

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«СКАДИ»

(ООО «СКАДИ»)

УТВЕРЖДАЮ

**КОМПИЛЯТОР ЯЗЫКА МУЛЬТИОБЕРОН СО СМЕННЫМИ
БЭКЕНДАМИ**

Руководство системного программиста

34185873.425510.003.32.М

(На 25 листах)

2023

ООО «СКАДИ»	Компилятор языка МультиОберон со сменными бэкендами Руководство системного программиста	Версия 1.3
-------------	--	---------------

РАЗРАБОТАЛ

34185873.425510.003.32.M	Стр. 2 из 25
--------------------------	--------------

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ О ПО.....	4
1.1 НАИМЕНОВАНИЕ.....	4
1.2 НАЗНАЧЕНИЕ И ФУНКЦИИ.....	4
1.3 НЕОБХОДИМЫЕ ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА	5
2 СТРУКТУРА ПО	6
2.1 Состав ПО	6
2.2 СВЯЗИ СОСТАВНЫХ ЧАСТЕЙ ПО.....	6
3 НАСТРОЙКА ПРОГРАММЫ.....	7
3.1 ИНСТАЛЛЯЦИЯ.....	7
3.1.1 Инсталляция для ОС Astra Linux, ОС Linux	7
3.1.2 Установка рабочего каталога для Linux	7
3.1.3 Инсталляция для ОС Windows	7
3.1.4 Установка рабочего каталога для Windows	8
3.2 ОПИСАНИЕ КОНФИГУРАЦИОННЫХ ФАЙЛОВ.....	8
3.3 ИСПОЛЬЗОВАНИЕ КОНСОЛЬНОГО РЕЖИМА	8
4 ПРОВЕРКА ПРОГРАММЫ	9
4.1 КОМПИЛЯЦИЯ ПРИМЕРОВ	9
4.2 ВЫПОЛНЕНИЕ ПРИМЕРОВ	9
5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ	11
5.1 КОМПИЛЯЦИЯ КОМПИЛЯТОРОВ И ОБОЛОЧЕК	11
6 СООБЩЕНИЯ СИСТЕМНОМУ ОПЕРАТОРУ	12
ПРИЛОЖЕНИЯ	13
Приложение 1 – Состав каталогов МультиОберон	13
Приложение 2 – Опции конфигурационных файлов	15
Приложение 3 – Опции командной строки	16
Приложение 4 – Перечень команд компилятора.....	18
Приложение 5 – Сообщения компилятора об ошибках	19
ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ.....	24
ЛИСТ ИЗМЕНЕНИЙ.....	25

ООО «СКАДИ»	Компилятор языка МультиОберон со сменными бэкендами Руководство системного программиста	Версия 1.3
-------------	--	---------------

1 ОБЩИЕ СВЕДЕНИЯ О ПО

1.1 НАИМЕНОВАНИЕ

Наименование изделия — Компилятор языка МультиОберон со сменными бэкендами «МультиОберон 1.3».

Сокращенное наименование: «МультиОберон».

Программа «МультиОберон» («MultiOberon») зарегистрирована в реестре программ для ЭВМ №2022669920 от 26.10.2022.

Наименование документа 34185873.425510.003.32.М, обозначение по ГОСТ 34.201-89.

МультиОберон является базовым инструментом сборки приложений «СКАДИ» и «СКЗА». «СКАДИ» представляет собой программную платформу средств комплексной автоматизации и диагностики. «СКЗА» представляет собой средства коммуникации зонной архитектуры. Описание языка МультиОберон приведено в документе 34185873.425510.001.35.М «Программная платформа АСУ ТП и диагностики. Описание языка МультиОберон»

1.2 НАЗНАЧЕНИЕ И ФУНКЦИИ

МультиОберон предназначен для кроссплатформенной сборки программного обеспечения на диалектах языка Оберон. Программа применяется для сборки систем, работающих в режиме онлайн с повышенными требованиями в части функциональной безопасности.

МультиОберон был предложен для реализации механизма редуцирования диалектов Оберона к минималистическим подмножествам:

Компонентный Паскаль->Оберон-2->Оберон-07->Оберон.

МультиОберон представляет собой масштабируемую технологию на основе систем ограничений с начальной точкой в виде синтаксиса Компонентного Паскаля.

МультиОберон позволяет устанавливать систему ограничений на требуемые диалекты Оберона. Кроме этого, возможна установка дополнительных ограничений соразмерно требованиям проекта. Компилятор обеспечивает гарантии соответствия кода системе ограничений.

МультиОберон позволяет осуществлять кроссплатформенную сборку приложений, работающую на разных аппаратных архитектурах и операционных системах. МультиОберон это компилятор языка Оберон с тремя различными бэкендами:

- Генератором нативного кода x86 для системы BlackBox;
- Транслятором Ofront в язык C;
- Генератором кода LLVM.

МультиОберон содержит средства трансляции в ANSI C для переноса как компилятора МультиОберона, так и разработанного с его помощью ПО на перспективные отечественные платформы и операционные системы. Для каждой вновь появившейся отечественной программно-аппаратной среды компилятор и ПО на МультиОбероне переносится по стандартной схеме через транслятор в ANSI C.

МультиОберон поддерживает полностью поддерживает концепцию модульного программирования. Каждый программный модуль является единицей хранения, компиляции, загрузки и выполнения.

МультиОберон запускается как в консольном режиме, так и из среды BlackBox.

МультиОберон предназначен для мульти-платформенной разработки. Процедуры одних модулей не зависят от платфо-специфических характеристик, а для других зависимость существует. Эти характеристики включают в себя интерфейсы библиотечных функций операционных систем, различные структуры данных для 32 и 64-битных реализаций.

МультиОберон имеет средства для кросс-платформенной разработки. Приложения на платформе с работающим компилятором могут переноситься на другие целевые платформы. В конфигурационных файлах МультиОберона содержится список настраиваемых опций. Среди указанных опций есть опции с указанием других целевых платформ.

МультиОберон может применяться на всех этапах разработки ПО для критически важных систем как системное программное средство.

1.3 НЕОБХОДИМЫЕ ТЕХНИЧЕСКИЕ И ПРОГРАММНЫЕ СРЕДСТВА

Для функционирования МультиОберона требуется наличие операционной системы семейств Linux, Raspberry Pi OS, Windows.

ПО МультиОберон предназначено для работы с операционным программным и аппаратным обеспечением, согласно таблице 1.

Архитектура+ОС	Разрядность	linux	Raspberry Pi OS	windows
X86	32	Binue	--	Binwe
X64	64	Binur	--	Binwr
ArmV71	32	--	Binu4	--
Aarch64	64	--	--	--

Таблица 1 – Поддерживаемые ОС и архитектуры

При выборе доверенных операционных систем рекомендуется использование ОС Astra Linux, начиная с версии 1.6. При выборе Windows рекомендуется использование отечественных аналогов.

2 СТРУКТУРА ПО

2.1 СОСТАВ ПО

МультиОберон запускается как в консольном режиме, так и из среды BlackBox. Для консольного и графического режимов реализованы варианты компиляторов:

- ombs и OmbCompiler – для нативного бэкенда;
- omfc и OmfCompiler – для транслятора в C;
- omfc и OmlCompiler – для бэкенда LLVM.

Для консольного режима реализованы оболочки, позволяющие осуществлять динамическую загрузку модулей:

- ombsh - для нативного бэкенда;
- omfsh – для транслятора в C;
- omlsh – для бэкенда LLVM.

Дополнительно для бэкенда LLVM используется статический компилятор llc и библиотека LLVMT.so.

Состав и содержание каталогов ПО представлены в приложении 1.

2.2 СВЯЗИ СОСТАВНЫХ ЧАСТЕЙ ПО

ПО МультиОберон включает в себя модули функциональности компиляции, входящие в состав всех реализаций компиляторов:

- интерфейс доступа к файлам документов .ods и текстовым .mob файлам;
- фронтенд компилятора в составе сканнера, парсера, построителя синтаксических деревьев;
- анализатора синтаксических ошибок;
- системы контроля ограничений.

ПО МультиОберон также включает в себя модули общей функциональности, входящие в состав все реализаций компиляторов и оболочек:

- механизмов поддержки динамической загрузки;
- механизмов поиска точки входа и выполнения динамически загруженных модулей;
- средств позиционирования динамических модулей в каталогах проекта;
- средств доступа к данным конфигурационных файлов.

Приложения МультиОберона, различающиеся бэкендами и видами оболочек, связаны наличием в каждом указанной общей функциональности.

3 НАСТРОЙКА ПРОГРАММЫ

3.1 ИНСТАЛЛЯЦИЯ

3.1.1 Инсталляция для ОС Astra Linux, ОС Linux

Разместить пакет ПО МультиОберон для соответствующей архитектуры в корневой каталог:

- multioberon_i386_1.3.0.deb для архитектуры X86;
- multioberon_amd64_1.3.0.deb для архитектуры X64;
- multioberon_armhf_1.3.0.deb для архитектуры ArmV71;
- multioberon_arm64_1.3.0.deb для архитектуры Aarch64.

Установить пакет ПО МультиОберон средствами ОС:

```
sudo apt install multioberon_i386_1.3.0.deb      (* X86 *)
sudo apt install multioberon_amd64_1.3.0.deb    (* X64 *)
sudo apt install multioberon_armhf_1.3.0.deb    (* ArmV71 *)
sudo apt install multioberon_arm64_1.3.0.deb    (* Aarch64 *)
```

3.1.2 Установка рабочего каталога для Linux

Для сборки приложений и программных модулей компилятором МультиОберон требуется создание рабочего каталога. Создать рабочий каталог нужно командами ОС:

```
>mkdir MyDir
>cd MyDir
```

Для корректной работы и доступа к системным модулям требуется установить рабочий каталог. Установить рабочий каталог можно вызовом соответствующей утилиты инсталлятора.

Для ArmV71 установка рабочего каталога выполняется следующим образом.

```
>/bin/sh /usr/local/bin/multioberon/omc4_install
```

Для Aarch64 установка рабочего каталога выполняется следующим образом.

```
>/bin/sh /usr/local/bin/multioberon/omc8_install
```

Для X86 установка рабочего каталога доступ к системе BlackBox при наличии установленной BlackBox. Установка рабочего каталога с BlackBox выполняется следующим образом.

```
>/bin/sh /usr/local/bin/multioberon/omce_install <path-to-blackbox>; (*
например ${НОМЕ}/bb *)
```

Установка рабочего каталога без BlackBox выполняется следующим образом.

```
>/bin/sh /usr/local/bin/multioberon/omce_install
```

Установка рабочего каталога для X64 без BlackBox выполняется следующим образом.

```
>/bin/sh /usr/local/bin/multioberon/omcr_install
```

Следует прописать путь PATH= /usr/local/bin/multioberon/Binu[arch] при необходимости.

3.1.3 Инсталляция для ОС Windows

Инсталляция 32-битной версии состоит в следующем. Разместить инсталлятор ПО MultiOberon_1_2_0_x86.msi для 32-bit на жестком диске. Установить ПО МультиОберон запуском инсталлятора.

Инсталляция 64-битной версии состоит в следующем. Разместить инсталлятор ПО MultiOberon_1_2_0_x64.msi для 64-bit на жестком диске. Установить ПО МультиОберон запуском инсталлятора.

3.1.4 Установка рабочего каталога для Windows

Для сборки приложений и программных модулей компилятором МультиОберон требуется создание рабочего каталога. Создать рабочий каталог нужно командами ОС:

```
>mkdir MyDir  
>cd MyDir
```

Для корректной работы и доступа к системным модулям требуется установить рабочий каталог. Установить рабочий каталог можно вызовом соответствующей утилиты инсталлятора.

Для X86 установка рабочего каталога доступ к системе BlackBox при наличии установленной BlackBox. Установка рабочего каталога с BlackBox выполняется следующим образом.

```
>omce_install <path-to-blackbox>; (* например "c:\Program Files  
(x86)\BlackBox Component Builder 1.7.2" *)
```

Установка рабочего каталога без BlackBox выполняется следующим образом.

```
>omce_install
```

Используйте /clang для работы с clang компилятором/линкером. По умолчанию используется gcc.

```
>omce_install /clang
```

Установка рабочего каталога для X64 без BlackBox выполняется следующим образом.

```
>omcr_install
```

Следует прописать путь доступа к каталогам инсталляции "c:\Program Files (x86)\MultiOberon" и "c:\Program Files\MultiOberon".

3.2 ОПИСАНИЕ КОНФИГУРАЦИОННЫХ ФАЙЛОВ

Конфигурации консоли представляет собой текстовые файлы с расширением .cfg вида:

- Omb.cfg – файл для консольного компилятора ombc;
- Omf.cfg – файл для консольного компилятора omfc;
- Oml.cfg – файл для консольного компилятора omfc.

Конфигурации среды BlackBox представляют собой .odc файлы документов с именами:

- Omb/Rsrc/Strings.odc – документ компилятора OmbCompiler;
- Omf/Rsrc/Strings.odc – документ компилятора OmfCompiler;
- Oml/Rsrc/Strings.odc – документ компилятора OmlCompiler.

Опции конфигурационных файлов описаны в приложении 2.

3.3 ИСПОЛЬЗОВАНИЕ КОНСОЛЬНОГО РЕЖИМА

Команды консоли состоят из имени приложения компилятора, имени команды и опций командной строки. Опции командной строки состоят из выражений типа –option для одинарных и –option value для двойных.

Приложение 3 содержит перечень опций командной строки. Приложение 4 содержит перечень команд компилятора.

Линковка используется для создания исполняемого файла для последующего запуска.

4 ПРОВЕРКА ПРОГРАММЫ

4.1 КОМПИЛЯЦИЯ ПРИМЕРОВ

Скрипт для компиляции и линковки всех примеров устанавливается при инсталляции в подкаталог В.

```
bwe_tomake.bat      (* Windows, X86 *)
bue_tomake.sh       (* Linux, X86 *)
```

Вспомогательный скрипт для удаления всех приложений с примерами находится там же.

```
bwe_toclean
bue_toclean.sh
```

Ниже приведен набор команд компиляции всех примеров, включенных в инсталляционный пакет. Вариант для ОС Windows.

```
?Pathname?\Binwe\ombc co -odc OmtestHelloWorld OmtestFormats OmtestDateTime
OmtestMkTraps OmtestHeap
?Pathname?\Binwe\ombc link -r OmtestHelloWorld
?Pathname?\Binwe\ombc link -r OmtestFormats
?Pathname?\Binwe\ombc link -r OmtestDateTime
?Pathname?\Binwe\ombc link -r OmtestMkTraps
?Pathname?\Binwe\ombc link -r OmtestHeap
?Pathname?\Binue\ombc co -odc OmtestHelloWorld OmtestFormats OmtestDateTime
OmtestMkTraps OmtestHeap
```

Вариант для ОС Linux.

```
mkdir -p Omtest/Cbue
?Pathname?\Binue\ombc link -r OmtestHelloWorld
?Pathname?\Binue\ombc link -r OmtestFormats
?Pathname?\Binue\ombc link -r OmtestDateTime
?Pathname?\Binue\ombc link -r OmtestMkTraps
?Pathname?\Binue\ombc link -r OmtestHeap
```

Первая команда списка компилирует все файлы примеров. Линковка создает требуемый набор приложений. Если мы опускаем `-r` опцию рекурсивной работы, команда линковщика должна быть аналогично следующей:

```
?Pathname?\Binwe\ombsh li $+Kernel Log Math Strings OStrings OLog HostConLog
Runner OmtestHelloWorld
?Pathname?\Binue\ombsh li $+Kernel Log Math Strings OStrings OLog HostConLog
Runner OmtestHelloWorld
```

В результате сборки появляются исполняемые файлы для всех примеров.

4.2 ВЫПОЛНЕНИЕ ПРИМЕРОВ

Простейший Hello, World запускается именем приложения без параметров.

```
>Omtest\Cbwe\OmtestHelloWorld
>Omtest/Cbue/OmtestHelloWorld
Hello, World
```

Форматирование строки, целых и действительных чисел запускается приложением `OmtestFormats`.

```
>Omtest\Cbwe\OmtestFormats
>Omtest/Cbue/OmtestFormats
```

Дата, время и печать текущих значений задержки осуществляется приложением OmtestDateTime.

```
>Omtest\Cbwe\OmtestDateTime  
>Omtest/Cbue/OmtestDateTime
```

Обработка трапов рантаймом приводит к аварийному завершению работы приложения OmtestMkTraps с выдачей на консоль диагностики ошибки и стека вызванных процедур.

Простой Assert задается параметром a.

```
>Omtest\Cbwe\OmtestMkTraps -trap a  
>Omtest/Cbue/OmtestKmTraps -trap a
```

Простой Halt задается параметром h.

```
>Omtest\Cbwe\OmtestMkTraps -trap h  
>Omtest/Cbue/OmtestKmTraps -trap h
```

Деление на ноль задается параметром z.

```
>Omtest\Cbwe\OmtestMkTraps -trap z  
>Omtest/Cbue/OmtestKmTraps -trap z
```

Разыменованное нулевого указателя задается параметром p.

```
>Omtest\Cbwe\OmtestMkTraps -trap p  
>Omtest/Cbue/OmtestKmTraps -trap p
```

Динамическая память и сборка мусора в интенсивном режиме осуществляется приложением OmtestHeap.

```
>Omtest\Cbwe\OmtestHeap  
>Omtest/Cbue/OmtestHeap
```

5 ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

5.1 КОМПИЛЯЦИЯ КОМПИЛЯТОРОВ И ОБОЛОЧЕК

Скрипт для компилирования компилятора ombc устанавливается при инсталляции в подкаталог В.

```
B\bwe_compiler_tomake.bat
B/bue_compiler_tomake.sh
```

Скрипт для компилирования минимальной оболочки ombsh представлен ниже:

```
B\bwe_sh_tomake.bat
B/bue_sh_tomake.sh
```

Скрипт для само компиляции omfc устанавливается при инсталляции в подкаталог В.

```
B\fwe_compiler_tomake.bat      (* X86 Windows *)
B\fwr_compiler_tomake.bat      (* X64 Windows *)
B/fue_compiler_tomake.sh       (* X86 *)
B/fur_compiler_tomake.sh       (* X64 *)
B/fu4_compiler_tomake.sh       (* ArmV71 *)
B/fu8_compiler_tomake.sh       (* Aarch64 *)
```

Скрипт для компиляции минимальной оболочки omfsh следующий:

```
B\fwe_sh_tomake.bat
B/fue_sh_tomake.sh
```

Скрипт для само компиляции omfc следующий:

```
B\lwe_compiler_tomake.bat      (* X86 Windows *)
B\lwr_compiler_tomake.bat      (* X64 Windows *)
B/lue_compiler_tomake.sh       (* X86 *)
B/lur_compiler_tomake.sh       (* X64 *)
B/lu4_compiler_tomake.sh       (* ArmV71 *)
B/lu8_compiler_tomake.sh       (* Aarch64 *)
```

Скрипт для компиляции минимальной 64-битной оболочки omlsh следующий:

```
B\lwr_sh_tomake.bat
B/lur_sh_tomake.sh
```

6 СООБЩЕНИЯ СИСТЕМНОМУ ОПЕРАТОРУ

Задание уровня трассировки осуществляется опцией `-tl`. Возможные значения уровня трассировки приведены в таблице 3-2. Значение по умолчанию равно 0.

Задание уровня печати осуществляется опцией `-pl`. Возможные значения уровня печати приведены в таблице 3-3. Значение по умолчанию равно 1.

Перечень сообщений компилятора об ошибках представлен в приложении 5.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1 – СОСТАВ КАТАЛОГОВ МУЛЬТИОБЕРОН

МультиОберон состоит из исполняемых файлов приложений и списка программных подсистем. Каталоги бинарных исполняемых файлов соответствуют правилу Bin[os][arch], таблица 1-1.

Архитектура+ОС	windows	unix (linux, ...)
X86	Binwe	Binue
X64	Binwr	Binur
ArmV71	--	Binu4
Aarch64	--	Binu8

Таблица 1-1 – Именование каталогов бинарных исполняемых файлов

Бинарные приложения состоят из компиляторов и оболочек (shells). Компиляторы МультиОберона включают в себя полную поддержку функций компиляции и следуют правилу om[backend]c. Оболочки МультиОберона поддерживают только динамическую загрузку и исполнение, файлы соответствуют правилу om[backend]sh. Имена бинарных приложений приведены в таблице 1-2.

	BlackBox	Ofront	LLVM
компилятор	ombc	omfc	omlc
оболочка	ombsh	omfsh	omlsh

Таблица 1-2 – Имена бинарных приложений

Программные подсистемы структурируются по по правилам подсистем BlackBox. Бэкенд следует правилу Om[backend]. Третья буква означает следующие варианты:

- Omc это компилятор и консоль МультиОберона;
- Omb является реализацией компилятора для бэкенда BlackBox;
- Omf является реализацией компилятора для бэкенда Ofront;
- Oml является реализацией компилятора для бэкенда LLVM. Использует библиотеку, подготовленную из LLVM 5.0.

Каждая подсистема имеет каталог Mod для исходников, Doci для документов, каталоги с кодовыми и символьными файлами.

Каталоги кодовых файлов следуют правилу C[backend][os][arch], таблица 1-3.

Архитектура+ОС	windows	Unix (Linux, ...)
X86 BlackBox portable	Code	Code
X86 Omb-specific	Cbwe	Cbue
X86 Ofront	Cfwe	Cfue
X64 Ofront	Cfwr	Cfur
X86 LLVM	Clwe	Clue
X64 LLVM	Clwr	Clur
ArmV71 Ofront	--	Cfu4
Aarch64 Ofront	--	Cfu8
ArmV71 LLVM	--	Clu4
Aarch64 LLVM	--	Clu8

Таблица 1-3 - Каталоги кодовых файлов

Программные подсистемы структурируются по правилам подсистем BlackBox. Символьные файлы и файлы использования расположены в символьных каталогах. Каталоги символьных файлов следуют правилу S[backend][os][arch], таблица 1-4.

Архитектура+ОС	windows	Unix (Linux, ...)
X86 BlackBox portable	Sym	Sym
X86 Omb-specific	Sbwe	Sbue
X86 OFront	Sfwe	Sfue
X64 OFront	Sfwr	Sfur
X86 LLVM	Slwe	Slue
X64 LLVM	Slwr	Slur
ArmV71 OFront	--	Sfu4
Aarch64 OFront	--	Sfu8
ArmV71 LLVM	--	Slu4
Aarch64 LLVM	--	Slu8

Таблица 1-4 - Каталоги символьных файлов

ПРИЛОЖЕНИЕ 2 – ОПЦИИ КОНФИГУРАЦИОННЫХ ФАЙЛОВ

Конфигурация состоит из одиночных параметров в виде option=value. Пример одиночного параметра.

```
spath=/home/ddag/mobdev/bb
```

Конфигурационные параметры приведены в таблице 2-1.

Опция	Описание	Примечание
path	Путь к импортируемым подсистемам (к каталогу модулей)	Не задано – означает путь к рабочему каталогу
spath	Теневой путь к импортируемым подсистемам	Здесь задаем путь к каталогу BlackBox
cc	Имя внешнего компилятора	gcc clang
lnk	Имя внешнего линкера	gcc clang
gcc_opt	Опции компиляции для gcc	Например, -O3 -fshort-wchar -fPIC
gcc_lnkopt	Опции линкера для gcc	Например, -O2 -lm -ldl
gcc_32	Опция компиляции для режима 32 бит	-m32
gcc_64	Опция компиляции для режима 64 бит	-m64
gcc_lnk32	Опция линкера для режима 32 бит	-m32
gcc_lnk64	Опция линкера для режима 64 бит	-m64
clang_opt	Опции компиляции для clang	Например, -O2
clang_lnkopt	Опции линкера для clang	Например, -O2 -ldl
clang_32	Опция компиляции для режима 32 бит	-m32
clang_64	Опция компиляции для режима 64 бит	-m64
clang_lnk32	Опция линкера для режима 32 бит	-m32
clang_lnk64	Опция линкера для режима 64 бит	-m64
llc	Приложение llc с полным путем	/usr/local/bin/multioberon/Binue/llc
llc_opt	Опция llc-компиляции	-O0 -march=x86

Таблица 2-1 – Список конфигурационных параметров

ПРИЛОЖЕНИЕ 3 – ОПЦИИ КОМАНДНОЙ СТРОКИ

МультиОберон использует следующие опции командной строки, таблица 3-1.

Опция	Описание	Возможные значения
-odc	Выбор файла .odc вместо .mob	-
-32	32-битный режим	-
-64	64-битный режим	-
-os	Выбор ОС для кросс-компиляции	Windows Linux
-r	Использование рекурсивного автоматического импорта	-
-h	Не включать HostConLog автоматически	-
-n	Перекомпилировать только новые файлы	-
-tl	Уровень трассировки	Таблица 2-2
-pl	Уровень печати	Таблица 2-3
-ht	Тип обработчика ошибок	Таблица 2-4
-wsd	Разрешение писать в каталог System	-
-o	Имя выходного файла	Имя файла
-path	Путь к импортируемым подсистемам	Каталоги для всех приложений
-spath	Теневой путь к импортируемым подсистемам	Каталоги для всех приложений

Таблица 3-1 – Перечень опций командной строки

Возможные значения уровня трассировки приведены в таблице 3-2. Значение по умолчанию равно 0.

Значение	Имя	Описание
0*	TRACE_LEVEL_NONE	Нет трассировки
1	TRACE_LEVEL_MIN	Печать сообщений об ошибках и важных сообщений
2	TRACE_LEVEL_NORM	Печать основных сообщений и дампов
3	TRACE_LEVEL_DETAILED	Детальная трассировка

Таблица 3-2 – Значения уровней трассировки

Возможные значения уровня печати приведены в таблице 3-3. Значение по умолчанию равно 1.

Значение	Имя	Описание
0	PRINT_LEVEL_NONE	Печать только сообщений об ошибках
1*	PRINT_LEVEL_DEFAULT	Печать выполняемой операции и сообщений об ошибках
2	PRINT_LEVEL_NORM	Подробная печать операции и сообщений об оибках
3	PRINT_LEVEL_DETAILED	Детальная печать

Таблица 3-3 – Значения уровней печати

Возможные значения типа обработчика ошибок приведены в таблице 3-4. Значение по умолчанию равно 0.

Значение	Имя	Описание
0*	HANDLER_TYPE_AUTO	Выбор обработчика по умолчанию
1	HANDLER_TYPE_DLINK	Обработчик с использованием структур DLink
2	HANDLER_TYPE_FP	Обработчик с использованием указателя фрейма FP
3	HANDLER_TYPE_STACK	Обработчик с алгоритмом анализатора стека

Таблица 3-4 – Значения типа обработчика ошибок

ПРИЛОЖЕНИЕ 4 – ПЕРЕЧЕНЬ КОМАНД КОМПИЛЯТОРА

Перечень команд компилятора МультиОберон представлен в таблице 4-1.

Из консоли	Из BlackBox	Описание
co[mpile]	Om?Compiler.CompileThis	Компиляция списка программных модулей
r[un]	--	Запуск на выполнение
ex[ecute]	--	Выполнение после компиляции и линковки
li[nk]	Om?Linker.LinkExe	Линковка исполняемого файла
tr[averse]	Om?Linker.TraverseModule	Рекурсивная печать списков импорта модуля
cl[ear]	Om?Linker.ClearFiles	Удаление сгенеренных файлов
bu[ild]	Om?Linker.BuildFiles	Сборка промежуточных файлов (кроме Omb)
te[st]	Om?Linker.TestModules	Запуск тестов модулей
be[nch]	Om?Linker.BenchModules	Прогон временных характеристик

Таблица 4-1 – Команды компилятора МультиОберон

ПРИЛОЖЕНИЕ 5 – СООБЩЕНИЯ КОМПИЛЯТОРА ОБ ОШИБКАХ

Перечень сообщений об ошибках фронтенда компилятора приведен ниже. Каждому коду ошибки соответствует строка сообщения.

```
eMsg[0] := 'undeclared identifier';
eMsg[1] := 'multiply defined identifier';
eMsg[2] := 'illegal character in number';
eMsg[3] := 'illegal character in string';
eMsg[4] := 'identifier does not match procedure name';
eMsg[5] := 'comment not closed';
eMsg[9] := '"=' expected';
eMsg[12] := 'type definition starts with incorrect symbol';
eMsg[13] := 'factor starts with incorrect symbol';
eMsg[14] := 'statement starts with incorrect symbol';
eMsg[15] := 'declaration followed by incorrect symbol';
eMsg[16] := 'MODULE expected';
eMsg[19] := '."' missing';
eMsg[20] := '",' missing';
eMsg[21] := ':" missing';
eMsg[23] := ')" missing';
eMsg[24] := ']" missing';
eMsg[25] := '}" missing';
eMsg[26] := 'OF missing';
eMsg[27] := 'THEN missing';
eMsg[28] := 'DO missing';
eMsg[29] := 'TO missing';
eMsg[35] := '",' or OF expected';
eMsg[36] := 'CONST, TYPE, VAR, PROCEDURE, BEGIN, or END missing';
eMsg[37] := 'PROCEDURE, BEGIN, or END missing';
eMsg[38] := 'BEGIN or END missing';
eMsg[40] := '"(' missing';
eMsg[41] := 'illegally marked identifier';
eMsg[42] := 'constant not an integer';
eMsg[43] := 'UNTIL missing';
eMsg[44] := '":=" missing';
eMsg[46] := 'EXIT not within loop statement';
eMsg[47] := 'string expected';
eMsg[48] := 'identifier expected';
eMsg[49] := '";" missing';
eMsg[50] := 'expression should be constant';
eMsg[51] := 'END missing';
eMsg[52] := 'identifier does not denote a type';
eMsg[53] := 'identifier does not denote a record type';
eMsg[54] := 'result type of procedure is not a basic type';
eMsg[55] := 'procedure call of a function';
eMsg[56] := 'assignment to non-variable';
eMsg[57] := 'pointer not bound to record or array type';
eMsg[58] := 'recursive type definition';
eMsg[59] := 'illegal open array parameter';
eMsg[60] := 'wrong type of case label';
eMsg[61] := 'inadmissible type of case label';
eMsg[62] := 'case label defined more than once';
eMsg[63] := 'illegal value of constant';
eMsg[64] := 'more actual than formal parameters';
eMsg[65] := 'fewer actual than formal parameters';
eMsg[66] := 'element types of actual array and formal open array differ';
eMsg[67] := 'actual parameter corresponding to open array is not an
array';
eMsg[68] := 'control variable must be integer';
eMsg[69] := 'parameter must be an integer constant';
```

```
eMsg[70] := 'pointer or VAR / IN record required as formal receiver';
eMsg[71] := 'pointer expected as actual receiver';
eMsg[72] := 'procedure must be bound to a record of the same scope';
eMsg[73] := 'procedure must have level 0';
eMsg[74] := 'procedure unknown in base type';
eMsg[75] := 'invalid call of base procedure';
eMsg[76] := 'this variable (field) is read only';
eMsg[77] := 'object is not a record';
eMsg[78] := 'dereferenced object is not a variable';
eMsg[79] := 'indexed object is not a variable';
eMsg[80] := 'index expression is not an integer';
eMsg[81] := 'index out of specified bounds';
eMsg[82] := 'indexed variable is not an array';
eMsg[83] := 'undefined record field';
eMsg[84] := 'dereferenced variable is not a pointer';
eMsg[85] := 'guard or test type is not an extension of variable type';
eMsg[86] := 'guard or testtype is not a pointer';
eMsg[87] := 'guarded or tested variable is neither a pointer nor a VAR-
or IN-parameter record';
eMsg[88] := 'open array not allowed as variable, record field or array
element';
eMsg[89] := 'ANYRECORD may not be allocated';
eMsg[90] := 'dereferenced variable is not a character array';
eMsg[92] := 'operand of IN not an integer, or not a set';
eMsg[93] := 'set element type is not an integer';
eMsg[94] := 'operand of & is not of type BOOLEAN';
eMsg[95] := 'operand of OR is not of type BOOLEAN';
eMsg[96] := 'operand not applicable to (unary) +';
eMsg[97] := 'operand not applicable to (unary) -';
eMsg[98] := 'operand of ~ is not of type BOOLEAN';
eMsg[99] := 'ASSERT fault';
eMsg[100] := 'incompatible operands of dyadic operator';
eMsg[101] := 'operand type inapplicable to *';
eMsg[102] := 'operand type inapplicable to /';
eMsg[103] := 'operand type inapplicable to DIV';
eMsg[104] := 'operand type inapplicable to MOD';
eMsg[105] := 'operand type inapplicable to +';
eMsg[106] := 'operand type inapplicable to -';
eMsg[107] := 'operand type inapplicable to = or #';
eMsg[108] := 'operand type inapplicable to relation';
eMsg[109] := 'overriding method must be exported';
eMsg[110] := 'operand is not a type';
eMsg[111] := 'operand inapplicable to (this) function';
eMsg[112] := 'operand is not a variable';
eMsg[113] := 'incompatible assignment';
eMsg[114] := 'string too long to be assigned';
eMsg[115] := 'parameter does not match';
eMsg[116] := 'number of parameters does not match';
eMsg[117] := 'result type does not match';
eMsg[118] := 'export mark does not match with forward declaration';
eMsg[119] := 'redefinition textually precedes procedure bound to base
type';
eMsg[120] := 'type of expression following IF, WHILE, UNTIL or ASSERT is
not BOOLEAN';
eMsg[121] := 'called object is not a procedure';
eMsg[122] := 'actual VAR-, IN-, or OUT-parameter is not a variable';
eMsg[123] := 'type is not identical with that of formal VAR-, IN-, or
OUT-parameter';
eMsg[124] := 'type of result expression differs from that of procedure';
eMsg[125] := 'type of case expression is neither INTEGER nor CHAR';
```

```
eMsg[126] := 'this expression cannot be a type or a procedure';
eMsg[127] := 'illegal use of object';
eMsg[128] := 'unsatisfied forward reference';
eMsg[129] := 'unsatisfied forward procedure';
eMsg[130] := 'WITH clause does not specify a variable';
eMsg[131] := 'LEN not applied to array';
eMsg[132] := 'dimension in LEN too large or negative';
eMsg[133] := 'function without RETURN';
eMsg[135] := 'SYSTEM not imported';
eMsg[136] := 'LEN applied to untagged array';
eMsg[137] := 'unknown array length';
eMsg[138] := 'NEW not allowed for untagged structures';
eMsg[139] := 'Test applied to untagged record';
eMsg[140] := 'untagged receiver';
eMsg[141] := 'SYSTEM.NEW not implemented';
eMsg[142] := 'tagged structures not allowed for NIL compatible var
parameters';
eMsg[143] := 'tagged pointer not allowed in untagged structure';
eMsg[144] := 'no pointers allowed in BYTES argument';
eMsg[145] := 'untagged open array not allowed as value parameter';
eMsg[150] := 'key inconsistency of imported module';
eMsg[151] := 'incorrect symbol file';
eMsg[152] := 'symbol file of imported module not found';
eMsg[153] := 'object or symbol file not opened (disk full?)';
eMsg[154] := 'recursive import not allowed';
eMsg[155] := 'generation of new symbol file not allowed';
eMsg[156] := 'incorrect symbol file processor';
eMsg[160] := 'interfaces must be extensions of IUnknown';
eMsg[161] := 'interfaces must not have fields';
eMsg[162] := 'interface procedures must be abstract';
eMsg[163] := 'interface records must be abstract';
eMsg[164] := 'pointer must be extension of queried interface type';
eMsg[165] := 'illegal guid constant';
eMsg[166] := 'AddRef & Release may not be used';
eMsg[167] := 'illegal assignment to [new] parameter';
eMsg[168] := 'wrong [iid] - [new] pair';
eMsg[169] := 'must be an interface pointer';
eMsg[177] := 'IN only allowed for records and arrays';
eMsg[178] := 'illegal attribute';
eMsg[179] := 'abstract methods of exported records must be exported';
eMsg[180] := 'illegal receiver type';
eMsg[181] := 'base type is not extensible';
eMsg[182] := 'base procedure is not extensible';
eMsg[183] := 'non-matching export';
eMsg[184] := 'Attribute does not match with forward declaration';
eMsg[185] := 'missing NEW attribute';
eMsg[186] := 'illegal NEW attribute';
eMsg[187] := 'new empty procedure in non extensible record';
eMsg[188] := 'extensible procedure in non extensible record';
eMsg[189] := 'illegal attribute change';
eMsg[190] := 'record must be abstract';
eMsg[191] := 'base type must be abstract';
eMsg[192] := 'unimplemented abstract procedures in base types';
eMsg[193] := 'abstract or limited records may not be allocated';
eMsg[194] := 'no supercall allowed to abstract or empty procedures';
eMsg[195] := 'empty procedures may not have out parameters or return a
value';
eMsg[196] := 'procedure is implement-only exported';
eMsg[197] := 'extension of limited type must be limited';
eMsg[198] := 'obsolete oberon type';
```

```
eMsg[199] := 'obsolete oberon function';
eMsg[200] := 'not yet implemented';
eMsg[201] := 'lower bound of set range greater than higher bound';
eMsg[202] := 'set element greater than MAX(SET) or less than 0';
eMsg[203] := 'number too large';
eMsg[204] := 'product too large';
eMsg[205] := 'division by zero';
eMsg[206] := 'sum too large';
eMsg[207] := 'difference too large';
eMsg[208] := 'overflow in arithmetic shift';
eMsg[209] := 'case range too large';
eMsg[210] := 'code too long';
eMsg[211] := 'jump distance too large';
eMsg[212] := 'illegal real operation';
eMsg[213] := 'too many cases in case statement';
eMsg[214] := 'structure too large';
eMsg[215] := 'not enough registers: simplify expression';
eMsg[216] := 'not enough floating-point registers: simplify expression';
eMsg[217] := 'unimplemented SYSTEM function';
eMsg[218] := 'illegal value of parameter (0 <= p < 128)';
eMsg[219] := 'illegal value of parameter (0 <= p < 16)';
eMsg[220] := 'illegal value of parameter';
eMsg[221] := 'too many pointers in a record';
eMsg[222] := 'too many global pointers';
eMsg[223] := 'too many record types';
eMsg[224] := 'too many pointer types';
eMsg[225] := 'illegal sys flag';
eMsg[226] := 'too many exported procedures';
eMsg[227] := 'too many imported modules';
eMsg[228] := 'too many exported structures';
eMsg[229] := 'too many nested records for import';
eMsg[230] := 'too many constants (strings) in module';
eMsg[231] := 'too many link table entries (external procedures)';
eMsg[232] := 'too many commands in module';
eMsg[233] := 'record extension hierarchy too high';
eMsg[235] := 'too many modifiers';
eMsg[240] := 'identifier too long';
eMsg[241] := 'string too long';
eMsg[242] := 'too many meta names';
eMsg[243] := 'too many imported variables';
eMsg[249] := 'inconsistent import';
eMsg[250] := 'code proc must not be exported';
eMsg[251] := 'too many nested function calls';
eMsg[254] := 'debug position not found';
eMsg[255] := 'debug position';
eMsg[260] := 'illegal LONGINT operation';
eMsg[265] := 'unsupported string operation';
eMsg[270] := 'interface pointer reference counting restriction violated';
eMsg[271] := 'keyword check failed';
eMsg[272] := 'keyword expected for use';
eMsg[273] := 'keyword is not applicable for use';
eMsg[274] := 'unexpected identifier for use';
eMsg[275] := 'type is not applicable for use';
eMsg[276] := 'procedure is not applicable for use';
eMsg[277] := 'attribute is not applicable for use';
eMsg[278] := 'keyword is not applicable for use in context';
eMsg[279] := 'unexpected RETURN position';
eMsg[280] := 'procedure with RETURN';
eMsg[281] := 'only local or global variables allowed';
eMsg[282] := 'nested procedure is not allowed';
```

```
eMsg[283] := 'recursion is not allowed';
eMsg[301] := 'implicit type cast';
eMsg[302] := 'guarded variable can be side-effected';
eMsg[303] := 'open array (or pointer to array) containing pointers';
eMsg[900] := 'never used';
eMsg[901] := 'never set';
eMsg[902] := 'used before set';
eMsg[903] := 'set but never used';
eMsg[904] := 'used as varpar, possibly not set';
eMsg[905] := 'also declared in outer scope';
eMsg[906] := 'access/assignment to intermediate';
eMsg[907] := 'redefinition';
eMsg[908] := 'new definition';
eMsg[909] := 'statement after RETURN/EXIT';
eMsg[910] := 'for loop variable set';
eMsg[911] := 'implied type guard';
eMsg[912] := 'superfluous type guard';
eMsg[913] := 'call might depend on evaluation sequence of params.';
eMsg[930] := 'superfluous semicolon';
eMsg[401] := 'bytecode restriction: no structured assignment';
eMsg[402] := 'bytecode restriction: no procedure types';
eMsg[403] := 'bytecode restriction: no nested procedures';
eMsg[404] := 'bytecode restriction: illegal SYSTEM function';
eMsg[410] := 'variable may not have been assigned';
eMsg[411] := 'no proofable return';
eMsg[412] := 'illegal constructor call';
eMsg[413] := 'missing constructor call';
```

ПЕРЕЧЕНЬ ТЕРМИНОВ И СОКРАЩЕНИЙ

АЭС	- Атомная Электростанция
КСА	- Комплекс средств автоматизации
ОС	- Операционная система
ПО	- Программное обеспечение
LLVM	- Low Level Virtual Machine – Виртуальная низкоуровневая машина
СКАДИ	- Средства комплексной автоматизации и диагностики
ЧМИ	- Человеко-машинный интерфейс
ICMP	- Internet Control Message Protocol
TCP	- Transmission Control Protocol
UDP	- User Datagram Protocol

