***Характеристика языка PHP***

**История PHP**

Когда в конце 90-х годов прошлого столетия мы были свидетелями взрывообразного развития Интернета, возникал вопрос: так на чем же писать WеЬ-приложения? В то время не быmо удобных языков программирования и впечатляющих фреймворков. На выбор быmо не так много альтернатив: Java, С и Perl. С# и платформа ASP.NET появятся много позже, не говоря уже о современных фреймворках, таких как Ruby on Rails и Django соответственно на языках Ruby, и Python. Последние появились, когда на РНР уже создавались огромные социальные сети вроде Facebook или "ВКонтакте".

В те далекие времена Интернет еще не был повседневным инструментом. В него не вкладывались огромные средства, правила поведения владельцев сайтов и посетителей не регулировались государственными законами, а сама Сеть не была доступна большей части населения страны. Сеть больше напоминала клуб энтузиастов с доходами выше среднего. В глазах большинства Интернет был скорее дорогой игрушкой, для самих энтузиастов он был зоной постоянных экспериментов. В таких условиях приложения редко жили больше полугода, а редизайны сайтов проводились чуrь ли не ежегодно. Разумеется, не было и многомилионных бюджетов на разработку и команд в несколько десятков человек.

Приложения нужно было писать быстро, иначе конкурировать в такой агрессивной среде было почти невозможно. Писать СGl-скрипты на С было практически неосуществимой задачей: слишком низкоуровневый язык, требующий значительных трудозатрат и предельной внимательности. В быстро меняющемся мире Интернета было, есть и будет о чем подумать, кроме выделения, возвращения памяти, нулевых указателях и переполнения буфера.

Поэтому первые СGl-сценарии были написаны на Perl, наиболее распространенном на тот момент скриптовом языке. Perl известен своими регулярными выражениями, которые оказались настолько удачными, что используются во множестве других языках.

Создатель языка Perl Ларри Уолл, лингвист по образованию, построил вокруг регуляр­ных выражений довольно странный с точки зрения программиста язык, основная идея которого заключалась в контекстном выполнении операций. Эксперимент оказался удачным, и язык стал событием в мире программирования. Множество энтузиастов и поклонников языка экспериментировали с Perl. Новый язык на долгие годы стал основ­ным скриптовым языком администрирования серверов. К моменту становления Интер­нета Perl был широко распространен среди сообщества разработчиков. Неудивительно, что первые Wеb-приложения создавались на Perl - на то время это был самый быстрый· способ, что было важно в отсутствии больших команд и сжатых сроках разработки.

В корпоративном секторе с большими бюджетами и командами заправлял язык Java, который представлял хорошую альтернативу С, т. к. обладал сборщиком мусора и автоматически решал проблему с выделением памяти. Однако его особенностью было значительное потребление оперативной памяти и довольно высокий вход для новичков - являясь даже не языком, а платформой, выполняющейся в любой операционной системе, а иногда даже без таковой, он обладает внушительным набором инструментом. Политика разработки языка заключается в сохранении полной обратной совместимости с ранними версиями. В результате язык тащит за собой огромный воз возможностей, удачных и неудачных решений. Корпорации могли себе позволить оплачивать дорогие серверы и внушительные команды, поэтому большие интернет-магазины, банковское программное обеспечение или внутрикорпоративные проекты часто разрабатывались и по сей день разрабатываются с использованием Java. Разумеется, на начальных этапах развития Интернет, когда им была охвачена лишь небольшая доля населения страны, как дорогостоящий инструмент разработки не мог получить широкое распространение.

Поэтому для энтузиастов, строящих Web, альтернатив было немного: "Конечно, Perl!". Однако Perl не приспособлен непосредственно для программирования WеЬ-приложений. Это в некотором роде универсальный язык, поэтому он не поддерживает напрямую того, чего бы нам хотелось, например - автоматическую отправку НТTР­заголовков или удобное управление cookies. Вторым недостатком Perl является его синтаксис, который плохо подходит для командной работы - слишком много равно­значных реализаций одних и тех же решений, в результате сообщество разделяется на слабовзаимодействующие субкультуры, плохо понимающие код не из своей среды.

Поэтому, когда в 1998 году появился РНР - язык, специально созданный для работы в Интернете, он получил широчайшее распространение. Сам проект начался ранее, в 1994 году. Первые две версии представляли собой варианты, написанные на Perl датским программистом Расмусом Лердорфом. Язык имел мало общего с современным РНР и больше походил на то, что сейчас принято называть языками шаблонов (вроде Smarty или HAML).

Третья версия языка, разработанная на С двумя израильскими программистами Энди Гутмансом и Зеевом Сураски, была построена на модульной основе, позволяющей сторонним командам создавать свои расширения, снабжающими язык новыми возможностями по взаимодействию с базами данных, обработке изображений, реализации алгоритмов шифрования.

Практически вслед за третьей версией в 1999 году была выпущена четвертая версия языка на движке, названном Zend Engine, обновленные версии которого используются до настоящего времени, определяя все достоинства и недостатки языка.

В 2005 году выходит следующая версия- РНР 5, построенная на движке Zend Engine 2. Ее главной особенностью стало введение новой концепции объектно­ ориентированного программирования.

Так как язык получил огромное распространение, а в его разработке участвовало множество энтузиастов и команд, преследующих самые разные цели, к этому времени было накоплено множество проблем с безопасностью, с синтаксисом и поддержкой многобайтовьrх кодировок. Поэтому следующей целью была назначена версия РНР 6, которая должна была решить как минимум проблему с поддержкой UTF-8 на уровне ядра.

Однако эта цель никогда не была достигнута, версия РНР 6, разрабатываемая с 2005 по 2010 год, так никогда не была выпущена. Изменения в интерпретаторе, подготовленные в рамках этой работы, были слиты с основной веткой 5.х и выпущены в виде релиза РНР 5.3, который содержал нехарактерное количество изменений для минорной версии. Все они, включая пространства имен, лямбда-функции, замыкания.

По мере роста популярности языка РНР росло количество компаний, которые использовали язык в качестве основного инструмента разработки. Всемирная энциклопедия Wikipedia также использует в качестве основного серверного языка РНР. Ряд компаний и их продукты получили мировое признание, в частности на РНР разработана крупнейшая в мире социальная сеть Facebook.

РНР, создававшийся в расчете на средние сайты, не выходящие за пределы одного сервера, испытывал серьезные проблемы с производительностью, потреблением памяти в таких гигантских Wеb-приложениях. Особенно остро это проявлялось на фоне языков­конкурентов. В эти годы на рынок последовательно выходили новые инструменты Wеb-разработки, такие как С# и платформа .NET корпорации Microsoft, сверхудобные фреймворки Ruby on Rails и Django, сверхбыстрый серверный вариант JavaScript - Node.js, компилируемый язык Go от компании Google. Выбрать было из чего, и там, где раньше безраздельно властвовал РНР, новые проекты все чаще и чаще начали создаваться с использованием других инструментов и языков, оттягивая интерес разработчиков от РНР.

С другой стороны произошел разворот в области Wеb-проектирования. Вместо небольших сайтов, разработка которых ведется несколько месяцев и которые затем ютятся по 500 штук на сервере в рамках виртуального хостинга, проекты стали укрупняться, и сайты в настоящий момент разрабатываются по году и больше, размещаются на нескольких десятках, а иногда и сотнях серверов. РНР проигрывал альтернативным технологиям и по производительности, и по потреблению памяти.

Разумеется, такая огромная компания, как Facebook, не могла мириться с положением дел, когда основной бизнес, приносящий миллиарды долларов, основан на языке, не совсем предназначенном для столь масштабных проектов, как социальная сеть. При том, что сообщество, ответственное за язык, несколько стагнирует. Замена языка, вероятно, не представляет труда для небольшого сайта, но не для огромной империи, которая обслуживается множеством дата-центров, в интеллектуальную собственность которой вложено множество человеко-лет.

Для того чтобы оценить масштаб трудностей, с которыми столкнулся Facebook, следует заметить, что РНР до недавнего времени оставался интерпретируемым языком, несмотря на то, что его код первоначально транслируется в байт-код и затем выполняется, он все равно проигрывает компилируемым языкам в скорости выполнения. РНР-программа может уступать в скорости в 1000 раз (три порядка) программам, разработанным на С или С++. Это не так важно, когда арендуется небольшая часть сервера для обслуживания нескольких сотен посетителей в сутки. Когда для обслуживания миллионов посетителей требуются сотни тысяч серверов, три порядка означает, что их могло быть пусть не в 1000, но все же как минимум в десятки, если не в сотню раз меньше.

Поэтому, захватив рынок, Facebook попытался увеличить эффективность работы собственных серверов и команды разработчиков. В связи с этим было предпринято несколько шагов.

В первую очередь в 2010 году был начат проект компилятора РНР-tо-С++, переводящего РНР-код в С++, названный НРНРс. Добившись значительного снижения нагрузки на собственные серверы, Facebook столкнулся с проблемой длительного цикла разработки приложения: код сначала разрабатывается на РНР, потом переводится на С++, далее компилируется, затем разворачивается на серверах, и все это требует времени.

Поэтому в недрах Facebook стартовал новый проект виртуальной машины, которая бы не тратила время на компиляцию, а сразу выполняла байт-код РНР с высокой эффективностью. Новый проект получил название HipHop Virtual Machine или сокращенно ННVМ. Первая реализация была представлена в 2011 году. ННVМ, подобно Java, конвертирует РНР-код в промежуточный байт-код, который компилируется и оптимизируется JIT компилятором в машинный код. ННVМ реализует множество низкоуровневых оптимизаций, не доступных на уровне НРНРс. Причем виртуальная машина HHVM постоянно совершенствуется. В ноябре 2012 года производительность ННVМ превзошла НРНРс. Несмотря на сложность реализации, ННVМ ведет себя точно так же, как Zеnd-реализация языка и так же доступна для любого желающего.

По следам Facebook последовала и социальная сеть "ВКонтакте", хорошо известная российским пользователям Интернета. В 2014 году разработчики этой социальной сети представили транслятор РНР-кода в С++, который назвали КittenPНP или сокращенно КРНР.

Если до этого момента Zеnd-вариант РНР представлял собой канон и стандарт языка, то с появлением мощных коммерческих игроков с сильными командами, ресурсами, а главное мотивацией улучшать эффективность РНР, перед сообществом встала проблема неполной совместимости различных реализаций РНР.

Таким образом, сложились как минимум две значимые и популярные ветви языка РНР: Zend и ННVМ. В июне 2014 года Facebook опубликовал проект спецификации языка РНР, который доступен на GitHub по адресу https://github.com/php/php-langspec/tree/master/spec. За основу взят Zеnd-вариант реализации языка. Возможно, в будущем спецификация дойдет до полноценного стандарта, который может быть взят за основу любым независимым разработчиком, пожелавшим создать собственную реализацию языка РНР.

Следующим этапом в погоне Facebook за эффективностью стала разработка нового диалекта РНР - Hack. Язык добавляет новые структуры и интерфейсы, а также статическую типизацию. Собственно Hack - это другой язык, который решает проблемы большой корпорации с большим количеством серверов.

Успехи социальных сетей в области создания альтернативных реализаций РНР побудило сообщество к выпуску новой, более эффективной версии РНР 7 на новом движке Zend Engine 3, релиз которой состоялся в декабре 2015 года. Титанические усилия команды разработчиков позволили ускорить движок РНР 7 в два раза, на фоне снижения потребления оперативной памяти. В версии РНР 7 значительно улучшена поддержка 64-битных операционных систем.

**Что нового в PHP 7**

Среди новинок PHP 7 следует отметить:

* новые операторы<=> и ??;
* обработка ошибок реализована через механизм исключений, помимо существующе­го класса Exception, введен дополнительный класс Error, который позволяет пере­хватывать стандартные ошибки РНР;
* анонимные классы;
* поддержка скалярных типов аргументов и возвращаемых значений в функциях;
* введен специальный синтаксис \u{00FFI для поддержки UТF-8 символов;
* массивы-константы;
* допускается использование конструкции list совместно с объектами, реализующими интерфейс ArrayAccess;
* предопределенный класс Intlchar для обеспечения подцержки символов UTF-8;
* класс Generator, представляющий генераторы РНР, получил метод getReturn (), позволяющий извлечь значение, возвращаемое ключевым словом return;
* делегирование генераторов с использованием ключевых слов yield from;
* регулярные Реrl-выражения дополнены новой функцией preg\_replace\_callback\_ array ();
* новые функции random\_int ()и random\_bytes ();
* параметры сессии теперь можно передавать через необязательный аргумент функции session \_ start {}.

Помимо нововведений, часть устаревших конструкций и расширений были окончательно удалены из РНР. Среди них можно отметить:

* альтернативные теrи <% и %>, а также <script language="php"> и </script>, которые можно было использовать вместо тегов <?php и ?>;
* регулярные РОSIХ-выражения (еrеg-функции), вместо них рекомендуется использовать регулярные Реrl-выражения;
* расширение mysql, вместо него следует использовать либо объектно-ориентированное расширение mysqli, а лучше еще более новое расширение РОО;
* не допускается использование конструктора в стиле РНР 4, когда имя метода совпадает с именем класса.

Контрольные вопросы:

1. Кто создатель языка PHP и в каком году был создан его прототип?
2. Какой язык взять за основу PHP?
3. Какая компания повлияла на развитие языка PHP?
4. Как называется современная версия движка?