

Скриптовые языки программирования

Доклад на семинаре по специальности

Студент гр.4057/2 Руцкий Владимир
20.10.2009

Содержание

- Введение
- Особенности скриптовых языков программирования
- Типы скриптов и примеры
- Заключение

Введение

Иерархия языков программирования

- Машинные коды, язык ассемблера. Работа с отдельными ячейками памяти, регистрами
- Компилируемые в машинные коды (C, C++). Работа с примитивными типами данных (числа, массивы, структуры/классы)
- Компилируемые в байт-код или интерпретируемые (Java, C#, Python). Более сложные типы данных и более сложные элементарные операции с ними
- Работающие со сложными типами данных (с таблицами, программами — SQL, shell)

Сложность элементарных операций

<здесь будет приведена диаграмма количества машинных команд необходимых для выполнения каждой элементарной операции в соответствующем классе языка программирования с предыдущего слайда>

Сложность написания программ

<здесь будет приведена статистика, сколько строк кода (и сколько времени) занимает решение какой-то определённой задачи, в различных классах языков программирования с предыдущего слайда>

Эволюция языков программирования

Чем выше уровень языка, тем

- Проще и быстрее разработка программы
- Медленнее работа программы

Вычислительная техника развивается столь стремительно, что применение сильно высокоуровневых языков программирования является оправданным во многих задачах.

Появление скриптов

- 1960-е годы. Shell-скрипты. Автоматизация работы человека-оператора по вводе команд на системах с разделением времени
- «Скриптовый язык», «язык сценариев» - язык записи «сценария», последовательности выполнения команд

Особенности скриптовых языков программирования

Назначение скриптовых языков программирования

Быстрое и простое связывание и управление готовыми объектами (функциями, программами)

- Для создания новых программ на основе существующих
- Для автоматизации различных рутинных операций
- Для быстрой разработки технологических тестов
- Для задания сценариев работы программы не программистами

Типы данных

- Бестиповые — более абстрактный и универсальный код
- Универсальные типы — возможность произвольного связывания различных компонент

Недостаток:

- Нетипизированность данных не позволяет выявить ошибочное использование переменных до начала выполнения скрипта

Особенности среды выполнения скриптов

Выполнение скриптов:

- Практически никогда не компилируются в машинные коды
- Компилируются в байт-код (обычно лишь для оптимизации скорости выполнения, кешируются)
- Чаще всего интерпретируются «на лету»

В виду машиннонезависимости часто являются кроссплатформенными

Особенности синтаксиса

- Чаще всего позволяет «построчное» выполнение кода — даёт возможность программирования «на лету», прямо во время выполнения программы
- Минимализм в конструкциях языка

Типы скриптов и примеры

Управление последовательностью команд ОС

- Shell — «оболочка» — интерфейс между функциями ядра/системы и конечным пользователем
 - GUI — graphical user interface
 - CLI — command-line interface
- Первые оболочки ОС — текстовые (CLI)

История Shell

- CTSS OS (Compatible Time-Sharing System) — одна из первых ОС с разделением времени (1961, MIT Computation Center)
- Louis Pouzin (родился в 1931 во Франции) участвовал в проектировании CTSS, разработал программу RUNCOM (1963/64), позволяющую выполнять команды в текущей директории. Первым ввёл термин «shell»

Развитие Shell

<TODO: пока не знаю точно о чем это>

Четыре поколения shell:

1. Thomson shell, Mashey shell
2. C-shell, Bourne shell
3. tcsh, ksh88
4. ksh93, bash, zsh, Microsoft Power Shell

ОС Unix shell: sh

<TODO: Изменить заголовок слайда, мне не удаётся сделать это в OpenOffice сейчас :)>

- **Thompson shell** — первая оболочка для Unix, разработал Ken Thompson в AT&T Bell Laboratories в 1971
- **sh** — *Bourne shell*, разработал Stephen Bourne в AT&T Bell Laboratories и был выпущен в 1977 как оболочка по умолчанию для Version 7 Unix. Заменял Thompson shell
- **bash** — *Bourne-again shell*, разработал в 1987 Brian Fox в рамках проекта GNU.
- bash — обратно совместимое надмножество над sh
- bash до сих пор продолжает развиваться и широко используется в качестве shell по умолчанию в различных Unix-like ОС

Пример работы с Unix-системой через CLI - bash (1/2)

```
bob@bob:~$ slogin balto -l bob
bob@balto's password:
Linux balto 2.6.24.2 #5 SMP Sat Jul 26 18:43:21 MSD 2008 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Fri Oct 16 03:14:01 2009 from bob
bob@balto:~$ ls
backup Desktop stuff
bob@balto:~$ ls -l
total 12
drwx----- 2 bob bob 4096 2008-08-22 01:50 backup
drwx----- 2 bob bob 4096 2008-12-23 21:51 Desktop
drwxr-xr-x 6 bob bob 4096 2009-10-16 03:12 stuff
bob@balto:~$ cat > hello.cpp
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}
```

Пример работы с Unix-системой через CLI - bash (2/2)

```
bob@balto:~$ g++ -Wall -o hello hello.cpp
bob@balto:~$ ls -l
total 28
drwx----- 2 bob bob 4096 2008-08-22 01:50 backup
drwx----- 2 bob bob 4096 2008-12-23 21:51 Desktop
-rwxr-xr-x 1 bob bob 10304 2009-10-16 03:16 hello
-rw-r--r-- 1 bob bob 71 2009-10-16 03:15 hello.cpp
drwxr-xr-x 6 bob bob 4096 2009-10-16 03:12 stuff
bob@balto:~$ ./hello
Hello, world!
bob@balto:~$ cat hello.cpp
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}

bob@balto:~$ rm hello
bob@balto:~$ ls -l
total 16
drwx----- 2 bob bob 4096 2008-08-22 01:50 backup
drwx----- 2 bob bob 4096 2008-12-23 21:51 Desktop
-rw-r--r-- 1 bob bob 71 2009-10-16 03:15 hello.cpp
drwxr-xr-x 6 bob bob 4096 2009-10-16 03:12 stuff
bob@balto:~$ logout
Connection to balto closed.
bob@bob:~$
```

Возможности bash

- Сейчас bash — полноценный язык программирования, по историческим причинам ориентированный на задание последовательности выполнения команд
- «Стандартной библиотекой» для bash является набор стандартных утилит Unix.

Возможности bash, как языка программирования

<переменные, конструкции if-then-else, for, while, switch, возможность задания функций, массивы, перенаправление ввода/вывода. Примеры>

Возможности bash в связке со стандартным набором программ Unix

<find, grep и некоторые примеры>

ОС MS Windows shell: cmd.exe

- **COMMAND.COM** — shell по умолчанию для ОС DOS, и CLI ОС MS Windows 9x/Me
<TODO: где и когда был разработан впервые>
- **CMD.EXE** — shell для ОС MS Windows выше Windows 2000, разработал Therese Stowell для MS Windows NT

Возможности cmd.exe

<TODO>

<+ Примеры>

Другие оболочки ОС

- <Другие Unix shells>
- <MS Power Shell>

Итог: современное использование оболочек ОС

<TODO>

Языки для обработки текстов

- Текст — цепочка символов — универсальный тип данных
- Алгоритмы, принимающие цепочку символов и возвращающие цепочку, можно связывать друг с другом в произвольном порядке
- Shell превосходно подходит для обработки цепочек символов: вход и выход программ — цепочки символов и shell имеет встроенную возможность для перенаправления вывода одной команды на вход другой

sed

- <История создания, возможности, примеры>

awk

- <История создания, возможности, примеры>

Web-скрипты

- Одно из современных использований Интернета — передача информации
- Чаще всего обмен информацией организован между парами клиент-сервер
- Генерация данных на сервере, подготовка к передаче клиенту, отображение данных на стороне клиента, приём данных от клиента — рутинные операции, общие для большинства схем взаимодействия клиент-сервер.

Серверные Web-скрипты

- <PHP, ASP, Ruby on rails, SMX>

Web-скрипты. Клиентские

- <JavaScript, VBScript>

Языки общего назначения

- <Perl, Python, Tcl, Ruby, Lua, Basic>

Задание графического пользовательского интерфейса

- `<Tk (Tcl/Tk)>`

Узконаправленные языки

- <MAXScript (3DSMax), MATLAB, R, Bison, консоль в некоторых играх (Quake, Half-Life), макросы MS Word/Excel, OpenOffice.>

Сравнение скриптовых языков программирования

<здесь будет сводная таблица характеристик
различных скриптовых языков
программирования, как Вы предложили>

Заключение

Скриптовые языки — набор удобных и надёжных инструментов для быстрого и простого решения широкого спектра задач

Источники информации

<будет заполнено позже>

Благодарю за внимание!