ar (Unix)

The **archiver**, also known simply as **ar**, is a Unix utility that maintains groups of files as a single archive file. Today, `ar` is generally used only to create and update static library files that the link editor or linker uses and for generating .deb packages for the Debian family; it can be used to create archives for any purpose, but has been largely replaced by `tar` for purposes other than static libraries.[3] An implementation of `ar` is included as one of the GNU Binutils.[2]

In the Linux Standard Base (LSB), `ar` has been deprecated and is expected to disappear in a future release of that standard. The rationale provided was that "the LSB does not include software development utilities nor does it specify .o and .a file formats."[4]

ar

Original author(s)	Ken Thompson, Dennis Ritchie (AT&T Bell Laboratories)
Developer(s)	Various open-source and commercial developers
Initial release	November 3, 1971
Operating system	Unix, Unix-like, V, Plan 9, Inferno
Platform	Cross-platform
Type	Command

archiver format

Filename extension	`.a, .lib, .ar`[1]
Internet media type	`application/x-archive`[1]
Magic number	`!<arch>`
Type of format	archive format
Container for	usually object files (`.o`, `.obj`)
Standard	Not standardized, several variants exist
Open format?	Yes[2]

Contents

File format details

The ar format has never been standardized; modern archives are based on a common format with two main variants, BSD and System V (initially known as COFF, and used as well by GNU, ELF, and Windows.)

Historically there have been other variants[5] including V6, V7, AIX (small and big), and Coherent, which all vary significantly from the common format.[6]

Debian ".deb" archives use the common format.

An ar file begins with a global header, followed by a header and data section for each file stored within the ar file.

Each data section is 2 byte aligned. If it would end on an odd offset, a newline ('\n', 0x0A) is used as filler.

File signature

The file signature is a single field containing the magic ASCII string `"!<arch>"` followed by a single LF control character (0x0A).

File header

Each file stored in an ar archive includes a file header to store information about the file. The common format is as follows. Numeric values are encoded in ASCII and all values right-padded with ASCII spaces (0x20).

Offset	Length	Name	Format
0	16	File identifier	ASCII
16	12	File modification timestamp (in seconds)	Decimal
28	6	Owner ID	Decimal
34	6	Group ID	Decimal
40	8	File mode (type and permission)	Octal
48	10	File size in bytes	Decimal
58	2	Ending characters	0x60 0x0A

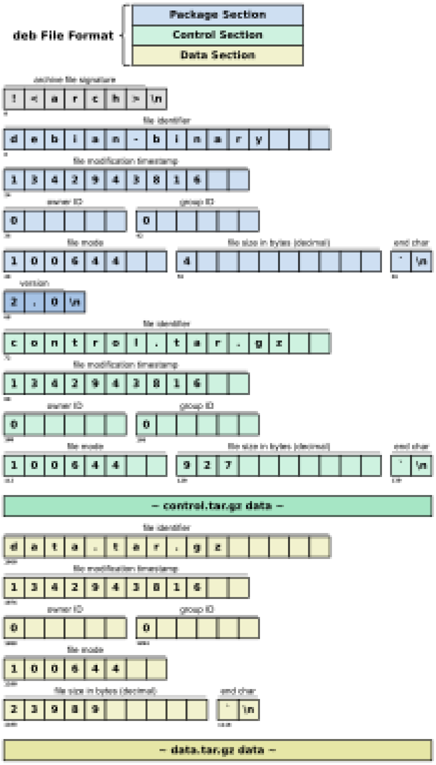

Diagram showing an example file structure of a .deb file

As the headers only include printable ASCII characters and line feeds, an archive containing only text files therefore still appears to be a text file itself.

The members are aligned to even byte boundaries. "Each archive file member begins on an even byte boundary; a newline is inserted between files if necessary. Nevertheless, the size given reflects the actual size of the file exclusive of padding."[7]

Due to the limitations of file name length and format, both the GNU and BSD variants devised different methods of storing long filenames. Although the common format does not suffer from the year 2038 problem, many implementations of the ar utility do and may need to be modified in the future to handle correctly timestamps in excess of 2147483647. A description of these extensions is found in libbfd.[8]

Depending on the format, many ar implementations include a global symbol table (aka armap, directory or index) for fast linking without needing to scan the whole archive for a symbol. POSIX recognizes this feature, and requires ar implementations to have an `-s` option for updating it. Most implementations put it at the first file entry.[9]

BSD variant

BSD ar stores filenames right-padded with ASCII spaces. This causes issues with spaces inside filenames. 4.4BSD ar stores extended filenames by placing the string "#1/" followed by the file name length in the file name field, and storing the real filename in front of the data section.[6]

BSD ar utility traditionally does not handle the building of a global symbol lookup table, and delegates this task to a separate utility named **ranlib**,[10] which inserts an architecture-specific file named `__.SYMDEF` as first archive member.[11] Some descendents put a space and "SORTED" after the name to indicate a sorted version.[12] A 64-bit variant called `__.SYMDEF_64` exists on Darwin.

Since POSIX added the requirement for the `-s` option as an replacement of ranlib, however, newer BSD ar implementations have been rewritten to have this feature. FreeBSD in particular ditched the SYMDEF table format and embraced the System V style table.[13]

System V (or GNU) variant

System V ar uses a '/' character (0x2F) to mark the end of the filename; this allows for the use of spaces without the use of an extended filename. Then it stores multiple extended filenames in the data section of a file with the name "//", this record is referred to by future headers. A header references an extended filename by storing a "/" followed by a decimal offset to the start of the filename in the extended filename data section. The format of this "//" file itself is simply a list of the long filenames, each separated by one or more LF characters. Note that the decimal offsets are number of characters, not line or string number within the "//" file. This is usually the second entry of the file, after the symbol table which always is the first.

System V ar uses the special filename "/" to denote that the following data entry contains a symbol lookup table, which is used in ar libraries to speed up access. This symbol table is built in three parts which are recorded together as contiguous data.

1. A 32-bit big endian integer, giving the number of entries in the table.
2. A set of 32-bit big endian integers. One for each symbol, recording the position within the archive of the header for the file containing this symbol.
3. A set of Zero-terminated strings. Each is a symbol name, and occurs in the same order as the list of positions in part 2.

Some System V systems do not use the format described above for the symbol lookup table. For operating systems such as HP-UX 11.0, this information is stored in a data structure based on the SOM file format.

The special file "/" is not terminated with a specific sequence; the end is assumed once the last symbol name has been read.

To overcome the 4 GiB file size limit some operating system like Solaris 11.2 and GNU use a variant lookup table. Instead of 32-bit integers, 64-bit integers are used in the symbol lookup tables. The string "/SYM64/" instead "/" is used as identifier for this table[14]

Windows variant

The Windows (PE/COFF) variant is based on the SysV/GNU variant. The first entry "/" has the same layout as the SysV/GNU symbol table. The second entry is another "/", a Microsoft ECOFF extension that stores an extended symbol cross-reference table. This one is sorted and uses little-endian integers.[5][15] The third entry is the optional "//" long name data as in SysV/GNU.[16]

Thin archive

The version of `ar` in GNU binutils and Elfutils have an additional "thin archive" format with the magic number `!<thin>`. A thin archive only contains a symbol table and references to the file. The file format is essentially a System V format archive where every file is stored without the data sections. Every filename is stored as a "long" filename and they are to be resolved as if they were symbolic links.[17]

Example usage

To create an archive from files `class1.o`, `class2.o`, `class3.o`, the following command would be used:

```
ar rcs libclass.a class1.o class2.o class3.o
```

Unix linkers, usually invoked through the C compiler `cc`, can read `ar` files and extract object files from them, so if `libclass.a` is an archive containing `class1.o`, `class2.o` and `class3.o`, then

```
cc main.c libclass.a
```

or (if libclass.a is placed in standard library path, like `/usr/local/lib`)

```
cc main.c -lclass
```

or (during linking)

```
ld ... main.o -lclass ...
```

is the same as:

```
cc main.c class1.o class2.o class3.o
```

See also

- .deb
- Archive formats
- List of Unix commands

References

1. "application/x-archive" (https://reposcope.com/mimetype/application/x-archive). Retrieved 2019-03-11.
2. "ar(1) - Linux man page" (http://linux.die.net/man/1/ar). Retrieved 3 October 2013.

3. "Static Libraries" (http://tldp.org/HOWTO/Program-Library-HOWTO/static-libraries.html). TLDP. Retrieved 3 October 2013.
4. Linux Standard Base Core Specification, version 4.1, Chapter 15. Commands and Utilities > ar (http://refspecs.linuxfoundation.org/LSB_4.1.0/LSB-Core-generic/LSB-Core-generic/ar.html)
5. Levine, John R. (2000) [October 1999]. "Chapter 6: Libraries" (https://archive.today/20161010051456/http://www.iecc.com/linker/linker06.html). *Linkers and Loaders* (https://www.iecc.com/linker/). The Morgan Kaufmann Series in Software Engineering and Programming (1 ed.). San Francisco, USA: Morgan Kaufmann. ISBN 1-55860-496-0. OCLC 42413382 (https://www.worldcat.org/oclc/42413382). Archived (https://archive.today/20121205032107/http://www.iecc.com/linker/) from the original on 2012-12-05. Retrieved 2020-01-12. Code: [1] (https://archive.today/20200114225034/https://linker.iecc.com/code.html)[2] (ftp://ftp.iecc.com/pub/linker/) Errata: [3] (https://linker.iecc.com/)
6. Manual page for NET/2 ar file format (http://www.freebsd.org/cgi/man.cgi?query=ar&sektion=5&manpath=4.3BSD+NET%2F2)
7. "ar.h" (http://www.unix.com/man-page/opensolaris/3head/ar.h/). *www.unix.com*. The UNIX and Linux Forums.
8. "bminor/binutils-gdb: archive.c" (https://github.com/bminor/binutils-gdb/blob/master/bfd/archive.c). *GitHub*.
9. `ar (https://www.opengroup.org/onlinepubs/9699919799/utilities/ar.html)` – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group
10. Manual page for NET/2 ranlib utility (http://www.freebsd.org/cgi/man.cgi?query=ranlib&manpath=4.3BSD+NET%2F2)
11. Manual page for NET/2 ranlib file format (http://www.freebsd.org/cgi/man.cgi?query=ranlib&sektion=5&manpath=4.3BSD+NET%2F2)
12. "ranlib.h" (https://opensource.apple.com/source/Libc/Libc-320/include/ranlib.h.auto.html). *opensource.apple.com*.
13. `ar(5) (https://www.freebsd.org/cgi/man.cgi?query=ar&sektion=5)` – FreeBSD File Formats Manual
14. "ar.h(3HEAD)" (https://docs.oracle.com/cd/E36784_01/html/E36873/ar.h-3head.html). *docs.oracle.com*. Oracle Corporation. 11 November 2014. Retrieved 14 November 2018.
15. Pietrek, Matt (April 1998), "Under The Hood" (https://web.archive.org/web/20070624034732/https://www.microsoft.com/msj/0498/hood0498.aspx), *Microsoft Systems Journal*, archived from the original (http://www.microsoft.com/msj/0498/hood0498.aspx) on 2007-06-24, retrieved 2014-08-23
16. "llvm-mirror/llvm: archive.cpp (format detection)" (https://github.com/llvm-mirror/llvm/blob/ad1310e/lib/Object/Archive.cpp#L593-L610). *GitHub*. Retrieved 10 February 2020.
17. "ar" (https://sourceware.org/binutils/docs/binutils/ar.html). *GNU Binary Utilities*.

External links

- `ar (https://www.opengroup.org/onlinepubs/9699919799/utilities/ar.html)`: create and maintain library archives – Commands & Utilities Reference, The Single UNIX Specification, Issue 7 from The Open Group
- `ar(1) (https://9p.io/magic/man2html/1/ar)` – Plan 9 Programmer's Manual, Volume 1
- `ar(1) (http://man.cat-v.org/inferno/1/ar)` – Inferno General commands Manual
- `ar(1) (https://linux.die.net/man/1/ar)` – Linux User Commands Manual
- `ar(1) (https://www.freebsd.org/cgi/man.cgi?query=ar&sektion=1)` – FreeBSD General Commands Manual
- `ar(1) (http://man.cat-v.org/unix_7th/1/ar)` – Version 7 Unix Programmer's Manual

- `ar(5) (https://www.freebsd.org/cgi/man.cgi?query=ar&sektion=5)` – FreeBSD File Formats Manual -- an account of Unix formats
- *The 32-bit PA-RISC Run-time Architecture Document, HP-UX 11.0 Version 1.0,* Hewlett-Packard, 1997.
  See *Chapter 4: Relocatable Libraries*. Available at [4] (https://web.archive.org/web/20110607140011/http://h21007.www2.hp.com/portal/StaticDownload?attachment_ciid=35ae0c53071b7110VgnVCM100000275d6e10RCRD&ciid=a0f9483d2d2b7110VgnVCM100000275d6e10RCRD) (devresource.hp.com)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Ar_(Unix)&oldid=1049238784"