

**Обработка на уеб заявки със CGI интерфейс
чрез използване на апарата на обобщени мрежи**

Веселина Бурева, Ивелина Вардева
Бургаски университет „Проф. д-р Асен Златаров”
vesito_ka@abv.bg, iveto@btu.bg

Резюме: В статията се разглежда схема за обработване на заявки, приложения и други ресурси при Common Gateway Interface програма. Използван е апаратът на обобщените мрежи за моделиране процеса на предаване на информация в CGI програмите.

Ключови думи: Обобщени мрежи, CGI програма, уеб сървър

1. Въведение

Разгледани са етапите на обработка, през които минава една заявка от уеб клиент. Използван е апаратът на обобщените мрежи за моделиране на процесите за обработка на заявката.

CGI (Common Gateway Interface, „общ шлюзов интерфейс”) осъществява връзка между външна програма и уеб сървър. Той е интерфейс към уеб сървъра, с помощта на който се увеличава функционалността му многократно. CGI скриптът е програма, която се изпълнява на сървъра, обикновено като отговор на запитване или заявка на потребителя. CGI програмата извършва обработките. Обикновено тя генерира данни под формата на уеб страница. CGI не се интересува от браузъра, който използва, [3].

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, браузърът използва HTTP. Когато уеб сървърът получи заявка, той изпълнява CGI програма, като ѝ предава изпратените от браузъра данни. Когато CGI приложението извърши обработката на данните, обикновено то генерира нови данни под формата на уеб страница, която връща на браузъра. Този изход може да бъде HTML, прост текст, графика или нещо друго.

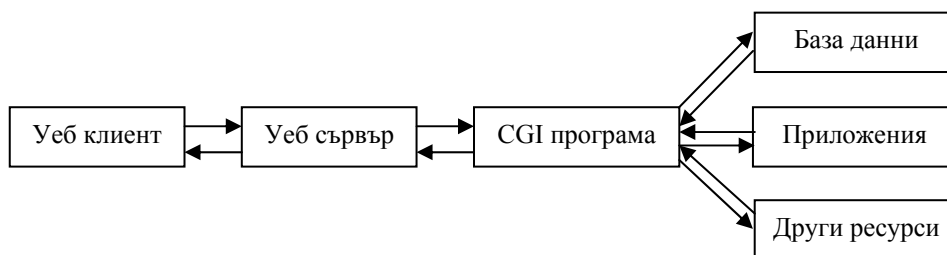
За да се стартира един CGI скрипт, най-често браузърът предава данните от текущата HTML форма на съответния скрипт. Той разчита полетата на формата (които му се предават кодирани по определен, строго фиксиран начин, за да могат да се предават през HTTP протокола), изпълнява специфични действия на базата на тези полета и генерира нова HTML страница, която се изпраща обратно на потребителя. Така може да се направи например четене от база данни или проверка на форма, която не може да бъде проверена публично чрез JavaScript (JavaScript кодът се намира в самия HTML и е достъпен за крайния потребител). Тъй като CGI скриптът

се изпълнява на сървъра, той има достъп до ресурсите, базите данни и файловата система на сървъра и може да генерира HTML страници, използвайки цялата информация, съдържаща се там. CGI скриптове могат да се пишат на всеки програмен език, но на практика се използват два езика: Perl и C. За да се напише CGI скрипт, е нужно да се спазват определени правила. За да се изпълни CGI скрипт е необходим сървър, който да има тази възможност, както и интерпретатор, който да изпълни скрипта, когато последният е писан на Perl. CGI се явява една от първите технологии за създаване на уеб приложения, изпълнявани от сървъра. Програмите, които съблюдават CGI стандарта, се наричат CGI програми.

От своя страна CGI програмите са два типа: компилирани и във вид на програмен (сорт) код. Компилираните CGI програми могат да бъдат написани на почти всеки език, който се компилира до изпълним код. 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. Сорт кодове могат да бъдат изпълнявани на сървъри, на които има инсталирани съответните интерпретатори. Кодът 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. Най-популярният език за писане на CGI е Perl. Той е интерпретируем език, а неговият интерпретатор се намира и в инсталацията на Linux, която е една от най-популярните операционни системи за сървърно администриране. CGI скриптовете са значително по-малки от CGI приложенията и заемат по-малко място на диска.

Съществен недостатък на CGI е, че става дума за самостоятелни приложения - всеки път, когато потребител се обърне към скрипта, в паметта се заделя ново място и се изпълнява ново копие на програмата. Това става проблем, например, когато в един и същи момент към скрипта се отправят няколко заявки – сървърът заделя твърде много памет (особено ако скриптът е по-голям), процесорно време и работата му може чувствително да се забави.

CGI обаче е много подходящ за сайтове, които не предвиждат горепосочената екстремна посещаемост, а и имат ограничен бюджет например операционната система Linux, която се разпространява безплатно, включва и интерпретатори за Perl и други езици, на които може да бъде писан CGI.



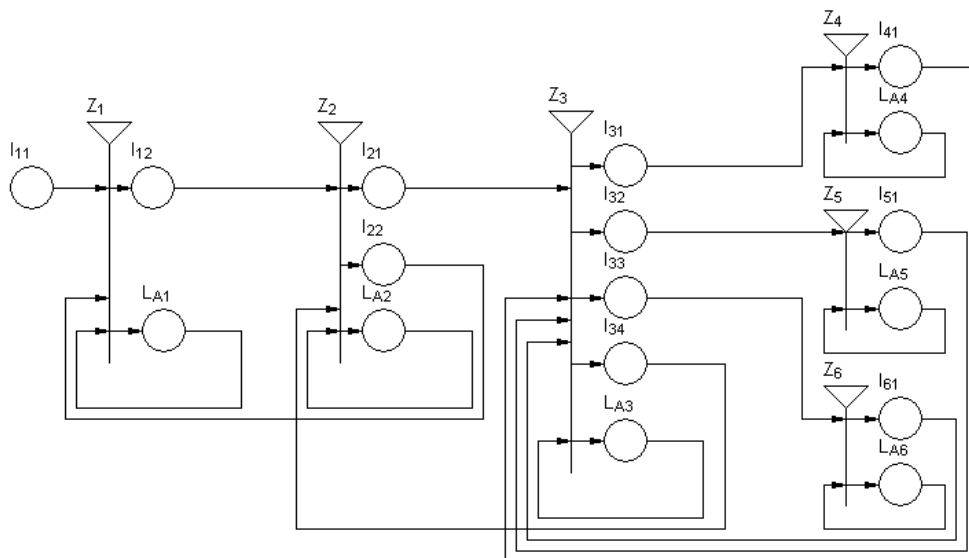
Фигура 1. Обща схема за обработка на уеб заявка през CGI

2. Обобщено мрежов модел

Първоначално има следните ядра в обобщената мрежа:

- в позиция $L_{A1} - \alpha_1$ – ядро с характеристика „уеб браузър на клиента“;
- в позиция $L_{A2} - \alpha_2$ – има ядра с начална характеристика $\langle 0 \rangle$, а следващите характеристики са агрегират през ОМ – натрупване на заявки за обработка в уеб сървър.
- в позиция $L_{A3} - \alpha_3$ – ядро с характеристика „CGI програма“.
- в позиция $L_{A4} - \alpha_4$ – ядро с характеристика „База данни“
- в позиция $L_{A5} - \alpha_5$ – ядро с характеристика „Приложения“
- в позиция $L_{A6} - \alpha_6$ – ядро с характеристика „Други ресурси“

Заявките на клиента постъпват в мрежата през позиция l_{11} , съдържаща ядро α_0 с характеристика „Има текуща заявка на клиента от браузъра на клиента“.



Фигура 2. Обобщено мрежови модел обработка на уеб заявка през CGI

Разработен е обобщен мрежови модел с въведено множеството от преходи A , които: $A = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6\}$, където преходите описват следните процеси:

- Z_1 = “Задачи извършени от уеб клиент”;
- Z_2 = “Задачи извършени от уеб сървър”;
- Z_3 = “Задачи извършени от CGI”;
- Z_4 = “Задачи извършени от База данни”;
- Z_5 = “Задачи извършени от Приложения”;
- Z_6 = “Задачи извършени от Други ресурси”.

Преходите имат описанието, дадено по-долу.

$$Z_1 = \langle \{l_{11}, l_{22}, L_{A1}\}, \{l_{12}, L_{A1}\}, R_1, \vee(l_{11}, l_{22}, L_{A1}) \rangle,$$

където:

$$R_1 = \begin{array}{c|cc} & l_{12} & L_{A1} \\ \hline l_{11} & false & true \\ l_{22} & false & true \\ L_{A1} & w_{A1,12} & true \end{array}$$

$w_{A1,12}$ = “Има заявка за уеб сървър”

Ядрото, постъпващо в позиция l_{12} , получава характеристика „заявка от клиента за обработка”.

$$Z_2 = \langle \{l_{12}, l_{34}, L_{A2}\}, \{l_{21}, l_{22}, L_{A2}\}, R_2, \vee(l_{12}, l_{34}, L_{A2}) \rangle,$$

където:

$$R_2 = \begin{array}{c|ccc} & l_{21} & l_{22} & L_{A2} \\ \hline l_{12} & false & false & true \\ l_{34} & false & false & true \\ L_{A2} & w_{A2,21} & w_{A2,22} & true \end{array},$$

$w_{A2,21}$ = “Има готова заявка”;

$w_{A2,22}$ = “Има изпълнена заявка за предаване към уеб браузър на клиент”.

Ядрата, постъпващи в позиции l_{21} и l_{22} , получават характеристики съответно: “Заявка от сървър за обработка” и “Има обработена клиентска заявка”.

$$Z_3 = \langle \{l_{21}, l_{41}, l_{51}, l_{61}, L_{A3}\}, \{l_{31}, l_{32}, l_{33}, l_{34}, L_{A3}\}, R_3, \vee\{l_{21}, l_{41}, l_{51}, l_{61}, L_{A3}\}, \rangle$$

където:

$$R_3 = \begin{array}{c|ccccc} & l_{31} & l_{32} & l_{33} & l_{34} & L_{A3} \\ \hline l_{21} & false & false & false & false & true \\ l_{41} & false & false & false & false & true \\ l_{51} & false & false & false & false & true \\ l_{61} & false & false & false & false & true \\ L_{A3} & w_{A3,31} & w_{A3,32} & w_{A3,33} & w_{A3,34} & true \end{array}$$

$w_{A3,31}$ = “Изискана е комуникация към База Данни”;

$w_{A3,32}$ = “Изискана е работа с уеб приложения”;

$w_{A3,33}$ = “Изискана е работа с други ресурси”;

$w_{A3,34}$ = “Изпълнена е CGI заявката”.

Влизащите ядра в позиции l_{31} , l_{32} , l_{33} получават съответно характеристики “Заявка към базата данни”, „заявка към приложения”, „заявка към ресурси”. След прехода

полученото ядро в позиция l_{34} излиза със следната текуща характеристика “Има изпратена заявка от CGI програма”.

$$Z_4 = \langle \{l_{31}, L_{A4}\}, \{l_{41}, L_{A4}\}, R_4, \vee(l_{31}, L_{A4}) \rangle,$$

където:

$$R_4 = \begin{array}{c|cc} & l_{41} & L_{A4} \\ \hline l_{31} & false & true \\ L_{A4} & w_{A4,41} & true \end{array},$$

$w_{A4,41}$ = “Има готова заявка”.

Постъпващото ядро β в позиция L_{A4} се слива с текущото ядро. Излизащото ядро от позиция l_{41} получава характеристика “Изпратена е заявка от базата данни”.

$$Z_5 = \langle \{l_{32}, L_{A5}\}, \{l_{51}, L_{A5}\}, R_5, \vee(l_{32}, L_{A5}) \rangle,$$

където:

$$R_5 = \begin{array}{c|cc} & l_{51} & L_{A5} \\ \hline l_{32} & false & true \\ L_{A5} & w_{A5,51} & true \end{array},$$

$w_{A5,51}$ = “има готова заявка”.

Постъпващото ядро γ в позиция L_{A5} се слива с текущото ядро. Излизащото ядро от позиция l_{51} получава характеристика “Изпратена е заявка от приложения”.

$$Z_6 = \langle \{l_{33}, L_{A6}\}, \{l_{61}, L_{A6}\}, R_6, \vee \{l_{33}, L_{A6}\} \rangle,$$

където:

$$R_6 = \begin{array}{c|cc} & l_{61} & L_{A6} \\ \hline l_{33} & false & true \\ L_{A6} & w_{A6,61} & true \end{array},$$

$w_{A6,61}$ = “Има готова заявка”.

Постъпващото ядро δ в позиция L_{A6} се слива с текущото ядро. Излизащото ядро от позиция l_{61} получава характеристика “Изпратена е заявка от ресурси”.

3. Заключение

В статията са разгледани основните етапи за разработване на подадена клиентска заявка към CGI програма. За решаването на проблемите по разработването на заявката е необходимо да се състави пълен анализ за разработка. Моделът позволява да се разглеждат различните етапи на заявката:

- подаване на заявката от уеб клиент към уеб server;
- предаване на заявката от уеб server към CGI програма;
- ориентиране и разработване на заявката;
- връщане на вече изпълнената заявка.

Литература

- [1] Atanassov, K. Generalized nets, World Scientific, Singapore, 1991.
- [2] Atanassov, K. On Generalized Nets Theory, “Prof. M. Drinov” Academic Publishing House, Sofia, 2007.
- [3] <http://www-it.fmi.uni-sofia.bg/courses/pc3/beginner/beginner9/content5.html>.
- [4] Guelich, S., S. Gundavaram, G. Birznieks, CGI Programming with Perl, Second Edition, O'Reilly Media, 2000.
- [5] Gundavaram, S., CGI Programming on the World Wide Web – First Edition March, O'Reilly Media, 1996 .
- [6] Кастро, Е., “Perl и CGI”, Инфодар, 2000.