准备工作

> 如有错误以及图片不清晰等问题请提交 issue，谢谢 ~
>
> 源路径：[https://github.com/HATTER-LONG/glibc_Learnging]

编译带有 Debug 信息的 Glibc

在 Linux 下想要调试 Glibc 源码有很多方法，（以 Ubuntu 为例）可以通过 apt 下载 Glibc 相关的编译信息，配合指定源码目录可以实现调试，缺点是无法尝试修改源码验证学习，因此本文采用编译出 Glibc 源码的方案。

```
# GDB 命令：set substitute-path  /build/glibc-eX1tMB/glibc-2.31 /home/layton/Tools/glibc-2.31

"configurations": [
    {
        "name": "Debug",
        "type": "gdb",
        "request": "launch",
        "target": "/home/layton/WorkSpace/GLibcDebug/build/test/automatedtools_unittests",
        "cwd": "${workspaceRoot}",
        "valuesFormatting": "parseText",
        # 如下 通过 Substitutions 制定映射源码路径
        "pathSubstitutions": {
            "/build/glibc-eX1tMB/glibc-2.31": "/home/layton/Tools/glibc-2.31"
        }
    }
]
```

编译 Glibc 源码：

1. 下载源码（以 Ubuntu 为例）：sudo apt source glibc-source;
2. 解压并创建编译目录，与源码目录同级;

```
❯ ls glibc-* -d
glibc-2.33  glibc-debug
```

3. 进入编译目录进行编译，2.33 源码如果不启用 --enable-obsolete-rpc 会在 install 时缺少 xdr_* 报错失败，注意指定 -O1 关闭优化;

> -O0 编译会失败，由于 glibc 中使用了内敛相关语法，gcc 编译器 -O1 才会开启内联优化。

```
cd glibc-debug

../glibc-2.33/configure --prefix=/home/layton/Tools/glibc-debug/usr --enable-obsolete-rpc --enable-obsolete-nsl --enable-debug=yes CFLAGS="-O1 -g" CPPFLAGS="-O1 -g"

mkdir usr && make all && make install
```

4. 工程中使用 cmake 指定 link_dir;

```
link_directories(/home/layton/Tools/glibc-debug/usr/lib)
```

5. 编译完成后使用 ldd 命令查看动态库;

```
~/WorkSpace/GLibcDebug/build/test main*
❯ ldd automatedtools_unittests
        linux-vdso.so.1 (0x00007ffdc1d4d000)
        libpthread.so.0 => /home/layton/Tools/glibc-debug/usr/lib/libpthread.so.0 (0x00007fa69e86f000)
        libspdlog.so.1 => /usr/lib/libspdlog.so.1 (0x00007fa69e7dd000)
        libfmt.so.7 => /usr/lib/libfmt.so.7 (0x00007fa69e7a6000)
        libc.so.6 => /home/layton/Tools/glibc-debug/usr/lib/libc.so.6 (0x00007fa69e5f5000)
        /lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2 (0x00007fa69e896000)
        libstdc++.so.6 => /usr/lib/libstdc++.so.6 (0x00007fa69e3df000)
        libgcc_s.so.1 => /usr/lib/libgcc_s.so.1 (0x00007fa69e3c2000)
        libm.so.6 => /home/layton/Tools/glibc-debug/usr/lib/libm.so.6 (0x00007fa69e28a000)
```