1. Please, see “list” folder of this GIT repo.
2. The scenarios are following:
   1. Need to provide more security. The shared library can be replaced by another one library with malicious code. Thus, someone can get an access to memory of process by injection of shared library.
   2. Protect an app for cases of updated shared library to new version or it’s removal or corruption.
   3. Save non-volatile memory. When the application using several functions of library that have a lot of functions. The linker will cut unused functions and it will save the space.
   4. Execute an application more quickly because invocation of shared library takes some time.
3. Please, see “ls\_l” folder of this GIT repo.
4. The SIGTRAP signal that informs debugger about breakpoint occurred. This signal causes by CPU debug exception. For IBM PC architecture it’s INT 3 IRQ. The debugger’s software modifies executable with replacing opcode of operation under breakpoint to the opcode of INT 3 IRQ.
5. I guess no. I did an experiment to check this just for curiosity. I set the breakpoint by GDB to the shared library, ran it, and ran an application from another console. As result nothing bad was happened.
6. When the user-space process started an IO operation with device and stopped while operation’s completion pending, the current operation on the device will be stopped during execution of another process in the system. The hardware interruption request will be handled by the handler of operation system. Since the interruption handling takes some time the current process of the single CPU system will be suspended. Thus, any process in the system unpredictably impacts to performance of another processes of system nevertheless of their priority.