Using the Assembler Command Line A

This chapter describes how to invoke the assembler and use the command-line options.

This chapter is organized as follows:

Assembler Command Line	*page 167*
Assembler Command Line Options	*page 168*
Disassembling Object Code	*page 169*

Assembler Command Line

Invoke the assembler command line as follows:

```
as [options] [inputfile] ...
```

Note – The language drivers (such as *cc* and *f77*) invoke the assembler command line with the `fbe` command. You can use either the `as` or `fbe` command to invoke the assembler command line.

The `as` command translates the assembly language source files, *inputfile*, into an executable object file, *objfile*. The Intel assembler recognizes the file name argument *hyphen* (-) as the standard input. It accepts more than one file name on the command line. The input file is the concatenation of all the specified files. If an invalid option is given or the command line contains a syntax error, the Intel assembler prints the error (including a synopsis of the command line syntax and options) to standard error output, and then terminates.

The Intel assembler supports `#define` macros, `#include` files, and symbolic substitution through use of the C preprocessor *cpp*. The assembler invokes the preprocessor before assembly begins if it has been specified from the command line as an option. (See the `-P` option.)

Assembler Command Line Options

`-D`*name*
`-D`*name=def*

When the `-P` option is in effect, these options are passed to the *cpp* preprocessor without interpretation by the `as` command; otherwise, they are ignored.

`-I`*path*

When the `-P` option is in effect, this option is passed to the *cpp* preprocessor without interpretation by the `as` command; otherwise, it is ignored.

`-m`

This new option runs `m4` macro preprocessing on input. The *m4* preprocessor is more powerful than the C preprocessor (invoked by the `-P` option), so it is more useful for complex preprocessing. See the *SunOS 5.x Reference Manual for x86* for a detailed description of the `m4` macro-processor.

`-o` *outfile*

Takes the next argument as the name of the output file to be produced. By default, the `.s` suffix, if present, is removed from the input file and the `.o` suffix is appended to form the output file name.

`-P`

Run *cpp*, the C preprocessor, on the files being assembled. The preprocessor is run separately on each input file, not on their concatenation. The preprocessor output is passed to the assembler.

`-Q[y|n]`

This new option produces the "assembler version" information in the comment section of the output object file if the `y` option is specified; if the `n` option is specified, the information is suppressed.

`-s`

This new option places all stabs in the `.stabs` section. By default, stabs are placed in `stabs.excl` sections, that are stripped out by the static linker ld during final execution. When the `-s` option is used, stabs remain in the final executable because `.stab` sections are not stripped out by the static linker `ld`.

`-U`*name*

When the `-P` option is in effect, this option is passed to the *cpp* preprocessor without interpretation by the `as` command; otherwise, it is ignored.

`-V`

This option writes the version information on the standard error output.

Disassembling Object Code

The `dis` program is the object code disassembler for ELF. It produces an assembly language listing of the object file. For detailed information about this function, see the `dis`(1) manual page.

A