Check if program was compiled with debug symbols

When running the objdump --syms command, I see much more than "*no symbols*" in the output (at least, for ***kernel objects***).

To check if there's debug info inside the kernel object, you can add the following at the end of the objdump command: | grep debug.

If this string is found, you know the kernel object contains debug information. If not, then it's a "clean" kernel object.

Example of a kernel module I've compiled **without** debug information:

geertvc@jimi:~/mystuff/kernels/linux-3.12.6$ **objdump --syms ./modules/lib/modules/3.12.6/kernel/drivers/i2c/busses/i2c-at91.ko | grep debug**

Example of that same kernel module I've compiled **with** debug information:

geertvc@jimi:~/mystuff/kernels/linux-3.12.6$ **objdump --syms ./modules/lib/modules/3.12.6/kernel/drivers/i2c/busses/i2c-at91.ko | grep debug**

```
00000000 l    d  .debug_frame  00000000 .debug_frame
00000000 l    d  .debug_info   00000000 .debug_info
00000000 l    d  .debug_abbrev 00000000 .debug_abbrev
00000000 l    d  .debug_loc    00000000 .debug_loc
00000000 l    d  .debug_aranges 00000000 .debug_aranges
00000000 l    d  .debug_ranges 00000000 .debug_ranges
00000000 l    d  .debug_line   00000000 .debug_line
00000000 l    d  .debug_str    00000000 .debug_str
00000010 l       .debug_frame  00000000 $d
```

As you can see, the first output returns nothing, while the second output returns lines with debug in it.

Note: in my case, the file command returned me "not stripped" in **both** debug and non-debug case. However, the difference in size of the kernel object was remarkable:

- approx. 16k without debug information
- approx. 137k with debug information

Clearly, the latter version had debug information inside.

My question: is the file command reliable in such cases? From what I've experienced, I rely on the **objdump --syms ... | grep debug c**ommand.