebo POST
- Data jsou odeslána, když uživatel stiskne odesílací tlačítko (submit)

Webové formuláře představují způsob, jak může uživatel komunikovat s aplikací. Tím, že vyplní formulář, zadává aplikaci vstupní data a nastavuje parametry.

Formuláře sestávají z různých HTML prvků a jejich vzhled je možné modifikovat pomocí CSS stylů.

Data z formuláře jsou odeslána na server všechna najednou po stisknutí odesílacího tlačítka. Data, která uživatel vyplnil jsou buď odeslána v URL jako dvojice “parametr=hodnota” (metoda GET) a jsou tedy viditelná, nebo jsou obsažena v těle HTTP Requestu (metoda POST).

Webový server – dynamické stránky

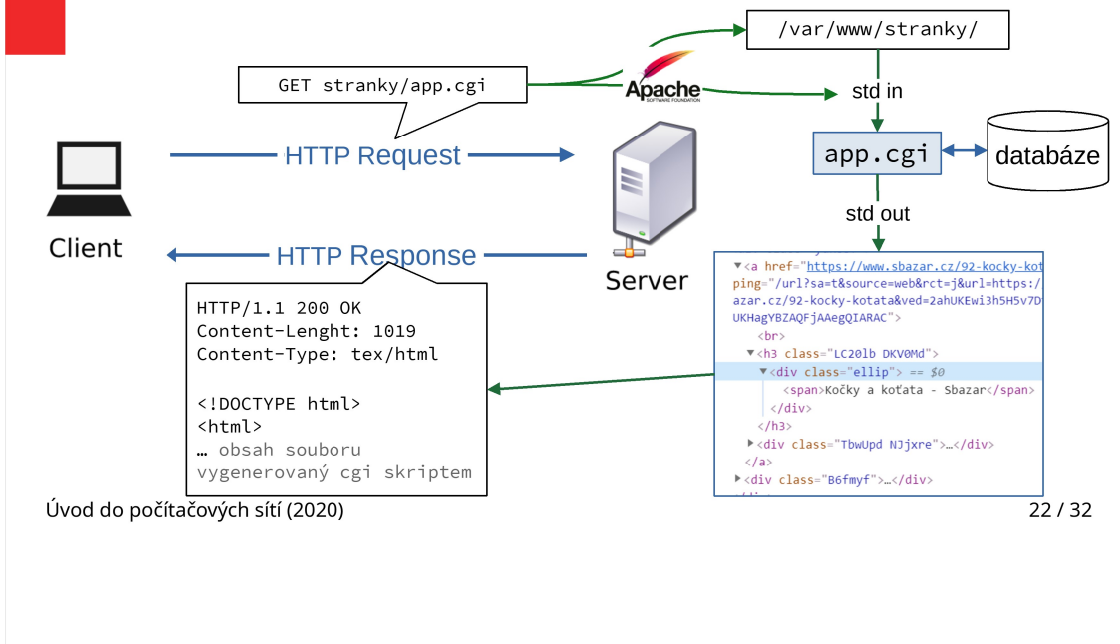
- Common Gateway Interface (CGI)
 - Jeden z prvních standardů pro generování dynamického obsahu na webu
 - Umožňuje serveru předat požadavek od klienta externí command-line aplikaci, která vrátí výsledek na výstup
 - Specifikuje jenom rozhraní
 - Samotná aplikace může být napsaná v libovolném programovacím jazyku
 - Vstupní data a hlavička (headers) se aplikaci předávají pomocí proměnných prostředí, HTTP POST se předává jako std. vstup

Základní princip server-side dynamicky generované webové stránky je založen na CGI. CGI je rozhraní pro propojení webového serveru (myšleno software) a command-line aplikace. Požadavky od klienta tak mohou být předány této externí aplikaci. Aplikace typicky zpracuje požadavek a webovému serveru vrátí vygenerovanou, již statickou, webovou stránku.

CGI specifikuje jenom rozhraní, command-line aplikace může využívat ke zpracování požadavku a vygenerování HTML libovolný programovací jazyk.

Samotné CGI bylo postupně vytlačeno efektivnějšími řešeními, jako je FastCGI nebo integrace modulů skriptovacích jazyků do webových serverů (viz dále).

Webový server – dynamické stránky – CGI



Vytvoření dynamické stránky na základě požadavku od uživatele pomocí CGI:

- Webový server (např. Apache) přijme HTTP Request.
- Podle hlavičky (Headers) Requestu nastaví proměnné prostředí (environment variables), obsah GET nebo POST Requestu je CGI programu (skriptu) předán na standardním vstupu.
- CGI skript dynamicky vygeneruje HTML stránku, může při tom používat například spojení s databází.
- CGI skript potom pošle vygenerovanou HTML stránku na standardní výstup.
- Hotová HTML stránka je webovým serverem odeslána jako HTTP Response.

Webový server – dynamické stránky – tři řešení

- FastCGI
 - CGI procesy jsou spuštěny a čekají na dotazy
 - Efektivnější interface
- Moduly skriptovacích jazyků
 - Speciální moduly, které jsou integrované do HTTP serveru
 - Interpretují skripty a starají se o jejich výstup
- Aplikace přímo obsahuje HTTP server (jako knihovnu)
 - dedikovaný webový server pro aplikaci

Úvod do počítačových sítí (2020)

23 / 32

Možná řešení server-side dynamického webu jsou v podstatě tři:

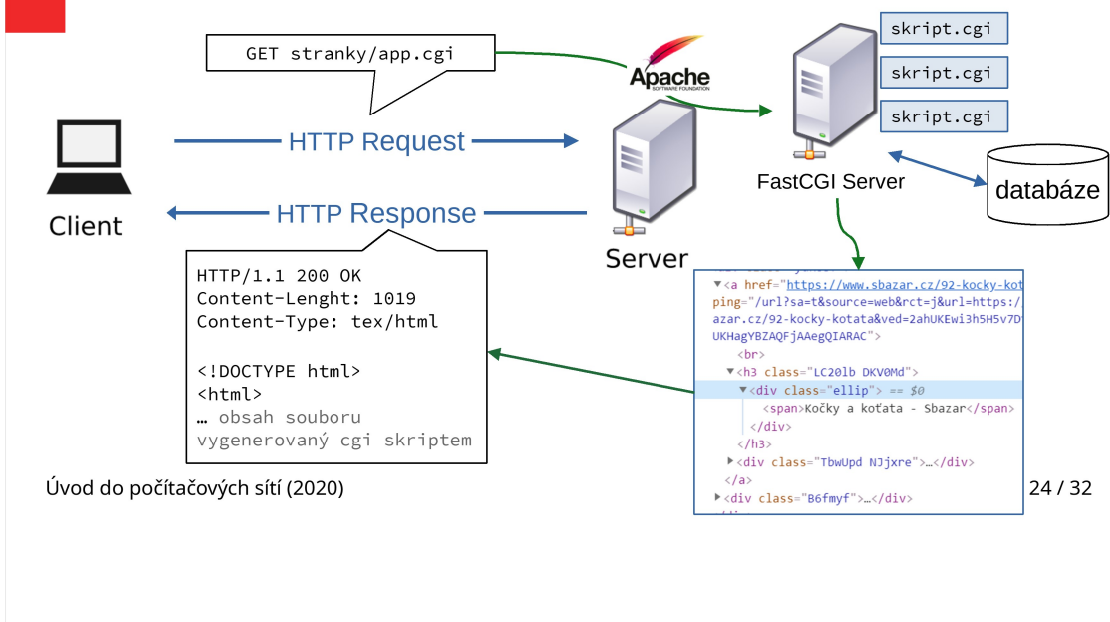
První možnost je použít vylepšenou verzi CGI – FastCGI. FastCGI vychází z původního CGI rozhraní. Vylepšení spočívá zejména v omezení zátěže a režie, které jsou spojené s komunikací mezi webovým serverem a CGI aplikací. Server tak zvládne ve stejném čase vyřídit více požadavků.

Druhá – velmi rozšířená – možnost je použít nějaký programovací jazyk, např. PHP nebo Python, jako modul integrovaný do HTTP serveru (Apache).

Třetí možností je vytvořit si vlastní plně customizovatelný webový server vytvořený na míru přímo pro zamýšlenou aplikaci, takovým příkladem může být třeba server pro streamování videa Wowza.

Princip fungování těchto tří přístupů bude dále rozebrán, bude popsán zejména rozdíl oproti situaci, kdy server poskytuje klientovi statickou HTML stránku.

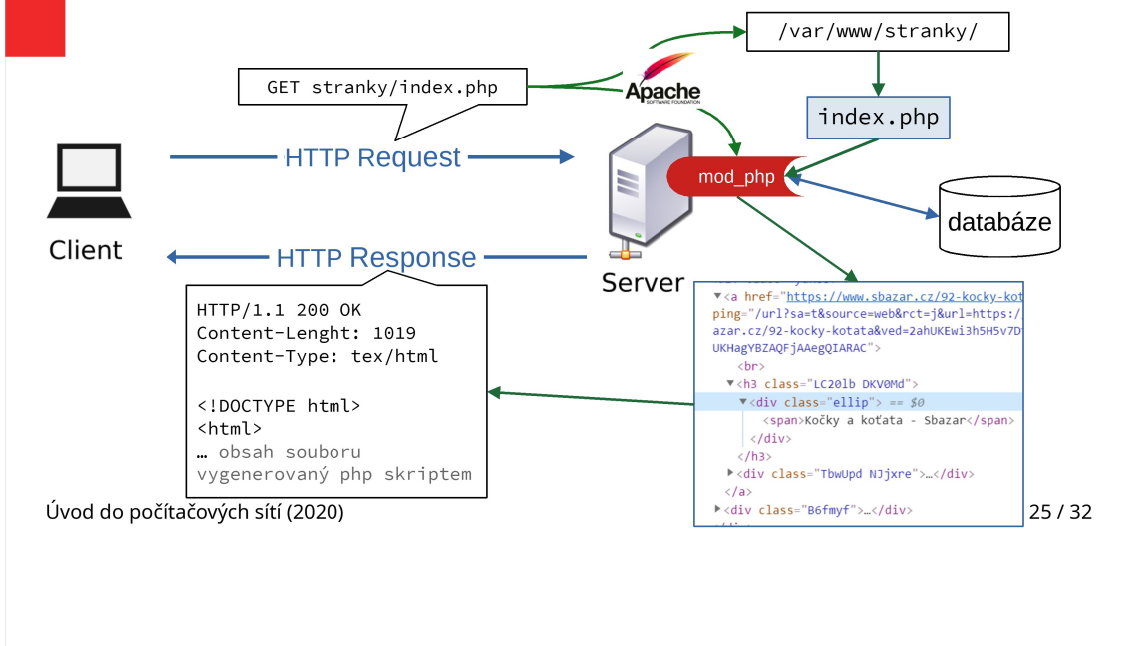
Webový server – FastCGI



Na fyzickém (hardwarovém) serveru běží kromě webového serveru ještě FastCGI server, který vyřizuje požadavky na dynamické stránky. Oddělení webového serveru a webových aplikací přináší různé výhody, např. webový server a webová aplikace mohou být restartovány nezávisle na sobě, FastCGI server může být více přizpůsobený konkrétním požadavkům, např. na zabezpečení. Různé typy příchozích požadavků také mohou být zpracovány specializovanými CGI skripty, které mohou být schopné tyto konkrétní požadavky zpracovat efektivněji.

- Apache předá HTTP Request FastCGI serveru, kde se CGI skripty postarají o vygenerování výstupu. FastCGI server také může při zpracování požadavku využít databázi.

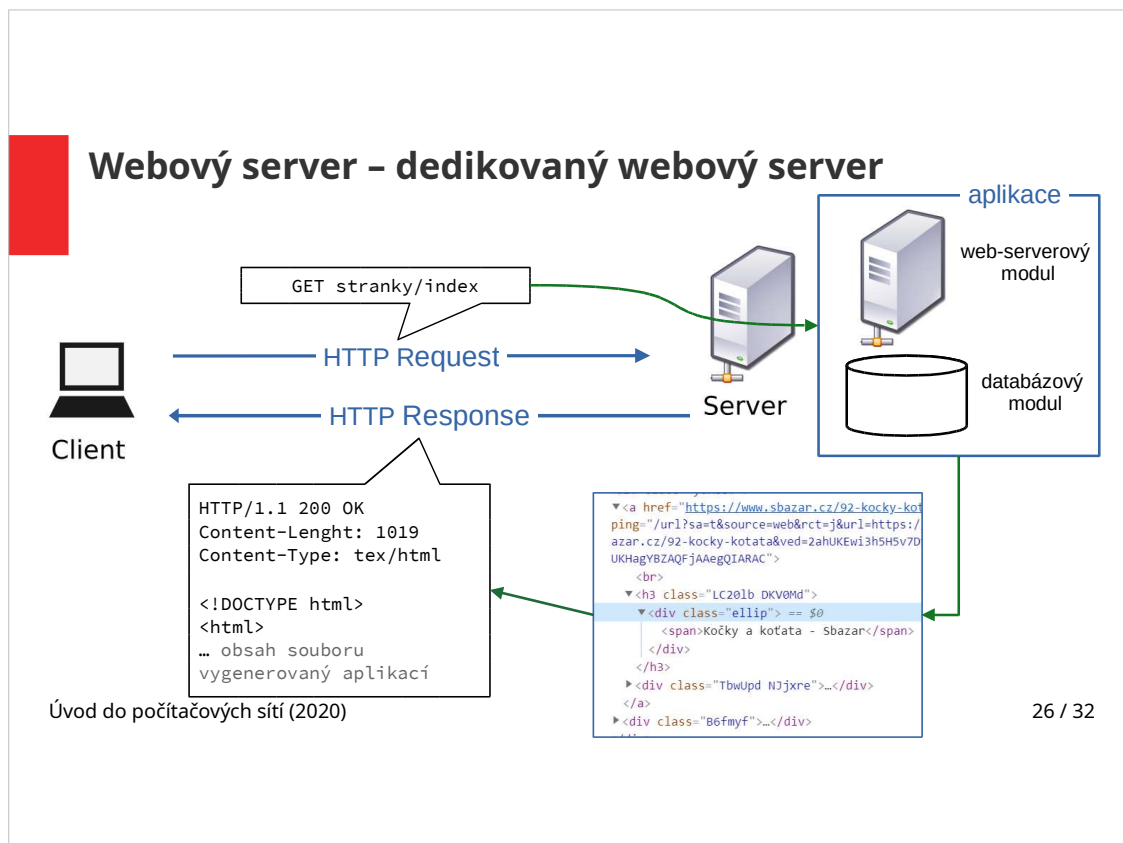
Webový server – moduly skriptovacích jazyků (PHP)



Modul skriptovacího jazyka je součástí webového serveru, např. mod_php, interpret jazyka PHP, je součástí HTTP webového serveru Apache. Tato integrace umožňuje lepší komunikaci mezi webovým serverem a programem, který zpracovává požadavek na dynamickou stránku.

- Po přijetí HTTP Requestu je nalezen příslušný PHP soubor
- Tento PHP soubor je interpretován pomocí modulu mod_php obsaženého ve webovém serveru.
- Modul vygeneruje HTML výstup. Tento modul může komunikovat s databází.

Pokud není zadáno konkrétní jméno souboru (.php), zobrazí se automaticky soubor s názvem index.php.



Složitější webové aplikace mohou mít natolik specifické požadavky na webový server, že se vyplatí naprogramovat specializovaný (dedikovaný) webový server, který je navržen přímo na míru dané aplikaci. HTTP server, a vše, co je potřeba pro vyřízení požadavků od klienta, je potom součástí této aplikace.

Příkladem tohoto řešení může být server pro streamování videa, který kromě streamování videí používá dynamické webové stránky jako uživatelské rozhraní pro nastavení různých možností streamování.

Klient – dynamické stránky

- JavaScript
 - Autor: Brendan Eich z Netscape (polovina 90. let)
 - Skriptovací jazyk pro webové prohlížeče
 - Syntaxe podobná jazyku C
 - Dynamicky typovaný
 - Objektově orientovaný, prvky funkcionálního programování
 - Slovo Java je součástí jeho názvu pouze z marketingových důvodů

Úvod do počítačových sítí (2020)

27 / 32

Dynamické stránky mohou být generovány čistě na straně klienta – přímo v prohlížeči. K tomuto účelu byl od počátku 90. let vytvářen programovací jazyk JavaScript.

JavaScript je plnohodnotný programovací jazyk, který může běžet i v jiných prostředích než je webový prohlížeč, funguje např. jako vestavěný skriptovací jazyk pro OpenOffice, Acrobat a Adobe Reader podporují spouštění JavaScriptu v pdf souborech. JavaScript je také možné využívat na straně serveru, např. jeho implementaci Node.js nebo Rhinola.

JavaScript nemá nic společného s programovacím jazykem Java, naopak, tyto jazyky jsou po všech stránkách velmi odlišné.

Klient – dynamické stránky

- JavaScript v prohlížeči
 - Sandbox, omezené možnosti
 - DOM (Document Object Model) API
 - Umožňuje manipulovat s HTML dokumentem a CSS styly
 - Úpravy se hned zobrazují
 - Event model pro obsluhu akcí uživatele
 - Ostatní API
 - Kontrola nad oknem prohlížeče, historií, URL
 - Animace, 3D grafika, zvuk
 - Ukládání dat (do prohlížeče)
 - Síť – HTTP requesty, WebRTC

JavaScript má přístup k objektovému modelu HTML stránky, která je načtená v prohlížeči. Stránka reaguje na akce uživatele, aniž by docházelo ke komunikaci se serverem.

Příklad takové dynamicky generované/modifikované stránky může být třeba vyskakovací okno s upozorněním, kontrola správnosti vyplnění formulářů, rozbalovací menu nebo zobrazení okna s velkým obrázkem po kliknutí na zmenšený obrázek (thumbnail).

Pomocí JavaScriptu je možné měnit strukturu HTML, atributy HTML prvků i jejich obsah. Je také možné měnit jejich vzhled nebo umístění pomocí nastavování jejich CSS vlastností. JavaScript dále může měnit okno prohlížeče, přejít na jiné URL, umožňuje pracovat s animacemi, atd.

Může také odesílat HTTP Requesty, nebo využívat WebRTC (Web Real-Time Communication) pro komunikaci v reálném čase (toto API poskytuje podporu např. pro telefonní hovory, video chat nebo přehrávání videa).

AJAX (/ˈeɪdʒæks/) – Asynchronous JavaScript and XML

- Kombinuje tři technologie
 - JavaScript
 - Asynchronní HTTP API integrované v prohlížeči
 - XML nebo jiný strukturovaný formát dat (JSON)
- Skript vyvolá HTTP požadavek – bez znovu-načtení (refreshování) celé stránky
- Požadavky jsou asynchronní, data zpracuje callback

AJAX je použití kombinace technologií pro získání dat ze serveru, aniž by bylo potřeba znovu načítat stránku.

Stránka se tak uživateli mění “pod rukama”, aniž by k tomu musel nějak přispět nebo aniž by o tom musel vůbec vědět. Z toho vyplývají také některá úskalí a bezpečnostní rizika použití AJAXu.

Asynchronní požadavky

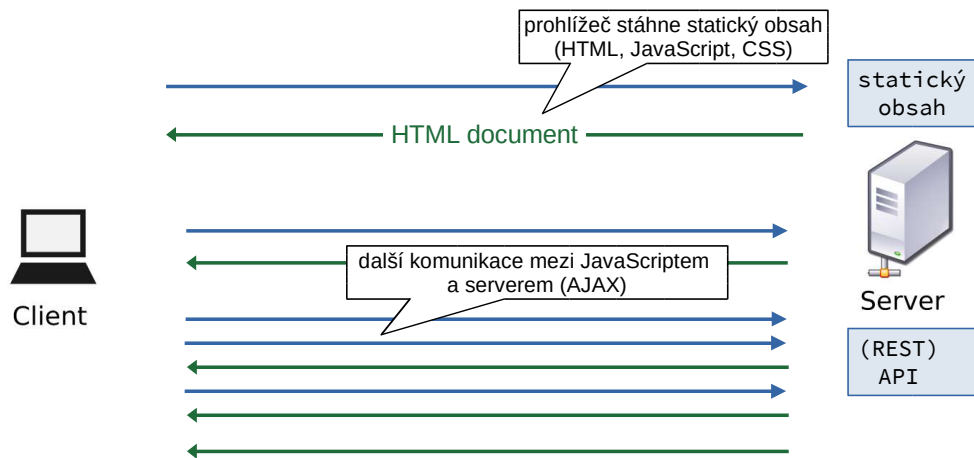
Určitě jste si někdy všimli, že během načítání se stránka může jevit jako prázdná, což se po chvíli samo změní (např. hlásí 0 zpráv, které se časem doplní). To je způsobeno asynchronním HTTP požadavkem JavaScriptu na získání dat ze serveru.

U synchronních operací program čeká, než se daná operace dokončí, potom dál pokračuje v provádění kódu. Při čekání program nic nedělá, z hlediska uživatele “zamrzl”.

Pokud se požadavek vyřizuje asynchronně, žádost se odešle a na výsledek se nečeká, program pokračuje dál. Mezitím, co se požadavek vyřizuje, tedy aplikace dál běží a reaguje na akce uživatele. Když je asynchronní operace dokončena, např. data jsou připravena a doručena aplikaci, zavolá se tzv. callback, který doručí data zpracuje.

Formát posílaných dat byl původně XML (značkovací jazyk podobný HTML, ale obecnější), nyní se nejčastěji používá formát JSON.

Webové aplikace - architektura



- Aplikace používající AJAX si typicky nejdříve načte statický obsah.
- Poté používá pro další získávání dat asynchronní HTTP požadavky pomocí JavaScriptu.

Aplikace často využívají pro komunikaci se serverem a získávání požadovaných dat API, konkrétně webové aplikace často používají REST (REpresentational State Transfer) API.

REST API je navrženo pro klient-server architekturu, je bezstavové, stav aplikace je vyjádřen tzv. **resource**em. Resource může být reprezenovan např. pomocí XML, HTML, JSONu, SVG, PDF.

Webové technologie

Klient

- HTML
- CSS
- JavaScript
- Flash



Internet

Server

- PHP, Python, ...
 - Posílání e-mailů
 - Zpracování formuláře
- databáze



- JavaScript (AJAX - Asynchronous JavaScript and XML)
- ~~Flash~~

Na závěr shrnutí webových technologií, o kterých jsme mluvili:

- HTML je značkovací jazyk, který definuje sémantický význam jednotlivých částí stránky, např. nadpis, odstavec; zobrazuje se v prohlížeči
- CSS – určuje vzhled stránky – podobu jednotlivých HTML elementů a jejich rozmístění; CSS zobrazuje prohlížeč
- JavaScript – plnohodnotný programovací jazyk, který umožňuje měnit vzhled a obsah HTML prvků v zobrazeném dokumentu; běží v prohlížeči
- Flash – již se nepoužívá
- PHP, Python, Java ... - serverové programovací jazyky, které generují výstup pro uživatele v podobě dynamické HTML stránky
 - s jejich pomocí je možné např. zpracovávat formuláře, posílat e-maily, ...
 - umožňují práci s databází
- AJAX – běží v prohlížeči, ale umožňuje získávání dat ze serveru bez nového načtení stránky
- Flash to umožňoval také

Otázky na závěr

- Vysvětlíte, jak se dostane webová stránka do prohlížeče.
- Co je URL a z jakých částí se skládá? Jaký je význam těchto částí?
- Popište komunikaci mezi klientem a serverem. Který protokol se používá?
- Jaký je rozdíl mezi statickou a dynamickou webovou stránkou?
- Jaké jsou dvě základní možnosti vytvoření dynamické stránky?
- Které webové technologie fungují v prohlížeči a které na serveru?
- Čím se vyznačují aplikace, které používají AJAX?