Wykład 1

Wprowadzenie do aplikacji WWW

wykład prowadzi: Maciej Zakrzewicz

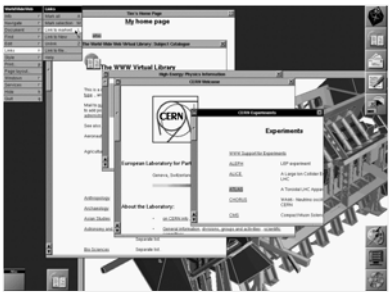
Plan wykładu

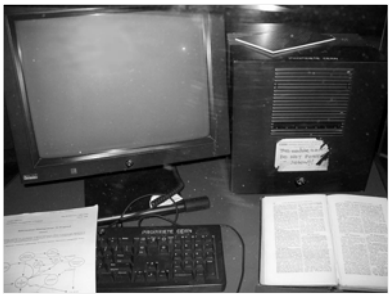
- Rys historyczny
- Składniki architektury WWW
  - klient HTTP
  - serwer HTTP
  - protokół HTTP
- Rozszerzona architektura WWW
  - aplikacja WWW
  - serwer aplikacji
  - aplikacje komponentowe
- Język HTML

Celem wykładu jest wprowadzenie do architektury WWW i problematyki aplikacji WWW. Wykład rozpocznie się od przedstawienia historii rozwoju technologii WWW. Następnie zostaną omówione składniki podstawowej architektury WWW: klient HTTP, serwer HTTP, protokół HTTP. W dalszej części wykładu skupimy się na pojęciu aplikacji WWW, serwera aplikacji oraz na komponentowych modelach aplikacji WWW. Na zakończenie przedstawimy podstawowe własności języka HTML.

Rys historyczny

- Projekt Tima Bernersa-Lee dla CERN (1989)
- Pierwsza przeglądarka - WorldWideWeb
- Pierwszy serwer WWW - httpd





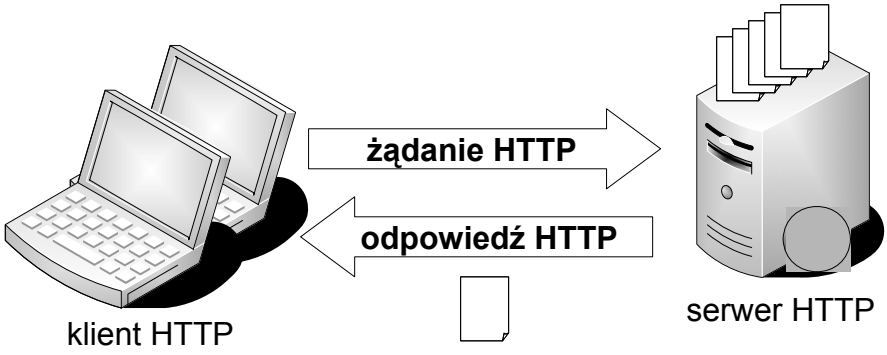
Za twórcę koncepcji World Wide Web (WWW) uznaje się Tima Bernersa-Lee, związanego ze szwajcarskim CERN (Organisation Européenne pour la Recherche Nucléaire), który od 1989 roku pracował nad wykorzystaniem koncepcji hipertekstu w celu udostępniania danych badawczych. Berners-Lee opracował pierwszą składnię języka HTML, zaimplementował pierwszy serwer WWW, nazwany httpd, a także pierwszą przeglądarkę WWW, nazwaną WorldWideWeb, która jednocześnie pełniła rolę edytora HTML.

Pierwszy serwer WWW został uruchomiony 6 sierpnia 1991 roku pod adresem http://info.cern.ch/. W roku 1994 Tim Berners-Lee założył przy MIT (Massachusetts Institute of Technology) organizację World Wide Web Consortium (W3C), która do dziś zajmuje się koordynacją rozwoju technologii WWW.

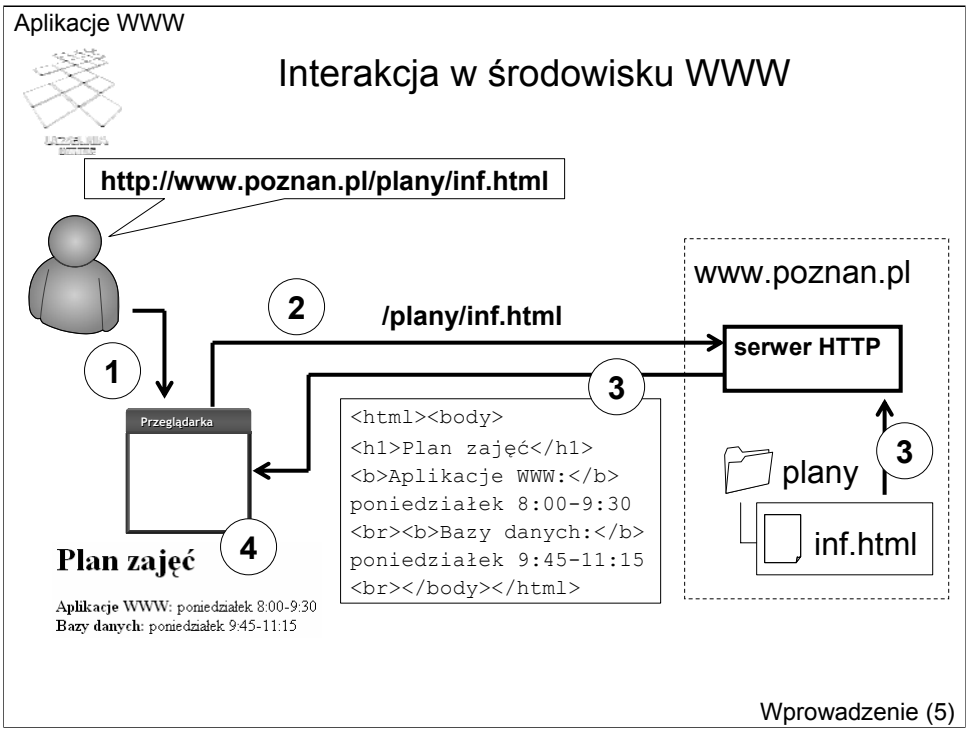
Na slajdzie umieszczono dwie fotografie. Lewa fotografia przedstawia wygląd pierwszej przeglądarki WWW - WorldWideWeb. Na prawej fotografii widoczny jest komputer NeXT, na którym pracował pierwszy serwer WWW - dziś znajduje się on w muzeum CERN w Meyrin. Fotografie pochodzą z Wikipedia (www.wikipedia.org).

Składniki architektury WWW

- Klient HTTP (przeglądarka WWW)
- Serwer HTTP (serwer WWW)
- Protokół HTTP



Architektura World Wide Web (WWW) jest przykładem architektury rozproszonej składającej się z dwóch funkcjonalnie rozdzielonych warstw: warstwy klienta HTTP i warstwy serwera HTTP. Komunikacja pomiędzy tymi warstwami jest realizowana za pośrednictwem protokołu HTTP. Serwer HTTP jest programem nieprzerwanie pracującym, obsługującym repozytorium dokumentów (np. HTML), które udostępnia sieciowym klientom HTTP. Klient HTTP jest programem użytkowym, który odpowiada za wysyłanie żądań pobrania dokumentów, wizualizację pobieranych dokumentów oraz obsługę interakcji z użytkownikiem końcowym.



W środowisku WWW żądania użytkowników końcowych są obsługiwane w następujący sposób:

1. Użytkownik końcowy przekazuje klientowi HTTP adres URL żądanego dokumentu, np. http://www.poznan.pl/plany/inf.html. Klient HTTP wyodrębnia z adresu URL adres DNS (lub IP) komputera, na którym pracuje oprogramowanie serwera HTTP, np. www.poznan.pl.

2. Klient HTTP otwiera połączenie TCP do serwera HTTP i wysyła komunikat żądania HTTP, zawierający nazwę i ścieżkę prowadzącą do żądanego dokumentu, np. /plany/inf.html.

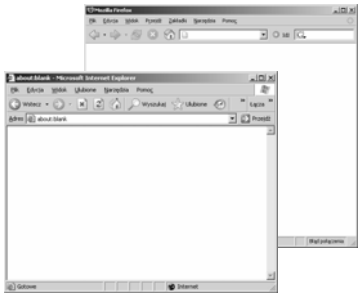
3. Serwer HTTP pobiera żądany dokument z systemu plików i wysyła komunikat odpowiedzi HTTP, do którego dołącza ten dokument, najczęściej zapisany w formacie HTML.

4. Klient HTTP wizualizuje dokument w formie graficznej i zamyka połączenie sieciowe z serwerem HTTP.

Aplikacje WWW

Zadania klienta HTTP

- Inicjowanie połączenia HTTP
- Pobieranie interfejsu użytkownika
- Prezentacja interfejsu użytkownika
- Interakcja z użytkownikiem
- Buforowanie odpowiedzi
- Kryptografia



Wprowadzenie (6)

Kluczowym elementem architektury WWW jest klient HTTP, nazywany również przeglądarką WWW (Web Browser). Klient HTTP jest programem użytkowym, który odpowiada m.in. za inicjowanie połączeń HTTP z serwerem HTTP, wysyłanie żądań pobrania dokumentów, odbieranie dokumentów od serwera HTTP oraz za ich wizualizację. Przykładami programów klientów HTTP są: Microsoft Internet Explorer (www.microsoft.com), Mozilla (www.mozilla.com), Mozilla Firefox (www.mozilla.com), Netscape (www.netscape.com), Opera (www.opera.com). Warto podkreślić, że funkcjonalność tych programów daleko wykracza poza wymaganą funkcjonalność klienta HTTP, zwykle mieszczą one w sobie także funkcje klienta FTP, klienta Gopher, itp.

Klient HTTP obsługuje również interakcję użytkownika końcowego z graficznym interfejsem użytkownika zawartym w pobranym dokumencie. Interfejs ten może zawierać np. takie elementy interakcyjne jak pola tekstowe, pola wyboru, przyciski, łącza, skrypty.

W celu skrócenia czasu odpowiedzi większość klientów HTTP buforuje pobierane dokumenty, zapisując je w lokalnym systemie plików, a następnie wykorzystuje do obsługi identycznych, powtórzonych żądań w przyszłości. Buforowanie dokumentów wymaga stosowania zaawansowanych mechanizmów kontroli spójności w celu uniknięcia ryzyka przedstawiania użytkownikowi nieaktualnej już wersji dokumentu.

W celu podniesienia bezpieczeństwa komunikacji, programy klientów HTTP umożliwiają szyfrowanie połączeń sieciowych z serwerami HTTP.

Zadania serwera HTTP

- Obsługa żądań HTTP
- Rejestracja żądań
- Uwierzytelnianie i kontrola dostępu
- Kryptografia
- Wybór wersji językowej wysyłanych plików

Drugim istotnym elementem architektury WWW jest serwer HTTP, nazywany również serwerem WWW. Serwer HTTP jest programem systemowym, nieprzerwanie pracującym na wyznaczonym komputerze, prowadzącym nasłuch sieciowy w celu odbioru żądań od klientów HTTP. Po otrzymaniu żądania HTTP, serwer HTTP pobiera z lokalnego systemu plików żądany dokument i wysyła go do klienta HTTP. Przykładami programów serwerów HTTP są: Apache (www.apache.org), Jigsaw (www.w3.org), Microsoft Internet Information Services (www.microsoft.com), Sun Java System Web Server (www.sun.com). Coraz częściej serwery HTTP stanowią standardowy składnik systemu operacyjnego.

Do dodatkowych zadań serwerów HTTP należą zwykle: rejestracja obsługiwanych żądań polegająca na ich zapisie w plikach dziennika (log files), uwierzytelnianie i kontrola dostępu użytkowników końcowych za pomocą nazwy i hasła, kryptograficzne szyfrowanie komunikacji sieciowej z klientem HTTP, automatyczny wybór odpowiedniej wersji językowej dokumentu, itp.

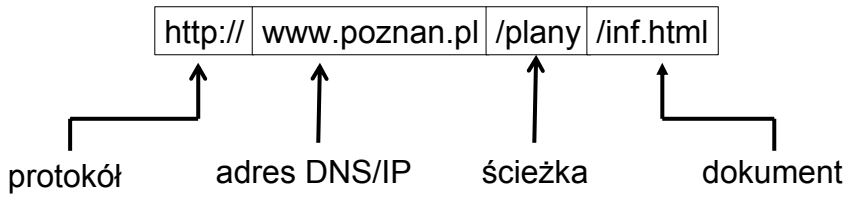
Protokół HTTP

- Oparty na TCP
- Komendy tekstowe
- Transmisja 8-bitowa
- Bezstanowy, bezsesyjny

Komunikacja pomiędzy klientami HTTP a serwerem HTTP jest realizowana za pomocą protokołu HTTP (Hypertext Transfer Protocol). HTTP jest prostym protokołem opartym na TCP, implementującym model żądanie-odpowiedź, korzystającym ze znakowych komend i komunikatów. Umożliwia przesyłanie zarówno dokumentów tekstowych, jak i binarnych. Połączenie HTTP pomiędzy klientem HTTP a serwerem HTTP ma charakter krótkotrwały - jest zamykane po zakończeniu pobierania dokumentu. Protokół ma charakter bezstanowy i bezsesyjny.

Adresy URL

- Wskaźnik do zasobu w sieci Internet



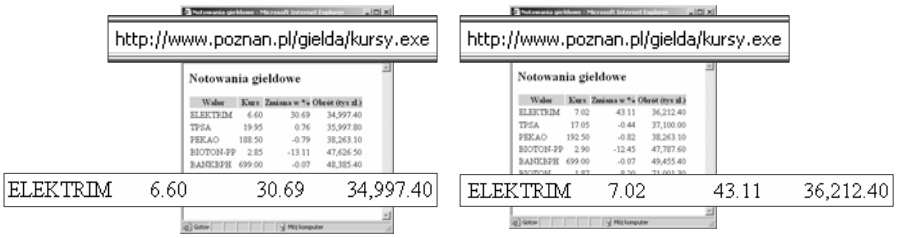
Dokumenty udostępniane przez serwery HTTP są identyfikowane za pomocą adresów URL (Uniform Resource Locator). Adres URL jest łańcuchem znakowym, który zawiera m.in: nazwę protokołu komunikacyjnego (np. HTTP, HTTPS), adres komputera na którym ulokowany jest serwer HTTP i opcjonalnie numer portu nasłuchu serwera HTTP, ścieżkę dostępu do dokumentu, nazwę dokumentu. W adresie URL mogą występować wyłącznie znaki alfanumeryczne i kilka znaków specjalnych. Pozostałe znaki, jak np. znaki spoza ASCII, znaki sterujące ASCII, znaki zarezerwowane ("$", "&", "+", ",", "/", ":", ";", "=", "?", "@") i tzw. znaki niebezpieczne (" ", "'", "<", ">", "£", "%", "{", "}", "|", "\", "^", "~", "[", "]", "`", cudzysłów) powinny być zapisane jako heksadecymalne kody poprzedzone znakiem "%" (URL Encoding). Pełna składnia adresów URL została opisana w RFC 1738.

Adresowanie URL może być stosowane nie tylko w odniesieniu do dokumentów udostępnianych przez serwery HTTP, ale też dla zasobów serwerów FTP, Gopher, Usenet News, itd.

Warto wspomnieć, że pomimo powszechności tej formy adresowania dokumentów, składnia URL wciąż jest przedmiotem zawziętej krytyki. Podstawowy zarzut wiąże się z obecnością fizycznej lokalizacji serwera HTTP w adresie dokumentu, co oznacza, że identyfikator dokumentu ulega zmianie gdy dokument jest przenoszony na inny serwer, oraz że identyczne dokumenty znajdujące się na różnych serwerach posiadają różne identyfikatory. Takie własności są sprzeczne z potocznym rozumieniem pojęcia identyfikatora. Mimo, iż zaproponowano również alternatywną metodę adresowania, nazwaną URI (Uniform Resource Identifier), to jednak w środowiskach WWW wciąż korzysta się z URL (RFC 3986).

Dokumenty statyczne i dynamiczne

- Dokument statyczny - gotowy do pobrania plik w systemie plików serwera HTTP
- Dokument dynamiczny - dokument generowany na żądanie przez program po stronie serwera HTTP
- Aplikacje WWW

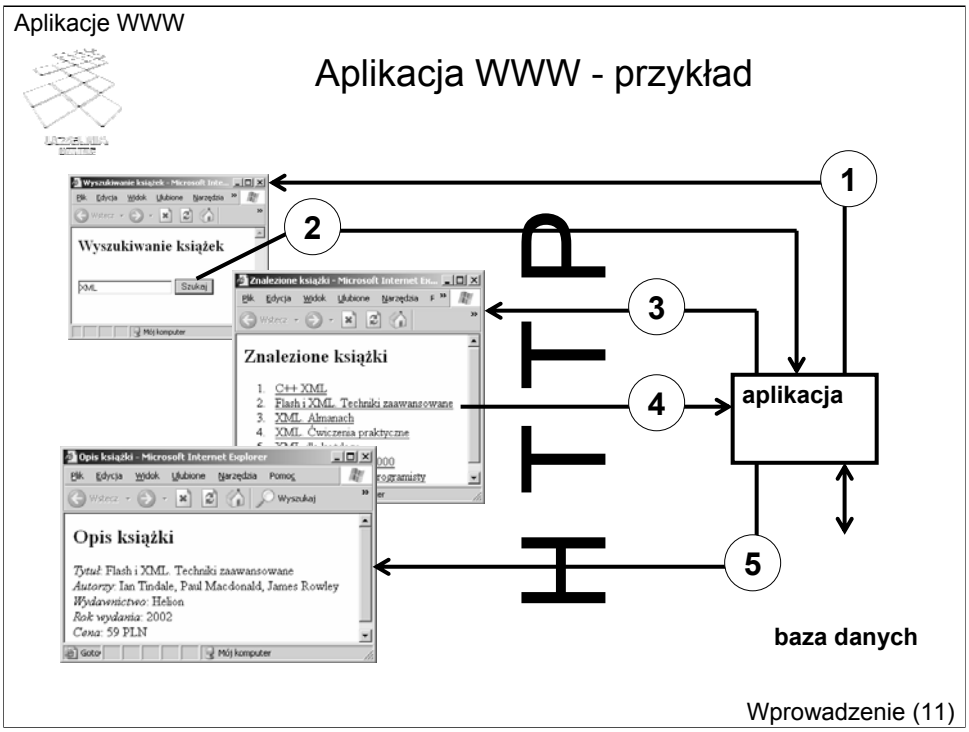


W pierwszych latach rozwoju technologii WWW wszystkie dokumenty udostępniane przez serwery HTTP były zapisane jako gotowe pliki w systemie plików serwera HTTP. Zapewniało to szybki dostęp do ich treści, lecz wymagało modyfikacji plików gdy zmianie ulegały opisywane przez nie dane. Dokumenty takie nazywano dokumentami statycznymi. W 1993 roku pojawiła się koncepcja automatycznego generowania dokumentów przez serwery HTTP (http://hoohoo.ncsa.uiuc.edu/cgi/). Zakładała ona, że po otrzymaniu żądania od klienta, serwer HTTP uruchamia program, który dopiero konstruuje dokument wynikowy. Program taki jest uruchamiany w odpowiedzi na każde żądanie klienta HTTP. Dokumenty generowane przez programy pracujące po stronie serwera HTTP nazywa się dokumentami dynamicznymi.

Przykład dokumentu dynamicznego został przedstawiony na slajdzie. Zauważmy, że dwa żądania zawierające identyczne adresy URL zwróciły dokumenty o różnej treści. Najbardziej prawdopodobnym wyjaśnieniem tego zjawiska jest to, że żądany dokument jest generowany automatycznie przez program znajdujący się po stronie serwera HTTP. Program ten, korzystając ze zmieniającej się zawartości źródła danych, zwrócił różne wyniki podczas każdego z wywołań.

Koncepcja automatycznego generowania dokumentów stała się inspiracją dla powstania nowej kategorii aplikacji komputerowych, nazywanych aplikacjami WWW (web applications) lub aplikacjami wielowarstwowymi (multitier applications). Aplikacje WWW to zestawy programów komputerowych znajdujących się po stronie serwera HTTP, które komunikują się z użytkownikiem końcowym za pomocą dokumentów dynamicznych obsługiwanych przez programy klientów HTTP. Zwykle aplikacje WWW wymagają obecności specjalnego środowiska uruchomieniowego nazywanego serwerem aplikacji. Serwer aplikacji stanowi część serwera HTTP lub jest z nim powiązany.

W dzisiejszym Internecie można znaleźć bardzo wiele zastosowań technologii dokumentów dynamicznych i aplikacji WWW. Wśród nich znajdują się: systemy bankowości internetowej, sklepy internetowe, serwisy aukcyjne, portale internetowe, systemy informujące o połączeniach lotniczych i kolejowych, itd.



Na slajdzie przedstawiono przykład funkcjonowania aplikacji WWW. Interakcja użytkownika z prostą aplikacją księgarni internetowej odbywa się w następujących krokach:

1. Klient HTTP otrzymuje od aplikacji WWW dokument zawierający formularz do wyszukiwania książek.

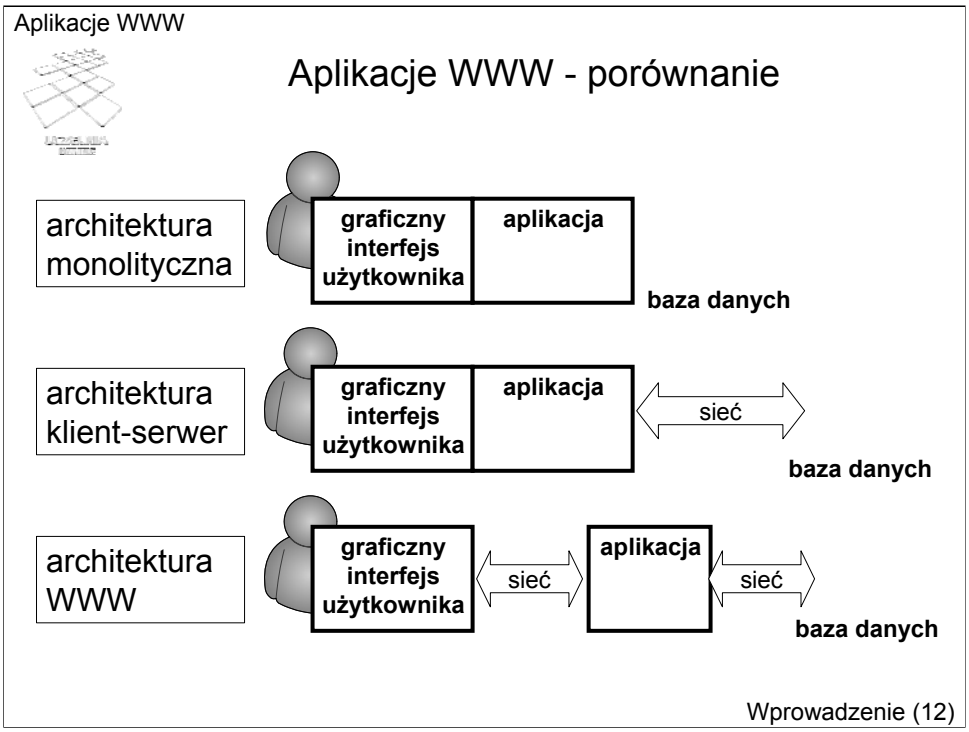
2. Użytkownik końcowy wprowadza tytuł szukanej książki i naciska przycisk "Szukaj". Klient HTTP wysyła żądanie do aplikacji, dołączając do żądania słowo kluczowe wprowadzone przez użytkownika.

3. Aplikacja WWW przeszukuje bazę danych w celu znalezienia tytułów książek zawierających podane słowo kluczowe. Aplikacja generuje dokument dynamiczny, w którym umieszcza znalezione tytuły. Dokument jest przesyłany do klienta HTTP i przedstawiany użytkownikowi końcowemu.

4. Użytkownik końcowy wybiera interesujący go tytuł za pomocą kliknięcia w łącznik. Aplikacja WWW otrzymuje kolejne żądanie.

5. Aplikacja WWW pobiera z bazy danych opis wybranej książki i generuje dokument dynamiczny, w którym umieszcza informacje szczegółowe o książce. Dokument jest przesyłany do klienta HTTP i przedstawiany użytkownikowi końcowemu.

Podkreślmy, że interakcja użytkownika końcowego z aplikacją WWW ma charakter dokumentowy, tzn. odpowiedzią na działanie użytkownika jest zawsze kompletny dokument generowany przez zdalną aplikację.

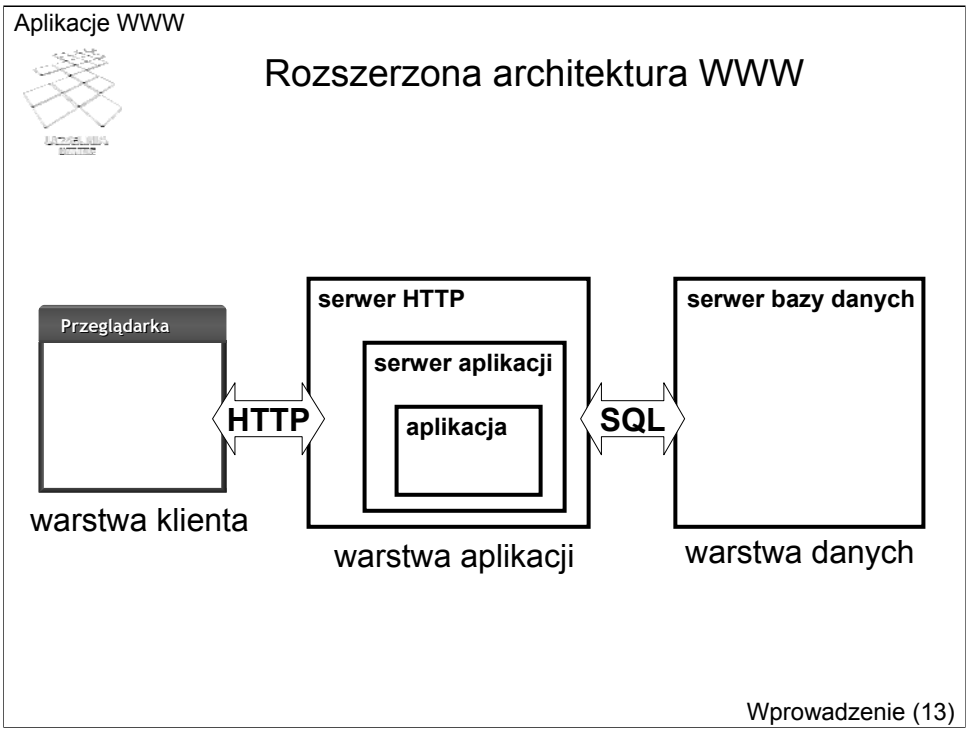


Na slajdzie dokonano historycznego porównania trzech architektur aplikacyjnych: architektury monolitycznej, architektury klient-serwer i architektury WWW:

1. Według architektury monolitycznej, oprogramowanie obsługi graficznego interfejsu użytkownika, oprogramowanie właściwej aplikacji oraz oprogramowanie dostępu do danych stanowią pojedynczy moduł programowy, zwykle uruchamiany jako pojedynczy proces systemu operacyjnego. Całość przetwarzania danych odbywa się na komputerze użytkownika końcowego. Rozwiązanie takie wyklucza współbieżność dostępu do danych przez wielu użytkowników oraz stawia wysokie wymagania wydajnościowe komputerowi użytkownik końcowego.

2. Według architektury klient-serwer, oprogramowanie obsługi graficznego interfejsu użytkownika i oprogramowanie właściwej aplikacji stanowią pojedynczy moduł programowy, zwykle uruchamiany jako pojedynczy proces systemu operacyjnego, natomiast oprogramowanie dostępu do danych jest umieszczone na oddzielnym, dedykowanym komputerze. Komunikacja pomiędzy aplikacją a bazą danych odbywa się przez sieć komputerową. Rozwiązanie takie umożliwia współbieżny dostęp do danych przez wielu użytkowników i wiele aplikacji oraz obniża wymagania wydajnościowe dla komputera użytkownika końcowego.

3. Według architektury WWW, na komputerze użytkownika końcowego funkcjonuje wyłącznie oprogramowanie obsługi graficznego interfejsu użytkownika, natomiast oprogramowanie właściwej aplikacji oraz oprogramowanie dostępu do danych znajdują się na oddzielnych, dedykowanych komputerach. Rozwiązanie takie umożliwia współbieżny dostęp do danych, współbieżny dostęp do aplikacji oraz drastycznie obniża wymagania wydajnościowe dla komputera użytkownika końcowego. Ponadto, jeśli potraktować oprogramowanie klienta HTTP jako składnik systemu operacyjnego komputera użytkownika końcowego, to architektura WWW praktycznie nie wymaga instalowania jakiegokolwiek oprogramowania aplikacyjnego na tym komputerze.



Realizacja aplikacji WWW wymaga zastosowania rozszerzonej architektury WWW, w skład której wchodzą trzy programowe warstwy funkcjonalne:

1. Warstwa klienta, odpowiedzialna za wizualizację graficznego interfejsu użytkownika i interakcję z użytkownikiem końcowym. Warstwa ta oparta jest na tradycyjnym kliencie HTTP.

2. Warstwa aplikacji, odpowiedzialna za generowanie dokumentów dynamicznych w odpowiedzi na żądania klientów. Warstwa ta składa się z serwera HTTP i z tzw. serwera aplikacji (application server), stanowiącego środowisko uruchomieniowe dla aplikacji generujących dokumenty dynamiczne.

3. Warstwa danych, odpowiedzialna za udostępnianie informacji osadzanych w dokumentach dynamicznych. Warstwa ta składa się z serwera bazy danych odpowiadającego na wywołania w języku SQL.

Aplikacje WWW

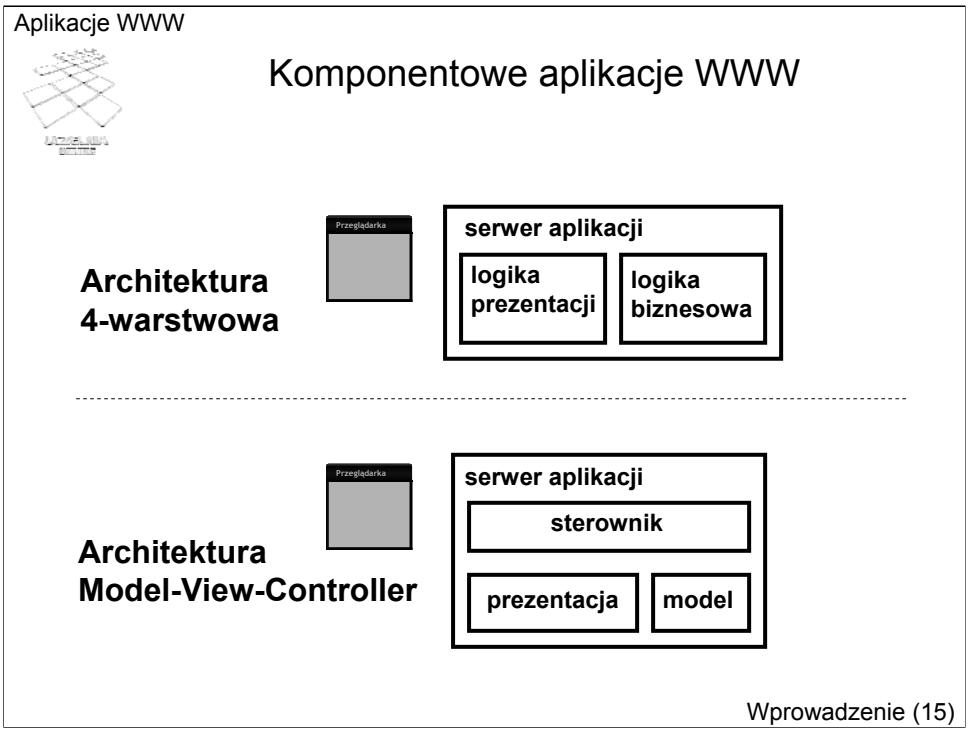
Serwer aplikacji

- Platforma dla uruchamiania aplikacji WWW
- Obsługa komunikacji z warstwą klienta i warstwą danych
- Usługi systemowe
  - transakcje
  - rejestracja żądań
  - autoryzacja dostępu
  - itd.

Wprowadzenie (14)

Kluczowym składnikiem rozszerzonej architektury WWW, umożliwiającej wykonywanie aplikacji WWW jest serwer aplikacji. Serwer aplikacji jest oprogramowaniem o charakterze systemowym, które odpowiada m.in. za obsługę komunikacji z warstwą klienta i warstwą danych. Dzięki temu, programista tworzący aplikacje WWW nie musi samodzielnie implementować kodu obsługi protokołu HTTP (z klientem HTTP) ani kodu obsługi komunikacji SQL (z serwerem bazy danych). Ponadto, serwery aplikacji zwykle wyręczają programistów z konieczności implementacji powtarzalnych, typowych funkcji aplikacyjnych, jak np. obsługa transakcji HTTP, rejestracja żądań w plikach dziennika, autoryzacja dostępu użytkowników do aplikacji, itd.

Rynek serwerów aplikacji jest bardzo duży, znajdują się na nim zarówno produkty całkowicie komercyjne, jak i produkty klasy open-source. Do najpopularniejszych serwerów aplikacji należą: BEA Weblogic (www.beasys.com), Borland Visibroker (www.borland.com), Caucho Resin (www.caucho.com), JBOSS (www.jboss.org), IBM WebSphere (www.ibm.com), Jakarta Tomcat (jakarta.apache.org), Oracle Application Server (www.oracle.com), Orion (www.orionserver.com), Sun Java Web Server (www.sun.com), W3 Jigsaw (www.w3.org), itd. Warto zaznaczyć, że często serwery aplikacji zawierają w sobie funkcjonalność serwera HTTP i dzięki temu potrafią w kompletny sposób obsłużyć wymagania warstwy aplikacji.



Aplikacje WWW są najczęściej budowane jako środowiska komponentowe, w których poszczególne komponenty rozdzielają między sobą nie tylko kroki procesów biznesowych, ale także rodzaje funkcji systemowych. Powszechnie wykorzystuje się dwa podejścia do separacji funkcji systemowych komponentów aplikacji WWW:

1. Architektura 4-warstwowa (4-tier architecture) zakłada, że komponenty aplikacji WWW dzielą się na dwie grupy: komponenty logiki prezentacji i komponenty logiki biznesowej. Komponenty logiki prezentacji odpowiadają za przyjmowanie żądań od klientów HTTP, wywoływanie funkcji komponentów logiki biznesowej, generowanie dokumentów dynamicznych i wypełnianie ich danymi przekazywanymi przez komponenty logiki biznesowej. Komponenty logiki biznesowej odpowiadają wyłącznie za realizację procesów biznesowych i komunikację z bazą danych. Architektura ta nazywana jest 4-warstwową, ponieważ definiuje 4 warstwy aplikacji WWW: klienta, logiki prezentacji, logiki biznesowej, danych.

2. Architektura Model-View-Controller zakłada, że komponenty aplikacji WWW dzielą się na trzy grupy: komponenty sterujące (controller), komponenty prezentacji (view) i komponenty modelu (model). Komponenty prezentacji odpowiadają za generowanie dokumentów dynamicznych i wypełnianie ich danymi przekazywanymi przez komponenty modelu. Komponenty modelu odpowiadają za realizację procesów biznesowych i komunikację z bazą danych. Komponenty sterujące odpowiadają za przyjmowanie żądań od klientów HTTP i koordynację ich obsługi, polegającą na wywoływaniu funkcji komponentów modelu i prezentacji. Architektura ta jest w zasadzie dalszym rozwinięciem architektury 4-warstwowej.

Zalety i wady aplikacji WWW

- Zalety
  - Niski koszt urządzeń dostępowych
  - Wygoda administrowania aplikacjami
  - Łatwość użytkowania
  - Ochrona własności intelektualnej
- Wady
  - Trudność wytwarzania oprogramowania
  - Uproszczony interfejs użytkownika
  - Koszt serwerów

Technologia aplikacji WWW ma wiele zalet, które wyróżniają ją na tle technologii aplikacji monolitycznych i aplikacji klient-serwer. Na uwagę pierwszoplanową zasługuje fakt, że użytkownik korzystający z aplikacji WWW posługuje się wyłącznie programem klienta HTTP, który zwykle cechuje się dość niskimi wymaganiami sprzętowymi. Dzięki temu, dostęp do aplikacji WWW jest możliwy nie tylko za pomocą prostej stacji roboczej, ale też za pomocą palmtopa czy telefonu komórkowego. Z drugiej strony, łatwość użytkowania programu klienta HTTP przekłada się na łatwość użytkowania aplikacji wykonanej w technologii WWW. Kolejny aspekt dotyczy wygody administrowania aplikacjami - ponieważ rezydują one na pojedynczym komputerze (serwerze aplikacji), to wszelkie prace instalacyjne, uaktualniające, pielęgnacyjne dotyczą tylko tego jednego komputera. Warte podkreślenia jest też zagadnienie ochrony własności intelektualnej twórców oprogramowania. Ponieważ oprogramowanie nie znajduje się na komputerze użytkownika końcowego, to zmniejsza się ryzyko jego kradzieży.

Niestety, wdrożenie technologii aplikacji WWW pociąga też za sobą pewne koszty i kompromisy. Wytwarzanie oprogramowania wymaga od programistów posiadania dodatkowych kwalifikacji. Ze względu na ograniczone możliwości interakcji z użytkownikiem, pewnemu upośledzeniu ulega konstrukcja graficznego interfejsu użytkownika. Nie należy również zapominać, że wdrożenie aplikacji WWW pociąga za sobą konieczność zakupu mocnego obliczeniowo serwera odpowiedzialnego za wykonywanie aplikacji, a często też konieczność zakupu komercyjnego oprogramowania serwera aplikacji.

Język HTML

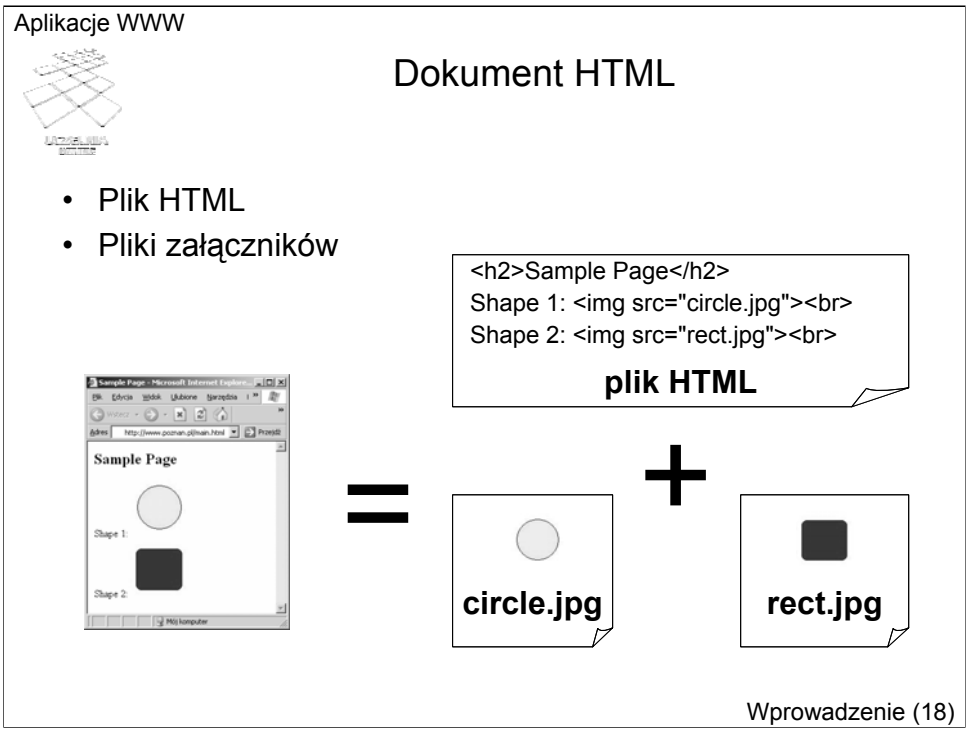
- Zapis treści i opis układu graficznego dokumentów
- Rozkazy formatujące zapisane w postaci znaczników
- Większość znaczników występuje w parach: znacznik otwierający i znacznik zamykający
- Znaczniki mogą posiadać atrybuty sterujące
- Komentarze: "<!--" i "-->"

Witamy w ****Poznaniu****

HTML (Hypertext Markup Language) to najważniejszy język definicji dokumentów dla klientów HTTP. Przyjmuje się, że został opracowany przez Tima Bernersa-Lee w roku 1990, częściowo w oparciu o język SGMLguid, który z kolei stanowił lokalną odmianę języka SGML stosowaną w CERN. Prawdopodobnie najstarszy dokument HTML dostępny w Internecie został utworzony 13 listopada 1990 roku: http://www.w3.org/History/19921103-hypertext/hypertext/WWW/Link.html (http://infomesh.net/html/history/early/).

HTML umożliwia zapis treści dokumentu i równocześnie opis jego układu graficznego. Dokument HTML to plik tekstowy, z ewentualnymi załącznikami, w którym znajduje się tekstowa treść przeplatana z rozkazami formatującymi, zapisanymi w formie tzw. znaczników (tags). Każdy znacznik HTML przyjmuje postać słowa kluczowego otoczonego ostrymi nawiasami (znakami "<" i ">"). Większość znaczników HTML występuje w parach, na które składają się znacznik otwierający i znacznik zamykający. Znacznik zamykający różni się od otwierającego wyłącznie znakiem ukośnika poprzedzającym słowo kluczowe. Znaczniki HTML mogą posiadać atrybuty sterujące, które wpływają na ich funkcjonowanie. Atrybuty sterujące są umieszczane wewnątrz znacznika, za słowem kluczowym. Język HTML dopuszcza też możliwość stosowania komentarzy, będących blokami tekstu ignorowanymi przez programy klientów HTTP. Komentarze otacza się znakami "<!--" i "-->".

Przykład prostego znacznika HTML przedstawiono w dolnej części slajdu. Znacznik "b" służy do wyświetlenia fragmentu tekstu z użyciem czcionki pogrubionej (bold). Znacznik ten występuje w parach: rozpoczyna tryb czcionki pogrubionej, a kończy tryb czcionki pogrubionej.



Dokument HTML może składać się z jednego lub wielu plików. Zestawy wieloplikowe są konieczne m.in. wtedy, gdy dokument zawiera elementy binarne w postaci np. obrazów graficznych lub gdy dokument ma tzw. strukturę ramkową. W takich przypadkach każdy obraz graficzny lub treść ramki są zapisane w odrębnych plikach, a główny plik HTML zawiera stosowne wskaźniki. Adresem URL wieloplikowego dokumentu HTML jest adres URL pliku HTML lub pliku definicji ramek. Za pobranie kompletnego zbioru plików odpowiada klient HTTP.

Na slajdzie przedstawiono przykład wieloplikowego dokumentu HTML.. Dokument składa się z pliku HTML, zawierającego opis układu graficznego dokumentu i jego treść tekstową, oraz z dwóch plików JPG zawierających obrazy graficzne osadzone w dokumencie. Powiązania plików graficznych JPG z plikiem HTML są implementowane poprzez osadzenie wskaźników adresowych w treści pliku HTML. Proces pobierania takiego dokumentu przez klienta HTTP przebiega w następujących krokach:

1. Użytkownik przekazuje klientowi HTTP adres URL pliku HTML, np. http://www.poznan.pl/main.html

2. Klient HTTP wysyła do serwera HTTP żądanie pobrania pliku HTML, np. o nazwie main.html

3. Klient HTTP otrzymuje od serwera HTTP plik HTML i analizuje jego zawartość w celu wyszukania wskaźników do plików załączników.

4. Klient HTTP wysyła do serwera HTTP żądania pobrania plików załączników, np. circle.jpg i rect.jpg.

5. Klient otrzymuje od serwera HTTP pliki załączników i wizualizuje kompletny dokument.

Warto podkreślić, że przedstawiona interakcja klienta HTTP z serwerem HTTP może wymagać trzykrotnego nawiązania połączenia HTTP, w przypadku stosowania protokołu HTTP 1.0, lub tylko jednokrotnego lecz wielożądaniowego, w przypadku stosowania protokołu HTTP 1.1 i mechanizmu Persistent Connections (patrz wykład "Protokół HTTP").

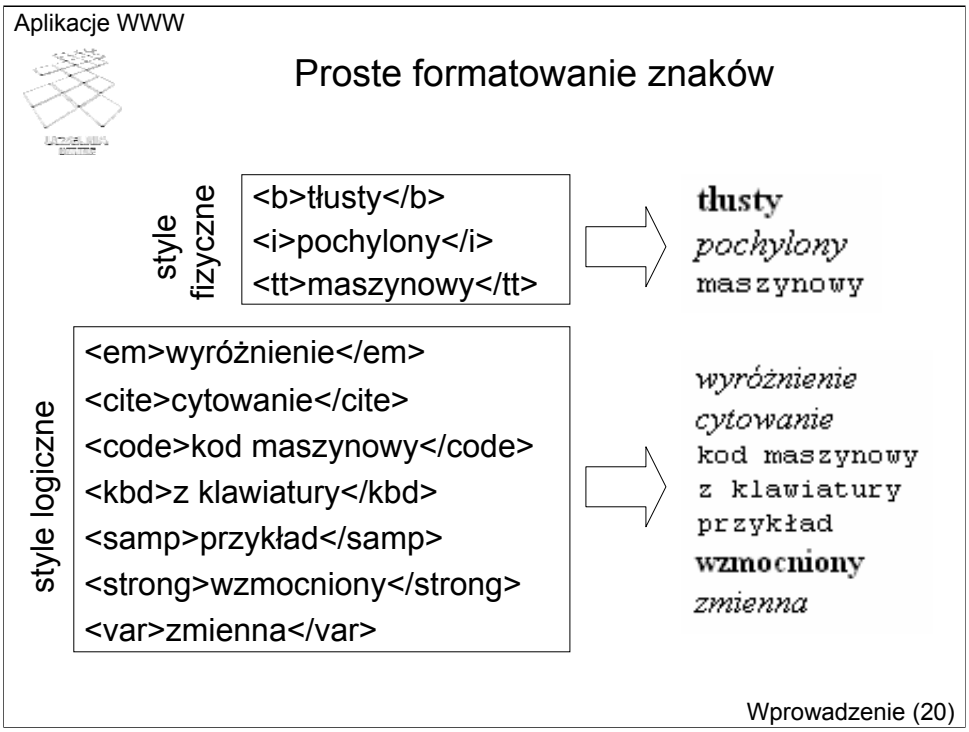
Struktura pliku HTML

```
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
 <HEAD>
  <TITLE>
    tytuł dokumentu
  </TITLE>
   pozostałe metadane
 </HEAD>
 <BODY>
   treść dokumentu
 </BODY>
</HTML>
```


Każdy plik HTML powinien posiadać dwumodułową strukturę, w której występuje nagłówek dokumentu (ang. head) i ciało dokumentu (ang. body). Nagłówek dokumentu jest otoczony znacznikami <head> oraz </head> i zawiera metadane opisujące dokument oraz jego tytuł. Ciało dokumentu jest otoczone znacznikami <body> oraz </body> i zawiera właściwą treść dokumentu widoczną dla użytkownika końcowego. Nagłówek wraz z ciałem są otoczone znacznikami <html> i </html>.

Dostępne na rynku programy klientów HTTP są dość tolerancyjne wobec naruszania przedstawionej struktury plików HTML, jednak ze względu na możliwości zautomatyzowanego przetwarzania plików HTML wskazane jest przestrzeganie tych zaleceń.

Nazwy plików HTML powinny posiadać rozszerzenie ".html" lub ".htm".

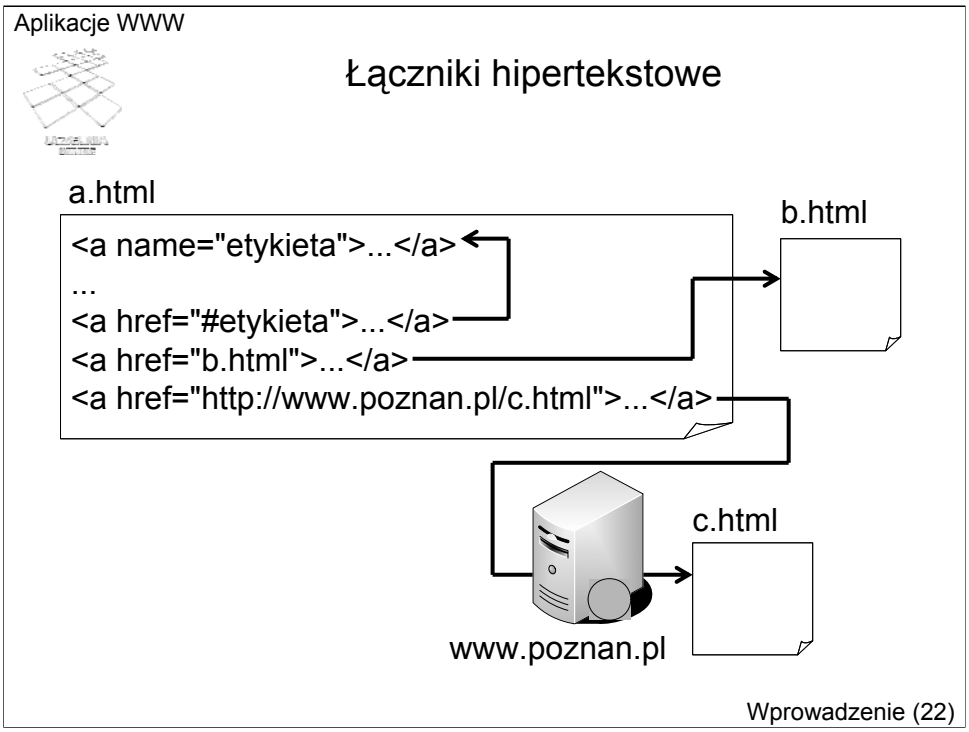


Do formatowania znaków służą dwie grupy znaczników HTML: znaczniki formatujące fizyczne i znaczniki formatujące logiczne. Znaczniki formatujące fizyczne pozwalają autorowi dokumentu na narzucenie określonego wyglądu tekstu, np. jego wytłuszczenia, pochylenia. Znaczniki formatujące logiczne umożliwiają wskazanie, że dany fragment tekstu powinien być traktowany odmiennie, lecz decyzję o sposobie wyróżnienia tekstu pozostawiają programowi klienta HTTP. Przykładowo, znacznik wyróżnienia jest zwykle interpretowany jako polecenie wyświetlenia znaków pismem pochyłym.

Znaki specjalne

&lt;	<
&gt;	>
&amp;	&
&copy;	©
&frac12;	½
&Oacute;	Ó
&euro;	€

Niektóre znaki alfabetu są znakami zastrzeżonymi w języku HTML: znak mniejszości, znak większości, znak "&", itp. W celu uzyskania w dokumencie znaków zastrzeżonych należy zastosować sekwencje specjalne HTML. Każda sekwencja specjalna rozpoczyna się znakiem "&", po którym następuje komenda, a następnie znak ";". Sekwencje specjalne mogą także służyć do uzyskiwania innych nietypowych znaków. Klika przykładów sekwencji specjalnych przedstawiono na slajdzie.



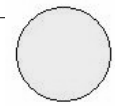
Jednym z fundamentalnych mechanizmów języka HTML jest mechanizm łącznika hipertekstowego (ang. hyperlink). Łącznik hipertekstowy jest elementem graficznego interfejsu użytkownika służącym do nawigacji wewnątrz pojedynczego dokumentu HTML lub pomiędzy różnymi dokumentami HTML. Do definiowania łączników hipertekstowych służy znacznik <a> użyty z atrybutem "href", natomiast znacznik <a> z atrybutem "name" służy do zdefiniowania adresu docelowego, tzw. kotwicy, do którego prowadzi łącznik hipertekstowy. Łącznik hipertekstowy może prowadzić do początku innego dokumentu HTML, do kotwicy wewnątrz tego samego dokumentu HTML lub do kotwicy wewnątrz innego dokumentu HTML. Pomiędzy otwierającym a zamykającym znacznikiem <a> umieszcza się tekst, który służy użytkownikowi do kliknięcia i wykonania skoku.

Wartość atrybutu "href" składa się z dwóch części, z których każda jest opcjonalna: adresu URL dokumentu docelowego i nazwy kotwicy w dokumencie docelowym. Nazwa kotwicy musi zostać poprzedzona znakiem "#".

Na slajdzie przedstawiono dokument zawierający trzy łączniki HTML: łącznik do kotwicy "etykieta" w tym samym dokumencie, łącznik do innego dokumentu znajdującego się w tym samym katalogu dyskowym i łącznik do innego dokumentu znajdującego się na innym komputerze.

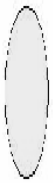
Załączniki graficzne

```
<img
  src="circle.jpg">
```

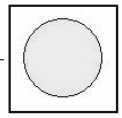


```
<img
  src="circle.jpg"
  alt="Kółko">
```


```
<img
  src="circle.jpg"
  width=50
  height=200>
```



```
<a href="b.html">
  <img  src="circle.jpg">
</a>
```



Język HTML umożliwia osadzanie w dokumencie załączników graficznych, np. fotografii, ikon, przycisków. Każdy załącznik graficzny jest odrębnym plikiem i jest odrębnie pobierany przez program klienta HTTP. Do osadzania załączników graficznych służy znacznik <img> z atrybutem "src", którego wartością jest adres URL pliku graficznego. Adres ten może być absolutny lub względny. Dodatkowo, znacznik <img> może być wyposażony w atrybuty "width" i "height", które pozwalają na zmianę rozmiarów oryginalnego obrazu graficznego. Ich wartości wyrażane są w pikselach. Kolejnym przykładem opcjonalnego atrybutu znacznika <img> jest "alt", który służy do dołączenia tekstowego komentarza do obrazu graficznego. Komentarz ten może być wyświetlany przez program znakowego klienta HTTP lub jako podpowiedź pojawiająca się po zatrzymaniu wskaźnika myszy.

Obrazy graficzne osadzone w dokumencie HTML mogą też pełnić rolę łączników hipertekstowych. W takim przypadku pomiędzy otwierającym a zamykającym znacznikiem <a> umieszcza się znacznik <img>. Wówczas kliknięcie obrazu graficznego powoduje skok do adresu docelowego.

Na slajdzie przedstawiono cztery przykłady użycia znaczników <img> wraz z wynikami ich działania.

Tabele

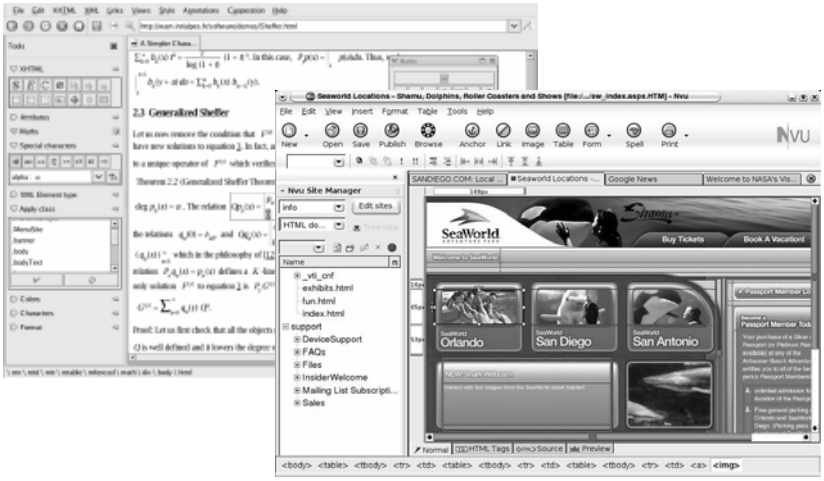
```
<table border=3>
 <tr><th>Miasto</th><th>Mieszkańcy</th></tr>
 <tr><td>Kraków</td><td>800.000</td></tr>
 <tr><td>Poznań</td><td>500.000</td></tr>
 <tr><td>Warszawa</td><td>1.900.000</td></tr>
</table>
```

Miasto	Mieszkańcy
Kraków	745.000
Poznań	585.000
Warszawa	1.638.000

Tabele są popularną konstrukcją w języku HTML, umożliwiającą nie tylko prezentację danych liczbowych, ale również sterowanie układem graficznym całego dokumentu. Do definiowania tabel służą m.in. znaczniki <table>, <tr>, <th> i <td>. Znaczniki <table> otaczają całą definicję tabeli. Znaczniki <tr> służą do definiowania zawartości kolejnych wierszy tabeli. Znaczniki <th> i <td> definiują komórki tabeli - odpowiednio: komórki tytułowe i komórki danych. Tabela może posiadać obramowanie, a jego grubość jest określana za pomocą atrybutu "border" znacznika <table>.

Na slajdzie przedstawiono przykładową definicję tabeli oraz jej obraz wyświetlony przez program klienta HTTP.

Narzędzia dla autorów



W celu ułatwienia tworzenia dokumentów HTM opracowano wiele programów narzędziowych nazywanych edytorami HTML, np. Nvu (http://www.nvu.com), Amaya (http://www.w3.org/Amaya), Netscape Composer (http://wp.netscape.com/communicator/composer). Programy te umożliwiają budowę dokumentów HTML w sposób przypominający opracowywanie tekstu za pomocą edytora tekstu, zwalniając autora z konieczności pamiętania i manualnego wprowadzania znaczników HTML. Ponadto, umożliwiają automatyczną kontrolę poprawności składniowej dokumentów, walidację łączników hipertekstowych, definicję arkuszy stylistycznych, itp.

HTML: zalety i wady

- Zalety
  - prostota składni
  - dostępność przeglądarek
- Wady
  - brak szablonów/wzorców
  - brak separacji formy i treści
  - ubogi graficznie

Wprowadzenie języka HTML rozpoczęło technologiczną rewolucję w Internecie, w wyniku której dziś użytkownicy posiadają dostęp do miliardów dokumentów. Niewątpliwymi zaletami tego języka są: prostota składni, przekładająca się na łatwość tworzenia dokumentów, oraz powszechna dostępność programów przeglądarek umożliwiających użytkownikom pobieranie i oglądanie dokumentów. Z drugiej strony, język HTML stał się źródłem wielu problemów wynikających z przyjętych założeń. Brak mechanizmów globalnego formatowania dokumentów HTML powoduje, że trudno jest zapewnić jednolitość graficzną w dużych internetowych systemach informacyjnych. Przemieszanie treści dokumentu ze znacznikami formatowania graficznego sprawia, że niezwykle trudno jest automatycznie przetwarzać lub przeszukiwać dokumenty HTML, a także oferować wiele wariantów graficznych tego samego dokumentu. Ponadto, możliwości graficznego formatowania dokumentów HTML są dość mocno ograniczone, np. w porównaniu z możliwościami funkcjonalnymi współczesnych edytorów tekstów.

Podsumowanie

- Podstawowe składniki architektury WWW to: klient HTTP, serwer HTTP, protokół HTTP
- Aplikacje WWW opierają się na automatycznym generowaniu dokumentów
- Aplikacje WWW wymagają serwera aplikacji
- Aplikacje WWW są zwykle komponentowe
- HTML
  - język znaczników
  - dokument HTML = plik HTML + załączniki
  - narzędzia edycyjne
  - liczne wady i ograniczenia

Podstawowa architektura WWW to architektura dwuwarstwowa, składająca się z klienta HTTP i serwera HTTP. Komunikacja pomiędzy klientem HTTP a serwerem HTTP jest realizowana za pomocą protokołu sieciowego HTTP. Rozszerzona architektura WWW to architektura trójwarstwowa, w której pojawia się aplikacja, nazywana aplikacją WWW, automatycznie generująca dokumenty w odpowiedzi na żądania klientów HTTP. Aplikacja WWW jest wykonywana w środowisku uruchomieniowym nazywanym serwerem aplikacji. Aplikacje WWW są zwykle budowane według modeli komponentowych.

Język HTML to język znaczników umożliwiający tworzenie multimedialnych dokumentów udostępnianych w sieci Internet. Pojęcie dokumentu HTML oznacza zwykle plik HTML, któremu mogą towarzyszyć pliki załączników, np. obrazów graficznych. Pomimo prostoty składni języka HTML, w praktyce do tworzenia dokumentów wykorzystuje się zaawansowane narzędzia edytorskie, pracujące w trybie WYSIWYG. Język HTML jest w obecnych czasach często oceniany negatywnie ze względu na liczne wady i ograniczenia.

Materiały dodatkowe

- "Uniform Resource Locators (URL)", RFC 1738
- "HTML 4.0 Specification", http://www.w3.org/TR/REC-html40
- "Kurs języka HTML - poradnik webmastera", http://webmaster.helion.pl/kurshtml/