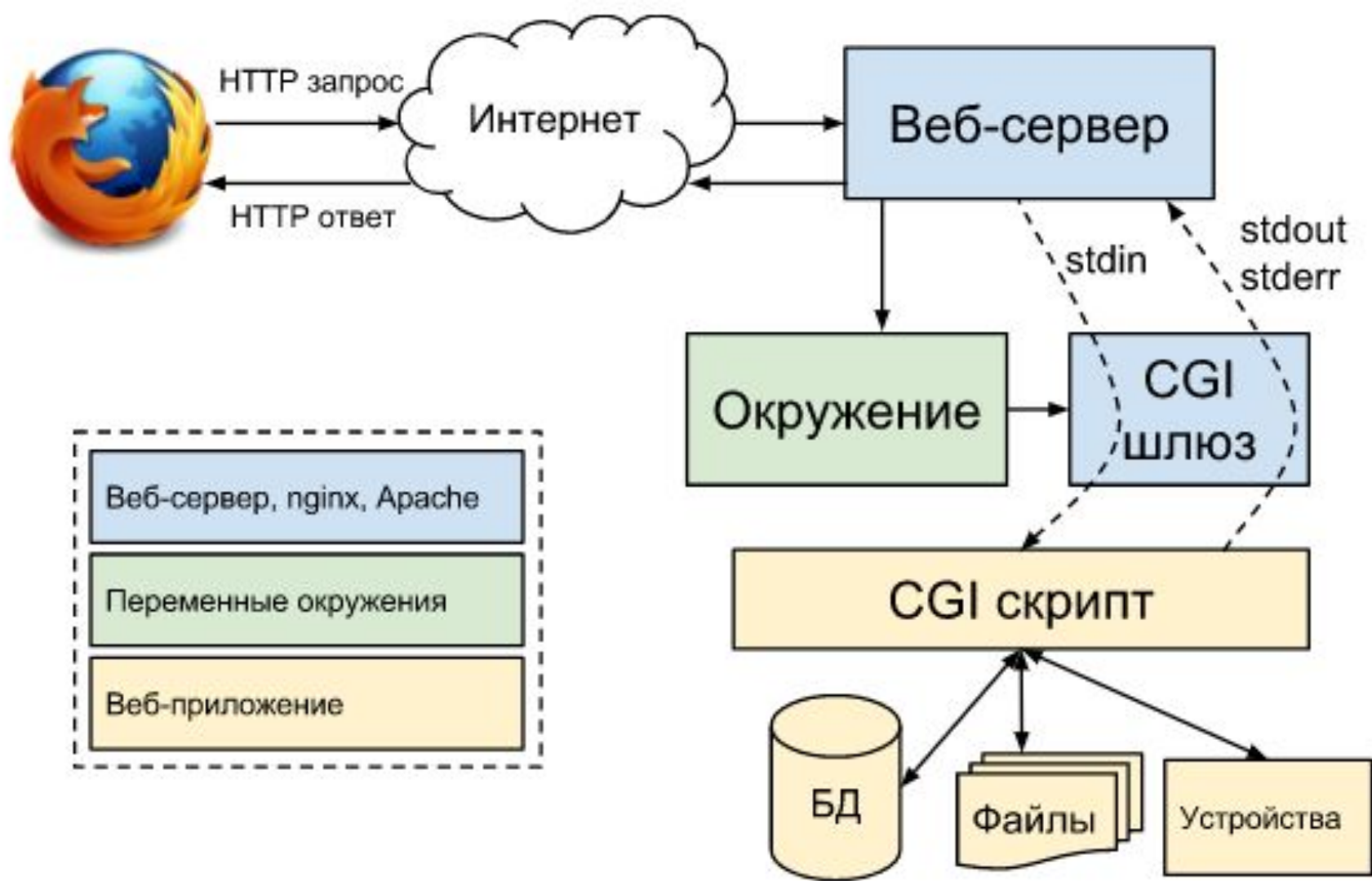


CGI

- **CGI** (от англ. Common Gateway Interface — «общий интерфейс шлюза») — стандарт интерфейса, используемого для связи внешней программы с веб-сервером. Программу, которая работает по такому интерфейсу совместно с веб-сервером, принято называть шлюзом, хотя многие предпочитают названия «скрипт» (сценарий) или «CGI-программа».
- Поскольку гипертекст статичен по своей природе, веб-страница не может непосредственно взаимодействовать с пользователем. До появления *JavaScript*, не было иной возможности отреагировать на действия пользователя, кроме как передать введенные им данные на веб-сервер для дальнейшей обработки. В случае CGI эта обработка осуществляется с помощью внешних программ и скриптов, обращение к которым выполняется через стандартизованный (см. RFC 3875: CGI Version 1.1) интерфейс — общий шлюз.



Сам интерфейс разработан таким образом, чтобы можно было использовать любой язык программирования, который может работать со стандартными устройствами ввода-вывода. Такими возможностями обладают даже скрипты для встроенных командных интерпретаторов операционных систем, поэтому в простых случаях могут использоваться даже командные скрипты.

# Как работает CGI?

Обобщенный алгоритм работы через CGI можно представить в следующем виде:

- Клиент запрашивает CGI-приложение по его URI.
- Веб-сервер принимает запрос и устанавливает переменные окружения, через них приложению передаются данные и служебная информация.
- Веб-сервер перенаправляет запросы через стандартный поток ввода (stdin) на вход вызываемой программы.
- CGI-приложение выполняет все необходимые операции и формирует результаты в виде HTML.
- Сформированный гипертекст возвращается веб-серверу через стандартный поток вывода (stdout). Сообщения об ошибках передаются через stderr.
- Веб-сервер передает результаты запроса клиенту.

# Области применения CGI

- Наиболее частая задача, для решения которой применяется CGI — создание интерактивных страниц, содержание которых зависит от действий пользователя. Типичными примерами таких веб-страниц являются форма регистрации на сайте или форма для отправки комментария. Другая область применения CGI, остающаяся за кулисами взаимодействия с пользователем, связана со сбором и обработкой информации о клиенте: установка и чтение «печенюшек»-cookies; получение данных о браузере и операционной системе; подсчет количества посещений веб-страницы; мониторинг веб-трафика и т.п.
- Эти возможности обеспечиваются тем, что CGI-скрипт может быть подключен к базе данных или обращаться к файловой системе сервера. Таким образом CGI-скрипт может сохранять информацию в таблицах БД или файлах и получать ее оттуда по запросу, чего нельзя сделать средствами HTML.

## Предупреждение

- CGI — это не язык программирования! Это простой протокол, позволяющий веб-серверу передавать данные через `stdin` и читать их из `stdout`. Поэтому, в качестве CGI-обработчика может использоваться любая серверная программа, способная работать со стандартными потоками ввода-вывода.

## Примеры

Пример на Python:

```
#!/usr/bin/python
print("""Content-Type: text/plain
Hello, world!""")
```

В этом коде строка `#!/usr/bin/python` указывает полный путь к интерпретатору Python.

Пример на Си:

```
#include <stdio.h>
int main(void) {
    printf("Content-Type: text/plain\n\n");
    printf("Hello, world!\n\n");
    return 0;
}
```

Строка `Content-type: text/html\n\n` — http-заголовок, задающий тип содержимого (mime-type). Удвоенный символ разрыва строки (`\n\n`) — обязателен, он отделяет заголовки от тела сообщения.

Все скрипты, как правило, помещают в каталог `cgi` (или `cgi-bin`) сервера, но это необязательно: скрипт может располагаться где угодно, но при этом большинство веб-серверов требуют специальной настройки. В веб-сервере Apache, например, такая настройка может производиться при помощи общего файла настроек `httpd.conf` или с помощью файла `.htaccess` в том каталоге, где содержится этот скрипт. Также скрипты должны иметь права на исполнение (`chmod +x hello.py`).

## Переменные окружения

Все CGI-приложения имеют доступ к переменным окружения, устанавливаемым веб-сервером. Эти переменные играют важную роль при написании CGI-программ. В таблице перечислены некоторые из переменных, доступных CGI.

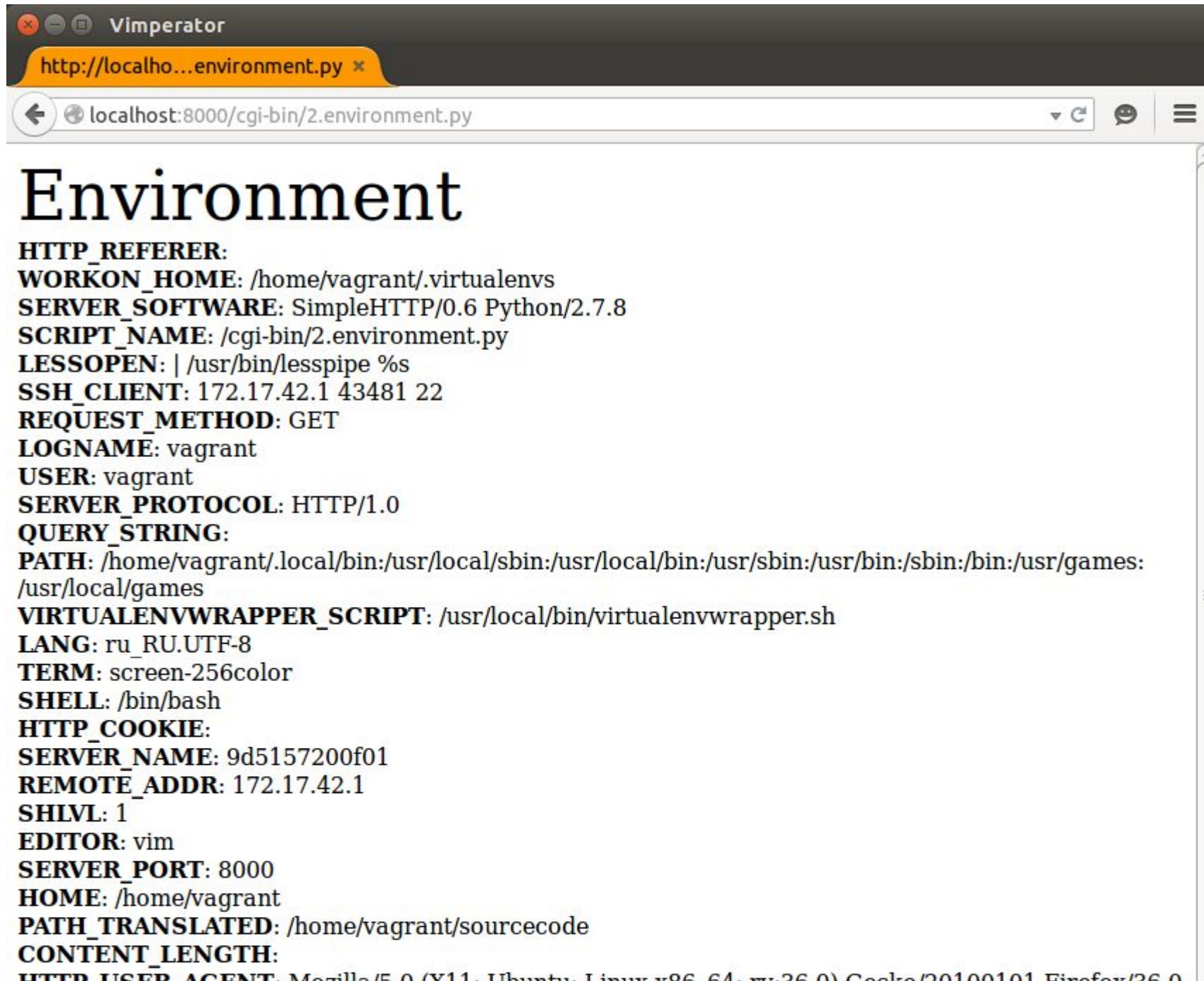
### Переменная Описание

CONTENT_TYPE	Тип данных, передаваемых на сервер. Используется, когда клиент отправляет данные, например, загружает файл.
CONTENT_LENGTH	Размер содержимого запроса. Эта переменная определена для POST-запросов.
HTTP_COOKIE	Возвращает набор «куков» в виде пар «ключ значение».
HTTP_USER_AGENT	Информация об агенте пользователя (браузере).
PATH_INFO	Путь к каталогу CGI.
QUERY_STRING	Строка запроса (URL-encoded), передаваемая методом GET.
REMOTE_ADDR	IP-адрес клиента, выполняющего запрос.
REMOTE_HOST	Полное имя (FQDN) клиента. (Если доступно)
REQUEST_METHOD	Метод, которым выполняется запрос. Чаще всего GET или POST.
SCRIPT_FILENAME	Полный путь к запрашиваемому скрипту (в файловой системе сервера).
SCRIPT_NAME	Имя скрипта.
SERVER_NAME	Имя сервера.
SERVER_ADDR	IP-адрес сервера.
SERVER_SOFTWARE	Информация о серверном ПО.

# Пример

- `#!/usr/bin/python`
- `import os`
- `print("Content-type: text/html\r\n\r\n")`
- `print("<font size=+10>Environment</font><br>")`
- `for param in os.environ.keys():`
- `print("<b>%20s</b>: %s<br>" % (param, os.environ[param]))`





The image shows a web browser window titled "Vimperator" with a single tab for "http://localho...environment.py". The address bar shows "localhost:8000/cgi-bin/2.environment.py". The main content area displays a list of environment variables in a plain text format.

**Environment**

**HTTP\_REFERER:**  
**WORKON\_HOME:** /home/vagrant/.virtualenvs  
**SERVER\_SOFTWARE:** SimpleHTTP/0.6 Python/2.7.8  
**SCRIPT\_NAME:** /cgi-bin/2.environment.py  
**LESSOPEN:** | /usr/bin/lesspipe %s  
**SSH\_CLIENT:** 172.17.42.1 43481 22  
**REQUEST\_METHOD:** GET  
**LOGNAME:** vagrant  
**USER:** vagrant  
**SERVER\_PROTOCOL:** HTTP/1.0  
**QUERY\_STRING:**  
**PATH:** /home/vagrant/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games  
**VIRTUALENVWRAPPER\_SCRIPT:** /usr/local/bin/virtualenvwrapper.sh  
**LANG:** ru\_RU.UTF-8  
**TERM:** screen-256color  
**SHELL:** /bin/bash  
**HTTP\_COOKIE:**  
**SERVER\_NAME:** 9d5157200f01  
**REMOTE\_ADDR:** 172.17.42.1  
**SHLVL:** 1  
**EDITOR:** vim  
**SERVER\_PORT:** 8000  
**HOME:** /home/vagrant  
**PATH\_TRANSLATED:** /home/vagrant/sourcecode  
**CONTENT\_LENGTH:**  
**HTTP\_USER\_AGENT:** Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:36.0) Gecko/20100101 Firefox/36.0

## **Преимущества CGI**

- Процесс CGI скрипта не зависит от Веб-сервера и в случае падения ни как не отразится на работе последнего
- Может быть написан на любом языке программирования
- Поддерживается большинством Веб-серверов

## **Недостатки CGI**

- Самым большим недостатком этой технологии являются повышенные требования к производительности веб-сервера. Дело в том, что каждое обращение к CGI-приложению вызывает порождение нового процесса, со всеми вытекающими отсюда накладными расходами. Если же приложение написано с ошибками, то возможна ситуация, когда оно, например, зациклится. Браузер прервет соединение по истечении тайм-аута, но на серверной стороне процесс будет продолжаться, пока администратор не снимет его принудительно.