NAME

objdump − display information from object files.

SYNOPSIS

objdump [**−a**|**−−archive−headers**]
[**−b** *bfdname*|**−−target=***bfdname*]
[**−C**|**−−demangle**[*=style*]]
[**−d**|**−−disassemble**]
[**−D**|**−−disassemble−all**]
[**−z**|**−−disassemble−zeroes**]
[**−EB**|**−EL**|**−−endian=**{big | little }]
[**−f**|**−−file−headers**]
[**−−file−start−context**]
[**−g**|**−−debugging**]
[**−h**|**−−section−headers**|**−−headers**]
[**−i**|**−−info**]
[**−j** *section*|**−−section=***section*]
[**−l**|**−−line−numbers**]
[**−S**|**−−source**]
[**−m** *machine*|**−−architecture=***machine*]
[**−M** *options*|**−−disassembler−options=***options*]
[**−p**|**−−private−headers**]
[**−r**|**−−reloc**]
[**−R**|**−−dynamic−reloc**]
[**−s**|**−−full−contents**]
[**−G**|**−−stabs**]
[**−t**|**−−syms**]
[**−T**|**−−dynamic−syms**]
[**−x**|**−−all−headers**]
[**−w**|**−−wide**]
[**−−start−address=***address*]
[**−−stop−address=***address*]
[**−−prefix−addresses**]
[**−−[no−]show−raw−insn**]
[**−−adjust−vma=***offset*]
[**−V**|**−−version**]
[**−H**|**−−help**]
objfile...

DESCRIPTION

objdump displays information about one or more object files. The options control what particular information to display. This information is mostly useful to programmers who are working on the compilation tools, as opposed to programmers who just want their program to compile and work.

objfile... are the object files to be examined. When you specify archives, **objdump** shows information on each of the member object files.

OPTIONS

The long and short forms of options, shown here as alternatives, are equivalent. At least one option from the list **−a,−d,−D,−f,−g,−G,−h,−H,−p,−r,−R,−S,−t,−T,−V,−x** must be given.

−a
−−archive−header
If any of the *objfile* files are archives, display the archive header information (in a format similar to **ls −l**). Besides the information you could list with **ar tv**, **objdump −a** shows the object file format of each archive member.

--adjust-vma=*offset*
When dumping information, first add *offset* to all the section addresses. This is useful if the section addresses do not correspond to the symbol table, which can happen when putting sections at particular addresses when using a format which can not represent section addresses, such as a.out.

-b *bfdname*
--target=*bfdname*
Specify that the object-code format for the object files is *bfdname*. This option may not be necessary; *objdump* can automatically recognize many formats.

For example,

```
objdump -b oasys -m vax -h fu.o
```

displays summary information from the section headers (**-h**) of *fu.o*, which is explicitly identified (**-m**) as a VAX object file in the format produced by Oasys compilers. You can list the formats available with the **-i** option.

-C
--demangle[=*style***]**
Decode (*demangle*) low-level symbol names into user-level names. Besides removing any initial underscore prepended by the system, this makes C++ function names readable. Different compilers have different mangling styles. The optional demangling style argument can be used to choose an appropriate demangling style for your compiler.

-g
--debugging
Display debugging information. This attempts to parse debugging information stored in the file and print it out using a C like syntax. Only certain types of debugging information have been implemented. Some other types are supported by **readelf -w**.

-d
--disassemble
Display the assembler mnemonics for the machine instructions from *objfile*. This option only disassembles those sections which are expected to contain instructions.

-D
--disassemble-all
Like **-d**, but disassemble the contents of all sections, not just those expected to contain instructions.

--prefix-addresses
When disassembling, print the complete address on each line. This is the older disassembly format.

-EB
-EL
--endian={big|little}
Specify the endianness of the object files. This only affects disassembly. This can be useful when disassembling a file format which does not describe endianness information, such as S-records.

-f
--file-headers
Display summary information from the overall header of each of the *objfile* files.

--file-start-context
Specify that when displaying interlisted source code/disassembly (assumes **-S**) from a file that has not yet been displayed, extend the context to the start of the file.

-h
--section-headers
--headers
Display summary information from the section headers of the object file.

File segments may be relocated to nonstandard addresses, for example by using the **−Ttext**, **−Tdata**, or **−Tbss** options to **ld**. However, some object file formats, such as a.out, do not store the starting address of the file segments. In those situations, although **ld** relocates the sections correctly, using **objdump −h** to list the file section headers cannot show the correct addresses. Instead, it shows the usual addresses, which are implicit for the target.

−H
−−help
Print a summary of the options to **objdump** and exit.

−i
−−info
Display a list showing all architectures and object formats available for specification with **−b** or **−m**.

−j *name*
−−section=*name*
Display information only for section *name*.

−l
−−line−numbers
Label the display (using debugging information) with the filename and source line numbers corresponding to the object code or relocs shown. Only useful with **−d**, **−D**, or **−r**.

−m *machine*
−−architecture=*machine*
Specify the architecture to use when disassembling object files. This can be useful when disassembling object files which do not describe architecture information, such as S−records. You can list the available architectures with the **−i** option.

−M *options*
−−disassembler−options=*options*
Pass target specific information to the disassembler. Only supported on some targets.

If the target is an ARM architecture then this switch can be used to select which register name set is used during disassembler. Specifying **−M reg-name-std** (the default) will select the register names as used in ARM's instruction set documentation, but with register 13 called 'sp', register 14 called 'lr' and register 15 called 'pc'. Specifying **−M reg-names-apcs** will select the name set used by the ARM Procedure Call Standard, whilst specifying **−M reg-names-raw** will just use **r** followed by the register number.

There are also two variants on the APCS register naming scheme enabled by **−M reg-names-atpcs** and **−M reg-names-special-atpcs** which use the ARM/Thumb Procedure Call Standard naming conventions. (Either with the normal register names or the special register names).

This option can also be used for ARM architectures to force the disassembler to interpret all instructions as Thumb instructions by using the switch **−−disassembler−options=force−thumb**. This can be useful when attempting to disassemble thumb code produced by other compilers.

For the x86, some of the options duplicate functions of the **−m** switch, but allow finer grained control. Multiple selections from the following may be specified as a comma separated string. **x86−64**, **i386** and **i8086** select disassembly for the given architecture. **intel** and **att** select between intel syntax mode and AT&T syntax mode. **addr32**, **addr16**, **data32** and **data16** specify the default address size and operand size. These four options will be overridden if **x86−64**, **i386** or **i8086** appear later in the option string. Lastly, **suffix**, when in AT&T mode, instructs the disassembler to print a mnemonic suffix even when the suffix could be inferred by the operands.

For PPC, **booke**, **booke32** and **booke64** select disassembly of BookE instructions. **32** and **64** select PowerPC and PowerPC64 disassembly, respectively.

For MIPS, this option controls the printing of register names in disassembled instructions. Multiple selections from the following may be specified as a comma separated string, and invalid options are

ignored:

`gpr-names=`*`ABI`*
Print GPR (general-purpose register) names as appropriate for the specified ABI. By default, GPR names are selected according to the ABI of the binary being disassembled.

`fpr-names=`*`ABI`*
Print FPR (floating-point register) names as appropriate for the specified ABI. By default, FPR numbers are printed rather than names.

`cp0-names=`*`ARCH`*
Print CP0 (system control coprocessor; coprocessor 0) register names as appropriate for the CPU or architecture specified by *ARCH*. By default, CP0 register names are selected according to the architecture and CPU of the binary being disassembled.

`hwr-names=`*`ARCH`*
Print HWR (hardware register, used by the `rdhwr` instruction) names as appropriate for the CPU or architecture specified by *ARCH*. By default, HWR names are selected according to the architecture and CPU of the binary being disassembled.

`reg-names=`*`ABI`*
Print GPR and FPR names as appropriate for the selected ABI.

`reg-names=`*`ARCH`*
Print CPU-specific register names (CP0 register and HWR names) as appropriate for the selected CPU or architecture.

For any of the options listed above, *ABI* or *ARCH* may be specified as **numeric** to have numbers printed rather than names, for the selected types of registers. You can list the available values of *ABI* and *ARCH* using the **--help** option.

-p
--private-headers
Print information that is specific to the object file format. The exact information printed depends upon the object file format. For some object file formats, no additional information is printed.

-r
--reloc
Print the relocation entries of the file. If used with **-d** or **-D**, the relocations are printed interspersed with the disassembly.

-R
--dynamic-reloc
Print the dynamic relocation entries of the file. This is only meaningful for dynamic objects, such as certain types of shared libraries.

-s
--full-contents
Display the full contents of any sections requested.

-S
--source
Display source code intermixed with disassembly, if possible. Implies **-d**.

--show-raw-insn
When disassembling instructions, print the instruction in hex as well as in symbolic form. This is the default except when **--prefix-addresses** is used.

--no-show-raw-insn
When disassembling instructions, do not print the instruction bytes. This is the default when **--prefix-addresses** is used.

−G
−−stabs
Display the full contents of any sections requested. Display the contents of the .stab and .stab.index and .stab.excl sections from an ELF file. This is only useful on systems (such as Solaris 2.0) in which `.stab` debugging symbol-table entries are carried in an ELF section. In most other file formats, debugging symbol-table entries are interleaved with linkage symbols, and are visible in the **−−syms** output.

−−start−address=*address*
Start displaying data at the specified address. This affects the output of the **−d**, **−r** and **−s** options.

−−stop−address=*address*
Stop displaying data at the specified address. This affects the output of the **−d**, **−r** and **−s** options.

−t
−−syms
Print the symbol table entries of the file. This is similar to the information provided by the **nm** program.

−T
−−dynamic−syms
Print the dynamic symbol table entries of the file. This is only meaningful for dynamic objects, such as certain types of shared libraries. This is similar to the information provided by the **nm** program when given the **−D** (**−−dynamic**) option.

−V
−−version
Print the version number of **objdump** and exit.

−x
−−all−headers
Display all available header information, including the symbol table and relocation entries. Using **−x** is equivalent to specifying all of **−a −f −h −r −t**.

−w
−−wide
Format some lines for output devices that have more than 80 columns. Also do not truncate symbol names when they are displayed.

−z
−−disassemble−zeroes
Normally the disassembly output will skip blocks of zeroes. This option directs the disassembler to disassemble those blocks, just like any other data.

SEE ALSO

nm (1), *readelf* (1), and the Info entries for *binutils*.

COPYRIGHT