GCC and Portability.**GCC и переносимость.**

Сам *GCC* стремится быть переносимым на любую машину, где **int** является как минимум **32-битным типом.**Он нацелен на машины с плоским *(несегментированным)* **(non-segmented)** адресным пространством данных с байтовой адресацией (*адресное пространство кода может быть отдельным).*Целевые **ABI** могут иметь **8-, 16-, 32-** или **64-битный** тип **int**. **char** может быть шире **8 бит**. GCC получает большую часть информации о целевой машине из описания машины, которое дает алгебраическую формулу для каждой инструкции машины. Это очень простой способ описать цельНо когда компилятору нужна информация, которую трудно получить таким образом, для описания машин были определены специальные параметры. Цель переносимости — уменьшить общую работу, необходимую для компилятора; это не представляло интереса само по себе. GCC не содержит машинно-зависимого кода,но он содержит код, который зависит от параметров машины, таких как порядок следования байтов.(независимо от того, имеет ли самый старший байт самый высокий или самый низкий адрес байтов в слове)а также наличие автоинкрементной адресации. На этапе генерации **RTL** часто необходимо иметь несколько *стратегий* для генерации кода для определенного вида синтаксического дерева,стратегии,применимые для различных комбинаций параметров.В результате для новой цели могут потребоваться дополнительные стратегии.Вы узнаете, если это произойдет, потому что компилятор вызовет аварийную остановку.К счастью, новые стратегии можно добавлять машинно-независимым способом, и они будут влиять только на те целевые машины, которым они нужны**.**