

## Wybrane technologie wykorzystywane do tworzenia stron internetowych

W procesie obsługi żądań użytkowników w sieciach komputerowych, w tym także w Internecie, realizowane są różne zadania cząstkowe. Użytkownicy, korzystając z oprogramowania klienckiego (np. przeglądarki internetowej) formułują żądania (np. poprzez wpisanie adresu strony). Następnie są one analizowane, przetwarzane i przesyłane do komputerów zawierających zasoby, które mają zostać zaprezentowane użytkownikom. Zasoby te mogą być różnego rodzaju: pliki, strony internetowe, dane przechowywane w bazach danych. Wybrane dane i informacje, po przetworzeniu, są kierowane do aplikacji użytkownika, który sformułował żądanie, gdzie następuje ich prezentacja w sposób dla niego zrozumiały.

Aplikacje internetowe są projektowane w oparciu o architekturę klient-serwer, której istota polega na tym, że część zadań składających się na realizację żądań użytkowników jest wykonywanych na komputerze klienta, część na innym komputerze (lub innych komputerach), który w tym zestawie pełni rolę serwera. Zależnie od tego, jak dużo czynności realizowanych jest na komputerze klienta, w architekturze tej możemy spotkać rozwiązania typu cienki klient (ang. *thin client*) oraz gruby klient (ang. *thick client*). Z pierwszym przypadkiem mamy do czynienia, gdy większość operacji wykonywana jest na serwerze, z drugim, gdy więcej zadań realizowanych jest na komputerze klienta. W skrajnym przypadku, w architekturze opartej na cienkim kliencie warstwa kliencka może być zredukowana jedynie do klawiatury umożliwiającej formułowanie żądań przez użytkowników i monitora, na którym wyświetlane są odpowiedzi serwera, a wszystkie zadania składające się proces przygotowania i prezentacji treści są wykonywane na serwerze.

Architektura klient-serwer może wystąpić w trzech postaciach, jako dwuwarstwowa, trójwarstwowa lub wielowarstwowa. W przypadku rozwiązania dwuwarstwowego składowanie i przetwarzanie danych oraz wykonywanie części aplikacji odbywa się w jednym module (na serwerze), a zadania obsługi żądań użytkowników oraz prezentacji są wykonywane w drugim (na komputerze klienta). W architekturze trójwarstwowej, która jest najczęściej stosowana, istnieją trzy moduły (warstwy): prezentacji, logiki biznesowej (pośrednia) oraz przechowywania i przetwarzania danych. Warstwa prezentacji jest usytuowana na komputerze klienta i jest odpowiedzialna za realizację zadań składających się na komunikację z użytkownikami, stanowi więc tzw. interfejs użytkownika. Warstwa danych jest przeznaczona do przechowywania i udostępniania danych gromadzonych w bazach

danych oraz różnego rodzaju plików. Jej zadaniem jest także przetwarzanie zapytań będących wynikiem żądań formułowanych przez użytkowników, a więc wyszukiwanie i przesyłanie danych w odpowiedzi na otrzymane zapytania. Warstwa pośrednia stanowi swego rodzaju łącznik między tymi dwoma. Przetwarza żądania użytkowników, kieruje je do odpowiednich systemów w warstwie danych, pobiera stamtąd dane oraz dokonuje niezbędnych obliczeń przed przesłaniem ich do warstwy klienta (komputera użytkownika). W przypadku architektury wielowarstwowej występują kolejne moduły, a zadania składające się na proces komunikacji podlegają dalszemu podziałowi.

Architektura klient-serwer istniała, zanim pojawił się Internet, systemy tego typu były więc tworzone wcześniej. Jednym z istotnych problemów była wtedy konieczność rozpowszechniania oprogramowania klienckiego wszystkim zainteresowanym użytkownikom. Programiści, tworząc tego typu systemy, pisali oprogramowanie dla serwera i umieszczali je na serwerze oraz dla klientów i rozsyłali je wszystkim zainteresowanym. Użytkownicy komputerów, którzy chcieli korzystać ze stworzonego w ten sposób systemu, musieli posiadać na swoich komputerach oprogramowanie klienckie. Jednak raz zaprojektowany i wdrożony system musiał być niejednokrotnie modyfikowany. Jeśli zmiana dotyczyła aplikacji działającej na serwerze, nie było problemu, po jej modyfikacji wszyscy użytkownicy, bez konieczności podejmowania jakiegokolwiek akcji, mieli dostęp do nowej wersji. Jeśli jednak zmianie uległo oprogramowanie klienckie, projektanci musieli zadbać o to, aby wszyscy użytkownicy otrzymali nową wersję a ponadto użytkownicy ci musieli ją zainstalować w miejsce poprzedniej, co niejednokrotnie było nie lada problemem, szczególnie, jeśli nie mieli oni w tej kwestii doświadczenia. Internet zmienił sytuację w tym zakresie. Obecnie tworzone aplikacje są zwykle oparte na technologiach internetowych, dzięki czemu projektanci systemów internetowych (czy intranetowych), które siłą rzeczy działają w architekturze klient-serwer, nie muszą martwić się o pisanie i dystrybucję aplikacji przeznaczonych dla klientów, co jest niewątpliwą zaletą tego rozwiązania. Użytkownicy komputerów mający dostęp do Internetu posiadają bowiem przeglądarkę internetową, która stanowi oprogramowanie klienckie. Pojawia się wprawdzie problem związany z różnorodnością samych przeglądarek i koniecznością pisania aplikacji tak, aby mogły być one poprawnie interpretowane przez wszystkie lub przynajmniej te najpopularniejsze, ale jest on znacznie mniej kłopotliwy, niż sytuacja, w której oprogramowanie klienckie musi być dostarczone wszystkim zainteresowanym użytkownikom, tym bardziej, że w podstawowych funkcjach przeglądarki nie różnią się od siebie lub różnią się nieznacznie.

Najczęściej wykorzystywaną usługą internetową jest *www* (ang. *web service*). Od momentu zgłoszenia żądania użytkownika do wyświetlenia zawartości strony internetowej cały proces przebiega w zależności od tego, czy konieczna jest komunikacja z bazą danych. Jeśli nie jest ona wymagana, plik, który ma zostać wyświetlony w przeglądarce, ma zwykle rozszerzenie *htm* (lub *html*). Po wpisaniu adresu strony żądanie zostaje przesłane do serwera, który wyszukuje odpowiedni plik oraz wszystkie inne, które są potrzebne do poprawnego wyświetlenia zawartości strony (np. pliki CSS<sup>1</sup>, skrypty JavaScript<sup>2</sup> lub rysunki i inne pliki pomocnicze) i przesyła je do przeglądarki, która jest odpowiedzialna za poprawne wyświetlenie zawartości otrzymanych w ten sposób zasobów.

Jeśli do poprawnego wyświetlenia strony niezbędne jest pobranie danych z bazy danych, wymagana jest komunikacja z SZBD. Proces jest wtedy nieco bardziej złożony. Załóżmy, że komunikacja z bazą danych odbywa się przy wykorzystaniu języka PHP<sup>3</sup>. Plik zawierający opis strony internetowej musi mieć wtedy rozszerzenie *php*<sup>4</sup> (por. rys. 1). W przeciwnym przypadku nie zostanie na serwerze uruchomiony tzw. interpreter<sup>5</sup> PHP odpowiedzialny za wykonanie kodu PHP. Po otrzymaniu żądania od przeglądarki internetowej serwer odnajduje odpowiedni plik (na rysunku 1 jest to *plik.php*). Jeśli ma on rozszerzenie *php*, uruchamiany jest automatycznie interpreter PHP, który wykonuje wszystkie instrukcje PHP znajdujące się w tym pliku. Jeśli wśród tych instrukcji jest żądanie połączenia z bazą danych, takie połączenie jest inicjowane. Następnie do bazy danych kierowane jest zapytanie (zwykle w języku SQL), jest ono wykonywane, a wybrane w ten sposób dane są zwracane do aplikacji znajdującej się w pliku PHP, gdzie są przetwarzane. W wyniku działania interpretera PHP powstaje plik, w którym nie ma już instrukcji PHP, pozostają natomiast znaczniki (zwane także elementami) HTML<sup>6</sup> i treść, która ma być wyświetlona w przeglądarce. Dalsze czynności są identyczne, jak w przypadku przetwarzania „czystego” pliku HTML: plik jest przesyłany do przeglądarki internetowej, która interpretuje i wyświetla jego zawartość. Zauważmy, że po stronie serwera wykonywany jest kod PHP oraz ma miejsce komunikacja z bazą danych, a przeglądarka internetowa otrzymuje i interpretuje HTML oraz, ewentualnie, pliki CSS, skrypty JavaScript, pliki multimedialne odtwarzane przez dodatkowe oprogramowanie, itd. (por. rys. 2).

---

<sup>1</sup> CSS – kaskadowe arkusze stylów zawierają informacje o sposobie wyświetlenia treści na stronie internetowej

<sup>2</sup> JavaScript to język skryptowy, którego funkcje są wykonywane przez przeglądarki internetowe

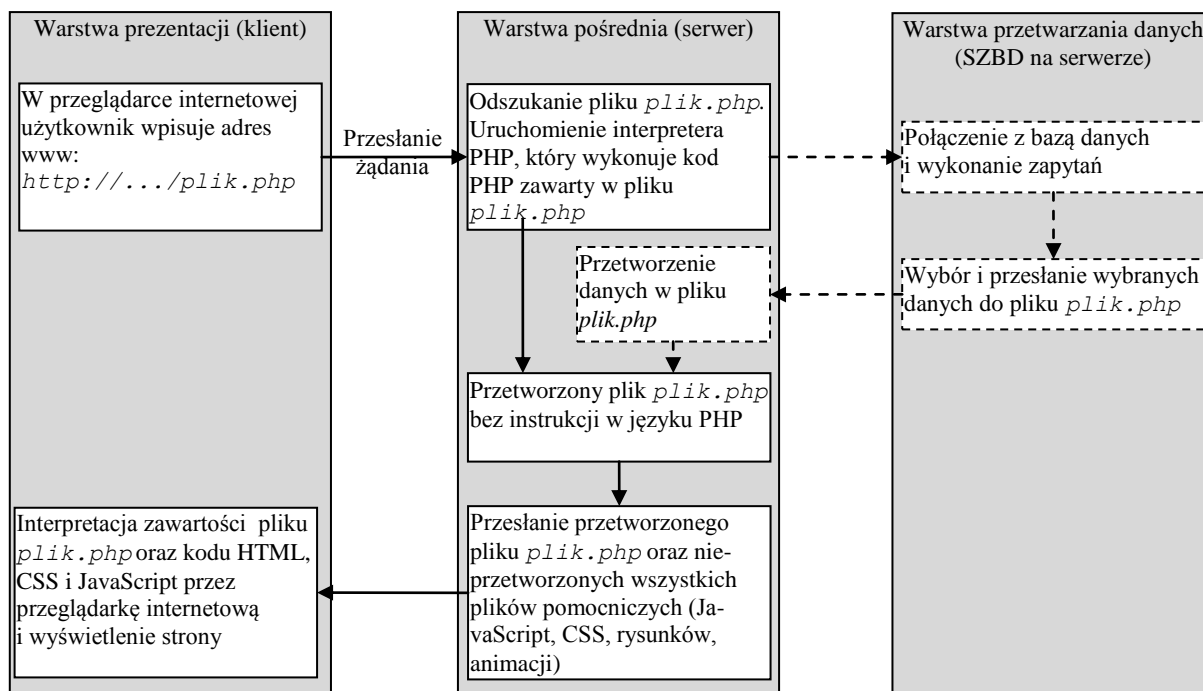
<sup>3</sup> PHP to język programowania, którego instrukcje wykonywane są na serwerze

<sup>4</sup> Jest to sytuacja najczęściej spotykana. Administrator serwera może jednak zdecydować inaczej i pliki PHP mogą mieć inne rozszerzenie

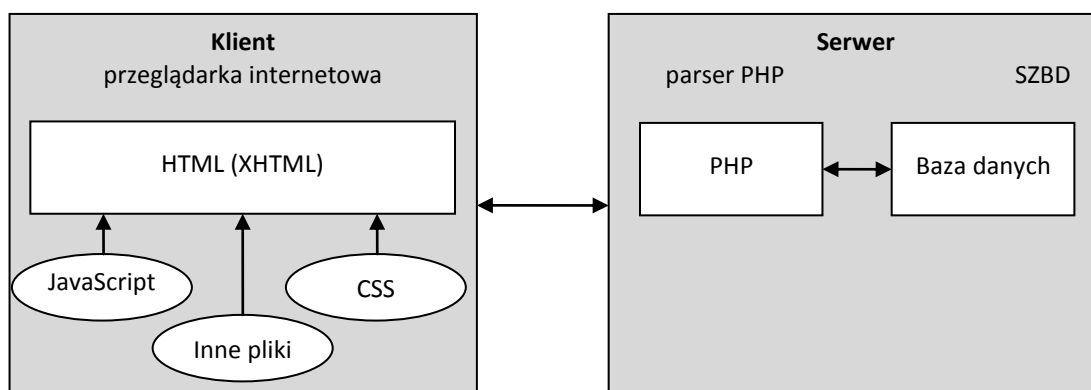
<sup>5</sup> Interpreter to aplikacja, która analizuje i wykonuje instrukcje zapisane w kodzie źródłowym programu napisanego w danym języku programowania

<sup>6</sup> HTML to język opisu stron internetowych

Rysunek 1. Zadania wykonywane przez poszczególne warstwy w procesie realizacji usługi www



Rysunek 2. Miejsce interpretacji i wykonania kodu w procesie obsługi usługi www



Do tworzenia stron internetowych wykorzystywany jest język HTML (lub XHTML). Plik zawierający treść wraz ze sposobem jej formatowania w przeglądarce może mieć wyłącznie format HTML (XHTML). Technologia CSS jest wykorzystywana do oddzielenia treści dokumentu od opisu sposobu jego wyświetlania – w takiej sytuacji w pliku HTML znajduje się to, co ma się pojawić na stronie (treść), a przy pomocy CSS jest opisane, jak to ma być sformatowane. Język skryptowy JavaScript jest używany przede wszystkim do dynamizacji stron internetowych. Funkcje napisane w tym języku wykonują różnorodne operacje w odpowiedzi na działania podejmowane przez użytkowników podczas przeglądania stron lub gdy wystąpią określone zdarzenia. Język ten może być także wykorzystany do sprawdzania poprawności danych wpisanych przez użytkowników w formularzach przed przesłaniem ich do przetwarzania na serwer. Interpretery wszystkich tych trzech technologii

(HTML, CSS, JavaScript) są wbudowane w przeglądarki internetowe i do ich poprawnego działania nie jest wymagane instalowanie dodatkowych narzędzi. Jeśli jednak strona internetowa zawiera elementy multimedialne (np. pliki stworzone w technologii Flash<sup>7</sup>), przeglądarka wyświetli ich zawartość pod warunkiem, że zainstalowane zostanie dodatkowe oprogramowanie (tzw. wtyczka, ang. *plug-in*). Jeśli zawartość strony internetowej ma być uzupełniana o dane pobierane z bazy danych, potrzebne są dodatkowe elementy: projektant musi posiadać konto i bazę danych na serwerze, na którym musi być ponadto zainstalowany interpreter języka programowania, którego instrukcje umożliwiają komunikację z bazami danych (np. PHP).

Przeglądarki internetowe interpretują i wyświetlają na ekranie zawartość plików otrzymanych od serwerów w odpowiedzi na żądania użytkowników. Pliki te mają specjalną budowę, są napisane w tzw. języku znaczników hipertekstu HTML (ang. *HyperText Markup Language*). Plik HTML, oprócz treści, która ma zostać wyświetlona, zawiera także informację, jak ta treść ma być sformatowana. Istnieją znaczniki otwierające (włączające) oraz zamykające (wyłączające). Wszystkie mają podobną budowę: otwierające posiadają nazwę oraz, opcjonalnie, parametry (zwane także atrybutami), zamykające jedynie nazwę. Nazwa otwierających umieszczona jest między znakami większości i mniejszości:

```
<nazwa parametry>
```

zamykające mają dodatkowo znak ukośnika / (ang. *slash*), umieszczony bezpośrednio po znaku mniejszości:

```
</nazwa>
```

Istnieje możliwość umieszczenia w pliku HTML znacznika z jednoczesnym zamknięciem, należy to wtedy zrobić w sposób następujący:

```
<nazwa parametry />
```

Jeśli na przykład, w odpowiedzi na żądanie użytkownika, przeglądarka otrzyma plik zawierający treść ograniczoną znacznikiem <p>:

```
<p>Ten tekst jest wyróżniony</p>
```

będzie to oznaczało, że tekst jest akapitem (znacznik <p> rozpoczyna akapit, oczywiście </p> kończy).

Większość parametrów umieszczanych wewnątrz znaczników składa się z trzech części: nazwy, znaku równości i wartości, która powinna być w cudzysłowie<sup>8</sup>. Jeśli atrybutów jest wiele, oddzielone są spacjami, np.:

---

<sup>7</sup> Flash to technologia tworzenia animacji z wykorzystaniem grafiki wektorowej

<sup>8</sup> W HTML nie jest konieczne umieszczanie wartości parametrów a cudzysłowie, w XHTML jest to obowiązkowe

```
<body style="background-color:red; color:yellow;">
```

W nomenklaturze HTML istnieje także pojęcie elementu, który oznacza grupę złożoną z następujących składników: znacznika otwierającego, atrybutów, wyświetlanej treści, znacznika zamykającego:

```
<znacznik_otwierajacy atrybut1="wartość1" atrybut2="wartość2" ... >  
Treść </znacznik_zamykajacy>
```

Terminologia związana z językiem HTML jest stosunkowo nowa, więc nie opracowano dotychczas jednolitego standardu dla nazywania poszczególnych jego składników. W niektórych opracowaniach pojęcie elementu jest utożsamiane ze znacznikiem, w innych znacznik oznacza to samo co tag. Czytając materiały dotyczące HTML, a także XML, należy sobie zdawać sprawę z tej wieloznaczności terminologicznej.

Aby stworzyć stronę internetową, wystarczy skorzystać z dowolnego edytora tekstowego (np. *Notatnika*), ale nie graficznego (np. *Worda*) i, używając elementów HTML, utworzyć plik tekstowy. Jeśli plik zostanie zapisany na dysku lokalnym, będzie go można wyświetlić przy pomocy przeglądarki zainstalowanej w systemie lokalnym. Aby stronę opublikować w Internecie trzeba mieć konto na serwerze o stałym IP i tam umieścić plik zawierający opis strony internetowej. Należy pamiętać aby plik miał rozszerzenie *htm* lub *html* – w ten sposób będzie związany (skojarzony) z domyślną przeglądarką zainstalowaną w systemie. Można oczywiście skorzystać z różnych edytorów HTML, które są dostępne w Internecie, a wiele z nich to oprogramowanie niekomercyjne.

W pierwszych latach rozwoju Internetu producenci przeglądarek prześcigali się w tworzeniu produktów odpornych na błędy w plikach HTML opisujących zawartość stron internetowych. W efekcie dokument HTML nie musi być pisany starannie, nieistotna jest m.in. wielkość liter w znacznikach, wartości parametrów nie muszą być pisane w cudzysłowie, kolejność zamykania znaczników jest dowolna, niektóre wręcz nie muszą być zamykane. To z pewnością ułatwiło pracę projektantom, doprowadziło jednak do dużej dowolności w tworzeniu i interpretacji kodu HTML. Co więcej, strona internetowa nie musi w ogóle zawierać znaczników. Wystarczy w edytorze tekstowym (np. w *Notatniku*) napisać dowolny tekst, zapisać plik na dysku nadając mu rozszerzenie *htm* i dwukrotnie klikając jego nazwę można wyświetlić zawartość pliku przeglądarce internetowej. Jednak znajomość HTML a także CSS i JavaScript oraz innych technologii jest niezbędna, jeśli chcemy tworzyć profesjonalnie wyglądające strony internetowe, zawierające, oprócz tekstu, takie elementy jak rysunki, animacje, menu, itd.

Założmy, że w systemie A przechowywane są dane zgodnie z modelem obiektowym, a w systemie B gromadzone są dane w bazie danych zaprojektowanej w oparciu o model relacyjny. Aby dane można było wymieniać między tymi systemami, lub łączyć i przetwarzać dane pobrane z obydwu, musi istnieć metoda pozwalająca na ich jednoznaczne kodyfikowanie. W tym celu opracowano i w 1996 roku opublikowano specyfikację rozszerzalnego języka znaczników XML (ang. *eXtensible Markup Language*). Stanowi on platformę wymiany danych pobieranych z różnych źródeł. Jeśli więc dane pobrane z systemów A i B zostaną przeformatowane do postaci XML-a, możliwe stanie się ich jednoczesne przetwarzanie i wzajemna wymiana. Oczywiście jest to warunek konieczny, ale niewystarczający, dane te bowiem muszą być jednakowo rozumiane nie tyle przez moduł je przetwarzający, co przez użytkowników, którzy są ich odbiorcami.

XML, podobnie jak HTML, jest technologią opartą na znacznikach. Jednak w XML-u pełnią one inną rolę niż w HTML. Są wykorzystywane jedynie do nazwania zasobów informacyjnych, przypisania im tzw. metadanych, nie są używane do formatowania wyglądu danych. Język XML, w porównaniu z HTML, jest ponadto bardziej poprawny składniowo. Reguły tworzenia dokumentów są bardziej rygorystyczne niż w HTML.

Z połączenia własności HTML i XML powstał język XHTML (ang. *eXtensible HyperText Markup Language*). Spełnia on taką samą rolę jak HTML – służy do opisu treści umieszczanych na stronach internetowych i sposobu ich formatowania, jest kompatybilny z HTML, co oznacza, że zawiera w większości te same tagi. Jednak dokumenty tworzone w XHTML muszą być poprawne składniowo na wzór dokumentów XML, a więc, w odróżnieniu od bardziej liberalnego HTML-a, dokumenty w XHTML posiadają m.in. następujące własności:

- wszystkie znaczniki są zamknięte,
- kolejność zamykania znaczników musi odwrotna do kolejności ich otwierania – ostatnio otwarty tag musi być zamknięty jako pierwszy. W HTML tagi mogą być zamykane w dowolnej kolejności, np.:

```
<a><b> Tekst </a></b>
```

podczas gdy w XHTML (podobnie jak w XML) jedynie poprawnym rozwiązaniem jest:

```
<a><b> Tekst </b></a>
```

- wartości parametrów muszą być ujęte w cudzysłowy,
- wielkość liter jest istotna (w XML i XHTML <PERSON> i <person> to dwa różne znaczniki, podczas gdy w HTML oznaczają to samo),

- w dokumencie musi być znacznik główny (w HTML może w ogóle nie być żadnego znacznika),
- ich struktura jest zgodna ze składnią XML.

Powstaje więc pytanie: czy pliki opisu stron internetowych tworzyć w oparciu o liberalny i bardziej odporny na błędy projektanta/programisty HTML, czy poprawniejszy składniowo XHTML. Odpowiedź nie jest jednoznaczna. Na pewno należy założyć, że tworzona witryna będzie w przyszłości korzystać z danych w XML-u oraz że będzie źródłem danych dla innych systemów, wtedy lepiej od początku używać XHTML, w przyszłości będzie mniej problemów związanych z koniecznością współdziałania z różnorodnymi systemami. Poza tym, dzięki kaskadowym arkuszom stylów (ang. CSS – *Cascading Style Sheets*) można oddzielić zawartość dokumentu od sposobu jej formatowania, co upraszcza plik HTML-owy i ułatwia jego analizę i modyfikację. Dzięki CSS można także dowolny plik XML-owy wyświetlić w przeglądarce internetowej w sposób opisany w zewnętrznym dokumencie CSS. Jeśli jednak stworzona zostanie witryna internetowa w HTML bez uwzględnienia własności XHTML, poza wątpliwym porządkiem wewnątrz pliku nie powinno być innych problemów, szczególnie tych związanych z wyświetlaniem zawartości.

Internetowa usługa www działa w oparciu o protokół komunikacyjny HTTP (ang. *HyperText Transfer Protocol*). Opisuje on zasady przesyłania dokumentów hipertekstowych, czyli plików zawierających zawartość stron internetowych. Zgodnie z tym protokołem adres internetowy ma następującą postać:

`http://serwer/katalog/plik`

W tym adresie *serwer* to nazwa serwera, do którego zostaje skierowane żądanie. Po niej umieszczona jest nazwa katalogu oraz pliku, który znajduje się w tym katalogu i który ma zostać przesyłany użytkownikom. Oprócz nazwy serwera wszystkie pozostałe składniki adresu są opcjonalne. Nazwę protokołu (*http://*) można pominąć, gdyż jest przyjmowana domyślnie – przeglądarka służy do wyświetlania zawartości stron internetowych, więc standardowo korzysta z protokołu HTTP. Jeśli plik znajduje się w katalogu ustawionym jako domyślny na serwerze przez administratora, nazwę katalogu także można pominąć. Jeśli plik nosi nazwę *index.htm* (lub *index.html* albo nawet *index.php*) jego nazwy też nie trzeba wpisywać. Reasumując: jeśli plik opisu strony internetowej ma domyślną nazwę i znajduje się na serwerze w domyślnym katalogu, użytkownik może pobrać i wyświetlić jego zawartość wpisując w przeglądarce internetowej jedynie nazwę serwera.

Aby udostępniać własne pliki w Internecie trzeba mieć konto na serwerze, który ma stały adres IP. Standardowo wszystkie pliki, które mają być dostępne w Internecie należy



umieszczać w katalogu *public\_html* (lub jego podkatalogach) znajdującym się w katalogu domowym. Zarówno katalog domowy, jak i *public\_html* oraz wszystkie jego podkatalogi, w których zapisywane są pliki, które mają być dostępne w Internecie, muszą mieć uprawnienia **711** (wszystkie dla właściciela oraz prawo wykonywania dla innych). Natomiast pliki muszą mieć uprawnienia **644** (wszystkie dla właściciela i prawo czytania dla pozostałych). Ponieważ takie uprawnienia są nadawane standardowo podczas tworzenia nowego katalogu lub pliku, projektant aplikacji internetowych zwykle nie musi się o to troszczyć. Jeśli, przykładowo, użytkownik o nazwie *s123456* ma konto na serwerze *abc.pl* i w katalogu *public\_html* utworzył podkatalog o nazwie *stronyInternetowe*, a w nim plik o nazwie *s.html*, to aby wyświetlić zawartość tego pliku w przeglądarce internetowej należy wpisać adres:

`http://abc.pl/~s123456/stronyInternetowe/s.html`

Jeśli plik znajduje się w katalogu *public\_html*, wystarczy:

`http://abc.pl/~s123456/s.html`

A jeśli plik będzie miał nazwę *index.htm*:

`http://abc.pl/~s123456`

Oczywiście w każdym przypadku nazwę protokołu *http://* można pominąć.