

# CGI программалау.

Қабылдаған: Картбаев Т.  
Орындаған: Алданияз Г.

## Жоспар:

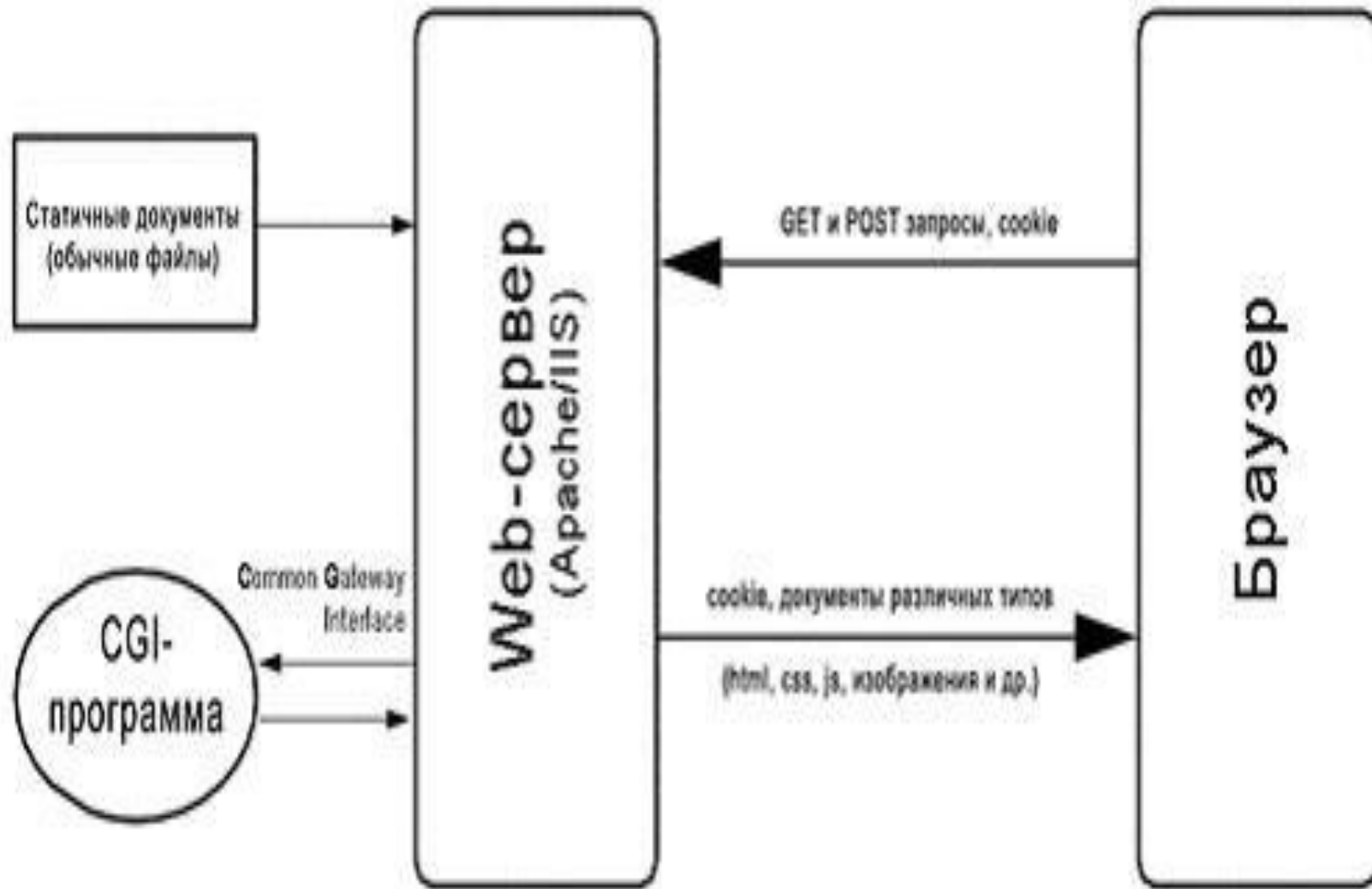
1. CGI -технологиясы.
2. CGI шлюзді интерфейсі.
3. Шлюздерге деректер жіберу.
4. Шлюз арқылы ақпаратты шығару.

## **CGI- технологиясы.**

Common Gateway Interface -веб-серверге сыртқы бағдарламасын қосу үшін пайдаланылатын стандартты интерфейс.

Біздің сұранысымызға сәйкес дайындалған динамикалық беттер “жылдам қалыптасты”. Мысалы кез-келген пікірлер кітабы сізге белгілі бір форманы көрсетеді бұнда сіз өз мәтініңізді қосасыз, келесіде осы бетті ашсаңыз онда жаңа хабарлама тұрады. Web-парағына динамикалық мазмұнды қосуға мүмкіндік жасайтын технологиялардың бірі CGI (Common Gateway Interface) болып табылады. Ол сол немесе басқа URL мен статистикалық документі емес программаны түсінуге мүмкіндік жасайды нәтижесінде нақты уақытта мәліметтер қалыптастырады. Мысалы егер сіз белгілі бір районда ауа райынын дер кезіндегі мәліметінін бергіңіз келсе онда сіз әр бір ретте жаңа бетті құруыңыз керек. Бұл CGI технологиясының негізінде жүзеге асыруы мүмкін. Серверде жұмыс істеу бастағанда CGI программасы қосылады, ол цифрлы өлшеуіш құралына айналып температура,қысым және т.б мәліметтер береді. Әр кезде осы адрестен мәлімет алу үшін байланысқаныңызда сіз сол уақыттағы мәліметті аласыз. Басқа мысал: егер сіз ізденіс жүйесінен белгілі мәлімет алғыңыз келсе онда CGI программасы жұмысының нәтижесін ізделінді адрестер жыйынтығы түрінде аласыз.

# Hyper Text Transfer Protocol



CGI программасын нақты уақыттағы Web сервердің бір бөлігі ретінде қарастыруға болады. Сервер тұтынушының сұрағын CGI программасына береді ол оларды өңдеп жұмыс нәтижесін тұтынушы экранына қайтарып береді. Клиент үшін адресіте URL статистикалық құжат па немесе CGI программа ма еш қандай айырмашылығы жоқ. CGI программалары жұмысының нәтижесі статистикалық құжат сияқты форматта болады. CGI терминін тек қана программа емес протокол ретін де түсінуге болады. Бұл жағдайда CGI Web сервер үшін стандартты тәсіл болып табылады— тұтынушы сұранысын бағдарламаға беру және одан мәлімет алу. Сервер мен оның қосымшасының арасындағы бір біріне мәлімет жіберу жөніндегі CGI протоколы HTTP протоколының бір бөлігі болып табылады. CGI программасының үлкен бөлігі CGI скрипталары болып табылады. Скрип дегеніміз интерпритацияланатын немесе басқа программаларымен жұмыс жасайтын ережелер жинағы. Perl, JavaScript тілдері тура осы скрипталық тілдер түрінде ойлап табылған. Олар сценариилер жазу тілдері деп те аталады. Негізінен CGI программасы скрипталық тілде және де компилирлық тілде жазылуы мүмкін. C,C++,Delphi.



CGI альтернативті технологиясы Microsoft компаниясының технологиясы болып табылады. Ол былай аталады Active Server Page (ASP) ол да сол принциппен құрылған: web- серверге қосылған скрипт, парақ тұтынушыға жіберуден бұрын серверде орындалады. Осы принциппен орындалатын басқа да бір қатар технологиялар бар. Динамикалық мазмұны сервер жағындағы қалыптасатын схемадан өзгеше динамикалық мазмұны тұтынушы жағында көрінеді. Соңғы жағдайда активті құжаттар web серверде және локальдік компьютерде сақталады. Онда белгілі бір есептеулер орындалады және осы есептеулер нәтижесі экранда көрсетіледі. Активті құжаттарды дайындауда түрлі технологиялар пайдаланады: бұл мәліметтер JavaScript-те Java апплеттерде.

## CGI шлюзді интерфейс

CGI – Common Gateway Interface спецификациясы NCSA тобымен ұсынылды. CGI интерфейсі арнайы www мүмкіндіктерін кеңейту үшін өңделді. CGI интерфейсінде қосылу әдісі қарапайым болып келеді. HTTP хаттамасында сипатталған кіру әдістері CGI интерфейсін жүзеге асыруда маңызды орын алады.

Web-серверлердің қолданбалы интерфейстері Web-сервердегі адресі кеңістікте динамикалық кітапханалар (Windows-тегі DDL және UNIX-те қолданатын объектілер) түрінде жүзеге асырылады. Web-серверлердің Netscape және Microsoft өндірушілері өздерінің API кеңейтілу серверлерін өңдеді: InternetInformationServer-ге арналған ISAPI және Netscape серверіне арналған NSAPI.

ISAPI-дің DDL-кітапханалары екі кіріс нүктеге ие болады:

- Get Extension Version – кеңейтілу түрінің номерін алу;
- HttpExtensionProc – кеңейтілудің main-процедурасының эквиваленті;

HttpExtensionProc-тағы ақпарат тек бір параметр көмегімен беріледі.

ISAPI-дің негізгі функциялары:

- `GetServerVariable` – қосымша ақпаратты аты бойынша сұрау;
- `ReadClient` – ақпаратты HTTP-сұранысынан ақпаратты санау;
- `WriteClient` – ақпаратты HTTP-клиентке жіберу;
- `ServerSupportFunction` – серверге орналасу және процестің жағдайы туралы ақпаратты қайтару.

NSAPI интерфейсі ISAPI интерфейсіне ұқсаса болып келеді, бірақ сервер конфигурациясымен тығыз байланысты және анағұрлым күрделі болып келеді. NSAPI интерфейсінің әрбір функция конфигурациясы Netsite конфигурациясының объектілі мәліметтер базасында берілу қажет.

NSAPI параметрлерінің блоктары HTML-формаларындағы диалогтық айнмалылардың берілуіне ұқсас name-Value параларына негізделеді.



## Шлюздерге деректерді жіберу.

Серверден шлюзге ақпараттық тапсырыс жөнінде деректерді жіберу үшін сервер командалық жолды және айнала айнымалыларын қолданады. Бұл айнала айнымалылары сервер шлюз бағдарламасын орындау уақытында орнатылады.

Әртүрлі әдістер үшін тапсырыстар.

Шлюздерге ақпарат келесі формада беріледі:

*аты=мәні&аты1=мәні1&...*

мұнда аты – айнымалы аты (FORM операторынан, мысалы), және мәні – оның нақты мәні.

Тапсырыс үшін қолданылатын әдіске байланысты бұл жол URL бөлімі ретінде (GET әдісі жағдайында) немесе HTTP тапсырыстың құрамы ретінде (POST әдісі) пайда болады.

Соңғы жағдайда, бұл ақпарат енгізудің стандартты ағымына шлюзге жіберілетін болады.

Енгізудің стандартты ағымының файлдық дескрипторына CONTENT\_LENGTH байт жіберіледі. Сонымен қатар сервер шлюзге CONTENT\_TYPE (жіберілетін деректер типі) жібереді. Сервер CONTENT\_TYPE деректер байтын жібергеннен кейін және шлюз оларды оқығаннан кейін файл соңының символын жіберуге міндетті емес.

## Мысал:

Мысал ретінде форманың POST (METHOD="POST") әдісімен жұмыс нәтижесін алайық. Жуықтап осылай `a=b&b=c` кодталған 7 байт алынған болсын.

Бұл жағдайда сервер `CONTENT_LENGTH` мәнін 7 тең деп және `CONTENT_TYPE` мәнін `application/x-www-form-urlencoded`-ке орнатады. Енгізудің стандартты ағымында шлюз үшін бірінші символ «a» болады, одан кейін кодталған жолдың қалдығы жүреді.

Командалық жолдың аргументтері.

Командалық жолда шлюз серверден алатыны:

бірінші параметр ретінде шлюз атынан кейін URL қалдығы (егер тек шлюз аты болған жағдайда, бірінші параметр бос болады), және іздеу скрипті үшін командалық жол қалдығы ретінде кілттік сөздер тізімі, немесе қосылған теңдік белгісімен (жұп позицияларда) және айнымалылардың сәйкестік мәндерімен (тақ позицияларда) формалар өрістерінің тізімделген аттары

Кілттік сөздер, форма өрістерінің аттары және мәндер шлюз командалық жолда қосымша өзгертулерді жүзеге асыру қажетінсіз ақпаратты сол күйінде алатындай Bourne shell кодтау ережелерімен сәйкес кодтары шешілген (HTTP URL кодтау форматынан) және қайта кодталған болып тасымалданады.

FORM операторының тапсырыстары.

FORM операторының тапсырыстары өріс атына жауап беретін әрбір параметр теңдік белгісімен бітетіндей, ал қалдық осы параметр мәні болатындай өңделеді. Мысалдар:

```
/htbin/script/x/y/z?name1=value1&name2=value2
```

келесідей шақырылады:

```
/.../script/x/y/z name1=value1 name2=value2
```

CGI айнала айнымалылары

Келесі айнала айнымалылары тапсырыс типі бойынша спецификалық болып табылмайды және барлық тапсырыстар үшін орнатылады.

SERVER\_SOFTWARE

Тапсырысқа жауап беретін (және шлюзді жүктейтін) ақпараттық сервердің аты және версиясы.

Форматы: аты/версиясы

SERVER\_NAME

Сервер жүктелген хост аты, DNS аты, немесе URL-де көрсетілген түрдегі IP адрес.

GATEWAY\_INTERFACE

Сервер құрастырылған кездегі CGI спецификациясының версиясы.

Форматы: CGI/версиясы

Келесі айнала айнымалылары әртүрлі тапсырыстар үшін спецификалық болып табылады, және шлюз шақырылуы алдында толтырылады.

SERVER\_PROTOCOL

Тапсырыс келген ақпараттық протоколдың аты мен версиясы.

Форматы: протокол/версиясы

SERVER\_PORT

Сұраныс жіберілген порттың нөмірі

REQUEST\_METHOD

Сұраныс үшін пайдаланған әдіс. HTTP үшін, ол “GET”, “HEAD”, “POST” және т.б.

PATH\_INFO

Клиент жіберген жол туралы ақпарат. Басқа сөзбен айтқанда, шлюзге қатынас вертуалды түрде де жүзеге асырылады. Оның арнайы қосымша ақпараттары бар. Ол ақпарат PATH\_INFO да беріледі.

PATH\_TRANSLATED

Сервер PATH\_INFO-ның өзгертілген нұсқасын жібереді. Ол өзінде вертуалды түрден физикалық түрге түрге түрлендірілген жолды қамтиды.

SCRIPT\_NAME

URL ды алу үшін пайдаланылатын шлюздік вертуалды жол.

QUERY\_STRING

Жоғарыда аталған шлюзге тікелей қатынасы бар URL ге жөнелтілетін ақпарат. Бұл ақпарат сұраныс жолы тәріздес. Ол өзгертілмеуі тиіс. Сұраныс жолына тәуелсіз бұл айнымалы міндетті түрде ақпаратта көрсетілуі тиіс.

REMOTE\_HOST

Сұраныс тудырып отырған хост аты. Егер серверде мұндай ақпарат жоқ болса, онда ол REMOTE\_ADDR ді орнатуы керек.

REMOTE\_ADDR

Сұраныс тудырып отырғын хосттың IP адресі.

AUTH\_TYPE

Егер сервер пайдаланушының идентификациясын қолдап, шлюзге кездейсоқ қатынастан қорғалған болса, онда бұл идентификация тәсілі пайдаланушыны тексеру үшін пайдаланады.

REMOTE\_USER

Пайдаланушының атын сақтау үшін пайдаланылады.

REMOTE\_IDENT

Егер HTTP сервер пайдаланушының идентификациясын RFC 931 ге келісімі қолдайтын болса, онда бұл айнымалы серверден алынған пайдаланушы есімін сақтайды.

CONTENT\_TYPE

Қосымша ақпараттары бар сұраныстардың түрін анықтау үшін пайдаланады.

Мысалы: HTTP, POST, PUT.

CONTENT\_LENGTH

Пайдаланушы жіберген ақпарат ұзындығы.

Оған қосымша егер сұраныста қосымша жолдар болса, онда олар айнамадыға жіктеледі, одан кейін тақырып аты жазылады. Кез-келген '-' символы '-' символына алмастырылады. Сервер өңделп аяқталған кез-келген тақырыпты алып тастай алады, мысалы: Authorization, Content-type, Content-length. Және де тақырып атының ұзындығы қажетті ұзындықтан асып кеткен болса да сервер оны жойып жібере алады. Ондай айнымалы мысалы HTTP\_ACCEPT бола алады. Ол CGI/1.0 спецификациясында анықталды. Ал басқа мысал ретінде User-Agent қарастыруға болады.

HTTP\_ACCEPT

HTTP тақырыптарында берілгендей, қолданушы өңдей алатын MIME типтерінің тізімі. Басқа протоколдар бұл ақпаратты басқа жерлерден алулары тиіс. HTTP спецификациясына сай бұл тізімдегі әр тип үтірмен ажыратылуы тиіс. Форматы: тип/ішкі тип, тип/ішкі тип.

HTTP\_USER\_AGENT

Клиент сұранысын жөнелту үшін қолданады. Форматы: бағдарлама/кітапхана нұсқасы/нұсқасы.

## **Шлюз арқылы ақпаратты шығару.**

Шлюз өзінің қорытындысын стандартты қорытындыға қосып жібереді. Ол қорытынды шлюз көмегімен генерацияланған құжат түрінде болуы мүмкін. Тәртіп бойынша шлюз қорытындысын шығарып оны кері қарай клиентке жөнелтеді. Бұл қатынастың басты ұтымдылығы шлюз HTTP/1.0 тақырыбын әр сұранысқа жіберіге міндетті емес.

### **Шығытын ағын тақырыбы**

Кейбі шлюздерге сервер арқылы емес клиентпен тікелей қатынас жасаған дұрысырақ. Мұндай шлюздерді өзге шлюздерден айыру үшін CGI талабы бойынша ол шлюздердің аты `prh-` префиксінен басталуы тиіс. Бұл жағдайда шлюзге синтаксиздік дұрыс жауапты клиентке қайтару міндеті жүктеледі.

## Синтаксиздік жіктелген тақырыптар

Шлюз қорытындысы кішкентай тақырыпшадан басталады. Ол HTTP дағыдай форматтағы мәтіндік жолдардан тұрады да бос жолмен аяқталады. ( алмастыру символынан немесе CR/LF ден басқа).

Сервер директивасына қатынсы жоқ тақырыптың кез-келген жолы клиентке міндетті түрде жіберілуі тиіс. Қазіргі кезде CGI сервер директивасының үш түрін ажыратады:

Content-type

MIME қайтарылатын құжат типі.

Location

Бұл серверге құжаттың өзі емес тек оның нұсқамасы ғана қайтарылғанда қолданылады.

Егер URL аргумент болса, онда сервер клиентке сұранысты қайта жіберуін өтінеді. Ал аргумент виртуалды жол түрінде болса, онда сервер клиентке осы жол арқылы алынғын құжатты кері қайтарады.

Status

Бұл директива серверге HTTP/1.0 статус жолын баяндау үшін қолданылады. Ол клиентке жөнелтіледі. Формат: nnn xxxxx, бұл жерде nnn 3 сандық статус-код, ал xxxxx мәселе жолы.



Мысалы:

Мәтіндік HTML конвертер бар деп алайық. Ол өз жұмысын аяқтағаннан кейін келесі қорытындыны стандартты қорытындыдан көрсетуі тиіс:

---- қорытынды басы----

Content-type: text/html

----қорытынды соңы----

Енді жоғарыда аталған серверден /path/doc.txt құжатын шығаратын шлюзді қарастырайық. Ал ол құжат клиентпен <http://server:port/path/doc.txt> арқылы сұраныс берілген болсын. Бұл жағдайда шлюзден шығатын қорытынды мынадай болмақ:

--- қорытынды басы---

Location: /path/doc.txt

--- қорытынды соңы---

Соңынан шлюз gopher серверге <gopher://gopher.ncsa.uiuc.edu/> сілтемесін жібереді. Бұл жағдайда шлюзден шығатын қорытынды мынадай болмақ:

--- қорытынды басы---

Location: <gopher://gopher.ncsa.uiuc.edu/>

--- қорытынды соңы---

Осылайша бағдарламашы бағдарлама құрған кезде CGI талаптарын қанағаттандырғаны жөн.

**НАЗАРЛАРЫҢЫЗҒА РАХМЕТ!!!**