6.31 Объявление атрибутов функций.

В GNU C, вы объявляете определённые вещи для функций, вызываемых в вашей программе, которые помогают компилятору оптимизировать вызовы функций и проверять ваш код с более заботливо.

Ключевое слово \_\_attribute\_\_ позволяет вам указывать специальные атрибуты при объявлении (declaration). Это ключевое слово сопровождается спецификацией атрибута внутри двойных скобочек. Следующие атрибуты в настоящее время определены для функций на всех платформах: (NB: смотри gcc.pdf). Некоторые другие атрибуты определены для функций на отдельных целевых платформах. Другие атрибуты, включая section, поддерживаются для объявлений переменных, меток и типов.

GCC плагины могут предоставлять их собственные атрибуты.

Вы можете также определить атрибуты с «\_\_» до и после каждого ключевого слова. Это позволяет вам использовать их в заголовочных файлах не заботясь о возможном макросе с тем же именем. Например, вы можете использовать \_\_noreturn\_\_ вместо noreturn.

Смотри раздел 6.33 [Синтаксис атрибутов], стр. 425, для более детальной информации о точном синтаксисе используемых атрибутов.

NB: далее будем описывать атрибуты в том порядке, в каком они будут встречаться в исходных текстах программ.

**noreturn.**

Несколько стандартных библиотечных функций, таких как abort или exit, не могут возвращать управление. GCC знает это автоматически. Некоторые программы определяют их собственные функции, которые никогда возвращают управление. Вы можете объявить их как noreturn, чтобы сообщить компилятору этот факт. Например:

void fatal() \_\_atribute\_\_ ((noreturn));

void

fatal(/\* … \*/)

{

/\* … \*/ /\* Вывести сообщение об ошибке \*/ /\* … \*/

exit(1);

}

Ключевое слово noreturn говорит компилятору — принять, что fatal не может вернуть управление. Он затем сможет оптимизировать её без оглядки на то, что произошло бы, если fatal всё же что-то вернул. Это делает код несколько лучше. Более того, это помогает избегать паразитные предупреждения не инициализированных переменных.

Ключевое слово noreturn не влияет на тракт исключения, когда оно происходит: функция, помеченная как noreturn всё ещё может возвращать управление вызывающему, выбрасывая исключение или вызывая longjump.

Не следует полагать, что регистры, сохранённые вызывающей функцией, восстановятся до вызова noreturn функции.

Для noreturn функции не имеет смысла иметь тип возвращаемого значения, отличного от void.

**section(“section-name”).**

Нормально, компилятор размещает код который он генерирует в секции text. Иногда, однако, вам нужны дополнительные секции, вам нужно, чтобы некоторые определённые функции располагались в специальных секциях. Атрибут section указывает, что функция живёт в отдельной секции. Например, объявление:

extern void foobar (void) \_\_attribute\_\_ ((section(“bar”)));

помещает функцию foobar в секцию bar.

Некоторые форматы файлов не поддерживают произвольные секции, так что атрибут section не доступен на всех платформах. Если вам необходимо отобразить весь контент модуля в отдельную секцию, следует рассмотреть возможности компоновщика вместо этого (NB: атрибута).

**naked.**

Этот атрибут доступен на ARM, AVR, MCORE, MSP430, NDS32, RL78, RX и порты SPU. Этот атрибут позволяет компилятору создавать необходимое объявление функции, в то же время позволяя телу функции быть ассемблерным кодом. Указанная функция не будет иметь последовательности пролога/эпилога, генерируемой компилятором. Только базовые ассемблерные операторы могут безопасно быть включены в «голые» функции (см. раздел 6.43.1 [базовый ассемблер], стр.449). Использование расширенного ассемблера или смеси базового ассемблера и кода на C всё же может работать, но они не могут быть в зависимости от надёжной работы и не поддерживаются (NB: оригинал: While using extended asm or a mixture of basic asm and C code may appear to work, they cannot be depended upon to work reliably and are not supported).