Использование более новых версий GCC и binutils с коллекцией портов FreeBSD

Martin Matuska <mm@FreeBSD.org >

Издание: 45050

Авторские права © 2009 The FreeBSD Documentation Project

FreeBSD это зарегистрированная торговая марка FreeBSD Foundation.

Многие из обозначений, используемые производителями и продавцами для обозначения своих продуктов, заявляются в качестве торговых марок. Когда такие обозначения появляются в этом документе, и Проекту FreeBSD известно о торговой марке, к обозначению добавляется знак «TM» или «(R)».

2014-06-13 taras.

Аннотация

В этой статье описано, как использовать более новые версии компиляторов GCC и набора утилит binutils из коллекции портов FreeBSD. Здесь также обсуждаются специализированные конфигурации GCC.

Содержание

1. Предисловие 1
2. Подготовка 2
3. Настройка портов на использование требуемой версии GCC 2
4. Влияние на производительность исполняемых файлов 4

1. Предисловие

Компилятор, поставляемый с FreeBSD 8.0, - это GCC версии 4.2.1. В базовую систему FreeBSD включены binutils версии 2.15. Это довольно-таки старые версии, и в них,

между прочим, отсутствует поддержка новых инструкций CPU, таких как SSSE3, SSE4.1, SSE4.2 и т.п. Ввиду лицензионных ограничений, новые версии этих приложений не будут включены в базовую систему. К счастью, есть возможность использовать более новую версию компилятора GCC (например, версию 4.4) установив его из коллекции портов FreeBSD.

2. Подготовка

2.1. Установка binutils из портов

Чтобы задействовать все новые функциональные возможности последней версии GCC, желательна установка наиболее новой версии binutils. Это не является обязательным, но без обновления binutils вы не получите поддержку новых инструкций CPU.

Для того, чтоб установить наиболее новую версию binutils используя коллекцию портов FreeBSD, выполните следующую команду:

```
# cd /usr/ports/devel/binutils && make install
```

2.2. Установка GCC из портов

В коллекции портов FreeBSD можно найти несколько новых версий GCC. В нашем примере мы устанавливаем стабильную версию 4.4. Однако, возможно установить предыдущую или более новую (развивающуюся) версию (например: lang/gcc43 или lang/gcc45).

Для установки одного из вышеупомянутых портов GCC, наберите следующую команду:

```
# cd /usr/ports/lang/ gcc44 && make install
```

3. Настройка портов на использование требуемой версии GCC

Для использования требуемой версии GCC, установленной из коллекции портов FreeBSD, требуется дополнительная конфигурация системы.

3.1. Изменения в `make.conf`

Добавьте следующие строки в `/etc/make.conf` (или исправьте существующие):

```
.if !empty(.CURDIR:M/usr/ports/*) && exists(/usr/local/bin/gcc44)
CC=gcc44
CXX=g++44
CPP=cpp44
```

```
.endif
```

В качестве альтернативы, можно установить вручную переменные окружения `${CC}` и `${CPP}`.

Примечание

Примеры, приведённые выше, сделаны для GCC версии 4.4. Для того, чтобы использовать `gcc43`, замените `"gcc44"` на `"gcc43"` и `"4.4"` на `"4.3"` и т.п.

3.2. Исправления в `libmap.conf`

Многие установленные из портов приложения и библиотеки скомпонованы с libgcc_s или libstdc++. В базовую систему уже включены эти библиотеки, но от более старой версии GCC (версии 4.2.1). Чтобы предоставить rltd (и ldd) правильные версии, добавьте следующие записи к `/etc/libmap.conf` (или измените существующие):

```
libgcc_s.so.1   gcc44/libgcc_s.so.1
libgomp.so.1    gcc44/libgomp.so.1
libobjc.so.3    gcc44/libobjc.so.2
libssp.so.0     gcc44/libssp.so.0
libstdc++.so.6  gcc44/libstdc++.so.6
```

Примечание

Пример выше приведен для GCC версии 4.4. Если вы используете `gcc43`, замените `"gcc44"` на `"gcc43"` и т.д. Следует отметить, что все вышеупомянутые библиотеки обратно совместимы с библиотеками, поставляемыми с базовой системой.

Предупреждение

Некоторые C++ программы могут перестать работать, если отсутствует корректная компоновка этих библиотек. Если невозможно скомпоновать их все, рекомендуется сделать это по крайней мере для libstdc++.so.

3.3. Специальные установки CFLAGS для коллекции портов

Для портов, не поддерживающих специальные установки CFLAGS, исправьте /etc/make.conf согласно следующему примеру:

```
.if !empty(.CURDIR:M/usr/ports/*) && exists(/usr/local/bin/gcc44)
CC=gcc44
CXX=g++44
CPP=cpp44
CFLAGS+=-mssse3
.endif
```

Возможно заменить полностью CFLAGS и/или определить CPUTYPE. Мы рекомендуем устанавливать CPUTYPE так как во многих портах флаги оптимизации выбираются, базируясь на значении этой переменной.

3.4. Исключение портов, которые не собираются с новой версией GCC

Для того, чтобы исключить порты, которые не собираются со специальной версией GCC, исправьте /etc/make.conf согласно следующему примеру:

```
.if !empty(.CURDIR:M/usr/ports/*) && exists(/usr/local/bin/gcc44)
.if empty(.CURDIR:M/usr/ports/net/openldap*)
CC=gcc44
CXX=g++44
CPP=cpp44
.endif
.endif
```

Пример, приведённый выше, отменяет принудительное использование gcc 4.4 для портов net/openldap*. Также возможно указать больше портов в одной строке:

```
.if empty(.CURDIR:M/usr/ports/net/openldap*) && empty(.CURDIR:M/usr/
ports/xxx/yyy) && ...
```

4. Влияние на производительность исполняемых файлов

Использование GCC версии 4.4 с разрешенным набором инструкций SSSE3 (если они поддерживаются CPU) может дать до 10% увеличения средней производительности исполняемых файлов. В некоторых тестах прирост производительности превышал 20% (например, в обработке мультимедийных данных).

В таблице, размещенной на http://people.freebsd.org/~mm/benchmarks/perlbench/, показано сравнение GCC в настоящее время включенного в базу FreeBSD, GCC версии 4.3 и GCC версии 4.4 с различными комбинациями CFLAGS. Для сравнений производительности использовался тестовый пакет perlbench.