Методология литературного программирования

Опыт русификации инструментальных средств

к. т. н. Жуков Илья Борисович ibzh@yandex.ru

Общие тенденции развития языков

• абстрагирование от аппаратного уровня, в том числе от способа представления данных;

•абстрагирование от операционной системы;

• поддержка средств разбиения программы на отдельные обозримые компоненты, со слабой связью между ними;

• программирование на уровне, соответствующем предметной области и логике решаемой задачи.

Функции литературного программирования

Инструментальные средства литературного программирования (ЛП) позволяют:

- осуществлять разбиение программы на отдельные обозримые части, которые можно иерархически вкладывать друг в друга;
- программировать, используя термины решаемой задачи;
- документировать программы.

Происхождение ЛП

Концепция ЛП предложена Д. Кнутом в конце 70-х начале 80-х.

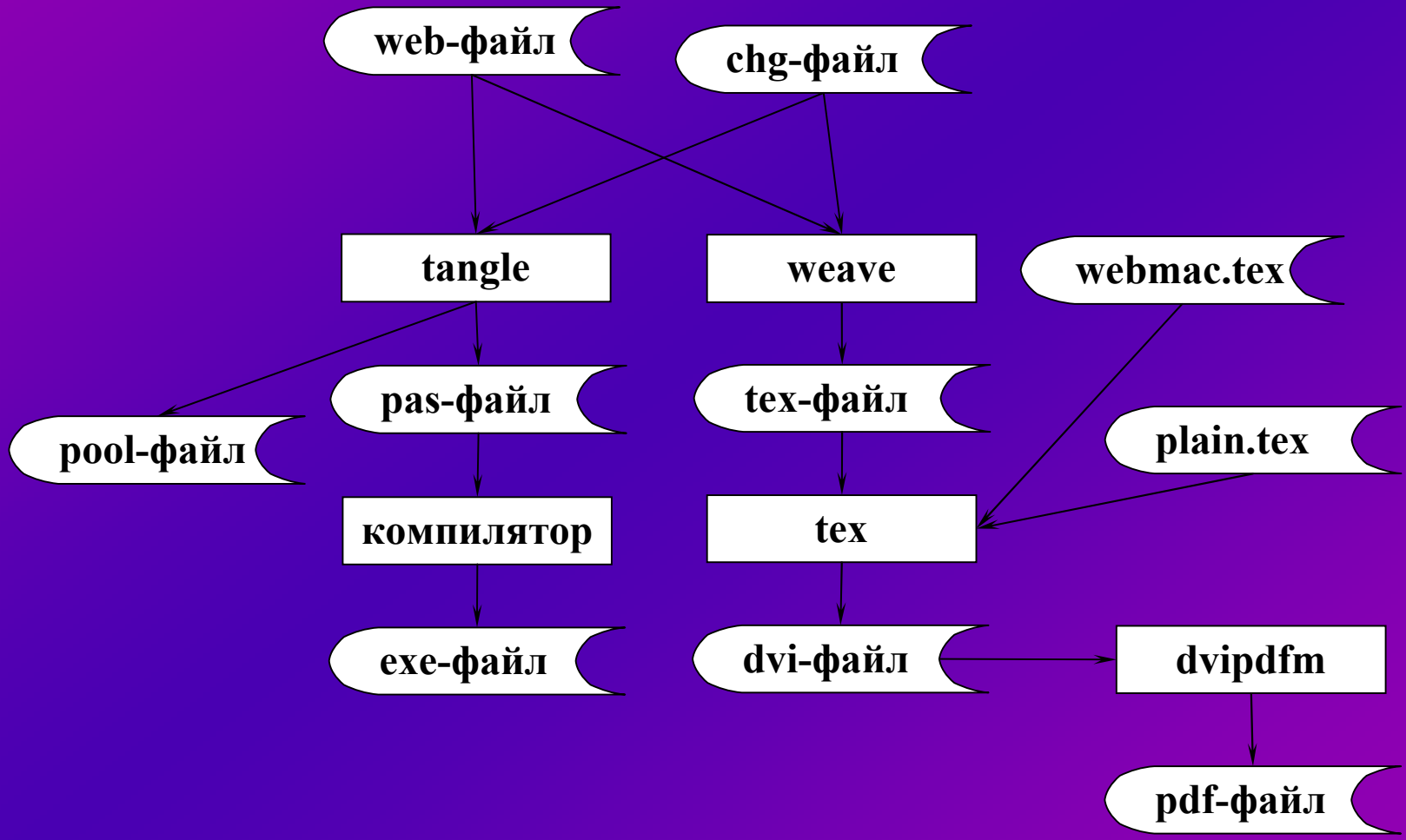
Она использовалась для разработки систем TeX, METAFONT и многих других.

Инструментальные средства ЛП также разработаны с использованием этого подхода.

Дональд Эрвин Кнут

автор концепции литературного программирования

Цикл обработки программы на ЛП



Реализуемые в ЛП подходы

ЛП позволяет осуществлять программирование

- сверху вниз,
- снизу вверх,
- а также постепенно дописывать код отдельных частей по мере его уточнения.

Есть возможность писать на псевдокоде:

if <условие> **then** <Действие 1> **else** <Действие 2>;

Конкретное содержание условия и действий можно описать в других частях программы.

Структура программы ЛП

LIMBO (Команды TeX, несколько строк.)

@* Раздел. Его описание на естественном языке.

@d макрос = = текст макроса

@p
Кусочек программы

@ Описание подраздела.

@<Название кусочка программы@>=
Кусочек программы

Вложенность кусочков программы

@ @<Часть программы 1@>=
    @<Часть 2 @>@;
    @<Часть 3 @>@;

@ @<Часть 2 @>=
    Команды

@ @<Часть 3 @>=
    Команды

@ @<Часть программы 1@>=
    Продолжение описания кусочка

@ @<Часть 3 @>=
    @<Часть 2 @>@;

Основные команды

- Раздел начинается с @* или @
- Сам знак @ вводится, как @@
- Текст |*операторы*| форматируется как кусочек кода

- Макроопределения начинаются с @d или @f

- Безымянные кусочки начинаются с @p
- Именованные кусочки начинаются с

  @< *имя кусочка* @>=

Форматирование фрагмента программы

```
@* Первый раздел программы.
Описание следующего кусочка
программы.

@d incr(#) = =  #:=#+1

@p
program a;
var x, y:integer;
begin
@<Some data input@>;
if x=1 then incr(y);
@<Result output@>;
end.
```

1. Первый раздел программы.
Описание следующего кусочка программы.

```
define incr(#) ≡  # ← # + 1
program a;
  var x, y:integer;
  begin  <Some data input 2>;
       if  x = 1 then incr(y);
       <Result output 4>;
  end.
```

Макрокоманды

- Константы

@d *имя* = *число* {*комментарии*}

- Обычный макрос

@d *имя* == *текст подстановки*

- Макрос с параметром

@d *имя*(#) == *текст подстановки*

- Форматирование задаётся командой

@f *имя* = *имя*

Указатели и перечни

Инструментальные средства ЛП формируют

- указатель идентификаторов;
- перечень кусочков;
- содержание (название разделов со звёздочкой)

Дополнительно в имени кусочка указывается номер раздела, где он начал определяться, а в конце этого раздела указываются номера всех разделов где этот кусочек используется и где определяется.

Команды ссылок

Если перед идентификатором поставить команду @!, то номер раздела в указателе будет подчёркнут.

Эта команда неявно вставляется после ключевых слов **var**, **procedure**, **program**, и некоторых других.

Ссылки в указатель идентификаторов можно вносить следующими командами:

@^ Ссылка печатается прямым шрифтом@>

@. Ссылка печатается моноширинным шрифтом@>

@: Ссылка печатается в формате, заданном \9 @>

Файл изменений

Используется для внесения изменений в программу без изменения основного web-файла. Состоит из последовательности разделов вида

Комментарии

@x *комментарии*

Строки web-файла, которые нужно заменить

@y *комментарии*

Текст замены

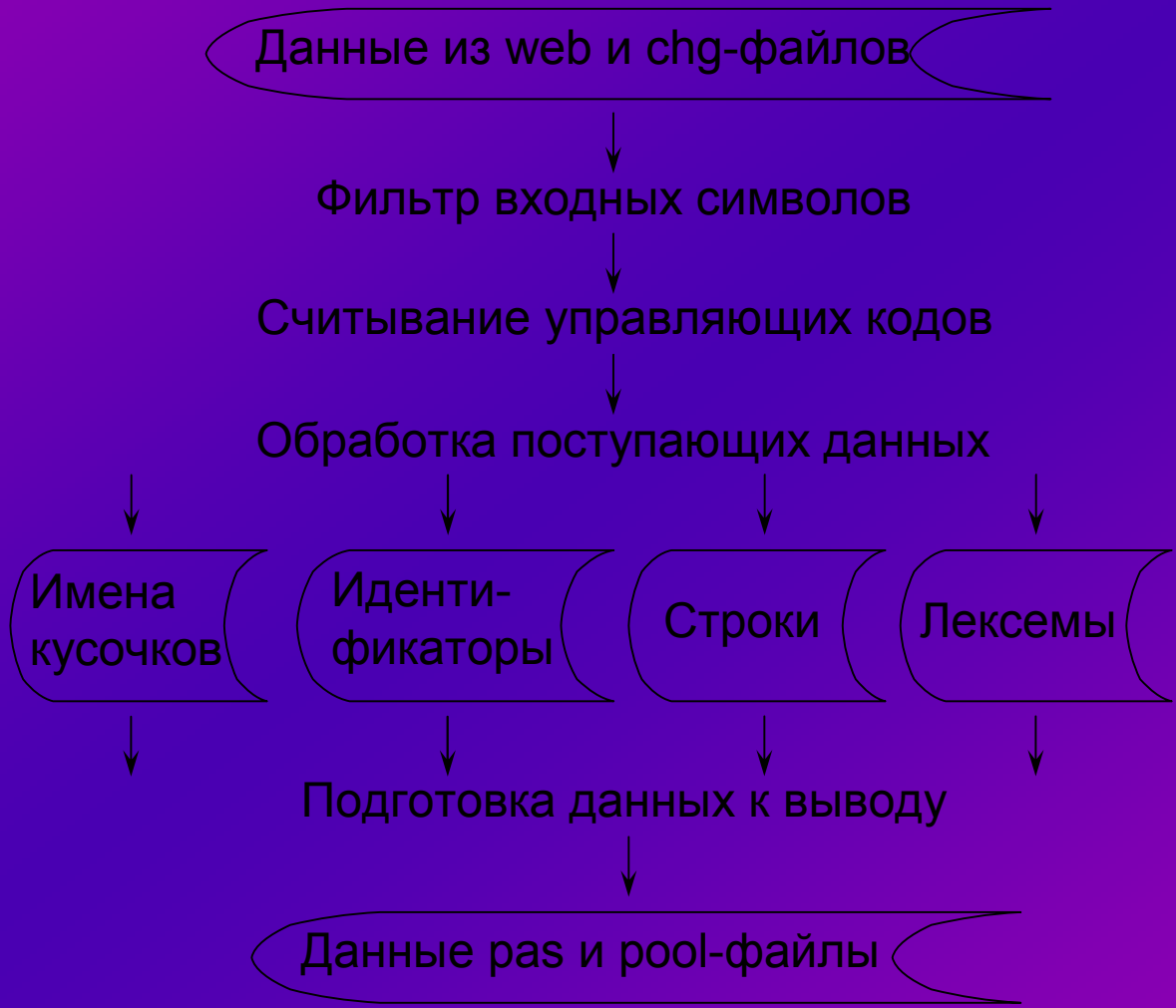
@z *комментарии*

Проблемы использования

Основные проблемы использования средств ЛП, разработанных Д. Кнутом:

- не обрабатывают знаки с кодами больше 126;
- не поддерживают расширения синтаксиса, доступные в современных компиляторах.

Поток информации в tangle



Новые возможности rutangle

- Пропускает практически любые коды ASCII.
- Выполняет транслитерацию идентификаторов, проверяет на совпадение с имеющимися латинскими.
- Не разрывает строки в выходном файле на знаке “#”.
- Позволяет записывать ассемблерные числа (200h).
- Не отбрасывает знак “_” в именах.
- Поддерживает командные скобки asm…end.
- Распознаёт однострочные комментарии //.

Новые возможности ruweave

- Пропускает практически любые коды ASCII.
- Корректно сортирует имена модулей в перечне модулей и имена идентификаторов

На данный момент ассемблерные вставки (asm…end) и однострочные комментарии не поддерживаются.

Реализация изменений в tangle и weave

Исходными для создания русифицированных версий послужили файлы tangle.web и weave.web из архива CTAN.

Изменения вносились через chg-файлы. Для этого использовалась программа tangle.exe из дистрибутива dostp22, размещённая в CTAN.

Среда программирования — свободно распространяемый компилятор free pascal.

Программам даны названия rutangle и ruweave. Все сообщения об ошибках переведены на русский язык, входным и выходным файлам можно давать русские имена.

Программы работают с файлами в кодировке CP866.

Программирование на русском языке

Можно определить ключевые слова и их формат

@d если == if

@d то == then

@f если == if

@f то == then

В web-файле	если угол = 360 то угол:=0;
В документе	**если** *угол* = 360 **то** *угол*←0;
В программе	if ugol=360 then ugol:=0;

Переработка стилевого файла

- Переведены выводимые на печать тексты на русский язык, изменены шрифты.
- Введены гиперссылки для ссылок на разделы
- Сделаны закладки для разделов со звёздочкой

Разработанная документация

- Краткое руководство по командам литературного программирования
- Учебный пример web-файла, показывающий основные средства литературного программирования

Практическое использование

- Подготовка переводов книг
  - TeX: The program
  - METAFONT: The program
- Разработка программы автоматического создания каталогов для промежуточных аттестаций магистров

Обзор других средств ЛП

Программа tie объединяет несколько chg-файлов ЛП для winword

Для других языков программирования:

- cweb — C/C++
- mweb — Matlab
- aweb — Ada
- funnelweb и nuweb — не зависят от языка
- и др.