методическое пособие

Обработка текстовых форматов данных и реализация компьютерных языков на Flex/Bison/C++/LLVM

© <dponyatov@gmail.com>

25 декабря 2015 г.

Оглавление

1	Необходимое программное обеспечение					
	1.1 Структура типового компилятора					
2	Типичная структура проекта					
	2.1 README.md 2.2 Makefile					

Применение

• обработка текстовых форматов данных (файлы САПР, исходные данные для расчетных программ)

- командный интерфейс для устройств на микроконтроллерах (управление человеко-читаемыми командами, передача пакетов данных любой структуры и типов)
- реализация специализированных скриптовых языков
- обработка исходных текстов программ (модификация, трансляция на другие языки программирования, универсальный язык шаблонов для ЯП с ограниченными или отсутствующими макросами)

Необходимое программное обеспечение

• Windows

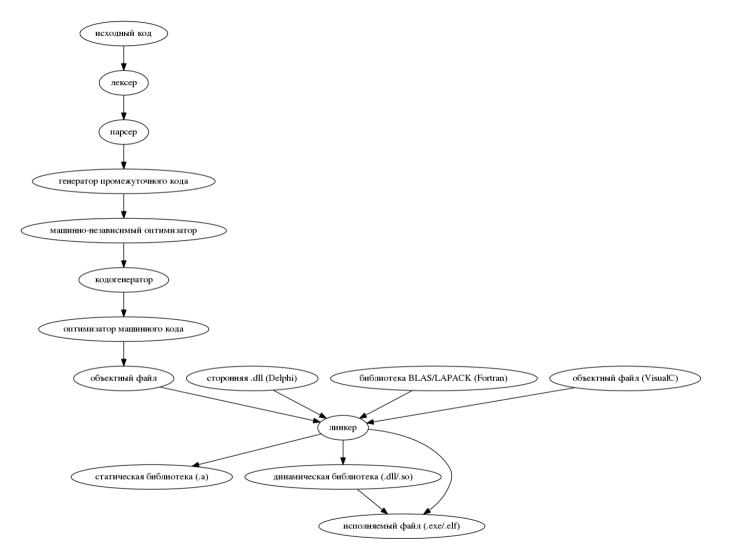
MinGW	http://www.mingw.org/	пакет компилятора и утилит GNU GCC toolchain $(g++, flex, bison, make)$
$\operatorname{git-scm}$	https://git-scm.com/	git-клиент
gvim	http://www.vim.org/download.php#pc	минималистичный редактор кода
		с самой простой подсветкой синтаксиса
		(на регулярках ??)
clang	http://llvm.org/releases/download.html	компилятор C/C++ на базе LLVM
llvm	???	сама библиотека LLVM

• Linux (суперкластер СГАУ "Сергей Королев") g++, flex, bison, make, git, llvm(-3.5), clang

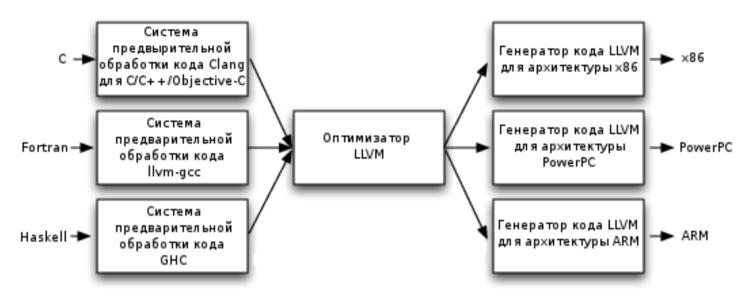
Глава 1

Структура компилятора

1.1 Структура типового компилятора



1.2 Архитектура LLVM



Глава 2

Типичная структура проекта

README.md	2.1	github	описание проекта на https://github.com/
Makefile	2.2	$_{\mathrm{make}}$	зависимости между файлами и команды сборки
lpp.lpp		flex	лексер ??
урр.урр		bison	парсер ??
hpp.hpp		g++/clang++	заголовочные файлы C^{++}
cpp.cpp		g++/clang++	C^{++} -код: ядро интерпретатора, компилятор,
			реализация динамических типов, пользовательский код
bat.bat		win32	запускалка gvim
rc.rc		windres	описание ресурсов: иконки приложения, меню,
logo.ico		windres	логотип в .ico формате
logo.png			логотип в .png (для github README)
$_{ m filetype.vim}$		(g)vim	привязка расширения файлов скриптов
$\operatorname{syntax.vim}$		(g)vim	синтаксическая подсветка для скриптов
.gitignore			список временных и производных файлов

2.1 README.md

```
# <логотип> <название>
(c) <имя> <email>
<лицензия>
<ссылка на проект на GitHub>
### <ссылки, дополнительная информация>
```

README.md

```
1# ![logo](logo.png) Mega script language
2
3 (c) Vasya Pupkin <pupkin@gmail.com>, all rights reserved
4
5 license: http://www.gnu.org/copyleft/lesser.html
6
7 GitHub: https://github.com/pupkin/megascript
```

2.2 Makefile

Опции сборки (win32|linux):

EXE суффикс исполняемого файла RES имя объектного файла ресурсов win32.exe TAIL опция команды **tail** число последних строк **MODULE.log**

Makefile

```
egin{array}{lll} 1 &\# EXE = .exe / \ 2 &\# RES = res.res / \ 3 &\# TAIL = -n17 / -n7 \end{array}
```

1 MODULE = \$ (notdir \$ (CURDIR))

Модуль заполняется автоматически по имени текущего каталога:

```
Makefile
```

```
Цель команды make по умолчанию: сборка и интерпретация тестового файла
```

Makefile

```
1 exec: ./$(MODULE)$(EXE)
2 ./$(MODULE)$(EXE) < $(MODULE).bI > $(MODULE).blog \
3 && tail $(TAIL) $(MODULE).blog
```

Литература

- [1] Книга Дракона (Dragon Book): Альфред Ахо, Моника С. Лам, Рави Сети, Джеффри Ульман Компиляторы: принципы, технологии и инструментарий
- [2] LLVM tutorial Серия статей на Habrahabr.ru
- [3] Компиляция. 1: лексер
- [4] Компиляция. 2: грамматики
- [5] Emden Gansner and Eleftherios Koutsofios and Stephen North **Drawing graphs with dot**