

## Основы языка Prolog на примере swi-prolog

Дудкин И.А

Большинство языков программирования делится на две большие группы: алгоритмические (процедурные) и декларативные. Классическими процедурными языками являются, например, семейство языков C, Pascal, Fortran и многие другие, хоть они и содержат в себе декларативные элементы (описание переменных).

Языки семейства Prolog имеют чистый декларативный характер. Они используются не для обработки данных, а для обработки фактов и правил. Факты представляют собой логические формулы, а база знаний (БЗ) задается их совокупностью. Логические методы обеспечивают получение новых фактов из фактов, представленных в БЗ.

В отличие от процедурных языков, где описание решения идет строго шаг за шагом, в языке Prolog программист описывает проблему и правила ее решения, а также набор фактов, связанных с этим решением. Поэтому языки данного семейства считаются языками более высокого уровня, чем C, Паскаль и другие. Явным подтверждением, сказанного выше утверждения, является факт того, что решение задачи на Prolog занимает меньше строк в компиляторе, чем, например, в C и Pascal, хоть и существуют исключения из этого правила.

Prolog может использоваться при разработке экспертных систем, а также для следующих задач:

- доказательства теорем и вывода решений в задачах;
- создания пакетов символьной обработки при решении уравнений, дифференцировании, интегрировании и т. д.;
- разработки упрощенных версий систем ИИ;
- создания естественно-языковых интерфейсов для существующих программ;
- перевода текстов с одного языка на другой, в том числе – с одного языка программирования на другой.

Prolog появился в 1973 г., когда группа исследователей из Марсельского университета под руководством Алана Колмероз, создала программу для доказательства теорем, которая была реализована на языке Фортран. Впоследствии этот продукт получил название Prolog. Первые годы Prolog не находил широкого практического применения. Однако в 1981 г. было объявлено о японском проекте создания ЭВМ пятого поколения, в основе программного обеспечения которых предполагалось использовать логическое программирование. Этот проект создания ЭВМ для обработки знаний вызвал большой резонанс во всем мире. Появились коммерческие и свободные реализации Prolog, такие как CProlog, Quintus Prolog, Silogic

Knowledge Workbench, Turbo Prolog и компилятор Swi-Prolog о котором дальше и пойдет речь.

В этой статье рассмотрим работу с Prolog в операционной системе Linux. Прежде чем начать работать с компилятором программы необходимо его установить. Для этого выполняем следующие действия: открываем командную строку и вводим:

```
% sudo apt-add-repository ppa:swi-prolog/stable
% sudo apt-get update
% sudo apt-get install swi-prolog
```

Следует отметить, что мы устанавливаем последний стабильный релиз, для бета-версии команды в консоли будут отличаться. Первая команда находит swi-prolog, вторая загружает его в систему, а третья непосредственно устанавливает ее в операционную систему.

После установки необходимо запустить интерпретатор, для этого в консоли вводим команду “pl”, а для остановки интерпретатора необходимо ввести “halt.”. Важно отметить то, что после команды в swi-prolog **всегда обязательно ставить точку**. Так же для остановки интерпретатора можно воспользоваться комбинацией горячих клавиш «Ctrl+c + e».

Теперь запустим файл и прокомпилируем его. Для этого существует такая последовательность действий:

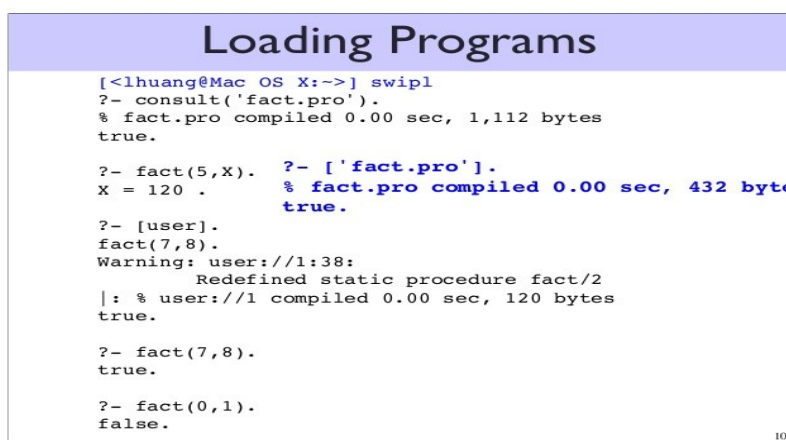
1. Сделать текущим каталог с программой.

2. Выполнить команду:

```
[имя_файла]. %для файла с именем имя_файла.pl.
```

```
[1]. %для файла с именем 1.pl.
```

3. В случае правильного выполнения предыдущих пунктов должна появиться надпись, которая информирует об удачном компилировании. Пример подобной надписи продемонстрирован на Рис 1.и Рис 2.



```

Loading Programs
[<lhuang@Mac OS X:~>] swipl
?- consult('fact.pro').
% fact.pro compiled 0.00 sec, 1,112 bytes
true.

?- fact(5,X).      ?- ['fact.pro'].
X = 120 .          % fact.pro compiled 0.00 sec, 432 bytes
                  true.

?- [user].
fact(7,8).
Warning: user://1:38:
         Redefined static procedure fact/2
|: % user://1 compiled 0.00 sec, 120 bytes
true.

?- fact(7,8).
true.

?- fact(0,1).
false.
```

Рис 1. Пример запуска файла с названием «fact.pro».

# SWI-Prolog

- We will use SWI-Prolog for the Prolog programming assignment  
– <http://www.swi-prolog.org/>
- After the installation, try the example program

```
?- [likes].                               Load example likes.pl
% likes compiled 0.00 sec, 2,148 bytes
Yes
?- likes(Mother,baby) .                    This goal cannot be proved, so it assumed to
No                                          be false (This is the so called Close World
                                          Assumption)
?- likes(sam, X) .
X = Steve ;
X = Paul ;
X = Samuel ;
```

Asks the interpreter to find more solutions

Рис 2. Пример запуска файла с названием «likes».

Теперь в кратком порядке опишем структуру языка. Основой служат факты, предикаты и правила. Факты- набор оперируемых нами данных, примером может послужить такая запись: «любит('Эллен',теннис).». Запись в кавычках должна содержать строго такую же структуру и лексику. Предикат – это логическая формула от одного или нескольких аргументов. Правила показывают отношения между фактами, эти правила позволяют делать совершенно независимые выводы по каждому запросу пользователя. Пример: «любит('Билл',Нечто):- любит('Том',Нечто).» .

Также существуют запросы. Запрос – это последовательность из одного предиката или множества предикатов, разделяемых запятыми (связка and) и завершающаяся точкой. С помощью запросов можно установить истинность соответствующего выражения. Приведем абстрактный пример: «?- P1,P2,...,Pn.», это означает: верно ли P1 и P2 и ...Pn?

Из этих структурных элементов комбинируется вся программа на языке Prolog.

Подводя итог, сделаем вывод, что язык Prolog является сильным инструментом в создании баз знаний и в программировании экспертных систем.

## Список литературы

1. Бурлаков, М. В. ЯЗЫК ЛОГИЧЕСКОГО ПРОГРАММИРОВАНИЯ ПРОЛОГ Методические указания к выполнению лабораторных работ/ М.В Бурлаков.- Редакционно-издательский отдел Отдел электронных публикаций и библиографии библиотеки Отдел оперативной полиграфии СПбГУАП 190000, Санкт-Петербург, ул. Б. Морская, 67.