Введение

Это краткое руководство содержит документацию по бинарным утилитам gnu (инструменты GNU для STM32 10.3-2021.10.20211105-1100) версия 2.36.1:

**ar**  Создание, изменение и извлечение из архивов

**nm** Список символов из объектных файлов

**objcopy** Копировать и переводить объектные файлы

objdump Отображение информации из объектных файлов

**readelf**  Отображение содержимого файлов формата ELF.

**size**  Список размеров разделов файла и общий размер

**strings** Список печатных строк из файлов

**strip**  Отменить символы

**elfedit**  Обновите заголовок ELF и свойство программы файлов ELF.

**c++filt** Символы **C++,** закодированные по алгоритму **Demangle**(в **MS-DOS** программа называется **cxxfilt**)

**addr2line**  Преобразование адресов в имена файлов и номера строк

**windres** Управление ресурсами Windows

**dlltool** Создайте файлы, необходимые для создания и использования библиотек динамической компоновки.

Этот документ распространяется на условиях лицензии GNU Free Documentation License.версия 1.3. Копия лицензии включена в раздел «Бесплатная документация GNU».Лицензия".

1 ar

**ar [-]p[mod] [--plugin name] [--target bfdname] [--output dirname] [--record-libdeps libdeps]**

**[relpos] [count] archive [member...]**

**ar -M [ <mri-script ]**

Программа gnu **ar** создает, модифицирует и извлекает из архивов. Архив — это отдельный файл, содержащий набор других файлов в структуре, которая позволяет извлекать исходные отдельные файлы (называемые элементами архива). Содержимое исходных файлов, режим (разрешения) **mode (permissions),** метка времени(timestamp), владелец и группа сохраняются в архиве и могут быть восстановлены при извлечении. gnu ar может поддерживать архивы, члены которых имеют имена любой длины; однако, в зависимости от того, как ar настроен в вашей системе, может быть наложено ограничение на длину имени элемента для совместимости с форматами архивов,поддерживаемыми другими инструментами. Если он существует, ограничение часто составляет 15 символов (типично для форматов, связанных с **a.out**) или 16 символов (типично для форматов, связанных с **coff**).**ar** считается бинарной утилитой, потому что архивы такого рода чаще всего используются в качестве библиотек, содержащих наиболее часто используемые подпрограммы. Поскольку библиотеки часто зависят от других библиотек, **ar** также может записывать зависимости библиотеки, если указан параметр **--record-libdeps.ar** создает индекс для символов, определенных в перемещаемых объектных модулях в архиве, когда вы указываете модификатор **‘s’**. После создания этот индекс обновляется в архиве всякий раз, когда ar изменяет свое содержимое (за исключением операции обновления **‘q’**). Архив с таким индексом ускоряет подключение к библиотеке и позволяет подпрограммам в библиотеке вызывать друг друга независимо от их размещения в архиве. Вы можете использовать **‘nm -s’**или **‘nm --print-armap’**, чтобы получить список этой индексной таблицы. Если в архиве отсутствует таблица, можно использовать другую форму **ar**, называемую **ranlib**, для добавления только таблицы.**gnu ar** может дополнительно создать тонкий архив, который содержит индекс символов и ссылки на исходные копии файлов-членов архива. Это полезно для создания библиотек для использования в локальном дереве сборки, где ожидается, что перемещаемые объекты останутся доступными, а копирование содержимого каждого объекта приведет только к пустой трате времени и места.После создания архива его формат нельзя изменить, не удалив его и не создав на его месте новый архив.Тонкие архивы также выравниваются, поэтому добавление одного тонкого архива в другой тонкий архив не приводит к его вложению, как это произошло бы с обычным архивом. Вместо этого элементы первого архива добавляются во второй архив по отдельности.

Пути к элементам архива записываются относительно самого архива. **gnu ar** спроектирован так, чтобы быть совместимым с двумя разными средствами. Вы можете управлять его активностью с помощью параметров командной строки, таких как различные варианты **ar** в системах **Unix**; или, если вы укажете единственную опцию командной строки **-M**, вы можете управлять ею с помощью сценария, поставляемого через стандартный ввод, например, программы **“librarian” MRI**.

* 1. Управление **ar** в командной строке

**ar [-X32\_64] [-]p[mod] [--plugin name] [--target bfdname] [--output dirname] [--recordlibdeps**

**libdeps] [relpos] [count] archive [member...]**

Когда вы используете **ar** в стиле **Unix**, **ar** настаивает на выполнении как минимум двух аргументов: *одной ключевой буквы*, определяющей операцию (необязательно сопровождаемой другими ключевыми буквами, определяющими модификаторы), и *имени архива*, с которым нужно действовать**.**

**gnu ar** позволяет смешивать код операции **p** и **флаги** модификатора **(modifier flags)mod** в любом порядке,в первом аргументе командной строки.При желании вы можете начать первый аргумент командной строки с дефиса.Ключевая буква **p** указывает, какую операцию выполнять; это может быть что-то из перечисленного ниже, но необходимо указать только один из них:

**‘d’** Удаление модулей из архива.

Укажите имена модулей, которые нужно удалить в качестве члена . . . ; архив остается нетронутым, если вы не укажете никаких файлов для удаления.Если вы укажете модификатор **‘v’**, **ar** перечислит каждый модуль, поскольку он был удален.

‘m’ Используйте эту операцию для перемещения элементов в архиве.Порядок элементов в архиве может повлиять на способ компоновки программ с использованием библиотеки, если символ определен более чем в одном элементе.Если с **m** не используются никакие модификаторы, все элементы, указанные вами в аргументах членов, перемещаются в конец архива; вместо этого вы можете использовать модификаторы **‘a’, ‘b’** или **‘i’**, чтобы переместить их в указанное место.

**‘p’** *Распечатайте* указанные элементы архива в стандартный выходной файл. Если указан модификатор **‘v’**, покажите имя члена перед копированием его содержимого в стандартный вывод. Если вы не укажете аргументы члена, будут напечатаны все файлы в архиве.

**‘q’** *Быстрое добавление***(Quick append);** Исторически сложилось так, что добавлялись файлы member . . . до конца архива,без проверки замены.Модификаторы **‘a’, ‘b’**,и **‘i’** на эту операцию не влияют; новые члены всегда помещаются в конец архива.Модификатор **‘v’**заставляет **ar** перечислять каждый файл по мере его добавления.Поскольку целью этой операции является скорость, реализации ar имеют возможность не обновлять таблицу символов архива, если она существует. Однако слишком много различных систем предполагают, что таблицы символов всегда обновлены, поэтому **gnu ar** перестроит таблицу даже при быстром добавлении.имеющиеся файлы в архиве и добавление новых в конец.

**‘r’**  Вставьте член файлов . . . в архив (с заменой). Эта операция отличается от **‘q’** тем, что любые ранее существовавшие элементы удаляются, если их имена совпадают с именами добавляемых. Если один из файлов, названных в файле member . . . не существует, ar отображает ошибку

сообщение и оставляет нетронутыми любые существующие элементы архива, соответствующие

это имя.По умолчанию новые члены добавляются в конец файла; но вы можете использовать одинмодификаторов **‘a’, ‘b’**,и **‘i’** ,чтобы запросить размещение относительно некоторых существующих член.Модификатор **‘v’**, используемый с этой операцией, вызывает строку вывода для каждого файла вставлен вместе с одной из букв **‘a’**или **‘r’** , чтобы указать, был ли файл

добавлен (старый член не удален) или заменен.