



Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ имени Н.Э.БАУМАНА

(национальный исследовательский университет)»

Факультет: Информатика и системы управления

Кафедра: Теоретическая информатика и компьютерные технологии

Лабораторная работа № 1

Раскрутка самоприменимого компилятора

Вариант 2

Работу выполнил

студент группы ИУ9-61

Бакланова А.Д.

Москва, 2020

Цель работы:

Целью данной работы является ознакомление с раскруткой самоприменимых компиляторов на примере модельного компилятора.

Исходные данные:

В качестве модельного выберем компилятор BeRo Tiny Pascal, разработанный Бенжамином Рузо (Benjamin Rosseaux) [1]. Входным языком компилятора является язык Pascal, совместимый с диалектами Delphi 7 и FreePascal ≥ 3.0 , а целевым языком — исполнимый код Win32.

Исходный текст компилятора составлен на языке Pascal, совместимом с подмножеством диалектов Delphi 7 и FreePascal ≥ 3.0 , при этом сам реализован на этом подмножестве. Тем самым, компилятор является самоприменимым.

Исходные данные для выполнения лабораторной работы в операционной системе Windows представлены следующим набором файлов:

btpc.pas — исходный текст компилятора BeRo Tiny Pascal;

btpc.exe — бинарная версия компилятора, полученная путём раскрутки;

hello.pas — программа, предназначенная для проверки работоспособности компилятора.

Индивидуальный вариант:

2	<p>Р5: не разрешать комментарием, начинающимся с (*, заканчиваться на }, а комметариям, начинающимся с {, заканчиваться на *).</p> <p>BeRo Tiny Pascal: Разрешать комментарием, начинающимся с (*, заканчиваться на }, а комметариям, начинающимся с {, заканчиваться на *).</p>
---	--

Задание:

Выполнение лабораторной работы заключается в осуществлении одного шага раскрутки самоприменимого компилятора BeRo Tiny Pascal и состоит из нескольких этапов:

1. добавление во входной язык компилятора **btpc** новых возможностей (см. таблицу 1) путём редактирования его исходного текста, в результате чего должен получиться файл **btpc2.pas** (следует сначала скопировать **btpc.pas** в **btpc2.pas** (на Linux: **btpc64.pas** в **btpc64-2.pas**), а потом вносить в него правки);
2. компиляция **btpc2.pas**, в результате которой должен получиться файл **btpc2.exe**;

3. проверка работоспособности **btpc2.exe** на небольшой программе, в которой обязательно должны использоваться новые возможности языка;
4. внесение изменений в **btpc2.pas**, связанных с использованием новых возможностей языка, и сохранение новой версии исходного текста компилятора в файле **btpc3.pas**;
5. завершение шага раскрутки путём компиляции **btpc3.pas** с помощью полученного на этапе 2 файла **btpc2.exe**;
6. разница между файлами **btpc.pas** и **btpc2.pas** (отображаемая командой `fc btpc.pas btpc2.pas`, на Linux: `diff -u btpc64.pas btpc64-2.pas`) должна демонстрировать изменения, внесённые в логику работы компилятора;
7. разница между файлами **btpc2.pas** и **btpc3.pas** (отображаемая командой `fc btpc2.pas btpc3.pas`, на Linux: `diff -u btpc64-2.pas btpc64-3.pas`) должна демонстрировать новые возможности языка.

Реализация:

На листингах 1 и 2 представлено добавление во входной язык компилятора новой возможности

```
754 end else if CurrentChar='(' then begin
755   ReadChar;
756   if CurrentChar='*' then begin
757     ReadChar;
758     LastChar:='-';
759     while (CurrentChar<>#0) and not (((CurrentChar=')') and (LastChar='*'))
       or (LastChar='}')) do begin
760       LastChar:=CurrentChar;
761       ReadChar;
762     end;
763     ReadChar;
764     GetSymbol;
```

Листинг 1 - при начале комментария с символов '(' разрешено закрывать его символом '}' или '*')

```
786 end else if CurrentChar='{ ' then begin
787   ReadChar;
788   LastChar:='-';
789   while (CurrentChar<>#0) and not (((CurrentChar=')') and (LastChar='*')) or
      (CurrentChar='}')) do begin
790     LastChar:=CurrentChar;
791     ReadChar;
792   end;
793   ReadChar;
794   GetSymbol;
```

Листинг 2 - при начале комментария с символа '{ ' разрешено закрывать его символами '*)' или '}'

Тестирование

Чтобы проверить работоспособность нового компилятора, написан тест продемонстрированный в Листинге 3.

```
program Hello;
begin
  WriteLn('Hello, student!');

  { ----- *)
  WriteLn('Work plz plz!');
  { }
  (* -----}
end.
```

Листинг 3 - тест для компилятора

При запуске программы с помощью `btpc3.pas` программа работает правильно.

```
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/lab1-2$ ./btpc64-2 <hello2.pas> hello3
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/lab1-2$ chmod +x hello3
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/lab1-2$ ./hello3
Hello, student!
Work plz plz!
```

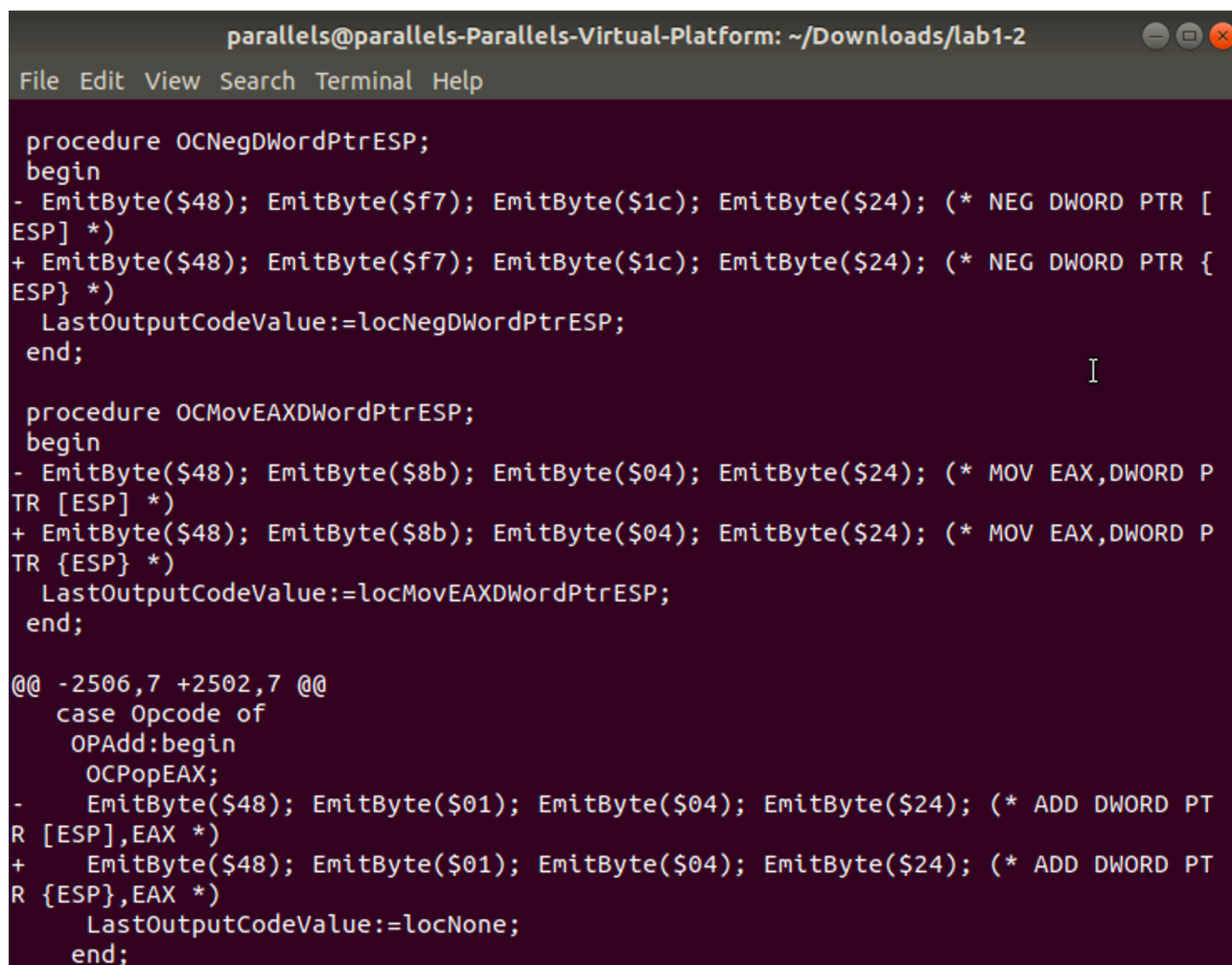
Далее создаем `btpc3.pas` и проверяем, что этот файл можно скомпилировать с помощью `btpc2.exe`

```
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/lab1-2$ ./btpc64-3 <btpc64-3.pas> btpc64-3a
parallels@parallels-Parallels-Virtual-Platform:~/Downloads/lab1-2$ ls -l
total 832
-rwxr-xr-x 1 parallels parallels 66996 Feb 15 04:07 btcp64-22
-rwxr-xr-x 1 parallels parallels 66996 Feb 15 04:14 btcp64-23
-rwxr-xr-x 1 parallels parallels 66929 Feb 15 05:07 btcp64-24
-rwxr-xr-x 1 parallels parallels 66810 Feb 11 19:45 btpc64
-rw-r--r-- 1 parallels parallels 43 Feb 15 05:50 btpc64-1
-rwxr-xr-x 1 parallels parallels 66926 Feb 15 05:37 btpc64-2
-rw-rw-r-- 1 parallels parallels 87916 Feb 15 05:36 btpc64-2.pas
-rwxr-xr-x 1 parallels parallels 66929 Feb 15 05:56 btpc64-3
-rw-r--r-- 1 parallels parallels 66929 Feb 15 05:56 btpc64-3a
```

При наличии синтаксической ошибки в коде компилятор в стандартный вывод записывает вместо двоичного кода сообщение об ошибке. Признаком того, что компиляция прошла неудачно является малый размер целевого файла (в данном примере `hello.exe`) — менее 100 байт. Для того, чтобы посмотреть размер файла, можно выполнить команду `ls -l`

Размер файла нормальный, ошибки нет.

Также, на втором этапе раскрутки пришлось исправлять некоторые комментарии, иначе компилятор не раскручивался из-за нововведений. Разница между btrc64-2.pas и btrc64-3.pas приведена ниже с помощью команды diff -u



```
parallels@parallels-Parallels-Virtual-Platform: ~/Downloads/lab1-2
File Edit View Search Terminal Help

procedure OCNegDWordPtrESP;
begin
- EmitByte($48); EmitByte($f7); EmitByte($1c); EmitByte($24); (* NEG DWORD PTR [
ESP] *)
+ EmitByte($48); EmitByte($f7); EmitByte($1c); EmitByte($24); (* NEG DWORD PTR {
ESP} *)
  LastOutputCodeValue:=locNegDWordPtrESP;
end;

procedure OCMovEAXDWordPtrESP;
begin
- EmitByte($48); EmitByte($8b); EmitByte($04); EmitByte($24); (* MOV EAX,DWORD P
TR [ESP] *)
+ EmitByte($48); EmitByte($8b); EmitByte($04); EmitByte($24); (* MOV EAX,DWORD P
TR {ESP} *)
  LastOutputCodeValue:=locMovEAXDWordPtrESP;
end;

@@ -2506,7 +2502,7 @@
  case Opcode of
    OPAdd:begin
      OCPopEAX;
-      EmitByte($48); EmitByte($01); EmitByte($04); EmitByte($24); (* ADD DWORD PT
R [ESP],EAX *)
+      EmitByte($48); EmitByte($01); EmitByte($04); EmitByte($24); (* ADD DWORD PT
R {ESP},EAX *)
      LastOutputCodeValue:=locNone;
    end;
```

Вывод

В ходе выполнения лабораторной работы, был получен опыт раскрутки самоприменимых компиляторов, возможность добавлять в компилятор дополнительные возможности.