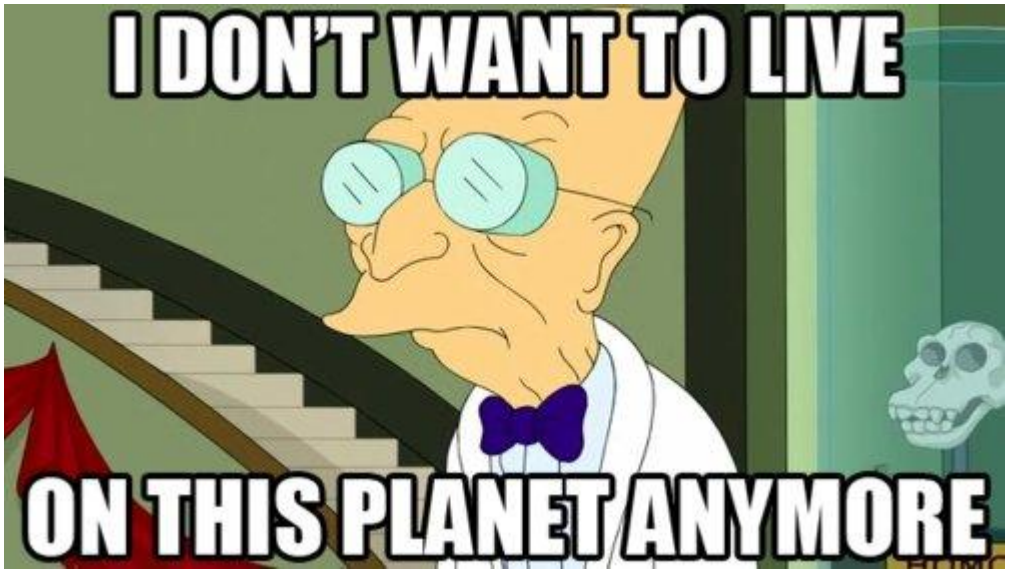
Прикладная разработка на C++
Лекция 1

Интерфейс CGI

C++ на HighLoad?



Билайн 11:22 92 %

highloadcup.ru

Highload Cup

Актуальная рейтинговая таблица

#	Участник
1	Алексей Дичковский (C++)
2	Иван Тямгин (C++)
3	Александр Шумский (C++)
4	Максим Андреев (C)
5	Oleg Kuznetsov (++C)
6	Никита Уваров (C++ bicycle)
7	Андрей Колышкин (C)
8	Алексей Акулович (Go, bicycl
9	Александр Харитонов (Java, C
10	Павел Кингсеп (C bicycle)

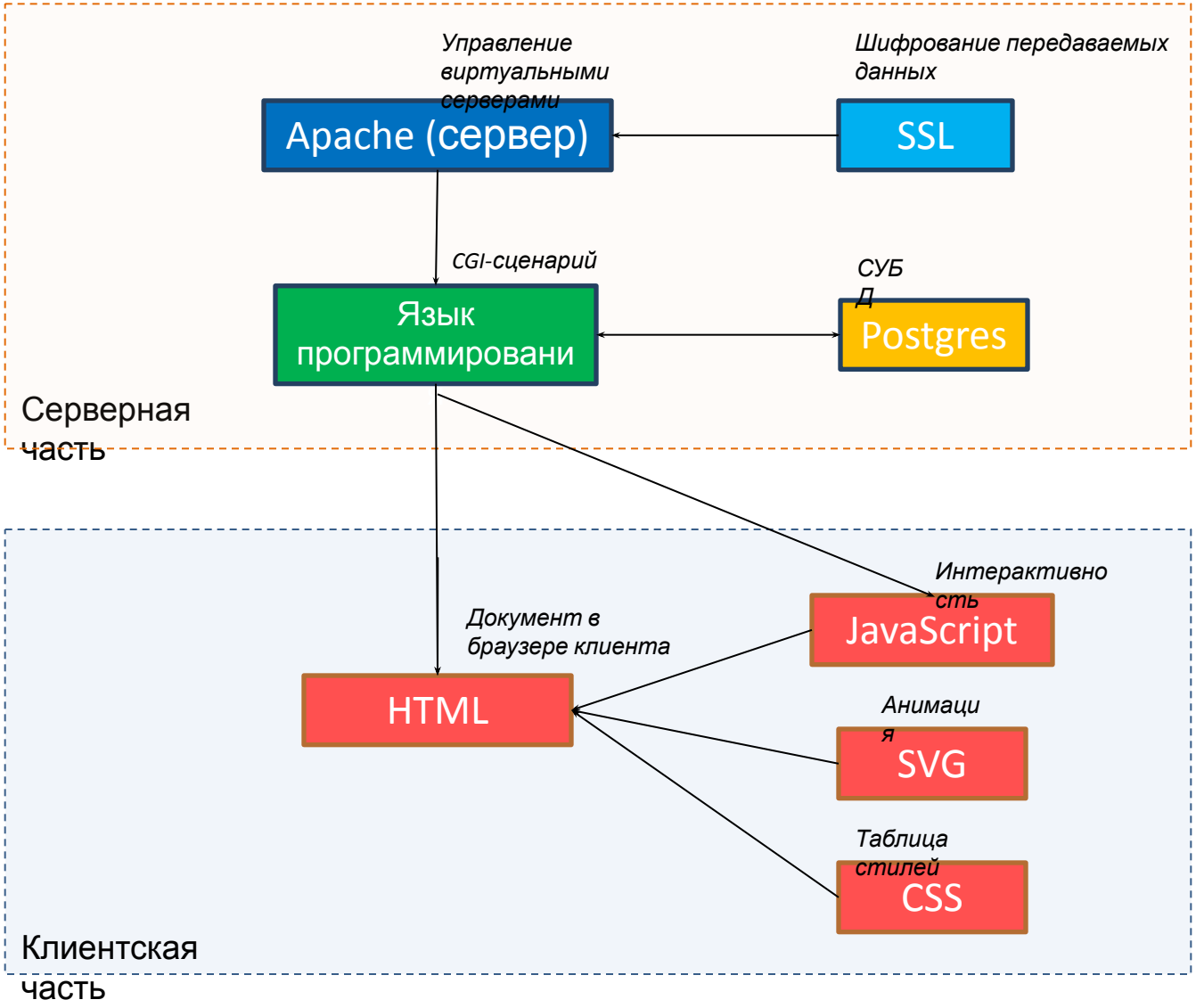
Динамические и статические страницы

- Динамическая страница — Web-страница, сгенерированная с помощью логически построенной программы в зависимости от запрошенных пользователем данных.
- Статическая страница является простым файлом, лежащим на сервере.

CGI

- **CGI** – (**Common Gateway Interface** – Общий интерфейс маршрутизации) служит для обеспечения связи внешней прикладной программы с Web-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, или «скриптом» (сценарием), «CGI-программой».
- Такая программа генерирует динамические страницы

Архитектура Web



Выполнение программы

- Обычно CGI-программы находятся в служебной директории “/cgi-bin”, однако это зависит от конфигурации Web-сервера.
- На нашем сервере (mati.su) CGI-программа может быть исполнена в случае, если её исполняемый файл имеет расширение .cgi и находится в директории Web-сервера.

Языки программирования

- Сам интерфейс разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода/вывода.
- На нашем сервере могут использоваться следующие скриптовые языки: Perl, PHP, Ruby, Python, shell-script, TCL.
- Установлены также компиляторы языков Assembler, C/C++, Pascal.

CGI-программа на языке C++

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Content-Type: text/html; charset=utf-8«
      << endl << endl;
    cout << "<p>Привет!</p>" << endl;
    cout << "<p>Ваш IP-адрес: " << getenv("REMOTE_ADDR")
<< ".</p>" << endl;
    cout << "<p>Ваш браузер: "
      << getenv("HTTP_USER_AGENT") << ".</p>" << endl;
    return -1;
}
```

Вывод


pavel.mati.su/test.cgi
Привет!
Ваш IP-адрес: 217.9.88.22.
Ваш браузер: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:68.0) Gecko/20100101 Firefox/68.0.

Компиляция программ

На языках C и C++:

- `g++ hello.cpp -owww/hello.cgi`

На языке Паскаль:

- `fpc hello.pas -owww/hello.cgi`

Принципы получения данных динамической страницей

- Через HTML-формы методами GET и POST
- Через HTTP-Cookies
- Через переменные окружения Web-сервера

Форма

Форма создаётся с помощью тега <form>, внутри неё могут быть любые необходимые теги, и характеризуется она следующими необязательными параметрами:

- адрес программы на веб-сервере, которая будет обрабатывать содержимое данных формы;
- элементами формы, которые представляют собой стандартные поля для ввода информации пользователем;
- кнопку отправки данных на сервер.

Атрибут action

```
<!DOCTYPE HTML>
<html>
<head>
 <meta charset="utf-8">
 <title>Данные формы</title> </head>
<body>
 <form action="/example/handler.cgi">
 <p>
  <input name="login">
  <input type="password" name="pass">
 </p>
 <p><input type="submit"></p>
 </form>
</body>
</html>
```

Указание метода передачи данных

- Для указания в форме метода передачи данных используется атрибут “method”, принимающий значения “GET” или “POST”.

Передача данных методами GET и POST

Метод GET

- Сохраняется в строке URL-адреса; адрес с запросом GET можно сохранить в закладках
- Кэшируется браузером
- Данные передаются в открытом виде и могут быть перехвачены
- В CGI передаётся через переменную окружения QUERY_STRING.

Метод POST

- «Невидим» для пользователя при отправке
- Передача данных происходит в теле запроса
- Способен передавать большие объёмы данных
- Способен передавать файлы
- В CGI передаётся через поток ввода.

Метод GET

http://yandex.ru/yandsearch?text=мемы+из+футурамы&lang=ru

CGI-сценарий

Запрос

В данном примере CGI-сценарию *yandsearch* передаётся строка запроса, внутри которой содержатся переменные “*text*” со значением “*мемы из футурами*” и “*lang*” со значением “*ru*”. Заголовок запроса начинается так:

GET /yandsearch?text=
%D0%BC%D0%B5%D0%BC%D1%8B%20%D0%B8%D0%B7%20%D1%84
%D1%83%D1%82%D1%83%D1%80%D0%B0%D0%BC%D1%8B &lang=ru
HTTP 1.1

Чередование методов GET и POST

```
POST /passport?mode=auth HTTP/1.1
Host: passport.yandex.ru
User-Agent: Mozilla/5.0 (Windows NT 6.1)
Gecko/20100101 Firefox/23.0
Content-Type: application/x-www-form-urlencoded
Content-Length: 62

from=passport&login=john&passwd=mypass&timestamp
=1379013756458
```

Отправка формы

```
<body>
 <form action="test.cgi" method="POST">
  <p><input name="user"></p>
  <p><input type="submit" value="Отправить"></p>
 </form>
</body>
```

Чтение данных, переданных методом POST

- Данные, передаваемые методом POST, обрабатываются CGI-программой путём чтения стандартного устройства ввода.

HTML-код:

```
<form action="test.cgi" method="POST">
    <input type="text" name="parameter" value="Какое-нибудь значение">
    <input type="submit" value="Send data">
</form>
```

C++ код:

```
string postData;
cin >> postData;
cout << "Значение POST-данных равно: " << str << endl;
```

Переменные CGI-окружения

- **CONTENT_LENGTH** – величина данных, переданных методом POST и подлежащих считыванию в стандартное устройство ввода.
- **DOCUMENT_ROOT** – абсолютный путь до директории Web-сервера, откуда выполняется CGI-сценарий.
- **HTTP_REFERER** – путь URL, откуда пришёл пользователь, запустив CGI-сценарий.

Переменные CGI-окружения

- **HTTP_USER_AGENT** – имя и версия клиента, используемого пользователем.
- **QUERY_STRING** – строка запроса, часть строки адреса после знака “?”. По сути данные, переданные методом GET.
- **REMOTE_ADDR** – IP-адрес клиента.
- **REQUEST_METHOD** – метод, с помощью которого клиент передаёт данные.

Переменные CGI-окружения

- **SCRIPT_NAME** – имя CGI-сценария, который выполняется в данный момент.
- **SERVER_NAME** – доменное имя или IP-адрес сервера, на котором выполняется CGI-сценарий.
- **SERVER_SOFTWARE** – тип сервера, на котором выполняется CGI-сценарий.

HTTP Cookies

- **HTTP Cookie (куки)** – небольшая порция текстовых данных, отправляемая веб-сервером и хранящаяся в браузере клиента. Браузер всякий раз при открытии страницы соответствующего сайта пересылает сохранённый фрагмент данных обратно веб-серверу через HTTP-заголовки.

HTTP Cookies

Куки используются для:

- аутентификации пользователя;
- хранения персональных предпочтений и настроек пользователя;
- отслеживания состояния сеанса доступа пользователя;
- ведения статистики о пользователях.

Установка Cookie

- В заголовке HTTP-ответа веб-сервера может содержаться указание браузеру сохранить куки:

```
HTTP/1.1 200 OK
Content-Type: text/html
Set-Cookie: name=value

Содержимое страницы
```

Установка Cookie

- Строка Set-Cookie, как правило, добавляется к HTTP-ответу не самим HTTP-сервером, а CGI-программой, работающей вместе с ним. HTTP-сервер только отправляет браузеру результат работы такой программы.

Чтение Cookie

- Строка Set-Cookie отправляется только тогда, когда сервер желает, чтобы браузер сохранил куки. В этом случае браузер запомнит строку name=value и отправит её обратно серверу с каждым последующим запросом.

```
GET /spec.html HTTP/1.1
Host: www.example.org
Cookie: name=value
```

- Значение Cookie может быть изменено повторной отправкой сервером «Set-Cookie».

Атрибуты Cookie

- Кроме пары «имя/значение» куки может содержать **срок действия**, **путь** и **доменное имя**, на которое оно распространяется. Пример:

```
Set-Cookie: name=newvalue; expires=date;
path=/; domain=.example.org.
```

Атрибуты Cookie

- Домен и путь говорят браузеру, что куки нужно отправлять обратно на сервер при запросах URL для указанного домена и пути. Если они не указаны, используются домен и путь запрошенной страницы.
- Дата истечения указывается в формате «Нед, ДД Мес ГГГГ ЧЧ:ММ:СС GMT». Например:
- `Set-Cookie: RMID=732423sdfs73242; expires=Fri, 31 Dec 2013 23:59:59 GMT; path=/; domain=.example.net`

Типы Cookie

- **Куки сессии** – существует только на то время, пока пользователь производит навигацию по сайту. Куки сессии создаётся автоматически, если не указан срок действия куки.
- **Постоянные куки** – существует до тех пор, пока не закончится срок действия куки. Например, если куки имеет атрибут Max-Age установленный на 1 год (например), то значение Cookie будет отправляться браузером на Web-сервер при каждом обращении в течение года.

Безопасность Cookie

- Куки легко перехватить и подменить (например, для получения доступа к учетной записи), если пользователь использует нешифрованное соединение с сервером.

Способы задания Cookie

1) Через клиентский JavaScript
2) Через прямую установку HTTP-заголовков на сервере

Cookie в CGI

- Получение Cookie в среде CGI происходит с помощью переменной окружения **HTTP_COOKIE**, которая в точности повторяет HTTP-заголовок клиента «**Cookie**».
- Формат Cookie имеет следующий вид:

```
name=value; name2=value2
```

Перенаправления

301 Moved Permanently — постоянный редирект.

302 Moved Temporarily — временный редирект. Значит, страница может быть возвращена по старому адресу.

Для изменения HTTP-статуса применяется псевдозаголовок “Status”:

```
cout << "Status: 302 Found" << endl;
cout << "Location: /form.html" <<
endl << endl;
```

Настройка Web-сервера Apache

- a2moden cgi
- systemctl restart apache2

Конфигурация файла .htaccess Web-сервера Apache

```
AddHandler cgi-script .cgi
Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
Require all granted
```

Необходимые компоненты

- std::string
- std::iostream
- std::vector или std::map
- Потоки cout, cin, cerr

Лабораторная работа

- Разработать библиотеку, способную принимать данные методами GET/POST и работать с HTTP-Cookie.
- **Написать CGI-программу, использующую разработанную библиотеку и реализующую работу с базой данных со следующими CGI-сценариями: добавление записи, удаление, просмотр списка, просмотр одной записи. Сохранить предыдущие введённые данные в форме добавления записи в Cookie.**
- Записи в базе данных хранятся построчно в файле, а отдельные поля записи разделены символами-разделителями.
- Ограничения: Использование только стандартных библиотек. Использовать Boost и др. библиотеки не допускается. Можно: STL и C++17.

Структура класса CGI

```
class CGI
{
    public:
    CGI();
    std::string httpGet(std::string name);
    std::string httpPost(std::string name);
    std::string getCookie(std::string name);
    std::string setCookie(std::string name, std::string value);
    ~CGI();
   private:
    …
}
```

Список литературы

- http://www.cplusplus.com/reference/string/string/
- https://www.youtube.com/watch?v=z7UtcqqQ1Pc - C++ and CGI
- Deitel P., Deitel H. C++ how to Program. – Pearson, 2016.
- Кейно П. П., Силуянов А. В. Разработка и внедрение интерпретатора декларативного языка моделирования Web-интерфейсов на высоконагруженных системах // Прикладная информатика. – 2015. – №. 1 (55).