

Об актуальной версии-ПЗ
системы программирования Рефал-5
и
задачах развития и поддержки Рефала

Андрей П. Немытых
Институт программных систем РАН
г. Переславль-Залесский

Совместное рабочее совещание ИПС РАН и МГТУ имени Н.Э. Баумана
по функциональному языку программирования Рефал

19 декабря 2025 г., Москва

Unicode-версия Рефала-5 под ОС Windows-10 / x86_64

(Windows-YY \geq Windows-10) / x86_64

- Основной разработчик:
 - Михаил Терюха (МГТУ имени Н.Э. Баумана)
- при активном и плодотворном обсуждении принципов переноса в среду ОС Windows (из среды ОС Linux) с:
 - Александром Коноваловым,
 - Антониной Непейвода.

Unicode-версия Рефала-5 под ОС Windows-10 / x86_64

cmd-консольное приложение:

- Использует кодировку **UTF-8**.
 - Активная кодовая страница должна быть установлена как 65001 (UTF-8).
 - Исходные тексты программ на Рефале-5 должны быть в кодировке UTF-8.
- Исполняемые модули выложены для свободного доступа:
http://www.botik.ru/pub/local/scp/refal5/ref5_win10_x86_64_exe_252309.zip
и:
http://refal.botik.ru/refal5/ref5_win10_x86_64_exe_252309.zip

Unicode-версия Рефала-5 / x86_64

В 2023 году была выложена Unicode-версия Рефала-5 под ОС Linux.

- Основной разработчик:
 - Михаил Терюха (МГТУ имени Н.Э. Баумана)
- Использует кодировку UTF-8:
 - Unicode-кодировка UTF-8 является базисной в среде ОС Linux.

Unicode-версия Рефала-5 / x86_64

UTF-8 vs. UTF-16

- ОС Linux:
 - базисная Unicode-кодировка **UTF-8**
- ОС Windows:
 - базисная Unicode-кодировка **UTF-16**,
 - точнее – UTF-16LE и **другие проблемы ...**

Unicode-версия Рефала-5 / x86_64

UTF-8 vs. UTF-16

- UTF-8:
 - множество неподвижных кодовых точек: множество всех ASCII-символов с кодом < 128 ;
 - $1 \leq \text{sizeof}(\text{UTF-8-код символа}) \leq 4$.
- UTF-16:
 - множество неподвижных кодовых точек является пустым;
 - $\text{sizeof}(\text{UTF-16-код символа})$ равен 2 или 4.

Unicode-версия Рефала-5 / x86_64

UTF-8 vs. UTF-16

- UTF-8:
 - множество неподвижных кодовых точек: множество всех ASCII-символов с кодом < 128 ;
 - \implies все исходные тексты, ранее работавшие в контексте ASCII-символов, работают в среде Unicode-символов, если не используют ASCII-символы с кодом ≥ 128 .
- UTF-16:
 - множество неподвижных кодовых точек является пустым;
 - \implies все исходные тексты, ранее работавшие в контексте ASCII-символов, требуют изменений при переходе в среду Unicode-символов.

Unicode-версия Рефала-5 / x86_64

Linux vs. Windows

Использование, насколько это возможно, **одной и той же логики работы с Unicode-символами** в разных операционных системах:

- **минимизирует накладные расходы** при работе с ASCII-символами с кодом < 128 ;
- **позволяет сохранить компактность исходные текстов** реализации Рефала-5;
- **упрощает поддержку** системы программирования Рефала-5;
- **упрощает разработку кроссплатформенных приложений** на Рефала-5:
 - например, результаты компиляции Рефал-программ в промежуточный язык сборки, полученные в одной ОС, могут быть использованы в среде другой ОС.

Unicode-версия Рефала-5 / x86_64

Windows-10 / О выборе компилятора.

- Компилятор Microsoft C/C++ x86_64 ориентирован на базисную логику Windows, т.е на кодировку UTF-16.
- **Компилятор MinGW** - Minimalist GNU for Windows (Copyright (C) 2024 Free Software Foundation, Inc.) производит исполняемый код, логика поведения которого максимально - насколько это логически возможно - имитирует поведение соответствующего исполняемого кода, который строит компилятор gcc в среде ОС Linux/Unix.
item gcc (x86_64-win32-seh-rev2, Built by MinGW-Builds project) 14.2.0

Unicode-версия Рефала-5 под ОС Windows-10 / x86_64

cmd-консольное приложение:

- Использует кодировку **UTF-8**.
 - Активная кодовая страница должна быть установлена как 65001 (UTF-8).
 - Исходные тексты программ на Рефале-5 должны быть в кодировке UTF-8.
- Исполняемые модули выложены для свободного доступа:
http://www.botik.ru/pub/local/scp/refal5/ref5_win10_x86_64_exe_252309.zip
и:
http://refal.botik.ru/refal5/ref5_win10_x86_64_exe_252309.zip

Поддержка Кириллицы и китайских иероглифов

Исходный текст программы и его результат работы.

```
Командная строка
c:\test\examples>type Chinese.ref
$ENTRY Go { = <Prout '會 倪 伯 粵 懷 '>
             <Prout "АЏБУЏА">; }

c:\test\examples>..\refc Chinese.ref

c:\test\examples>..\refgo Chinese
會 倪 伯 粵 懷
АЏБУЏА

c:\test\examples>
```

Греческий алфавит

test.ref / В.Ф. Турчин

- Живая демонстрация ...

Unicode-версия Рефала-5 / x86_64

Windows-10 / О выборе консоли.

Качество (и возможность) отражения Unicode-символов зависит от выбранной консоли и подгруженных к ней шрифтов.

- Эксперименты показывают, что версия фактически работает под ОС Windows-7, но её стандартная cmd-консоль умеет корректно отражать только Unicode-символы локали, зафиксированной при инсталляции системы.
- Ошибки cmd-консоли есть и в актуальной версии Windows-11.

Церковно-славянский шрифт

Фрагмент исходного текста программы. Ошибка в консольной команде type:

```

Командная строка
c:\test>type examples\Sirin.ref
$ENTRY Go {
  = <Prout '\n' <Length Plural 'Люби́ дѡброе собесѣдованіе, ѿ слы́хъ же бесѣды ѡдалайса:\nзане́ '
    'ни волшебникъ, ни разбѣйникъ, ни гроборазоритель тако родѣшася,\n'
    'но тако научѣшася ѿ чловѣквѣ, растлѣнныхъ ѡмомъ ѿ сатаны.'>>

  <Prout '\n' <Length Plural 'И ѡкоже тѡло, аше хлѣба не прѣиметь, живо бѣти не мѡжетъ:\n'
    'тако и дѡша, аше не прѣиметь пици своея, мертвѣ естъ. Ибо чловѣкъ сѣгѡбѣ
    естъ,\n'

    'и зъ дѡшѣи и тѣла. И сегѡ ради глаголаше Спсѣ:\n '
    'не ѡ хлѣбѣ единомъ живѣ бѣдетъ чловѣкъ.'
    >>;

/*      Люби́ дѡброе собесѣдованіе, ѿ слы́хъ же бесѣды ѡдалайса:\nзане́ '
    'ни волшебникъ, ни разбѣйникъ, ни гроборазоритель тако родѣшася,\n'
    'но тако научѣшася ѿ чловѣквѣ, растлѣнныхъ ѡмомъ ѿ сатаны.'>> ;

~
*/

/*
Для уточнения букв церковно-славянского алфавита:
«и въ слнци оучиниса ѡко мѣцъ . и зъ рогъ ѡго ѡко ѡгль жаровь исхожаше»

*/
}

```

Церковно-славянский шрифт

Функция печати Prout Рефала-5 вывела текст корректно:

```
Командная строка  x  +  v

c:\test>refc examples\Sirin.ref
c:\test>refgo examples\Sirin

Люби́ дѡброе собесѣдованіе, ѿ слы́хъ же бесѣдъ ѡудалайса:
ѡанѣ ни волшѣбникъ, ни разбѡйникъ, ни гроборазѡритель тако родишася,
но тако научишася ѿ человекъ, растлѣнныхъ ѡмомъ ѿ сатаны.

В этой цитате 196 символов, включая пробелы и символы перевода строки.

И ѡкоже тѣло, аще хлѣба не пріиметь, живо быти не мѡжетъ:
такѡ и дѡшя, аще не пріиметь пищи своея, мертвѡ естъ. Ибо человекъ сѡгѡубъ естъ,
и ѡ дѡшй и тѣла. И сегѡ ради глаголаше Спсѡ:
не ѡ хлѣбѣ единомъ живѣ бѡдетъ человекъ.

В этой цитате 246 символов, включая пробелы и символы перевода строки.

c:\test>
```

Unicode-версия Рефала-5 / x86_64

UTF-8 vs. UTF-16 / Проблемы общения с внешним миром

- ОС Linux:
 - базисная Unicode-кодировка **UTF-8**;
 - базисная файловая система **EXT4**;
- ОС Windows:
 - базисная Unicode-кодировка **UTF-16**,
 - точнее – UTF-16LE и **другие проблемы ...**
 - базисная файловая система **NTFS**.

Имена файлов и имена и значения переменных среды (environment variables) и т.п. хранятся в базисных кодировках.

Unicode-версия Рефала-5 / x86_64

Решение проблем общения с базисной файловой системой ОС Windows:

- Реализованы функции перекодировок:
 - UTF-8 \implies UTF-16
 - UTF-8 \impliedby UTF-16
- Область определения встроенной процедуры Рефала-5 Open:
расширена кодировками:
 - UTF-16LE, UTF-16BE;
- Встроенные процедуры ввода/вывода Рефала-5 расширены на чтение/запись в кодировках UTF-16.

Имена файлов и имена и значения переменных среды (environment variables) и т.п. хранятся в базисных кодировках.

Unicode-версия Рефала-5 / x86_64

Неожиданная проблема общения входных точек исполняемых модулей, построенных компилятором MinGW, и командной строки `cmd` –

```
c : \test> refgo Unicode_Arg1 Unicode_Arg2
```

```
main (int argc, char * argv []) {  
    .....  
}
```

не могут договориться, несмотря на переключение активной кодовой страницы:

- `main` получает (после применения некоторой маски) символы в кодировке конкретной локали, которая установлена при инициализации ОС Windows; в нашем случае – Windows-1251;
- замена `char` на `wchar_t` не решает проблемы.

Unicode-версия Рефала-5 / x86_64

Проблема общения входных точек исполняемых модулей, построенных компилятором MinGW, и командной строки cmd –

```
c:\test> refgo Unicode_Arg1 Unicode_Arg2
```

```
main (int argc, char * argv []) {  
    .....  
}
```

не могут договориться, несмотря на переключение активной кодовой страницы.

- Документации по этому вопросу не найдено ни у Microsoft C/C++ x86_64 (в Windows), ни в MinGW.

Unicode-версия Рефала-5 / x86_64

Проблема общения входных точек исполняемых модулей, построенных компилятором MinGW, и командной строки cmd.

```
void r_wargv_init(void) {  
# ifdef BY_GCC_MINGW_WIN  
    r_si = 0;  
    __wgetmainargs( &r_wargc, &r_wargv, &r_wenvp,  
                   _CRT_glob, &r_si  
                   );  
# endif  
    return; // Linux  
}
```

Имена исходных модулей с суррогатными парами
Unicode-символов

Исходный текст программы и результат его работы.

```
Командная строка
c:\test>type examples\Exoticus_世_𐄂_𐄃.ref
$ENTRY Go {
  = <Prout "The name of the executed refal-program" ('Exoticus_世_𐄂_𐄃.ref')
  " includes utf-16 surrogate symbols!";
}

c:\test>refc examples\Exoticus_世_𐄂_𐄃.ref

c:\test>refgo examples\Exoticus_世_𐄂_𐄃
The name of the executed refal-program (Exoticus_世_𐄂_𐄃.ref) includes utf-16
surrogate symbols!
```

Вызов cmd-консоли из Рефал-программы

Исходный текст программы и результат его работы.

```
Командная строка - refgo.exe
c:\test>chcp 65001
Active code page: 65001

c:\test>type examples\system.ref
$ENTRY Go {

    = <Prout <System 'chcp 65001 & echo привет 世_ノ_界 & pause'>>;

}

c:\test>refc examples\system.ref

c:\test>refgo examples\system
Active code page: 65001
привет 世_ノ_界
Press any key to continue . . .
```

Другие примеры

- Живая демонстрация ...

Список задач №1: Анализ и преобразование программ

- Разработка языка сообщений об ошибках в Рефал-программах:
 - удобного для автоматического анализа средствами Рефала;
 - развитие на этой основе парсера `prefal`.
- Разработка языка псевдо-комментариев компиляции и документации Рефал-программ:
 - удобного для автоматического анализа средствами Рефала;
 - реализация автоматического документирования Рефал-программ.
- Опциональное включение в компилятор Рефала:
 - простейших инструментов суперкомпиляции;
 - упрощенных версий алгоритмов суперкомпиляции;
 - абстрактного интерпретатора встроенных функций Рефала.

Список задач №2: Интерпретаторы и алгоритмы

- Развитие библиотеки (или реализация новых библиотек), написанных на Рефале:
 - интерпретатор разумного алгоритмически полного подмножества языка JavaScript:
 - ориентированный на преобразование строковых данных;
 - алгоритмы классической комбинаторики в словах, включая:
 - алгоритмы Хмелевского для решения некоторых классов уравнений в словах,
 - алгоритм Маканина перечисления решений уравнений в словах,
 - алгоритм Пландовского перечисления решений уравнений в словах.
 - интерпретатор разумного алгоритмически полного подмножества языка Prolog.

Список задач №3: Арифметика

- Развитие библиотеки (или реализация новых библиотек), написанных на Рефале:
 - интерпретатор арифметических выражений и арифметических схем, включающих:
 - действительные числа с плавающей запятой,
 - стандартные математические функции;
 - алгоритмы упрощения и решения систем и классов систем линейных уравнений с целыми, рациональными коэффициентами и коэффициентами с плавающей запятой.

Список задач №4: Анализ и форматирование текстов

- Разработка простого языка графов и
 - реализация на языке Рефал компилятора из этого языка в LaTeX-язык рисунков/графов.
- Разработка на языке Рефал инструментов автоматического форматирования:
 - русскоязычных и англоязычных комментариев.
- Реализация на языке Рефал:
 - инструментов поиска файлов, аналогичных соответствующим утилитам в ОС Linux;
 - инструментов аналогичных инструментам из пакета pdftk.