Глобальные переменные

Компилятор вычисляет необходимый размер глобальной памяти в момент компиляции. Реально память под такие переменные выделяется в процессе загрузки программы в память. Если места под глобальные переменные не хватает, то программа в память не загрузится.

Раздельная компиляция. Иногда необходимо иметь глобальные переменные на весь проект (несколько модулей). Однако если такая потребность возникла – стоит пересмотреть архитектуру проекта. Как это сделать:

main.c : main(){ a=6; }. hello.c : int a=3; void hello(){}. При компиляции и сборке такой программы возникает проблема, main не знает что такое a. Решаются такие проблемы заголовочными файлами: hello.h : int a; Проблемы возникнут на этапе линковки/компоновки. И в main.c и в hello.c будет a. Линковщик не сможет произвести операцию разрешения имён. Добавляем в hello.h : extern int a; Ключевое слово extern сообщает компилятору, что следующие за ним типы и имена переменных объявлены где-то в другом месте. Это сделано для ускорения и упрощения раздельной компиляции, то есть все глобальные переменные определяются в одном файле, а в остальных файлах блок объявлений копируется и везде добавляется слово extern int a; …

Вообще использовать глобальные переменные не очень хорошо: сложно следить за логикой работы программы, сложно следить за отличием глобальных имён в нескольких файлах исходного кода (особенно если проект большой и работает несколько человек). Если имена совпадут, то на этапе линковки будет проблема, также это может возникнуть при использовании библиотек, особенно тех, которые распространяются без исходных кодов, в виде объектных файлов. Короче говоря, при прочих равных, глобальные переменные лучше не использовать.