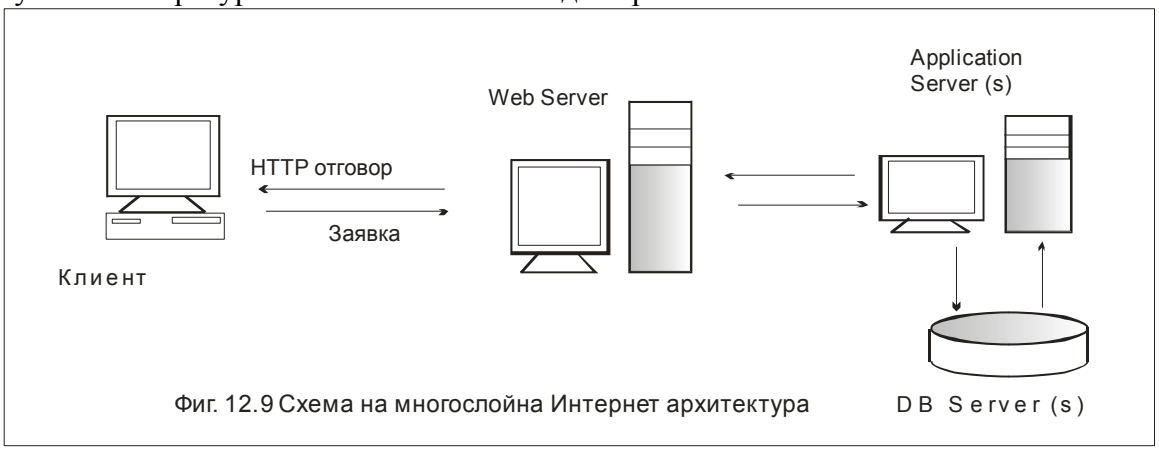
Лекция 5а

Динамично програмиране във Web

1. Връзка между клиента и сървъра в Интернет

Връзката между клиента и сървъра се основава на Web технология. Структурата й съдържа две части. Първата част обхваща всичко, което се намира между клиента и HTTP (Web) сървъра. Тази част е стандартизирана, независима е от платформите, основава се на Интернет услугите, и е в състояние да поддържа мрежи с нисък дебит. Втората част се отнася до всичко, което се намира след HTTP сървъра. В тази част се намират елементи и се прилагат технологии, специфични за класическата схема на клиент-сървър технологията (Фиг. 12.3). В средата, представена на фиг. 12.9, всеки отделен елемент има специфична функция. Клиентът управлява потребителския интерфейс и контролира въведените данни, с което се предотвратява излишния трафик по мрежата. Web сървърът свързва данните между клиента и сървъра за данни. В частта сървър за данни се извършват обработки над данните и заявките (обикновено SQL) и се осъществява връзка със сървъра на базата от данни за достъп до данните. От страна на сървърите обработките могат да бъдат разпределени между няколко физически машини, а не върху една единствена, с което се облекчава тяхното натоварване и се увеличават ресурсните им способности да обработват значително количество заявки.



Фиг. 12.9 Схема на многослойна Интернет архитектура

Действията на едно приложение в средата на многослойната архитектура се състоят от последователност от операции:

1. Установява се връзка между HTTP сървъра и Web клиента и се извлича съответната страница;
2. Данните се въвеждат в HTML формуляри, изобразени от Web браузър;
3. Скриптов език контролира на място въведената информация (JavaScript, VBScript);
4. Заявката се изпраща към HTTP сървър;
5. Обработват се данните от заявката в сървъра и се генерират SQL заявки;
6. Достъп до данните посредством сървъра на БД (SQL сървър);
7. Генерира се HTML страница с получените данни. Страниците се генерират от сървъра по начин, позволяващ да бъдат интерпретирани от клиента;
8. Изпращат се резултатите при клиента и се визуализират от браузър.

Предимствата на многослойната архитектура пред класическата схема клиент-сървър са способността на един клиент, независимо от неговата платформа, да контактува конкурентно с произволна конфигурация на сървър за БД посредством Web технология или HTTP сървър. Съвременните Web браузери предоставят лесен за усвояване интерфейс и относително прости средства за програмиране и достъпни за широк кръг от потребители. Промени в

структурата на базата данни, или процедурите за обработка на данните при сървъра, или промяна в неговата конфигурация, не оказват влияние на клиентската част, нещо почти немислимо при класическата схема.

2. Web сървъри

Кодът на Web сървърите може да бъде свободен (с отворен код) или лицензионен (със затворен код). Свободният код е безплатен и може свободно да се разпространява, променя и приспособява за определени цели. Лицензионният софтуер за Web сървъри също може да бъде безплатен (в редки случаи), но по-важното е, че потребителят няма достъп до програмния код и не може да се променя и адаптира към специфични нужди.

Към Web сървърите със свободен код се отнасят: *Apache*, *Boa*, *Red Hat Content Accelerator* (за Linux), *Thttpd*, *Mathopd* (за Linux и UNIX). По-популярни от лицензионните Web сървъри са: *Microsoft IIS* (за Windows), *IBM*, *Zeus* (за Linux и UNIX), *IPlanet* (за приложения с Java), *Stronghold*.

Основни функции на Web сървърите

Web сървърите изпълняват множество функции, които могат да се обобщят в следните направления:

- **Контрол на достъпа**. Потребителите трябва да могат да използват само тези ресурси на Web сървъра, за които имат разрешение. Всеки потребител може да ползва само тези файлове, които му принадлежат. Контролирането на достъпа става по различни начини, като настройка на подходящи разрешения за файлове и директории и прилагането на ограничения, свързани с името на хоста/IP адреса.

- **Обработка на страници от страна на сървъра**. Обработката е процес, при който Web сървъра заменя имената на полетата със съответните стойности от информацията, въведена от потребителя. След като обработи документа, Web сървърът го изпраща на клиента.

- **Поддържане на журнали**. Web сървърите използват механизъма за поддържане на журнали за администриране на работата а анализиране на проблемите, възникващи по време на обслужване на клиентските заявки. Обикновено журналите се поддържат за наблюдение и регистриране на успешните и неуспешните опити за достъп и грешките.

- **Изпълнение на CGI скриптове и други програми**. Web сървърите изпълняват CGI скриптове или програми. Те се използват за обработка на информацията, която се въвежда в Internet формите от клиентите.

Многонишкови и многопроцесни Web сървъри.

С постъпване на HTTP заявка, Web сървърът започва да търси заявеният от клиента ресурс. Докато сървърът е зает, същите или други клиенти могат да изпратят нови заявки. Web сървърът може да пренебрегне или да обработи успоредно тези заявки.

Web сървърите, които пренебрегват или нареждат на опашка получените нови заявки, се наричат *еднонишкови*. Това означава, че те не могат да се справят с натоварения при Web сървъра трафик. Тези сървъри обаче са много добри за сайтове, които имат слабо натоварен или умерен трафик, защото осигуряват много добра скорост на обработка на заявките. Добри примери за еднонишкови Web сървъри са thttpd, Medusa и Zeus.

Web сървърите, които обработват едновременно новопостъпилите заявки, изпълняват тази задача по два начина. Те могат да стартират нов процес или нова нишка на първоначалния процес. Сървърите, които започват нов процес за всяка нова заявка, се наричат *многопроцесни*, докато тези които стартират нова нишка – *многонишкови* Web сървъри.

IIS (Internet Information Services) на Microsoft е пример за многонишков Web сървър. Сървърът Apache за платформата UNIX е типичен пример за многопроцесен Web сървър. Поради ограниченията на операционната система Windows (липса на поддържка за разклоняване), Apache работи в многонишков режим, когато е инсталиран за Windows.

Web сървърите изпълнява привидно проста функция. Те се стартират заедно с операционната система и слушат (приемат) заявките, които някой е направил от Web пространството, отговарят на тези заявки и доставят продходящите Web страници. Поради това непрекъснатата работа на Web сървърите е от съществено значение. Съществуват множество сървъри, но двата доминиращи на парзара са Web сървърите Apache и Internet Information Server (IIS).

Web сървър Apache

Apache е HTTP сървър за публични домейни, разработен от Роб МакКул в Националния център за свръхизчисления (National Center for Supercomputing Applications) към университета в Илинойс. Скоро след появата на сървъра множество специалисти започват да създават свои собствени разширени версии към него. През 1994 година се организира група програмисти, които започват да се свързват с потребители от цял свят и да обменят информация за развитието на сървъра. По-късни се учредява група *Apache* (април 1995 година), която официално публикува версия на Apache 0.6.2. През декември 1995, след някои фундаментални промени и добавяне на много нови възможности, е подготвена нова версия – Apache 1.0.

Името на сървъра произтича от фразата "*a patchy*", защото програмистите, които написват кода на Apache, правят подобрения в кода посредством „пачове" (кръпки).

Apache Web Server е най-използваният в момента Web сървър. Той, подобно на операционната система *Linux*, скриптовия език *PHP* и сървърът за бази данни *MySQL* (които са често използвана комбинация от програмни средства във Web програмирането), е с отворен код. Apache може да работи с външни модули и всеки който има познания може да създаде код който да увеличи функционалността му.

По данни на NetCraft близо 60% процента от всички използвани Web сървъри се падат на Apache. Някои от предимствата му са - стабилност, бързина, лесно добавяне на допълнителни възможности, възможност за лесно преконфигуриране и не на последно място - той е безплатен.

Една от най-забележителните особености на Apache е, че може да се ползва от почти всички компютърни платформи. В началото Apache е бил свързан най-често с Unix, но вече не е така. Apache не само се ползва с повечето, ако не всички варианти на Unix, но също така и с Linux, Windows 2000/NT/9x и много други сървърни операционни системи като Amiga OS 3.x и OS/2. Apache работи най-добре с Unix и Linux, но и версиите за работа с Windows са стабилни и сигурни.

Apache предлага много възможности, включително индексиране, псевдоними, управление на дъщерни процеси, докладване за HTTP грешки, сървър ориентирани карти, онлайн наръчници и др.

Основните характеристики на Web сървърът са:

- поддръжка на HTTP 1.1 протокол. Apache е един от първите Web сървъри който интегрира HTTP 1.1 протокола. Apache е напълно съвместим с HTTP 1.1 протокола и в същото време е обратно съвместим със стария HTTP 1.0. Преди HTTP 1.1 Web браузъра е трябвало да чака за отговор от Web сървъра преди да подаде нова заявка. С появата на HTTP 1.1 Web браузъра може да изпраща паралелни заявки. Това допринася за по-бързото изпълнение на заявките.

- Просто но мощно конфигуриране. Apache няма графичен потребителски интерфейс за администратора. Той има основен конфигуриращ файл наречен httpd.conf, който се използва за конфигуриране на сървъра.
- Поддържа виртуални хостове.
- Поддържа HTTP идентификация. Apache поддържа Web базирана идентификация
- Поддържа PHP. Този език е станал широко използван и Apache предоставя добра поддръжка на PHP чрез mod_php модула.
- Поддържа Java.
- Интегрира Perl.
- Apache предоставя големи възможности за проследяване статуса на сървъра и потребителя.

Web сървърът Apache е разработен на модулен принцип, което гарантира възможности за адаптиране на сървъра. Модулната архитектура означава, че лесно може да се увеличи функционалността на Web сървъра, като се разширяват или съкращават възможностите му според потребностите. В ядрото на сървъра се съдържа стандартен комплект модули, необходими за нормалното функциониране на сървъра. Към този комплект могат да се добавят нови модули за разширяване на възможностите му.

Една от важните характеристики на Apache е възможността да се добавя поддръжка на определени програмни езици чрез инсталиране на съответни модули. Например за поддръжка на скриптовия език PHP трябва да се инсталира модула за поддръжка на този език. Най-често използваните модули за Web сървърът Apache са следните:

- mod_cgi. Осигурява поддръжка за CGI.
- mod_perl. Осигурява поддръжка на програмния език Perl.
- mod_pyhton. Осигурява поддръжка на интерпретатора на език Python.
- mod_php.Осигурява поддръжка на PHP.
- mod_javascript. Вгражда поддръйка на програмен език JavaScript.
- mod_serv. Осигурява поддръжка за Java сървлети.

Web сървър IIS във Windows

IIS 6.0 за Windows Server 2003 осигурява интегрирани, надеждни и сигурни възможности на Web сървър за intranet или Internet. IIS е стабилна и сигурна платформа за изпълнение и на динамични мрежови приложения. Той осигурява сигурен хост за Web сайтове в Internet, хост и управление на FTP сайтове и обслужване на Web news или E-mail чрез използване на Network News Transfer Protocol (NNTP) и Simple Mail Transfer Protocol (SMTP).

IIS 6.0 поддръжа ASP.NET, XML, и Simple Object Access Protocol (SOAP) за разаработване на Web приложения.

Основните характеристики на IIS 6.0 са:

- Надеждност. IIS 6.0 използва нова архитектура за организация на заявките и изолиране на приложенията за работа в отделена собствена среда и собствен процес. Това обкръжение предпазва Web приложението от намеса от други процеси и по този начин се намалява времето за спиране и рестартиране на процеса.

- Мащабируемост. IIS 6.0 включва нов драйвер за HTTP парсване и кеширане, който е специално разработен за увеличаване на работоспособността и обхвата на сървъра. Това се изразява в разширяване на следните възможности:

- Брой на Web сайтовете, които един IIS 6.0 сървър може да хоства
- Брой на конкурентните процеси, активирани в даден момент
- Скорост на стартиране и изключване на Web сървъра и отделни сайтове
- Брой на едновременните заявки, които Web сървърът може да обслужва.

- **Сигурност.** IIS 6.0 осигурява подобрена сигурност в сравнение IIS 5.0. За да се намалят атаките към системата, IIS 6.0 не се инсталира по подразбиране при инсталацията на Windows Server 2003. След инсталацията на системата, администраторът трябва ръчно да инсталира IIS 6.0. Когато IIS 6.0 се инсталира, той е заключен по подразбиране, така че може да обслужва само статично съдържание. Чрез стартиране на Web Service Extensions в IIS Manager, администратора може да активира или деактивира IIS функционалността, в зависимост от нуждите на потребителите

Истинските възможности на Web сървъра на Microsoft IIS (Internet Information Service) се проявява в сървърните платформи на Windows, например във Windows Server 2003.
Версията на IIS от състава на Windows XP Professional не се явява толкова мощна и гъвкава. Принципното ограничение се състои в това, че IIS позволява създаването само на един Web сайт и един сървър FTP. Освен това, се допускат не повече от 10 едновременни връзки по протокола TCP. Тъй като за връзка с някои Web страници трябват няколко съединения TCP, това означава, че реално с един сайт са в състояние да работят не повече от 7 потребители едновременно. За да работят повече от 10 потребителя едновременно е необходимо да се инсталират няколко сървъра Web или FTP.
За разработки във Web наличието на сървъра IIS във Windows XP Professional дава възможност да се разработват и тресират програми и да се пренасят на по-голям сървър. В неголеми организации и предприятия от среден размер IIS може да се използва за създаването на възлов интранет – локални мрежи, основани на Интернет технологиите. Ако е необходим FTP – сайт за съхранение на общите файлове, IIS под управление на Windows XP Professional осигурява всичко необходимо за неговото създаване и работа.

3. Програмиране във Web

Може да се постигне изключително много чрез използване на текст, изображения и мултимедия за разработване на Web страници. Въпреки това, за да се разработи истинско професионално Web приложение с интерактивни възможности, е необходимо и въвеждане на програмен код.
Съществуват възможности за разработване на интерактивни елементи в HTML чрез използване на така наречените формуляри (форми). Формулярите представляват текстови полета и падащи менюта, позволяващи на потребителите да въвеждат информация. Те се създават лесно както всички елементи в HTML. Формулярите са част от езика и са описани почти във всички по-подробни справочници. Приемането на информацията от потребителите е просто, но по-важно е какво да се прави тази информация.
Обработката на информацията изисква CGI-скриптове и приложни програми. CGI (Common Gateway Interface) е програма, която изпраща данните на друга програма, създадена от разработчиците на приложения. Скриптовете, създадени от CGI, се изпълняват върху сървъра, където се намира Web страницата. За да се стартира CGI скрипт, е необходимо разрешение от системния администратор, отговарящ за сървъра, на който се намира Web страницата. Много сървъри имат предварително подготвени формуляри и CGI пакети за често срещани задачи, като например броя на посетителите на сайта, регистрация на потребители на сайта и други.
Макар че простотата и гъвкавостta на HTML бяха ключът към успеха на Web, той има и сериозни ограничения. За преодоляване на тези ограничения се разработват няколко големи проекта – *Dynamic HTML* (динамичен HTML) и *XML.* Динамичният HTML е разширение на HTML, което осигурява по време на връзката със сървъра да се изпраща многослойна информация от потребителите. Потребителят първоначално вижда само част от информацията. Останалата част може да се покаже след известно време или след като потребителят предприеме определени действия – и всичко това, без да се осъществява повторна връзка със сървъра.

Разширяването на HTML с XML е друга промяна, която придвижва работата с Web още по-напред. Той позволява във Web страниците да се вграждат сложни структури от данни, създаване на приложения, управлявани от данните и предоставянето им по Web.

4. Динамични Web страници

Съдържанието на дадена Web страница може да бъде *статично* или *динамично*. В ранните години на Internet, Web страниците се създаваха с HTML и се изпращаха до клиента във вида в който са създадени, без да се променят. Поради тази причина те се наричат статични. Развитието на Internet доведе до създаване на динамично съдържание на Web страниците и днес популярността на всеки Web сайт зависи до голяма степен от динамичното съдържание в него. Съдържанието на динамичните Web страници може да бъде променяно, докато потребителят все още комуникира със сървъра.

По дефиниция динамични Web страници са страници, предоставящи елементи със съдържание или представяне, които не са зададени изрично, а зависят от определена обработка или изпълнение на програмен код. Страница, която съдържа само HTML код, не може да бъде динамична, защото и съдържанието и представянето са предварително дефинирани от кода на страницата.

Страници, които включват например функции на JavaScript, предаващи съобщения към потребителя при натискане на бутон, са динамични, защото съобщението се появява или не в зависимост от решението на потребителя. В случая програмният код се изпълнява от браузъра, към който се изпраща страницата. Програмен код може да се изпълнява и от сървъра при подготовката на Web страницата, която се изпраща към потребителя. Трябва да се има пред вид, че докато всички динамични Web страници съдържат някакъв скрипт елемент (програмен код), то не всички страници, които съдържат JavaScript са динамични. Например, ако JavaScript кодът се използва, само за да се визуализира даден текст, това не е динамична страница. Само Web страници, които имат зависещ от някакви условия елемент, могат да се разглеждат като динамични.

Съществуват два основни механизъма за организиране на динамични Web страници: динамично съдържание генерирано при клиента (от браузъра) и динамично изпълнение извършвано от Web сървъра. Затова динамичните страници могат да бъдат Сървър-базирани или Браузър-базирани Web страници.

Сървър-базирани динамични страници.

Сървър-базирани Web страници могат да бъдат реализирани по два основни начина: генериране на HTML документ (Web страница) в резултат на изпълнение на програмно приложение от страна на сървъра или генериране на динамично съдържание в резултат на вграждане на програмен код в самия HTML документ, който се интерпретира от сървъра.

Тези страници използват скриптове (програмен код) или специални програмни приложения на сървъра за генериране на съдържанието на дадена страница, която се изпраща към браузера. За целта се използват специализирани програмни езици и технологии като PHP, ASP, JSP, Cold Fusion и други.

Предимства на сървър-базираните страници. Сървър-базираните скриптове позволяват на разработчиците да се възползват от процесорната сила на Web сървърите. Сървърите обикновено са по-бързи и по-мощни от машините, използвани от средния Web клиент. Тази увеличена процесорна сила означава, че сървър-базираните динамични Web страници могат да разполагат с по-големи ресурси и да предоставят по-голямо потенциално съдържание на страниците.

Обработката от страна на сървъра е единственият начин за съхраняване и извличане на информация във и от база от данни. Данните могат да се организират по-добре и да се съхраняват непрекъснато във файлова система или релационни бази от данни, отколкото

това е възможно с използването на клиент-базираните скриптове. Това е особено важно за по-големите сайтове, които имат множество страници и използването на база от данни може да намали значително разходите по поддържане на сайта.
Обработката от страна на сървъра е независима от браузера. Сървърите изпращат обикновен HTML код и по този начин се избягват проблемите със съвместимостта с различните типове браузери.
Трудности при сървър-базираните страници. Информацията и обработката, която извършва сървърът, са без гражданство и не е известно дали две последователни заявки са от един и същ потребител или от различни. Това изисква всяка заявка да се третира като самостоятелна, а това е свързано с допълнителна обработка.
Сървър-базираната обработка се случва, само след като е изпратена заявка към сървъра. Това означава, че дори и най-добрите и бързи програмни скриптове могат да действат в малкия интервал от момента, в който потребителят е активирал връзката или е изпратил форма, до момента, когато динамичната страница е върната от сървъра. Тази обработка също така елиминира възможността за всяка промяна на страницата, след като тя е изпратена към браузера.

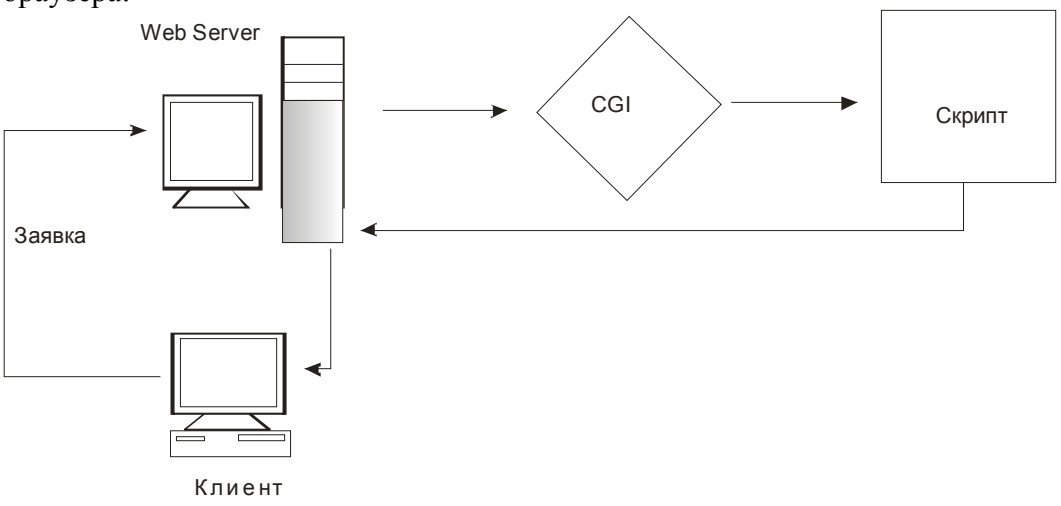
Браузър-базирани динамични страници.
Браузер-базираните динамични страници използват скриптове, които се включват в HTML страниците с инструкции за самия браузър, който генерира съдържание, зависещо от действия на потребителя. Повечето динамични страници базирани на браузърна обработка използват JavaScript, но се използват и други скрипт езици, като Jscript или VBScript.
Преимущества на Браузър-базирани динамични страници. За разлика от сървър-базираната обработка, браузър-базираната може да продължи и след като страницата се е заредила в клиентския компютър. Съдържанието и изобразяването на динамичните елементи на страницата не са ограничени от изпратения от сървъра HTML код. Скриптовете в страницата могат да накарат изображението да се появява и изчезва, да променят позицията и форматирането на страницата. Те имат и други функции, които не са възможни при използването на сървър-ориентирани скриптове.
Тъй като обработката се ръководи от браузъра, браузър-базираните динамични форми не се отразяват върху честотната лента на клиента (натоварването на връзката със сървъра). Функционалността, която определя връзката между отделните страници, поръчвани от клиента се управлява изцяло от текущата страница от браузера.
Трудности на Браузър-базирани динамични страници. Тъй като обработката на кода се извършва от браузъра, трябва да се отчитат възможностите на браузъра. Не всички браузъри имат еднакви набори от функции, а някои дори нямат никакви възможности да обработват скриптове. Това поставя разработчиците пред избор да ограничават функционалността на страниците за тесен кръг от потребители или да губят значително време в създаване на алтернативни скриптове за различните браузъри.
Повечето браузъри не могат да постигнат скоростта за първична обработка и гъвкавостта, които постигат сървър-базираните скриптове. В сравнение със сървър-базираната обработка, браузър-базираните динамични елементи са тромави и сложни.
Накрая, браузър-базираната обработка не позволява информацията да се съхранява по някакъв постоянен във времето начин. Могат да се съхраняват само малки количества текст на външни носители на потребителя. Съхранението на сложна информация или събиране на информация от повече от един потребител е невъзможно само с браузър-базирани скриптове.

CGI приложения.
CGI (Common Gateway Interface – общ портал за връзка) не е отделен език, а множество от спецификации, които позволяват на Web сървъра да комуникира с други приложения и

програми. Това е стандарт, който определя начина, по който Web сървърите могат да използват външни програми. Всеки език за писане на програми или скриптове, който съответства на CGI, може да се използва за създаване на CGI приложения (CGI програми). Най-разпространените езици, използвани за CGI включват tcl, AppleScript, C, Java, Visual Basic и Perl.

CGI програмите се изпълняват в реално време. Това дава възможност за генериране на динамично съдържание. Заявката за URL адреса на дадена CGI програма се формира след като потребителят избере хипервръзка или подаде формуляр. За да направи заявката, браузерът използва HTTP. Когато Web сървърът получи заявка, той изпълнява CGI програма, като й предава изпратените от браузъра данни. Когато CGI приложението извърши обработката на данните, обикновено то генерира нови данни под формата на Web страница, която връща на браузера. Технологията на взаимодействие между браузъра, Web сървъра и CGI приложението е показана на фиг.12.10.

За разлика от стандартните програмни езици CGI не е език за програмиране, а е интерфейс или набор от правила. Той работи върху Web сървъра, като осигурява начин за комуникация и за изпращане на данни от Web страница (Web формуляр, скрипт) към програма, която извършва обработка. Тази програма изпраща резултатите от обработката отново към Web сървъра и ако те са правилно форматирани, той ги изпраща като HTML документ към браузера.



Фиг. 12.10 Схема на работа на CGI приложение

CGI програмите могат да се разделят на два вида: *CGI приложения* и *CGI скриптове*. CGI приложенията са изпълними програми в двоичен код, които могат да се изпълнят самостоятелно от команден ред. Тези програми са написани на стандартен език от високо ниво като C, Pascal, Fortran и други които са компилирани предварително (статични *CGI приложения*) или се компилират когато бъдат извикани (динамични *CGI приложения*).

CGI скриптовете съдържат изходен (source) код на програмите. Този код се компилира при всяко тяхно извикване. Тъй като те представляват програмен код те не съдържат вградени библиотечни функции и зависят изцяло от компилатора за осигуряване на библиотечна поддръжка. *CGI скриптовете* са значително по-малки от *CGI приложенията* и заемат по-малко място на диска.

Механизъм Server-Side Includes (SSI)

Web сървърът Apache подържа механизъм наречен *server-side includes (SSI)*, който се използва за добавяне на динамично съдържание в HTML страници. SSI са HTML етикети, които Web сървърът може да тълкува и изпълнява. Тези етикети се поставят в HTML документите във вид на коментари. Когато Web сървърът срещне тези коментари по време на изпълнение на дадена заявка, той изпълнява указанията дадени в тях. За да изпълнява Web

сървърът SSI той трябва да бъде настроен за такава дейност. Това става чрез използване на директиви във файла httpd.conf. Често използван начин за конфигуриране на SSI е задаване на разширение на HTML файловете, което дава указание на сървъра да анализира документите във файловете и да търси в тях SSI етикети. Обикновено използваното разширение е *shtml*.
Основните действия, които могат да се изпълнят при сървъра с помощта на SSI са следните:
- **Показване на съдържание при зададени усл**овия. SSI осигурява проста if/else структура, която може да се използва за задаване на определени условия в програмния код на Web страниците.
- **Извикване на CGI програми**. С помощта на SSI може да се добавят конфигурационни директиви, които извикват CGI програма по време на работа на сървъра.
- **Включване на външни файлове**. SSI етикети могат да се използват за включване на данни от външни файлове, които се обработват, и резултатите се включат в изготвяния от сървъра документ, изпращан на клиента.
- **Достъп до променливи от обкръжението на система**та. С помощта на етикети от SSI може да се осигури достъп до променливи от обкръжението на операционната система.
Кога се използва CGI и кога SSI? И CGI и SSI се използват за създаване на динамични Web страници. Използването им зависи от динамичността, която трябва да се реализира за съответната страница. Ако е необходима минимална динамичност, по-доброто решение е да се използва SSI. Когато е необходима сериозна обработка на данни и логика, тогава се използва CGI.

Сървър-ориентирани скриптове
Използването на бази от данни във Web приложенията е свързано с процес на посредничество. Web формите събират информация, въведена от потребителите и я предават на Web сървъра. След това информацията трябва да се подаде към базата данни, но на Web сървъра трябва да се укаже къде да намери базата данни и кои таблици в нея да актуализира. В този случай се използва скрипт-език от страна на сървъра. Сървър-ориентираните скриптове позволяват на данните от информационен източник, като релационна база данни, Web форма или файлова система да бъдат изпратени на или от Web сървър. Най-често използваните скрипт-езици са:
Perl. Perl е най-старият сървър-ориентиран език за скриптове. По тази причина той е доста разпространен и с него има много разработени приложения. Недостатък на Perl и на всички CGI езици е, че Web сървърът трябва да стартира нов процес за всяко използване на CGI кода.
ASP (Active Server Page). ASP е приносът на Microsoft в общността на скриптовете за сървърна обработка. Поддръжката на техническата документация, предоставяна от Microsoft, прави ASP добър избор за разработчици, които използват Web сървъри на Microsoft. Като съставна част от Web услугите, ASP работи бързо и има универсална приложимост. Недостатък на ASP е, че малко от характерните му приложения са валидни за сървърите, които не са на Microsoft. Технологията ASP е основен компонент на платформената технология .NET Framework разработвана от Microsoft.

JSP (Java Server Pages). JSP се популяризира като универсален език за програмиране, който позволява на разработчика да ‘пише еднократно’ за всички операционни среди. От наименованието става ясно, че JSP включва Java код във Web страници. Тъй като програмният език Java е създаден като многоплатформен език, той се използва за програмиране на различни Web сървъри. Java е може би най-трудният за изучаване от скриптовите езици (той е пълноценен програмен език), но той осигурява голяма гъвкавост на приложенията.

По принцип всяка Web страница може да бъде генерирана в резултат на изпълнение на обикновена Java програма (Java сървлет), но е по- удобно и бързо да се пише Java скрипт в HTML, който добавя динамични елементи. Освен това, така може лесно да се раздели външния вид на страницата от динамичното съдържание. Web дизайнерът може да напише HTML частта (статичната страница) и да остави място на програмиста да вмъкне динамичното съдържание с Java код.

ColdFusion. Това е по-малко скриптов език и повече разширение на HTML. ColdFusion използва HTML подобни етикети, които правят динамично съдържание, комуникират с бази от данни и взаимодействат с обекти от файлова система. Специален сървър на ColdFusion обработва ColdFusion файловете и изпълнява необходимите действия. Необходимостта от специален сървър е недостатък, но близостта му до HTML го прави лесен за използване.

PHP (Hypertext Preprocessor). PHP е скрипт-машина, която комбинира множество готови Web инструменти и Form Interpreter (интерпретатор на форми) за генериране на бърз, лесен и с отворен код език за писане на скриптове. PHP работи практически на всеки Web сървър с малки различия в кода. Най-голямото предимство на PHP е, че е безплатен с отворен изходен код. Той не изисква специален сървър както при ASP и ColdFusion. Той е по-бърз от JSP и по-лесен за изучаване от Perl.
В момента PHP се използва от милиони домейни по целия свят и популярността му продължава да расте. С PHP може да създават и редактират файлове, да се събира и обработва информация от формуляри, да се изпращат данни с електронна поща, да се управляват записи в бази данни, да се съхраняват данни в променливи по време на сесия и други. PHP предлага висока скорост на изпълнение и използва много малка част от системните ресурси и не забавя работата на хост машината.

Вграждането на сървър-базиран код в стандартна HTML страница се извършва лесно. Обикновено текстовият файл на Web страниците, съдържащи скриптов код, който трябва да се интерпретира от Web сървъра се записват със специфично разширение, в зависимост от типа на използвания скриптов език. Ето един пример за използване на Active Server Pages (ASP):

ASP код в HTML страница
Една обикновена HTML страница: `hello_world.html има вида:`

```
<html>
<head><title>Hello World</title></head>
<body>
        Hello World
</body></html>
```

Когато е налице ***IIS (Internet Information Server)*** или друг Web сървър, поддържащ ASP горната програма може да бъде записана с използване на сървър-ориентиран скрипт. Създава се текстов файл, който трябва да има разширение '.asp', например `hello_world.asp`**:**

```
<%@ Language=VBScript%>
<html>
<head><title>Hello World</title></head>
<body>
                <% Response.Write("Hello World")%>
</body></html>
```

Това е файл, използващ сървър-ориентиран скрипт и може да се обработи от Web сървър. Три неща са променени в новия файл:
- Името на файла завършва с `.asp` вместо с `.HTML`. Така Web сървърът се информира, че страницата вероятно съдържа програмен код, който трябва да се обработи от него. В този

случай, вместо директно да подава страницата на браузъра при въвеждане на нейния URL, Web сървърът проверява преди всичко наличието на тагове от типа <% ... %>. Тези тагове се използват за записване на програмен код с помощта на скриптов език. ASP обикновено използва скриптовия език VBScript. Ако сървърът открие такива тагове, той разглежда съдържанието им като програмен код, интерпретира (изпълнява) този код и полученият резултат от програмата изпраща на браузера. във вид на генериран HTML код.

- Първата част от кода, който открива Web сървъра е <%@ Language=VBScript%>. Оттук Web сървърът разбира, че програмният език е VBScript. Това може да се изпусне, защото по подразбиране IIS предполага, че ASP програмите се пишат на VBScript.

-Следващата част от кода, която ще открие сървърът, е реда, съдържащ <%Response.Write("Hello World")%>.

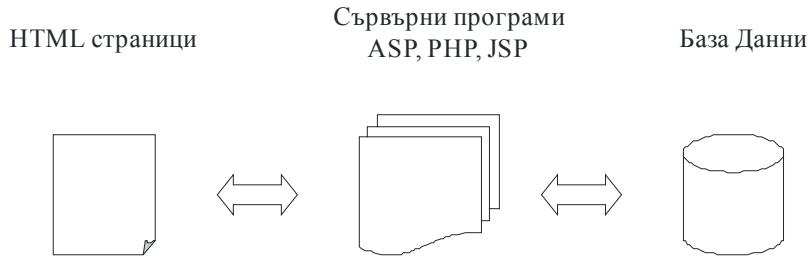
Командата Response.Write() наподобява командата Print() от другите езици за програмиране и скриптиране. В контекста на ASP, този код уведомява сървъра да изпрати на браузъра низа "Hello World".

Резултатът от показаните по-горе страници .asp и .html е аналогичен: Web страница с изписани в нея думи "Hello World".

Въпреки това, двете страници са абсолютно различни. Когато браузърът поиска от Web сървъра страница .html, сървърът предава страницата на браузъра, без да извършва никакви действия. Когато Web сървърът срещне страница с разширение .asp, той проверява дали има малки програми за изпълнение (скриптове). Това означава, че част или цялото съдържание на страницата .asp ще се генерира след обработка от страна на сървъра.

За изпълнение на .asp страница е необходимо да се избере технология, която да се използва за реализацията на проекта и софтуерна и хардуерна платформи, на която ще се базира интернет страницата. Тези неща са взаимно свързани и техният избор е важен за крайния продукт. Когато няма регистриран домейн или собствен интернет сървър, изборът е ограничен до предлаганите в интернет безплатни интернет сървъри и предлаганите от тях софтуерни платформи за сървърните програми. Лесно могат да се намерят безплатни сървъри, използващи PHP, Java Server Pages (JSP) и Active Server Pages (ASP).

Най-често, когато се използва сървър-ориентирана обработка се използват многослойни интернет страници, включващи слой за база данни, среден слой – сървърни програми, осъществяващи бизнес логиката и презентационен слой за визуализиране при клиента (фиг. 12.11).



Фиг. 12.11 Схема на многослойна Web страница

Като база данни на безплатните сървъри се предлага MySQL, обикновено в комбинация с PHP или Java Server Pages за средния слой сървърни програми и Apache като web server. И всичко това на операционна система Linux.

За Windows приложения може да се избере сървър, който предлага база данни, например MS Access и сървърно програмиране на ASP – Visual Basic и Microsoft© IIS (Internet Information Server) или MS Personal Web Server.