

# Oml МультиОберон для LLVM Быстрый Старт

Copyright © 2019, by [Dmitry Dagaev](#)

Oml is the instance of MultiOberon compiler with LLVM backend. Used prepared library with LLVM 5.0.  
Oml это реализация компилятора МультиОберон с бэкендом LLVM. Использует подготовленную библиотеку на основе LLVM 5.0.  
Версия 0.8 24-May-2019

## Как начать работу из Командной Строки.

### 1 Инсталляция

#### 1. Предусловия.

Oml не использует другие программные средства. Выполняйте все вышеперечисленные команды из корневого каталога Mob-master. **Распакуйте «здесь» Blwe.zip в корневой каталог Mob-master, получится Blwe. Распакуйте «здесь» Blwr.zip в корневой каталог Mob-master, получится Blwr.**

### 2 Компиляция примеров

```
Blwe\OmlSh co OmtestHelloWorld
```

Новый символьный файл создается с соответствующим сообщением, затем OmtestHelloWorld.mod компилируется в 32-битовый Omtest/Clwe/HelloWorld.bc. Может быть задан не один, а список файлов для компиляции.

```
Blwe\OmlSh ru OmtestHelloWorld
```

Ru[n] – Выполнение. Выполнить 32-битовый OmtestHelloWorld.bc как динамически загруженный модуль с помощью оболочки Oml Shell.

```
Blwe\OmlSh ex OmtestHelloWorld
```

Ex[ecute]. Execute означает 32-битовую компиляцию и выполнение OmtestHelloWorld.mod с помощью оболочки Oml Shell.

```
Blwe\OmlSh co OmtestHelloWorld: OmtestFormats: OmtestDateTime: OmtestMkTraps: OmtestHeap:
```

32-битовая компиляция всех перечисленных примеров выполняется командой выше.

```
Blwr\OmlSh co OmtestHelloWorld
```

Новый символьный файл создается с соответствующим сообщением, затем OmtestHelloWorld.mod компилируется в 64-битовый Omtest/Clwr/HelloWorld.bc. Может быть задан не один, а список файлов для компиляции.

```
Blwr\OmlSh ru OmtestHelloWorld
```

Run Ru[n] – Выполнение. Выполнить 64-битовый OmtestHelloWorld.bc как динамически загруженный модуль с помощью оболочки Oml Shell.

```
Blwr\OmlSh ex OmtestHelloWorld
```

Ex[ecute]. Execute означает 64-битовую компиляцию и выполнение OmtestHelloWorld.mod с помощью оболочки Oml Shell.

```
Blwr\OmlSh co OmtestHelloWorld: OmtestFormats: OmtestDateTime: OmtestMkTraps: OmtestHeap:
```

64-битовая компиляция всех перечисленных примеров выполняется командой выше.

### 3 Выполнение примеров

#### 3.1. Самый простой пример - Hello, World

```
Blwe\OmlSh ru OmtestHelloWorld
```

Печать строковых, целых и действительных чисел

```
Blwe\OmlSh ru OmtestFormats
```

#### 3.2 Печать даты, времени и реализация задержек

```
Blwe\OmlSh ru OmtestDateTime
```

#### 3.3 Травы – обработка нештатных ситуаций

Простой Assert

```
Blwe\OmlSh ru OmtestMkTraps -trap a
```

Простой Halt

```
Blwe\OmlSh ru OmtestMkTraps -trap h
```

Деление на ноль

```
Blwe\OmlSh ru OmtestMkTraps -trap z
```

Обращение к памяти по нулевому указателю

```
Blwe\OmlSh ru OmtestMkTraps -trap p
```

#### 3.4 Работа с динамической памятью и сборка мусора

```
Blwe\OmlSh ru OmtestHeap
```

### 4 Набор примеров

Предусловия.

Для компиляции и линковки в бинарники требуются средства разработки для языка C. В данном пакете эти средства не поставляются и представлены не будут: ни Visual Studio, ни MinGW, ни CMake. Вы можете устанавливать средства разработки C или модифицировать скрипты сборки под установленные уже средства на Вашем компьютере. В скриптах используется следующее:

- gcc, ar – для lwe (oFront-Windows-32bit),
- clang – для lwr (oFront-Windows-64bit).

```
lwe_tomake
```

Создает все исполняемые файлы набора примеров для 32-бит

```
lwe_toclean
```

Удаляет все исполняемые файлы набора примеров для 32-бит

```
lwr_tomake
```

Создает все исполняемые файлы набора примеров для 64-бит

```
lwr_toclean
```

Удаляет все исполняемые файлы набора примеров для 64-бит

### 5 Выполнение примеров как приложений

#### 5.1. Самый простой пример - Hello, World (64-bit)

```
Omtest\Clwr\OmtestHelloWorld.exe
```

Печать строковых, целых и действительных чисел

```
Omtest\Clwr\OmtestFormats.exe
```

#### 5.2 Печать даты, времени и реализация задержек

```
Omtest\Clwr\OmtestDateTime
```

#### 5.3 Травы – обработка нештатных ситуаций

Простой Assert

```
Omtest\Clwr\OmtestMkTraps -trap a
```

Простой Halt

```
Omtest\Clwr\OmtestMkTraps -trap h
```

Деление на ноль

```
Omtest\Clwr\OmtestMkTraps -trap z
Обращение к памяти по нулевому указателю
Omtest\Clwr\OmtestMkTraps -trap p
```

#### 5.4 Работа с динамической памятью и сборка мусора

```
Omtest\Clwr\OmtestHeap
```

## 6 Сборка бинарников OmlSh

Предусловия.

Для компиляции и линковки в бинарники требуются средства разработки для языка C. В данном пакете эти средства не поставляются и представлены не будут: ни Visual Studio, ни MinGW, ни CMake. Вы можете устанавливать средства разработки C или модифицировать скрипты сборки под установленные уже средства на Вашем компьютере. В скриптах используется следующее:

- gcc, ar – для lwe (oFront-Windows-32bit),
- clang – для lwr (oFront-Windows-64bit).

```
lwe_compiler_tomake
```

Создает все исполняемые файлы 32-бит оболочки OmlSh

```
lwe_compiler_toclean
```

Удаляет все исполняемые файлы 32-бит оболочки OmlSh

```
lwr_compiler_tomake
```

Создает все исполняемые файлы 64-бит оболочки OmlSh

```
lwr_compiler_toclean
```

Удаляет все исполняемые файлы 64-бит оболочки OmlSh

## Как начать работу из Black Box

### 1 Инсталляция

#### 1.1. Предусловия.

**Установите LLVMT.dll из ~/Blwe в место расположения BlackBox.exe**

Oml использует LLVM 5.0 в виде библиотеки LLVMT.dll.

#### 2 Скомпилируйте доступ к LLVM:

```
^Q DevCompiler.CompileThis LlvmC LlvmForAArch64 LlvmForAMDGPU LlvmForARM
LlvmForBPF LlvmForHexagon LlvmForLanai LlvmForMips LlvmForMSP430 LlvmForNVPTX
LlvmForPowerPC LlvmForSparc LlvmForSystemZ LlvmForX86 LlvmForXCore LlvmNative
LlvmRefs
```

#### 3 Скомпилируйте модули ниже:

```
^Q DevCompiler.CompileThis OmcTarget OmcCRuntime OmcHooks OmcDialog OmcOPM OmcOPT
OmcOPU OmcOPB OmcOPS OmcOPP OmcDump OmcParams OmcOdcSource OmcTxtSource
OmcRuntimeStd OmcDialogStd OmcDialogConsole OmcCompiler OmcHostDialog OmcConsole
OmlOPG OmlOPL OmlOPF OmlOPC OmlOPV OmlBackEnd OmlCompiler OmlConsole
```

## 2 Компиляция примеров

#### 2.1. Скомпилируйте примеры для 32-bit:

```
^Q OmlCompiler.CompileThis +HostConLog OmtestHelloWorld: OmtestFormats:
OmtestDateTime: OmtestMkTraps: OmtestHeap:
Ожидаемый результат в каталоге ~/Omtest/Clwe/: OmtestHelloWorld.ll OmtestHelloWorld.bc
OmtestFormats.ll OmtestFormats.bc OmtestDateTime.ll OmtestDateTime.bc OmtestMkTraps.ll
OmtestMkTraps.bc OmtestHeap.ll OmtestHeap.bc
```

#### 2.2. Скомпилируйте примеры для 64-bit:

```
^Q OmlCompiler.CompileThis -bits 64 +HostConLog OmtestHelloWorld: OmtestFormats:
OmtestDateTime: OmtestMkTraps: OmtestHeap:
```

Ожидаемый результат в каталоге ~/Omtest/Ciwr/: OmtestHelloWorld.ll OmtestHelloWorld.bc  
OmtestFormats.ll OmtestFormats.bc OmtestDateTime.ll OmtestDateTime.bc OmtestMkTraps.ll  
OmtestMkTraps.bc OmtestHeap.ll OmtestHeap.bc

### 3 Авто-компиляция Oml Shell

#### 3.1 Скомпилируйте 32-битный доступ к LLVM:

```
^Q OmlCompiler.CompileThis LlvmC LlvmForAArch64 LlvmForAMDGPU LlvmForARM  
LlvmForBPF LlvmForHexagon LlvmForLanai LlvmForMips LlvmForMSP430 LlvmForNVPTX  
LlvmForPowerPC LlvmForSparc LlvmForSystemZ LlvmForX86 LlvmForXCore LlvmNative  
LlvmRefs
```

#### 3.2 Авто-компиляция 32-битной консоли OmlSh.

```
^Q OmlCompiler.CompileThis -System Syslwe -Host Hostlwe -options lb -directories  
Llwe LlvmNative OmcTarget OmcCRuntime OmcDialog OmcHooks OmcTxtSource  
OmcDialogConsole OmcRuntimeStd OmcOPM OmcOPT OmcOPB OmcOPU OmcOPS OmcOPP  
OmcParams OmcConsole OmcDump OmcShell OmlOPG OmlOPL OmlOPF OmlOPC OmlOPV  
OmlBackEnd OmlLoader OmlSh:
```

#### 3.3 Скомпилируйте 64-битный доступ к LLVM.

```
^Q OmlCompiler.CompileThis -bits 64 LlvmC LlvmForAArch64 LlvmForAMDGPU  
LlvmForARM LlvmForBPF LlvmForHexagon LlvmForLanai LlvmForMips LlvmForMSP430  
LlvmForNVPTX LlvmForPowerPC LlvmForSparc LlvmForSystemZ LlvmForX86 LlvmForXCore  
LlvmNative LlvmRefs
```

#### 3.4 Авто-компиляция 64-битной консоли OmlSh.

```
^Q OmlCompiler.CompileThis -System Syslwr -Host Hostlwr -options lb -directories  
Llwr -bits 64 LlvmNative OmcTarget OmcCRuntime OmcDialog OmcHooks OmcTxtSource  
OmcDialogConsole OmcRuntimeStd OmcOPM OmcOPT OmcOPB OmcOPU OmcOPS OmcOPP  
OmcParams OmcConsole OmcDump OmcShell OmlOPG OmlOPL OmlOPF OmlOPC OmlOPV  
OmlBackEnd OmlLoader OmlSh:
```

### 8 Выгрузка компилятора Oml

```
^Q DevDebug.UnloadThis OmlCompiler OmlBackEnd OmlOPV OmlOPC OmlOPF OmlOPL OmlOPG  
OmcCompiler OmcDialogStd OmcRuntimeStd OmcOdcSource OmcParams OmcDump OmcOPP  
OmcOPS OmcOPU OmcOPB OmcOPT OmcOPM OmcDialog OmcHooks OmcCRuntime OmcTarget
```

### 9 Журнал изменений

may 2019 original MultiOberon pre-version 0.8 released

Use it and enjoy! - Ўъsalos y disfr talos! - Bonne utilisation - Приятного использования - Powodzenia - Viel  
SpaЯ

Дагаев Дмитрий Викторович  
[dvdagaev@yahoo.com](mailto:dvdagaev@yahoo.com)