6.2. About debugging symbols

Most programs and libraries are, by default, compiled with debugging symbols included (with gcc option -g).

When debugging a program or library that was compiled with debugging information included, the debugger can give you not only memory addresses but also the names of the routines and variables.

But the inclusion of these debugging symbols enlarges a program or library significantly. To get an idea of the amount of space these symbols occupy, have a look at the following:

- a bash binary with debugging symbols: 1200 KB
- a bash binary without debugging symbols: 480 KB
- Glibc and GCC files (/lib and /usr/lib) with debugging symbols: 87 MB
- Glibc and GCC files without debugging symbols: 16 MB

Sizes may vary a little, depending on which compiler was used and which C library. But when comparing programs with and without debugging symbols, the difference will generally be a factor between 2 and 5.

As most people will probably never use a debugger on their system software, a lot of disk space can be regained by removing these symbols .

To remove debugging symbols from a binary (which must be an a.out or ELF binary), run **`strip --strip-debug filename`**. Wildcards can be used to treat multiple files (use something like **`strip --strip-debug $LFS/tools/bin/*`**).

For your convenience, Chapter 9 includes one simple command to strip all debugging symbols from all programs and libraries on your system. Additional information on optimization can be found in the hint at http://www.linuxfromscratch.org/hints/downloads/files/optimization.txt.