Реализация общей поддержки среды исполнения для интерпретатора языка PostScript

Дмитрий Поздин

444 группа

29 мая 2014 г.

Научный руководитель: Д.Ю. Булычев

Предметная область

Язык PostScript :

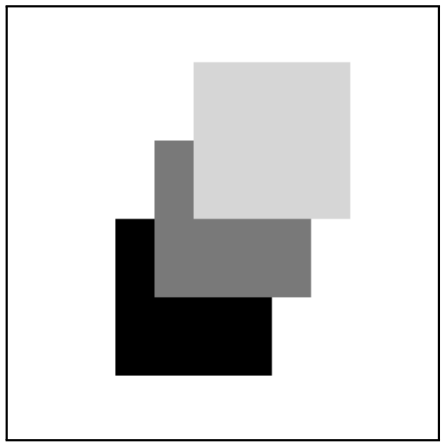
- интерпретируемый
- стековый
- поддержка векторной графики
- управление принтерами

Особенности PostScript

- Постфиксная запись операторов
- Последовательная обработка программы
- Стеки операндов, словарей, графический и исполнения
- Нет зарезервированных слов – всё находится в словарях
- Много операторов, работающих с графикой и шрифтами

Пример программы на языке PostScript

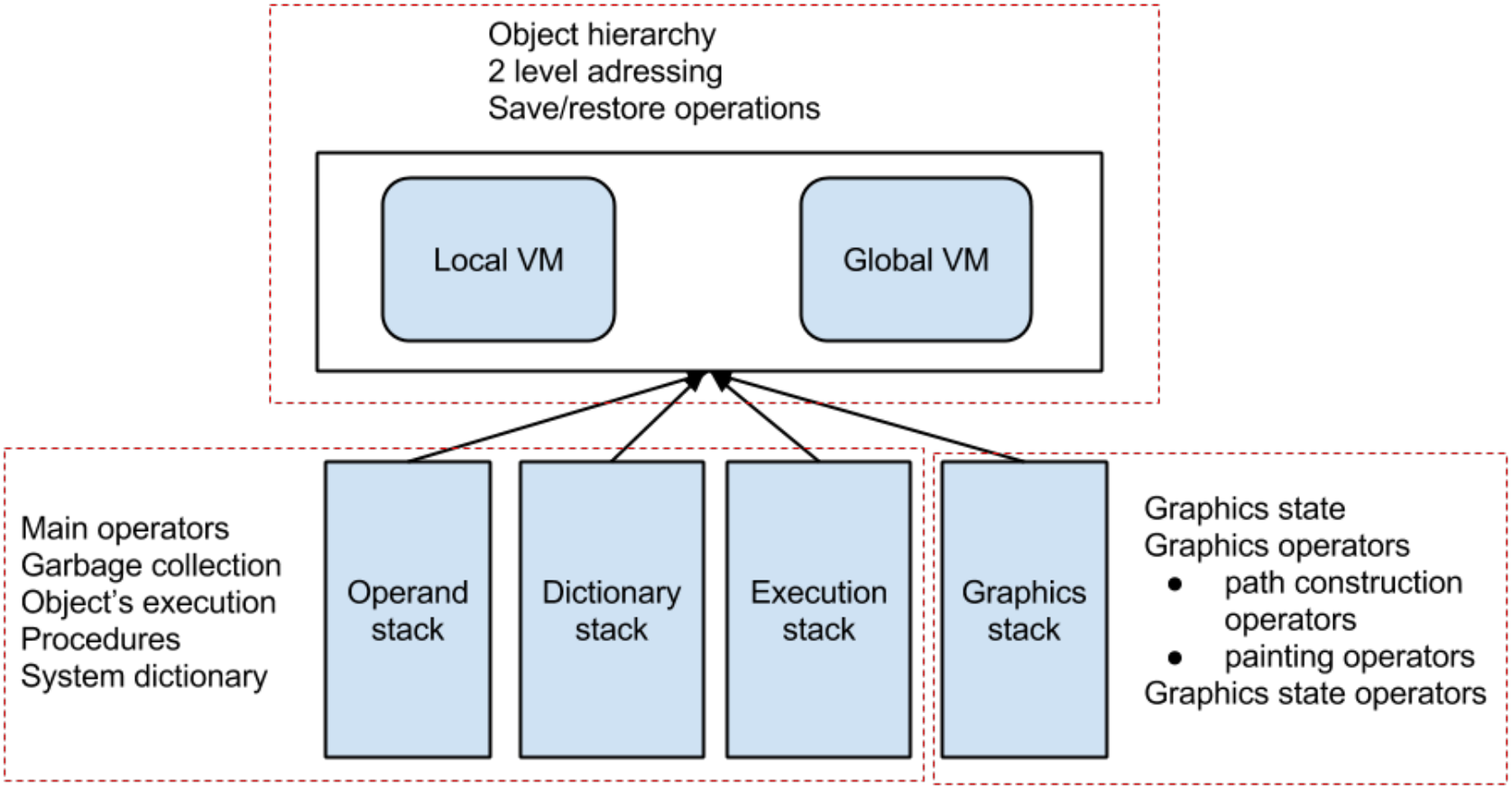
```
/box { 72 0 rlineto 0 72 rlineto -72 0 rlineto closepath} def
252 324 moveto box 0 setgray fill
270 360 moveto box 0.4 setgray fill
288 396 moveto box .8 setgray fill
showpage
```



Постановка задачи

- Реализация общей поддержки времени исполнения интерпретатора языка PostScript
- Предполагаемые результаты:
  - реализация стеков
  - реализация основных структур данных
  - реализация механизма исполнения
  - реализация основных операторов
  - реализация сборки мусора

Общая схема интерпретатора



Структуры данных

Словари:

- Ключи словаря - любые объекты
- Вместо hashCode реализовано compareTo и equals
- АВЛ-дерево
- Неизменяемое
- Переиспользование старого дерева при изменении
- Системный словарь с основными операторами

Массивы:

- Могут разделять значения
- ArrayElement содержит объект, элемент массива

Процедуры

- описывают контекст исполнения
- хранятся на стеке исполнения
- могут быть именованными или безымянными
- вызов:
  - по имени
  - по операторам for, forall, repeat, loop, if, ifelse, exec
- по операторам for, forall, repeat, loop, if, ifelse, exec
- могут быть прочитаны из файла (запуск программы)
- оператор exit для прерывания процедур

Исполнение

Первая реализация:

- рекурсивный вызов через оператор exec
- нельзя было использовать loop и exit и контролировать вложенность исполнения.

Вторая реализация:

- стек исполнения
- исполнение текущего объекта верхней процедуры в стеке
- поиск по имени в словарях
- поддержка вложенности
- вызов встроенных операторов

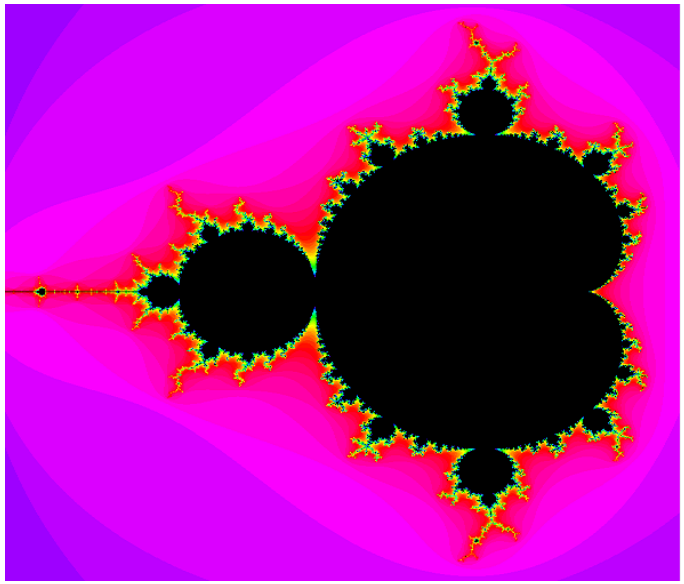
Сборка мусора

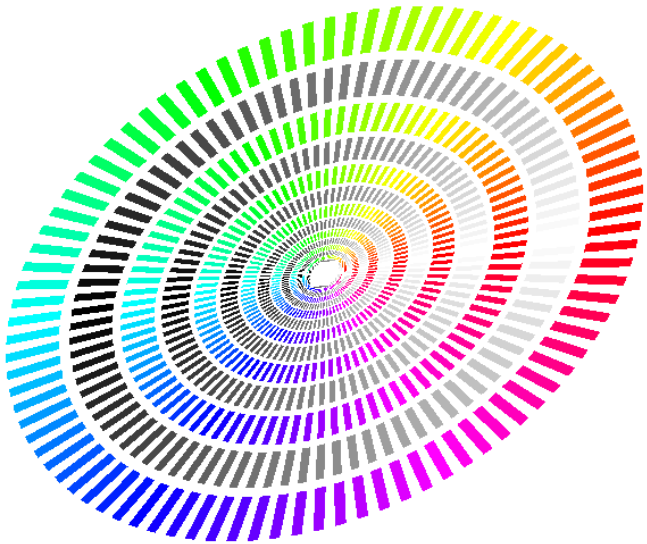
- Оптимизация работы интерпретатора
- Локальная память
- Построение корневого множества
- Поиск в стеках всех доступных объектов
- Удаление неиспользуемых объектов

Операторы

- работа со стеками
- арифметика, отношения, логика
- строки, массивы, словари
- приведение типов, изменение атрибутов
- виртуальная память (save/restore)
- управление (for, if ...)
- раннее связывание (bind)
- графика

Примеры





Результаты

Реализованы:

- стеки операндов, словарей, графических состояний и исполнения
- сборка мусора в локальной памяти
- механизм исполнения объектов языка
- процедуры
- основные структуры (массивы, строки, словари)
- системные словари и большинство встроенных операторов