

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 12:56:21
Уникальный программный ключ:
65ab2aa0d384efe8480e6a4c688eddbc475e411a

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Дюконинова
« 24 » 12 2017 г.



ТРАССИРОВКА И ОТЛАДКА ПРОГРАММ В СРЕДЕ VISUAL PROLOG

Методические указания по выполнению лабораторной работы по дисциплине «Функциональное и логическое программирование» для студентов направления подготовки 09.03.04 «Программная инженерия»

Курск 2017

УДК 004.652

Составители: В.Г. Белов, Т.М. Белова

Рецензент

Кандидат технических наук, доцент кафедры программной инженерии ЮЗГУ И.Н. Ефремова

Трассировка и отладка программ в среде Visual Prolog: методические указания по выполнению лабораторной работы по дисциплине "Функциональное и логическое программирование" для студентов направления подготовки 09.03.04 "Программная инженерия" / Юго-Зап. гос. ун-т; сост.: В.Г. Белов, Т.М. Белова, – Курск, 2017. – 12 с.: ил. 14.

Рассмотрены методы трассировки и отладки программ в среде **Visual Prolog**.

Материал предназначен для студентов направления подготовки 09.03.04 «Программная инженерия» специализации «Разработка программно-информационных систем»

Подписано в печать *24.12.17*. Формат 60x84 1/16.
Усл. печ. л. *9,6*. Уч.-изд. л. *9,5*. Тираж 100 экз. Заказ *4378*. Бесплатно.
Юго-Западный государственный университет

305040, Курск, ул.50 лет Октября, 94.

Содержание

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ	4
2 СРЕДСТВА ОТЛАДКИ СИСТЕМЫ Visual Prolog	4
2.1 Подключение отладчика	5
2.2 Запуск отладчика	6
2.3 Загрузка программы	6
2.4 Меню Views	6
2.5 Меню Run	7
2.6 Расстановка контрольных точек	8
2.7 Просмотр переменных	8
2.8 Просмотр фактов	9
2.9 Опции контроля стека	9
2.10 Окно Trap Points	9
2.11 Окно Backtrack Points	9
2.12 Просмотр на машинном уровне	10
2.13. Опции отладчика	11
3 Порядок выполнения лабораторной работы	12
4 Содержание отчета	12
5 Контрольные вопросы к защите лабораторной работы	12

1 ЦЕЛЬ ЛАБОРАТОРНОЙ РАБОТЫ

Целью лабораторной работы является приобретение навыков в трассировке и отладке программ на Visual Prolog.

2 СРЕДСТВА ОТЛАДКИ СИСТЕМЫ VISUAL PROLOG

Отладка – это процесс нахождения ошибок в программе. Подсчитано, что коррекция ошибок в программе забирает около 90% общего времени разработки. Это доказывает, что хорошие инструментальные средства для отладки очень важны.

Visual Prolog Debugger for Windows 32 является основным средством отладки программ, написанных на Visual Prolog.

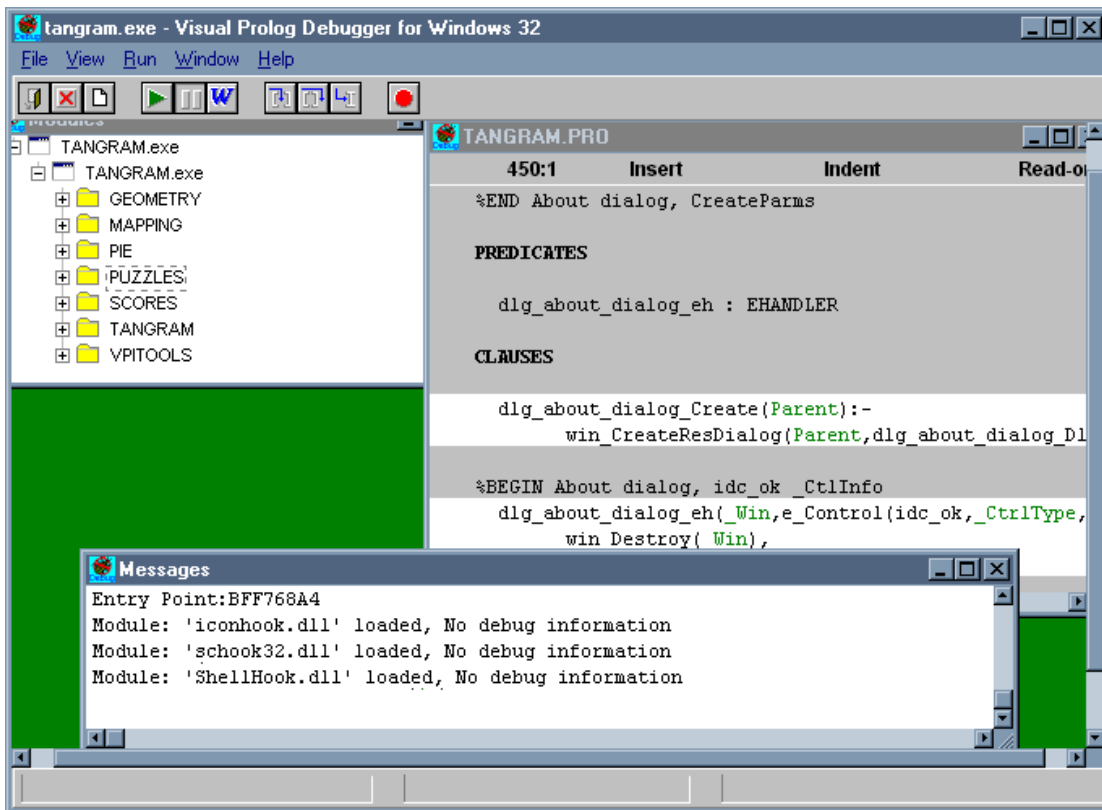


Рис.1. Отладчик Visual Prolog

Средства трассировки и отладки подразделяются на четыре основных категории: блок синтаксического контроля, блок трассировки, директивы компилятора и сообщения об ошибках в процессе выполнения программ.

Блок синтаксического контроля:

Блок синтаксического контроля работает в паре со встроенным редактором. При попытке пользователя компилировать исходный текст программы, содержащий ошибку, система выведет сообщение об ошибке в окне редактирования и поместит курсор в позицию с неверным синтаксисом.

Блок трассировки:

Имеются очень мощные средства трассировки, позволяющие :

- отслеживать по шагам выполнение программы;

- расставить точки трассировки для отслеживания выполнения определенных фрагментов программы;
- трассировать только некоторые указанные предикаты;
- выполнять трассировку в режиме оптимизации;
- интерактивно отключать и включать трассировку;

Директивы компилятора:

Третьим свойственным видом отладки является группа директив компилятора, позволяющих устанавливать тип и уровень компилятора ошибок и сообщений о них, обрабатываемых компилятором применительно к программе; сюда входят:

- предупреждения, вызываемые в тех случаях, когда в программах содержатся недетерминированные выражения.
- список предикатов программ, плюс их контекст, размер, домены аргументов и структуры потока являются ли они детерминированными и используются ли они.

Ошибки при выполнении:

Четвертый метод фиксации ошибок

- сообщения об ошибках в ходе выполнения программы. После успешной компиляции программы в ее логике может порождаться ошибка, обнаруживаемая только в ходе выполнения программы. Если такая ошибка возникает, выдается ее номер и указывается место возникновения.

Любой программист знает, что ошибки при выполнении программ представляют собой наихудший для выявления вид ошибок. Поскольку логика программы не может быть в полной мере проверена без прогона последней, средства выдачи сообщений об ошибках построены так, чтобы максимально облегчить отладку выполняющейся программы.

2.1 Подключение отладчика

Для подключения отладчика надо запустить диалог опций компилятора (compiler options). Это делается путем выбора в главном меню Visual Prolog опции Options/Project(.VPR)/Compiler Options. Появится следующее диалоговое окно:

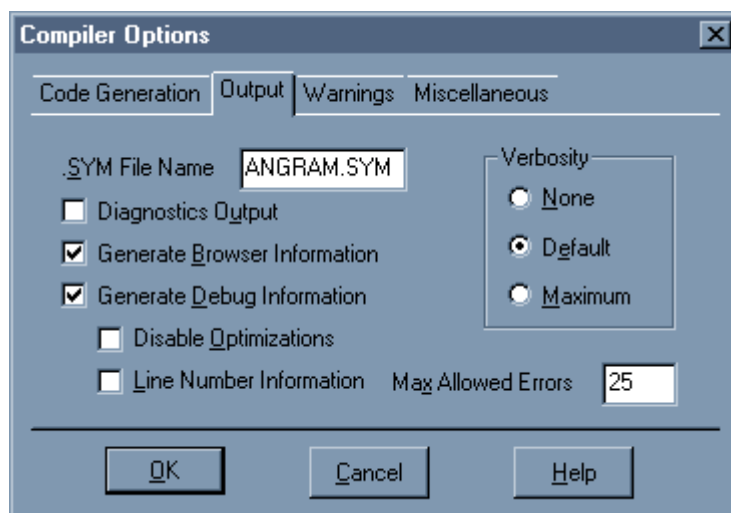


Рис. 2. Диалоговое окно Compiler Options

Включение "Generate Debug Information" заставит компилятор генерировать .DEB файлы в .OBJ директории. Эти .DEB файлы используются отладчиком для расположения переменных, фактов и т.п. во время выполнения программы.

2.2 Запуск отладчика

Когда Visual Prolog устанавливается на компьютер, программа установки установит иконку для отладчика. Для запуска отладчика необходимо запустить файл BIN\WIN\32\VIPDEBUG.EXE. К настоящему времени есть только версия WIN32 отладчика, которая может работать под Win95 и Windows NT.

2.3 Загрузка программы

Для загрузки программы, которую необходимо отладить, нужно нажать кнопку Start, находящуюся на панели инструментов.

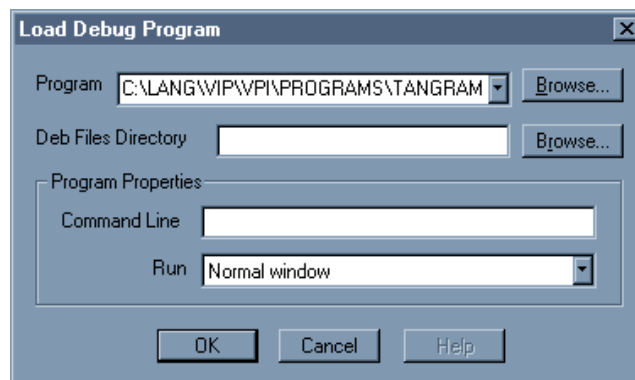


Рис. 3. Загрузка программы в Отладчик.

В поле "Command line" записывается строка с аргументами командной строки для отладки программы. Также может быть определено "Run"-поле, в нем можно выбрать способ прогона программы: полный, уменьшенный или расширенный.

Когда программа уже загружена в отладчик, входная точка и другая информация (какие модули были загружены, какой из них может быть отлажен, и т.п.) могут быть рассмотрены в Message -окне внизу экрана.

Для запуска программы используйте меню Run или кнопки на панели инструментов.

2.4 Меню Views

В отладчике могут открываться несколько окон одновременно с помощью меню Views. Рассмотрим подробнее, какие окна могут вызываться с помощью меню Views.

Окно Modules: в нем можно увидеть структуру файлов проекта, представленную в виде дерева. При двойном щелчке по имени файла, раскрывается его структура, а при двойном щелчке по имени предиката происходит переход в окно текста программы на место расположения выбранного предиката.

Для того чтобы увидеть текст программы, необходимо, чтобы программа была написана на Visual Prolog.

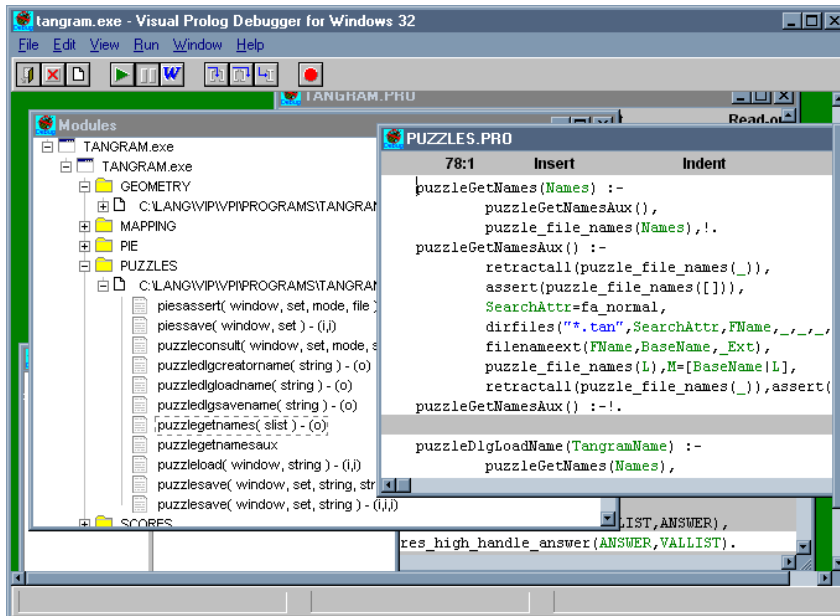


Рис. 4. Работа с окном Modules

Команда меню View->Source at Current position ставит курсор в текущую позицию указателя

В коде, находящимся на белом фоне, возможно установление контрольных точек с помощью наведения курсора в нужную позицию и двойного щелчка мышью в этом месте

С помощью правой кнопки мыши или меню Edit возможно копирование выделенного фрагмента в буфер обмена или вставка из буфера, имеющегося в нем фрагмента.

2.5 Меню Run

Меню Run используется для контроля выполнения отлаживаемой программы.

Опции меню:

Run (F9) Производит запуск отлаживаемой программы.

Break Program Остановка выполнения программы

Trace Into Пошаговая трассировка программы

Step Over Пропуск выполнения данного вызова предиката.

Run to Cursor Трассировка программы до выделенного фрагмента.



Рис. 5. Меню Run

2.6 Расстановка контрольных точек

Расстановка контрольных точек необходима для изоляции ошибок или для проверки части программы, которая работает корректно. Для того чтобы поставить контрольную точку, нужно дважды щелкнуть левой кнопкой мыши по строке, где будет контрольная точка. После запуска, отлаживаемая программа будет выполняться только до контрольной точки.

Если используется несколько контрольных точек, то их расположение и параметры можно посмотреть при вызове меню Views->Break Points

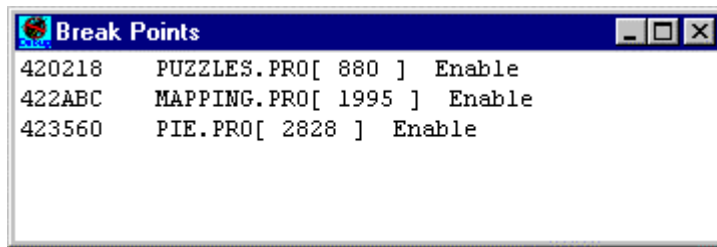


Рис. 6. Меню Break Points.

2.7 Просмотр переменных

При отладке программы бывает необходимо посмотреть текущие значения переменных. Это можно сделать, вызвав окно View->Local Variables.

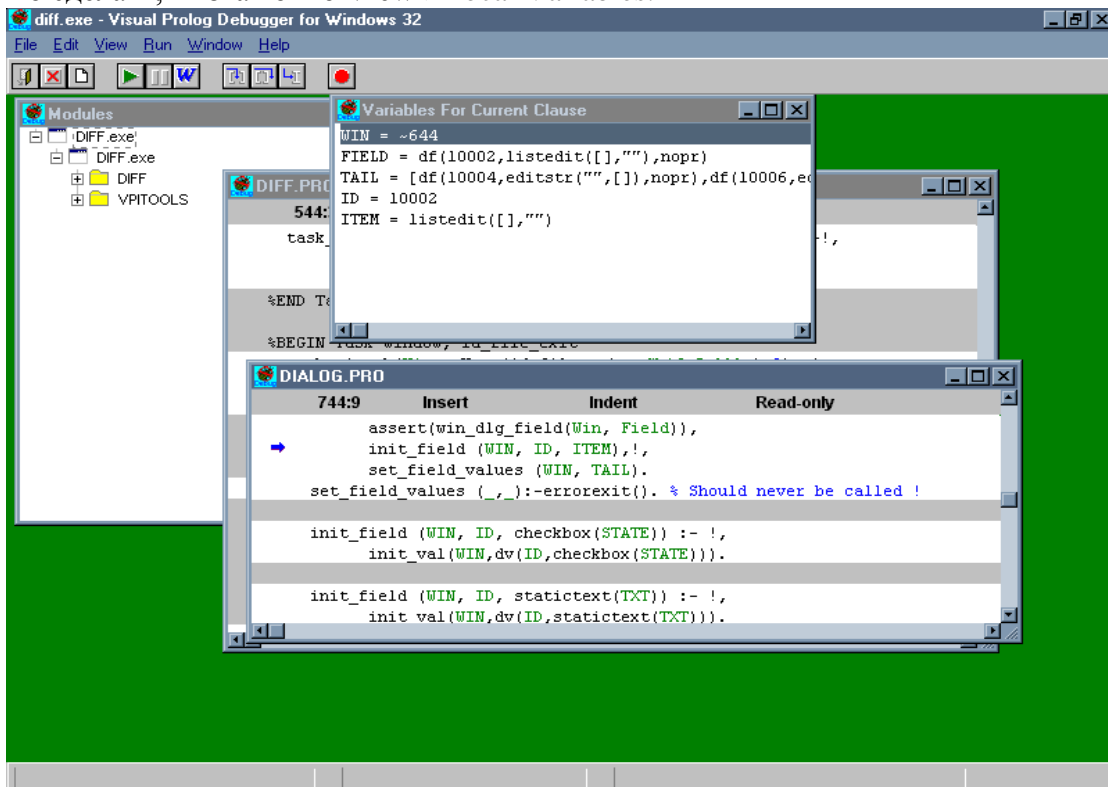


Рис. 7. Использование окна Local Variable для отладки программы

2.8 Просмотр фактов

Иногда бывает необходимо проверить правильность фактов при отладке программы. Для этого используется вызов меню Views->Facts.

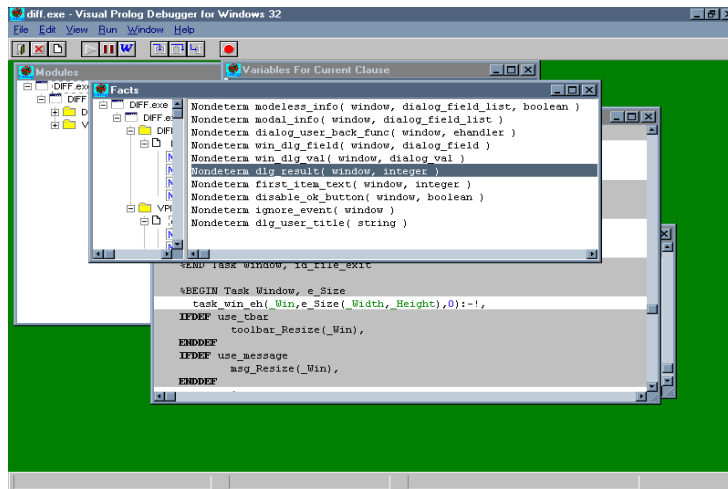


Рис.8. Вызов окна фактов.

2.9 Опции контроля стека

Окно контроля стека вызывается с помощью выбора опции Views->Call Stack.

Окно Call Stack показывает список предикатов, которые уже были вызваны во время отладки программы. Содержимое окна может быть очищено с помощью всплывающего меню, вызываемого при щелчке правой кнопкой мыши.

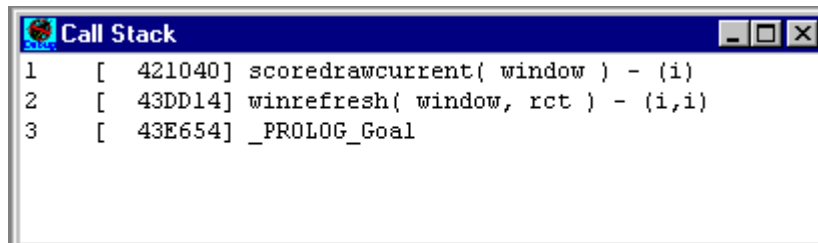


Рис. 9. Окно Call Stack

2.10 Окно Trap Points

Окно Trap Points показывает список контрольных точек отлаживаемой программы. Двойной щелчок по контрольной точке показывает код, где расположена контрольная точка.

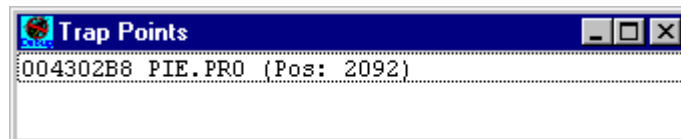


Рис.10. Окно Trap Points.

2.11 Окно Backtrack Points

Окно Backtrack Points показывает список точек backtrack-инга отлаживаемой программы. Двойной щелчок по точке backtrack-инга показывает код, где расположена контрольная точка.

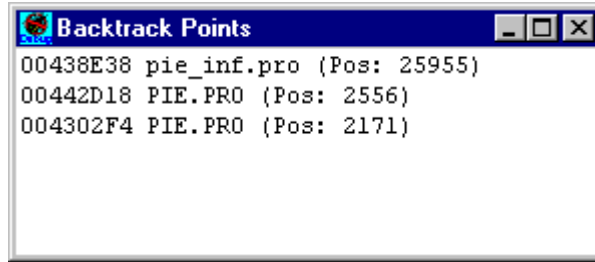


Рис.11 Окно Backtrack Points:.

2.12 Просмотр на машинном уровне

Имеется возможность просмотра ассемблерных кодов непосредственно в PDC Debugger, содержания регистров или памяти. Это необходимо, если требуется просмотреть код, написанный не на Прологе.

Дизассемблер

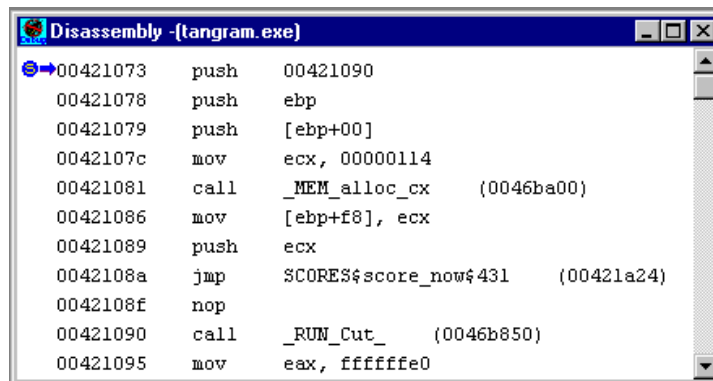


Рис.2:Просмотр ассемблерного кода

Память

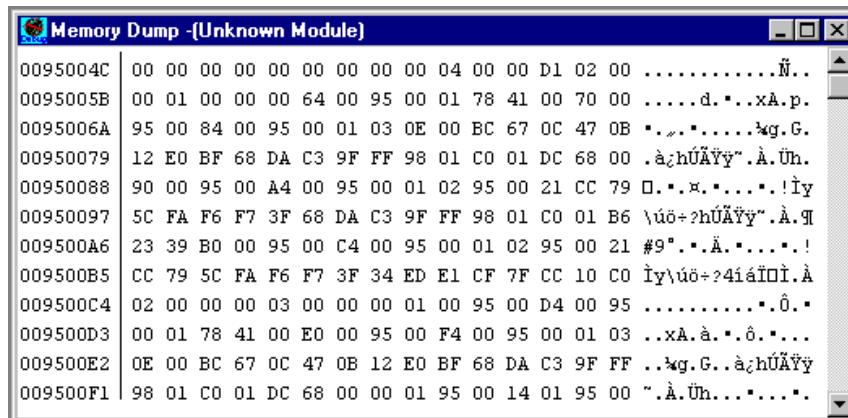


Рис.12. Просмотр памяти.

Регистры

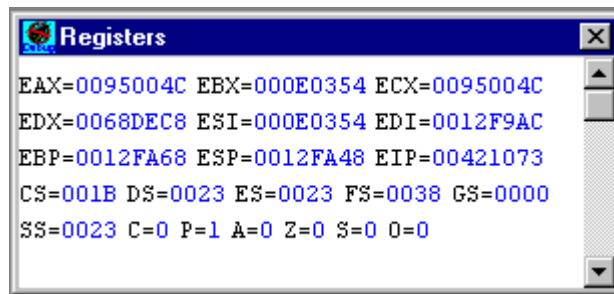


Рис.13. Просмотр машинных регистров.

2.13. Опции отладчика

Опции отладчика вызываются из меню Views->Configurations

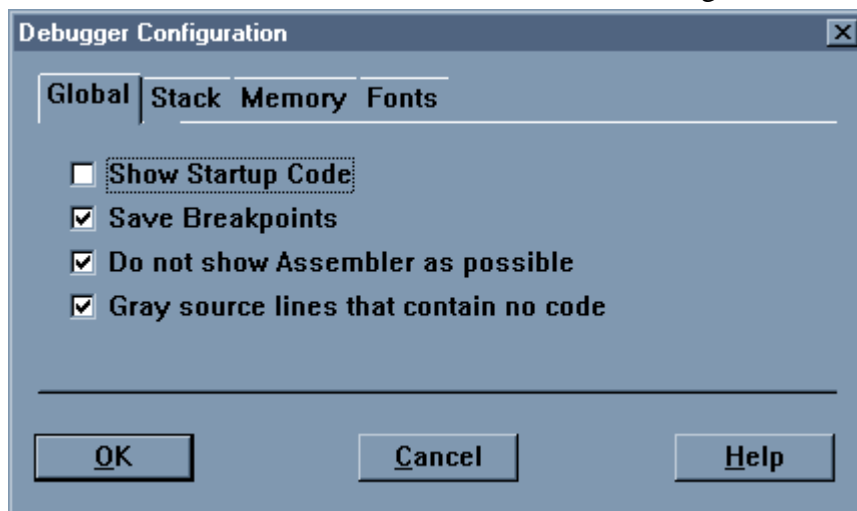


Рис. 14. Опции отладчика.

Show Startup Code – допускается отладка программы перед запуском Goal.

Scanning Call Stack from Goal – если выбрана, то окно Call Stack будет содержать только те вызовы, которые произошли после вызова Goal.

Save Breakpoints – сохраняет текущий проект в ".ini" файл вместе с контрольными точками.

Gray source lines that contain no code – линии отлаживаемой программы делаются серыми там, где не могут стоять контрольные точки.

Fonts – изменение шрифтов при работе в отладчике.

3 ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

Для выполнения лабораторной работы необходимо проверить все режимы работы при трассировке и отладке. Результат трассировок в различных режимах необходимо отобразить в отчете.

4 СОДЕРЖАНИЕ ОТЧЕТА

Отчет должен содержать результат трассировок в различных режимах.

5 КОНТРОЛЬНЫЕ ВОПРОСЫ К ЗАЩИТЕ ЛАБОРАТОРНОЙ РАБОТЫ

1. Какие средства трассировки и отладки существуют на Visual Prolog?
2. Какие виды ошибок существуют?
3. Как подключить и загрузить отладчик в Visual Prolog?
4. Чем отличается запуск программы с помощью пункта **Trace Into** от **Step Over**.
5. Для чего и каким образом выполняется просмотр переменных и фактов?
6. Для чего используются окна Disassembly, Memory View и Register View?
7. Каким образом в программу можно вставить контрольную точку?