Смирнов Александр Евгеньевич
Балашовский филиал Саратовского Государственного Университета,
г. Балашов
smirnov@balashov.san.ru

ЯЗЫК ПРОГРАММИРОВАНИЯ PERL

Всё чаще и чаще человек имеет дело с Internet. У многих организаций есть сайты и электронные адреса, посредством которых происходит общение с клиентами, как действительными, так и потенциальными, и предлагаются продукты производства либо квалифицированная помощь по поиску интересующей информации. Помимо этого, организации, предоставляющие доступ к Internet физическим лицам (провайдеры), предлагают пользователям электронные почтовые ящики. Но со временем, возможности переписки и поиска информации WWW в сети становится недостаточно даже обычному человеку. Это связано со стремлением создать свои Web-страницы или даже Web-сайты, где было бы возможно выразить свою точку зрения на различные явления и процессы в мире или просто найти своих единомышленников. Со временем этим пользователям становится недостаточно обычных тегов HTML. И тогда предпринимаются попытки оживить свои Web-сайты использованием языков создания клиентских сценариев (VBScript и JavaScript).

Впоследствии появляются всё новые и новые запросы. Это, например, пообщаться с клиентом. Для этого нужны такие программы, как счётчик посещений, гостевая книга, какой-либо социологический опрос, да и неплохо бы создать базу данных, дающую возможность предоставлять различную информацию, интересующую пользователей. Для всего этого недостаточно знания HTML, VBScript и JavaScript, так как они описывают только технологию взаимодействия сервера и клиента.

Вообще, без способности получать ввод от пользователей и давать информацию, Web был бы полностью статической средой. Информация была бы доступна только в формате, определенном автором. Это подорвало бы одну из вычислительных возможностей вообще: интерактивную информацию. Например, вместо того, чтобы принуждать пользователя к просмотру нескольких документов, книгу или словарь, будет лучше позволить пользователю определить ключевые слова по интересующей их тематике. Пользователи могут настраивать представление данных, а не полагаться на жесткую структуру, определенную поставщиком содержимого.

Термин "сервер Web" может ввести в заблуждение, потому что он может относиться и к физической машине и программному обеспечению, используемому в ней для взаимодействия с Internet-браузерами. Когда браузер запрашивает заданный адрес Web, он сначала подключается к машине через Интернет, посылая программному обеспечению Web-сервера запрос о документе. Это программное обеспечение работает непрерывно, ожидая поступления таких запросов и отвечая соответствующим образом.

Хотя серверы могут посылать и получать данные, у самого сервера функциональные возможности ограничены. Например, наиболее примитивный сервер может только отослать требуемый файл на браузер. Сервер обычно не знает, что делать с тем или иным дополнительным вводом. Если Internet-провайдер не сообщает серверу, как обработать такую дополнительную информацию, вероятнее всего сервер проигнорирует ввод.

Для того чтобы сервер смог выполнять и другие операции, помимо поиска и пересылки файлов на Internet-браузер, нужно знать, как расширить функциональные возможности сервера. Например, сервер Web не может производить поиск в базе данных, на основе ключевого слова, введенного пользователем, и возвращать несколько документов соответствия, если такая способность не была каким-то образом запрограммирована для сервера.

Всё вышесказанное влечёт необходимость изучения языков программирования, с помощью которых было бы возможно писать программы, работающие на сервере.

Программы связываются с сервером по протоколу CGI.

CGI - Common Gateway Interface (Общий Шлюзовой Интерфейс) является стандартом интерфейса (связи) внешней прикладной программы с информационным сервером типа HTTP, Web сервер. Обычно гипертекстовые документы, извлекаемые из WWW серверов, содержат статические данные. С помощью CGI можно создавать CGI-программы, называемые шлюзами, которые во взаимодействии с такими прикладными системами, как система управления базой данных, электронная таблица, деловая графика и др., смогут выдать на экран пользователя динамическую информацию. Программа-шлюз запускается WWW сервером в реальном масштабе времени. WWW сервер обеспечивает передачу запроса пользователя шлюзу, а она в свою очередь, используя средства прикладной системы, возвращает результат обработки запроса на экран пользователя.

CGI имеет несколько преимуществ перед более новыми технологиями.

- Он универсален и переносим. Можно написать приложение CGI, используя практически любой язык программирования на любой платформе. Некоторые из альтернатив, такие как API сервера, ограничивают некоторыми языками и которые очень сложны для изучения.

- Маловероятно что, клиентские технологии типа Java, заменят CGI, потому что имеются некоторые приложения, для которых серверные приложения намного лучше подходят для выполнения.

- Многие из ограничений CGI - это ограничения HTML или HTTP. По мере развития стандартов Интернета в целом, развиваются и возможности CGI.

Преимущества CGI-скриптов перед JavaScript и Java.

- Так как программа выполняется сервером, нет никакого значения, какой у посетителя браузер - древний Lynx или новейший Internet Explorer. Нет никаких ошибок и сообщений о них.

- Сgi-скрипты позволяют реализовать гораздо более широкий набор функций.

- Код сgi-скрипта закрыт для конкурентов.

Как работает CGI-скрипт?

Начнем с того, что браузер (после того, как был набран URL) соединяется по протоколу HTTP с указанным сервером и просит у него нужный файл, примерно так:

```
GET /~raaa/cgi-bin/guestbbok.cgi HTTP/1.0
```

Вот это самое главное в запросе. Далее идет посылаемая браузером информация о себе и о том, что более подробно ему надо. (Например Ассерт: */*). Ну и если запрошен простой файл, например .html, то если такой файл есть, то сервер отошлет браузеру ответ:

```
HTTP/1.0 200 Okay
Content-Type: text/html
```

```
<HTML>
<BODY>
.....
</BODY></HTML>
```

В ответе, состоящем из заголовка и тела, в заголовке содержится код возврата и информация о типе содержимого. Далее после пустой строки (она нужна чтоб отделить заголовок от тела) идет информация из самого документа, по заданному URL <HTML><BODY>. Вот в принципе и весь WWW. А что если нужно внести в этот процесс что-нибудь по настоящему интерактивное, динамическое? Просто нужно в запрашиваемом URL указать специальную программу (CGI) и то, что эта программа выдаст то и отправить браузеру. Сервер запускает .cgi программу и она, например, обработав данные формы, заносит пользователя куда-нибудь в свою базу данных.

Программа-шлюз может быть закодирована на языках C/C++, Fortran, Perl, TCL, Unix Schell, Visual Basic, Apple Script и на любой другой системе программирования, поддерживающей написание сgi-приложений.

Но в большинстве случаев CGI скрипты пишутся на языке Perl (Practical Extraction and Report Language — практический язык извлечений и отчетов) — очень удобном языке, впитавшем из других все лучшие черты. Для Perl родной платформой является Unix, поэтому пользователям PC он мало известен. Он является интерпретируемым, то есть программы на Perl не требуют компиляции. Это очень удобно: пользователь отлаживает программу на своей домашней машине (где, к примеру, установлена Windows), а затем просто копирует ее на сервер (где, скорее всего, установлен один из клонов Unix - FreeBSD, Linux, Sun Solaris, HP-UX и т.п.). При смене провайдера проблем с переносом скриптов также не будет.

Если же пользователь решит использовать компилятор, например, C/C++, то ему нужно будет компилировать исходный текст программы на сервере, для чего необходим telnet-доступ к серверу, который провайдеры давать очень не любят.

Perl применяется также для обработки потоков информации. Изначально он был создан для ОС Unix, но впоследствии, его стали переносить и на другие платформы: MS Windows, MS-Dos, OS/2, MacOS и прочие.

История Perl.

Отцом языка Perl является Ларри Вол (Larry Wall). Он разработан в 1986 году для создания отчетов о содержании многочисленных текстовых файлов в среде UNIX. Поскольку существующие средства не подходили для решения такой задачи, Вол (Wall) изобрел новое средство для ее решения. Название Perl означает практический язык для извлечения и составления отчетов. Вол продолжал добавлять различные возможности к Perl и сделал его доступным для общего пользования. Популярность Perl возрастает до сих пор, и он стал излюбленным средством многих программных пакетов. Умение программировать на Perl имеет возрастающую важность при составлении резюме.

Широкий спектр возможностей.

Perl предоставляет разработчику широкий спектр возможностей для создания кратких и эффективных программ. Ниже приведены некоторые наиболее замечательные черты Perl:

1. Ассоциативные массивы, которые индексируются программами с использованием нецелых ключей.
2. Автоматическое преобразование типов между целыми числами, числами с плавающей точкой и строками.
3. Автоматическое преобразование размера массивов.
4. Функции для преобразования бинарных данных.
5. Широкая поддержка так называемых регулярных выражений, которые программы используют для поиска, замены и других операций, связанных с разбором текста.
6. Функции вывода/ввода файлов.
7. Функции форматированного вывода, наподобие функции C, с добавлением к ним способности генерации отчетов на основе шаблонов (template).
8. Полный набор операторов C, с добавлением также операций по сравнению строк.
9. Функции для обработки списков, которые поддерживают стеки, очереди и другие данные списочных типов.
10. Функции системного сервиса.
11. Богатый набор операторов и структур управления, включая подпрограммы

Использование PERL как фильтра данных.

Средства UNIX-систем в значительной мере основаны на концепции о том, что программа служит фильтром данных, поток которых поступает на

вход, а отфильтрованные данные - на выход. Таким образом, программа-фильтр отбрасывает нежелательные данные. MS DOS также поддерживает концепцию фильтра данных. Утилита `grep` в среде UNIX представляет собой классический пример фильтра данных. Эта утилита сканирует входной поток данных в поиске строк, которые соответствуют заданному шаблону. Программа направляет на выход те строки, которые соответствуют этому шаблону. Она отфильтровывает, отбрасывает строки, не удовлетворяющие шаблону. Perl идеально подходит для построения фильтров данных. Действительно, вы можете создать простую версию утилиты `grep`, используя следующий короткий скрипт на языке Perl:

```
$pattern = shift(@ARGV); # берет командную строку
while (<>)
    { # читать строку из входного потока
    print if (/$pattern/); # output line if it
matches
    }
```

В этом случае скрипт просто просматривает входные данные строка за строкой. Если строка содержит текст, определенный первым аргументом командной строки, то программа печатает такую строку.

Использование языка PERL как шлюза безопасности.

Вопросам безопасности придается большое значение как в программировании CGI, так и при решении других задач, связанных с программным обеспечением для сетей. Часто бывает необходимо защитить файлы и другие ресурсы системы от угрозы со стороны пользователей, которая может проистекать от их неаккуратности или даже злого умысла. Это особенно важно для Web-серверов, а также других серверов, таких как FTP-сервер, которые подключены к Internet, где наличие злоумышленных пользователей хорошо известно. Одним из путей защиты системы от атак таких злоумышленников служит создание шлюзов безопасности, через которые пропускаются входные данные. В этом случае только данные, оцененные шлюзовой программой как безопасные, передаются в систему. Традиционно большая часть серверов Internet исполняется под системой UNIX и написана на языке C. Неверное использование программистами указателей на языке C ведет к нарушению безопасности работы системы. Одним из достоинств Perl, существенным при написании надежных шлюзовых программ, служит тот факт, что переменные строкового типа изменяют свою длину автоматически в зависимости от потребности. Строковые переменные содержат столько символов, сколько им присваивает скрипт. Perl не позволяет программе писать данные в одну переменную таким образом, чтобы нарушить целостность данных другой переменной. Имеется также специальная версия языка Perl, которую называют `taintperl`, в которой осуществляется проверка зависимостей между данными и предотвращается выполнение системой команд по передаче данных серверу от источника, не заслуживающего доверия. Если испорченные или некорректные данные поступают на вход программы `taintperl`, то она

помечает все значения командной строки, переменных окружения и входные данные как испорченные, предотвращая их дальнейшую передачу серверу и возникновение фатальной ошибки.

Все сказанное выше обуславливает необходимость изучения, по крайней мере основ языка программирования, тем более что это направление ещё только развивается в России, и в будущем эти знания обязательно будут востребованы. В настоящее время ведется работа по созданию курса лабораторных работ, который даст школьникам и студентам необходимые знания, а так же навыки работы на этом языке. Предполагается, что данный курс будет содержать 16 работ, охватывающий все основные аспекты, а именно: синтаксис языка, переменные, операторы и приоритеты, встроенные функции, подпрограммы, модули.

Немаловажным фактором актуальности исследования является ещё и то, что литературы по данной тематике на русском языке очень мало.