10. Вызов GNU C Препроцессора

Обычно нет необходимости вызывать C препроцессор отдельно: он вызывается автоматически C компилятором. Однако иногда требуется запустить препроцессор без компилятора.

C препроцессор запрашивает два имени файла в качестве аргументов. Один из них является входным файлом, а другой - выходным. Препроцессор считыват входной файл вместе с другими файлами, указанными директивой '#include'. Получаемые при обработке данные поступают в выходной файл.

Как входным так и выходным файлами может быть параметр '-'. Если он указан вместо имени входного файла, то все данные считываются со стандартного ввода, если же он указан вместо выходного файла, то все выходные данные поступают на стандартый вывод. Также если имя выходного файла или оба имени файла опущены, то для ввода и вывода вместо отсутствующих имен файлов используется стандартный ввод и вывод.

Далее следует список опций, используемых C препроцессором. Эти опции также могут быть указаны при компиляции программ, так как они автоматически передаются препроцессору при вызове компилятора.

'-P'

Предотвращает генерацию строк, начинающихся с символа '#', вместе с номером строки. Это может быть полезно при обработке препроцессором какого-либо текста, не являющегося C программой.

'-C'

Не удаляет комментарии и передает их в выходной файл. Комментарии в аргументах макросов поступают на вывод перед подстановкой макросов.

'-traditional'

Иммитирует работу более ранней версии препроцессора, что противопоставляется стандарту ANSI C.

При макроподстановках игнорируется наличие символов двойной и одинарной кавычки. Имя макроаргумента заменяется на его значение, даже если он является строчной или символьной константой.

Значение макроса может завершаться в середине строковой или символьной константы. Константа завершается в тексте программы после макроподстановки.

Конец строки завершает строковую или символьную константу без сообщения об ошибке.

Комментарии заменяются пустым значением. (В ANSI C комментарии заменяются пробелом.)

Не производится обработка чисел. Значение '1.0e+4' рассматривается как три значения: '1.0e', '+' и '4'.

Не запрещается использование макросов в их макроопределениях. Поэтому использование любого рекурсивно заданного макроса неизбежно приводит к ошибке.

Символ '#' не имеет какого-либо специального значения в макроопределении.

Текст программы, находящийся в конце значения макроса может выполняться вместе с текстом, находящимся после вызова макроса, образуя единую конструкцию, что невозможно в ANSI C.

Символ '\' в макроаргументе указывает на синтаксический смысл следующего символа.

'-trigraphs'

Обработка последовательностей trigraph стандарта ANSI. Это последовательности, состоящие из трех символов и начинающиеся со строки '??'. Они определены стандартом ANSI C для обозначения отдельных символов. Например, значением '??/' является '\', поэтому значением ''??/n'' будет символьная константа, вкючающая символ перевода строки. GNU C препроцессор не поддерживает все программы в стандарте ANSI C, пока не указана опция '-trigraphs'.

'-pedantic'

Выдаются предупреждающие сообщения в соответствии со стандартом ANSI C, например в случаях, если какой-либо другой текст кроме комментариев стоит после директив '#else' или '#endif'.

'-pedantic-errors'

То же, что и '-pedantic', только выдаются сообщения об ошибках, вместо предупреждений.

'-Wtrigraphs'

Выдается предупреждение, если обнаруживается trigraph последовательность (предполагается, что указана опция '-trigraphs').

'-Wcomment'

Выдается предупреждение, если обнаруживается последовательность '/*' внутри комментария.

'-Wall'

То же, что и '-Wtrigraphs' и '-Wcomment' (но не '-Wtraditional').

'-Wtraditional'

Выдается предупреждение, если встречаются отдельные конструкции, имеющие различия в обычном и в ANSI C.

'-I DIRECTORY'

Имя каталога DIRECTORY добавляется к началу списка каталогов, где производится поиск подключаемых файлов. Это может быть использовано для вставки нестандартного подключаемого файла со стандартным именем, потому как эти каталоги просматриваются перед системными каталогами с подключаемыми файлами. При использовании более одной опции '-I' указанные каталоги просматриваются слева направо, затем просматриваются системные каталоги.

'-I-'

Все каталоги, указанные опцией '-I' до указания опции '-I-' просматриваются только при выполнении директивы '#include "FILE"' и не используются директивой '#include <FILE>'.

Если после опции '-I-' указаны дополнительные каталоги опцией '-I', то эти каталоги используются всеми директивами '#include'.

При использовании опции '-I-', текущий каталог не просматривается в первую очередь директивой '#include "FILE"'. Поэтому текущий каталог просматривается только в случае его указания опцией '-I.'. Если указать обе опции '-I-' и '-I.' то можно проследить, какие каталоги просматриваются до текущего, а какие - после.

'-nostdinc'

Стандартные системные каталоги не используются для поиска подключаемых файлов. Используются только каталоги, указанные опциями '-I' (а также текущий каталог, если он указан).

'-nostdinc++'

Не производится поиск подключаемых файлов в стандартных каталогах C++, а для поиска используются остальные стандартные каталоги. (Эта опция применяется при построении библиотеки libg++.)

'-D NAME'

Определяется макрос с именем NAME и значением '1'.

'-D NAME=DEFINITION'

Определяется макрос с именем NAME и значением DEFINITION. Не существует никаких ограничений на значение DEFINITION, но если препроцессор вызывается из оболочки или подобной программы, то следует использовать специальный синтаксис для предотвращения передачи символов со специальным значением, таких как пробелы, оболочке. Если передается более чем один параметр '-D' для одного и того же значения NAME, то используется значение, стоящее правее всех.

'-U NAME'

Макрос с именем NAME не определяется. Если передаются оба параметра '-U' и '-D' для одного имени, то макрос не определяется, так как приоритет параметра '-U' выше.

'-undef'

Все нестандартные макросы не определяются.

'-A PREDICATE(ANSWER)'

Создается утверждение с именем PREDICATE и значением ANSWER.

Возможно использование опции '-A-' для предотвращения создания всех утверждений. Эта опция также уничтожает все макросы, указывающие на тип используемой системы.

'-dM'

Вместо вывода результата обработки, выводится список директив '#define' для всех определенных макросов при работе препроцессора, включая заранее определенные макросы. Это позволяет выяснить определения всех макросов в используемой версии препроцессора. Предполагая, что файл 'foo.h' не существует, следующая команда

```
touch foo.h; cpp -dM foo.h
```

выявит значения всех заранее определенных макросов.

'-dD'

То же, что и '-dM' за исключением двух аспектов: заранее определенные макросы не включаются и выводятся как все директивы '#define', так и результат обработки. Весь вывод поступает в стандартный файл вывода.

'-M [-MG]'

Вмето вывода результата обработки, выводится информация в формате команды 'make', описывающая зависимости основного исходного файла. Препроцессор выводит информацию для 'make', которая включает в себя имя объектного файла для данного исходного файла, запятую, и имена всех подключаемых файлов. Если подключаемых файлов довольно большое количество, то строка разбивается на несколько с помощью последовательности '\'-newline.

Опция '-MG' указывает, что отсутствующие подключаемые файлы являются генерируемыми файлами и предполагается что они находятся в одном каталоге с исходными файлами. Она должна быть указана вместе с опцией '-M'.

Эта возможность используется для автоматического обновления make-файлов.

'-MM [-MG]'

То же, что и '-M', только используются файлы, указываемые директивой '#include "FILE"'. Системные подключаемые файлы, указываемые директивой '#include <FILE>', не используются.

'-MD FILE'

То же, что и '-M', только вся информация записывается в файл с именем FILE. Используется в дополнение к стандартной компиляции файла. При указании этой опции не запрещается стандартная компиляция в отличие от опции '-M'.

При вызове gcc аргумент FILE не указывается. Gcc создает файлы путем замены ".c" на ".d" в конце имени исходного файла.

При использовании Mach возможно применение утилиты 'md' для объединения нескольких файлов в один, который затем можно использовать с командой 'make'.

'-MMD FILE'

То же, что и '-MD' за исключением того, что используются подключаемые файлы пользователя, а не системные подключаемые файлы.

'-H'

Выводятся имена всех используемых подключаемых файлов.

'-imacros FILE'

Обработка файла FILE в качестве ввода вне зависимости от вывода перед обработкой стандартного входного файла. Так как вывод при обработке файла FILE не используется, то единственным применением этой опции является определение макросов в файле FILE для их последующего использования при обработке исходного файла.

'-include FILE'

Обработка файла FILE в качестве входного с последующим включением всех данных, полученных при обработке, в основной входной файл перед его обработкой.

'idirafter DIR'

Имя каталога DIR добавляется ко второму пути поиска подключаемых файлов. Каталоги, указанные во втором пути, просматриваются в том случае, если требуемый подключаемый файл не был найден в каталогах, указанных в основном пути поиска (он задается опцией '-I').

'-iprefix PREFIX'

Определяет значение PREFIX, как префикс для последующих опций

'-iwithprefix'.

'-iwithprefix DIR'

Имя каталога DIR добавляется ко второму пути поиска подключаемых файлов. Имя каталога создается путем объединения значений PREFIX и DIR, где значение PREFIX дополнительно указывается опцией '-iprefix'.

'-isystem DIR'

Каталог DIR добавляется в начало второго пути поиска подключаемых файлов, помечая его как системный каталог. Таким образом, этот каталог используется также как и стандартные системные каталоги.

`-lang-c'

`-lang-c89'

`-lang-c++'

`-lang-objc'

`-lang-objc++'

Указывается язык исходного файла. Опция '-lang-c' указывается по умолчанию. Она допускает использование комментариев C++ (комментарии, начинающиеся со строки '//' изаканчивающиеся в конце этой строки). Опция '-lang-c89' запрещает использование комментариев C++. При указании опции '-lang-c++' обрабатываются комментарии C++ и используются дополнительные каталоги для подключаемых файлов C++. Опция '-lang-objc' допускает использование директивы объектного C '#import'. Опция '-lang-objc++' использует все возможности расширений объектного C и C++.

Эти опции создаются драйвером компилятора 'gcc', но не передаются через командную строку при вызове 'gcc' до тех пор пока не будет указана опция драйвера '-Wp'.

'-lint'

Производится поиск команд, заключенных в комментарии, для программы проверки 'lint' с их последующим включением в текст программы с префиксом '#pragma lint'. Например, комментарий '/* NOTREACHED */' после обработки препроцессором становится строкой '#pragma lint NOTREACHED'.

Использование этой опции возможно только при прямом вызове 'cpp'. 'gcc' не передает эту опцию из своей командной строки.

'-$'

Запрещается использование символа '$' в идентификаторах. Это используется для согласования со стандартом ANSI. 'gcc' автоматически передает эту опцию препроцессору при указании опции '-ansi', но сам 'gcc' не распознает опцию '-$'. Ее нужно указать препроцессору отдельно для ее применения без дополнительного воздействия опции '-ansi'.

Вперед Назад Содержание