嵌入式软件开发技术与工具实验报告四

黎炜桁

一、实验目的

- 学会下载安装Linux及相关工具到目标机或目标机仿真环境。
- 学会使用交叉编译器编译应用程序，并能够将可执行程序下载到目标机运行。
- 学会使用交叉调试工具gdb-gdbserver调试应用程序。

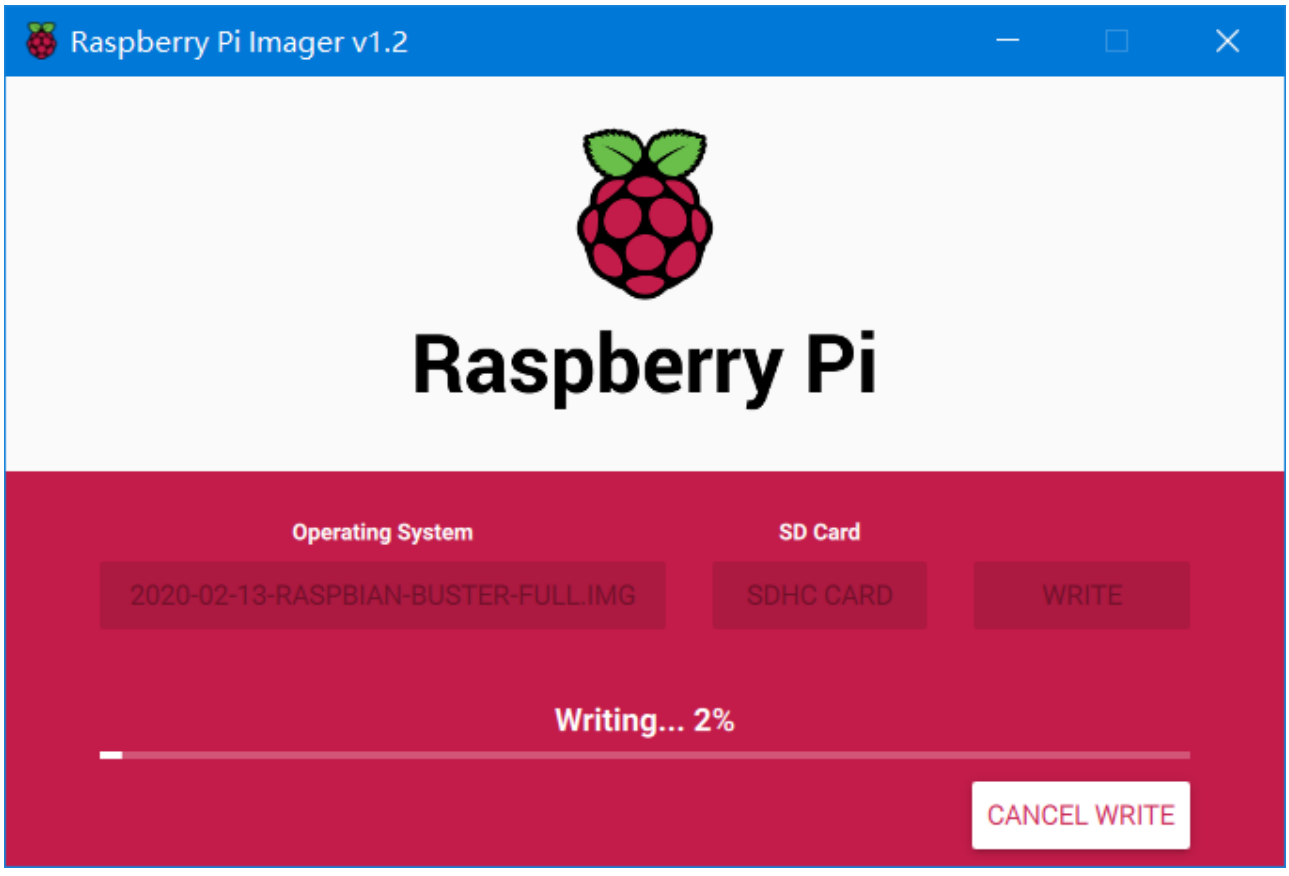
二、实验内容

1. Host-Target方案1：下载安装Linux系统到开发板，其中包括bootloader、内核、模块、库、文件系统等；建立host-target连接，常见连接有SSH，VNC，Putty。
2. Host-Target方案2：下载安装目标机仿真环境qemu，安装qemu树莓派系统。
3. 构建并测试开发板交叉编译器。
4. 建立gdb-gdbserver交叉调试环境并能够调试应用程序。

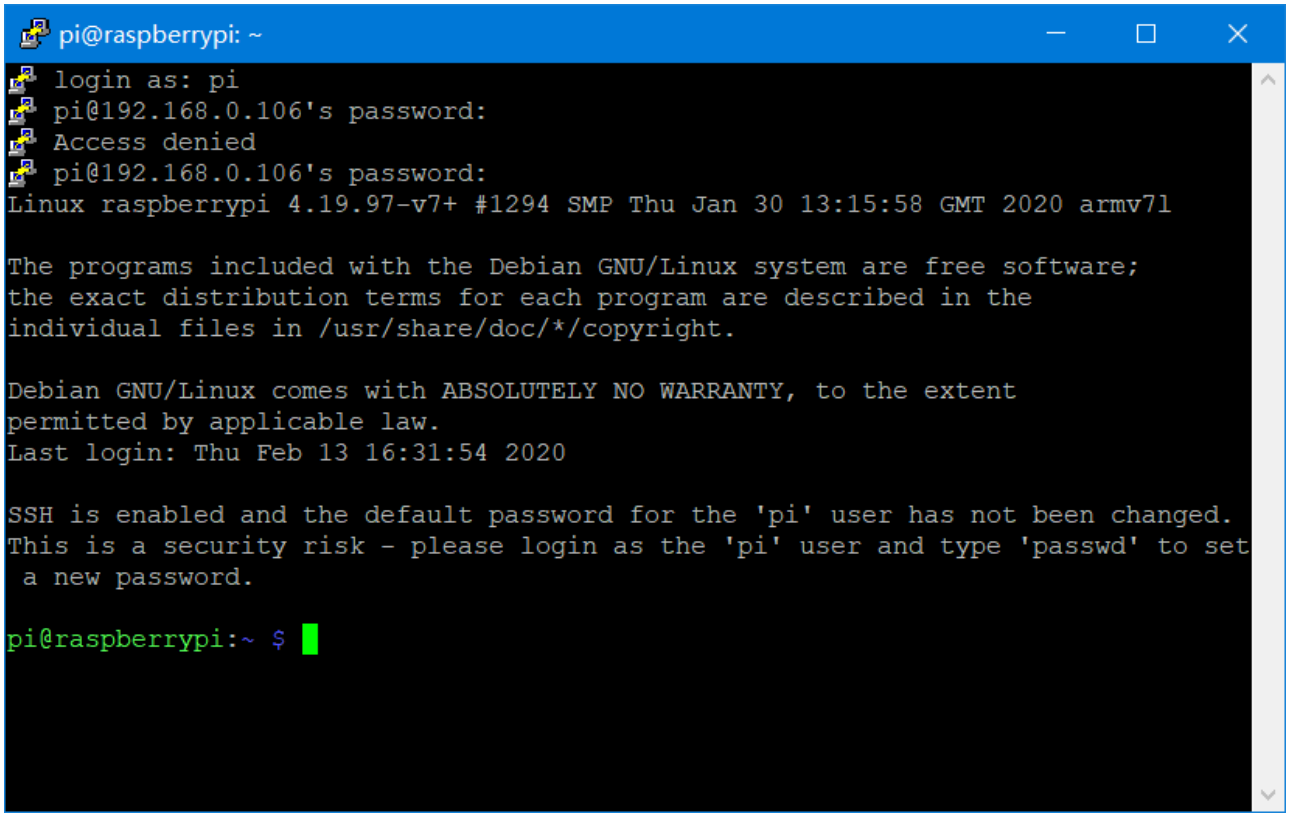
三、实验过程与结果

1. 下载安装Linux系统到开发板

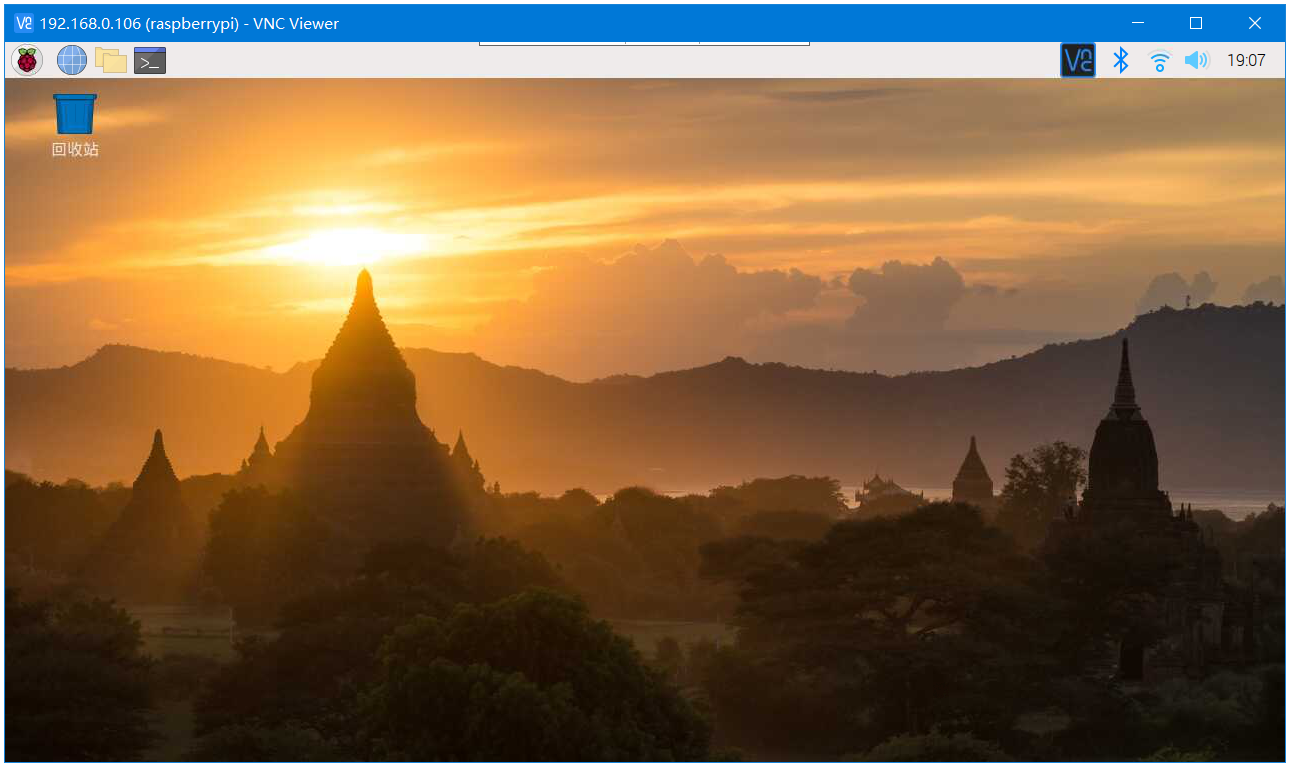
- 从树莓派官网下载Linux镜像，写入SD卡中



- 在SD卡根目录创建ssh文件，使能ssh，使用Putty连接树莓派目标端



- `sudo raspi -config`打开VNC功能，使用VNC viewer远程登陆图形化界面，完成树莓派各项配置



2. 构建并测试开发板交叉编译环境

- 下载交叉编译工具tools.git
- 将编译器路径加入到系统环境变量,并使用`source ~/.bashrc`应用设置。

```
.bashrc (~) - VIM
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# export PATH=/home/liweiheng/rasppi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnue
abihf-raspbian-x64/bin
export PATH=$PATH:/usr/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian
-x64/bin
# export PATH=$PATH:$HOME/rasppi/tools/arm-bcm2708-linux-gnueabi/bin
                                                              121,1         底端
```

- 输入`arm-linux-gnueabihf-gcc -v`查看交叉编译gcc版本，输出正确版本信息，表明交叉编译环境配置成功。

```
root@liwieheng-ThinkPad-X1-Carbon-5th: /usr/tools/arm-bcm2708
arm-linux-gnueabihf-gcov-5
root@liwieheng-ThinkPad-X1-Carbon-5th:/usr/tools/arm-bcm2708# arm-linux-gnueabih
f-gcc -v
Using built-in specs.
COLLECT_GCC=arm-linux-gnueabihf-gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc-cross/arm-linux-gnueabihf/5/lto-wrapper
Target: arm-linux-gnueabihf
Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 5.4.0-6ubu
ntu1~16.04.9' --with-bugurl=file:///usr/share/doc/gcc-5/README.Bugs --enable-lan
guages=c,ada,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-
5 --enable-shared --enable-linker-build-id --libexecdir=/usr/lib --without-inclu
ded-gettext --enable-threads=posix --libdir=/usr/lib --enable-nls --with-sysroot
=/ --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --w
ith-default-libstdcxx-abi=new --enable-gnu-unique-object --disable-libitm --disa
ble-libquadmath --enable-plugin --with-system-zlib --disable-browser-plugin --en
able-java-awt=gtk --enable-gtk-cairo --with-java-home=/usr/lib/jvm/java-1.5.0-gc
j-5-armhf-cross/jre --enable-java-home --with-jvm-root-dir=/usr/lib/jvm/java-1.5
.0-gcj-5-armhf-cross --with-jvm-jar-dir=/usr/lib/jvm-exports/java-1.5.0-gcj-5-ar
mhf-cross --with-arch-directory=arm --with-ecj-jar=/usr/share/java/eclipse-ecj.j
ar --disable-libgcj --enable-objc-gc --enable-multiarch --enable-multilib --disa
ble-sjlj-exceptions --with-arch=armv7-a --with-fpu=vfpv3-d16 --with-float=hard -
-with-mode=thumb --disable-werror --enable-multilib --enable-checking=release --
build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=arm-linux-gnueabihf --pr
ogram-prefix=arm-linux-gnueabihf- --includedir=/usr/arm-linux-gnueabihf/include
Thread model: posix
gcc version 5.4.0 20160609 (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.9)
root@liwieheng-ThinkPad-X1-Carbon-5th:/usr/tools/arm-bcm2708#
```

- 建立hello.c文件，验证交叉编译正确性

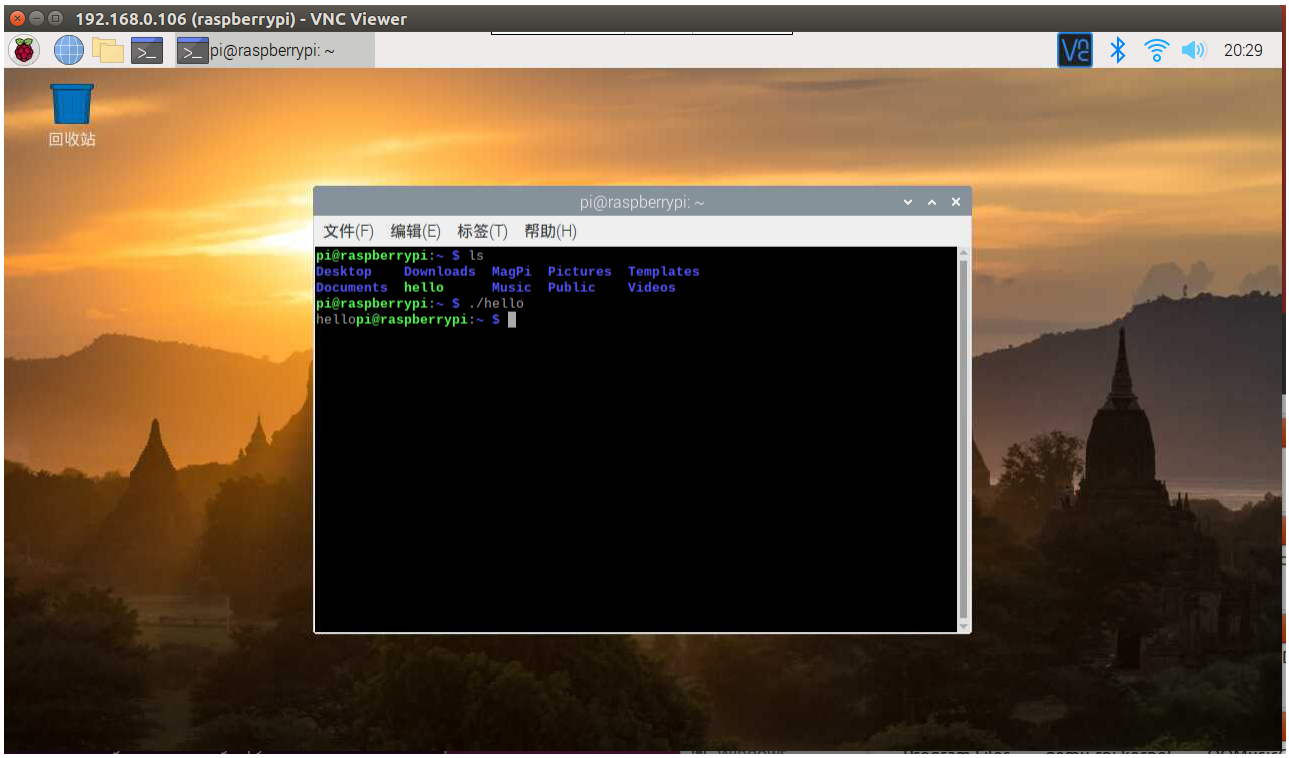
```
#include<stdio.h>

int main(){
```

```
        printf("hello");
        return 0;
    }
```

- 使用交叉编译工具进行编译，获得输出文件hello

```
liwieheng@liwieheng-ThinkPad-X1-Carbon-5th:~/rasppi$ arm-linux-gnueabihf-gcc hel
lo.c -o hello
liwieheng@liwieheng-ThinkPad-X1-Carbon-5th:~/rasppi$ ls
hello  hello.c
liwieheng@liwieheng-ThinkPad-X1-Carbon-5th:~/rasppi$
```

- 传输至树莓派运行，运行成功

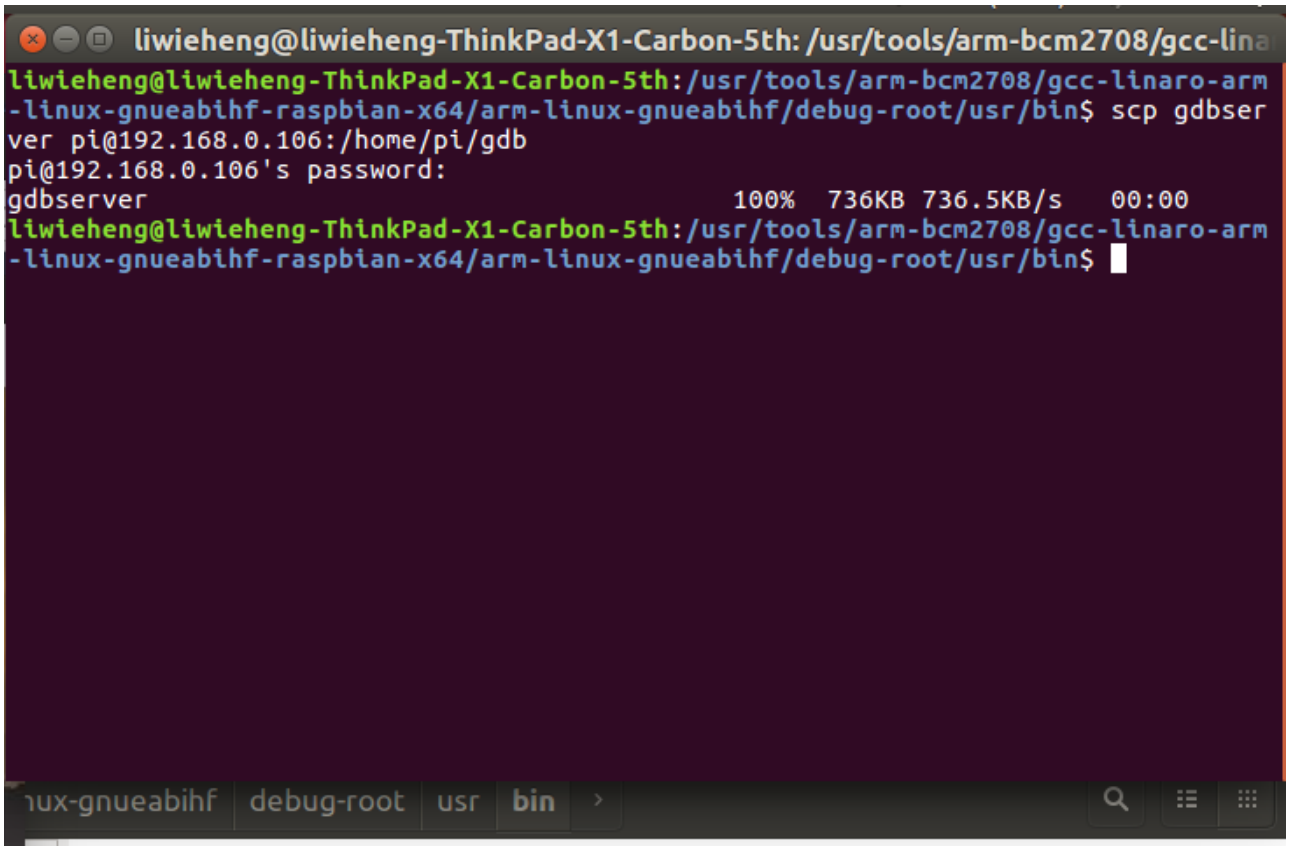


- 总结

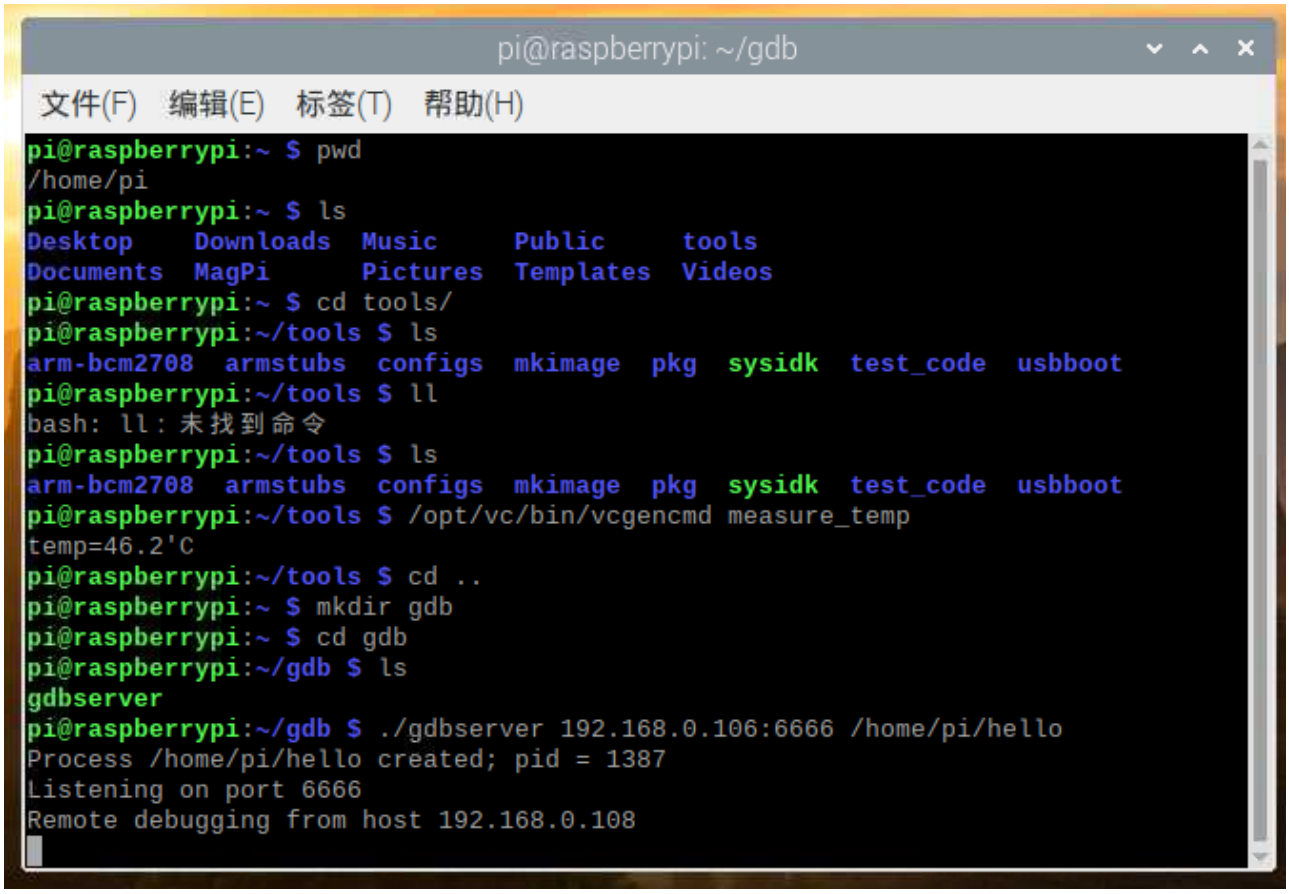
  交叉编译产生的执行文件不可在主机端运行，下载到目标端后运行成功，说明交叉编译工具是针对目标端环境进行编译的。

3. 建立gdb-gdbserver交叉调试环境

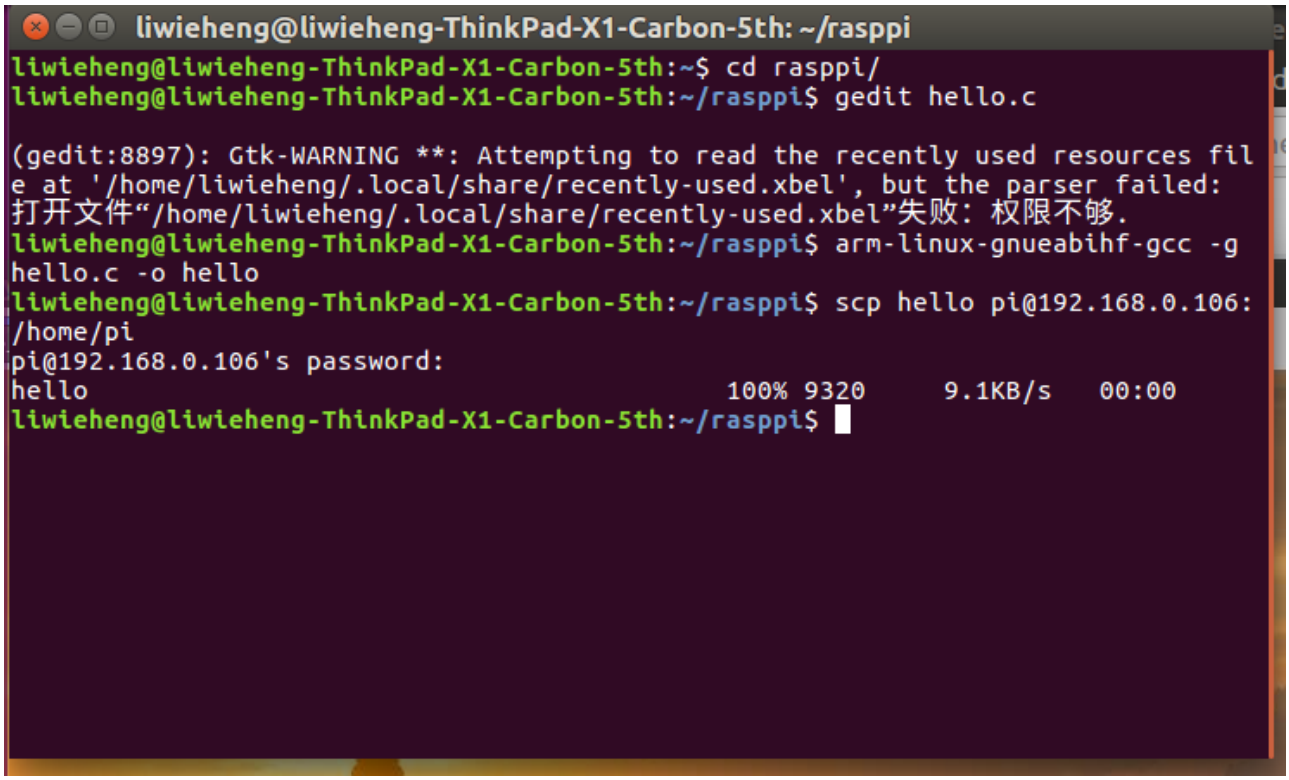
- 下载的交叉编译工具链中已包含了gdb-gdbserver相关工具，位于`tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian-x64/arm-linux-gnueabihf/debug-root/usr/bin` 目录下，可直接下载到树莓派目标端上即可使用



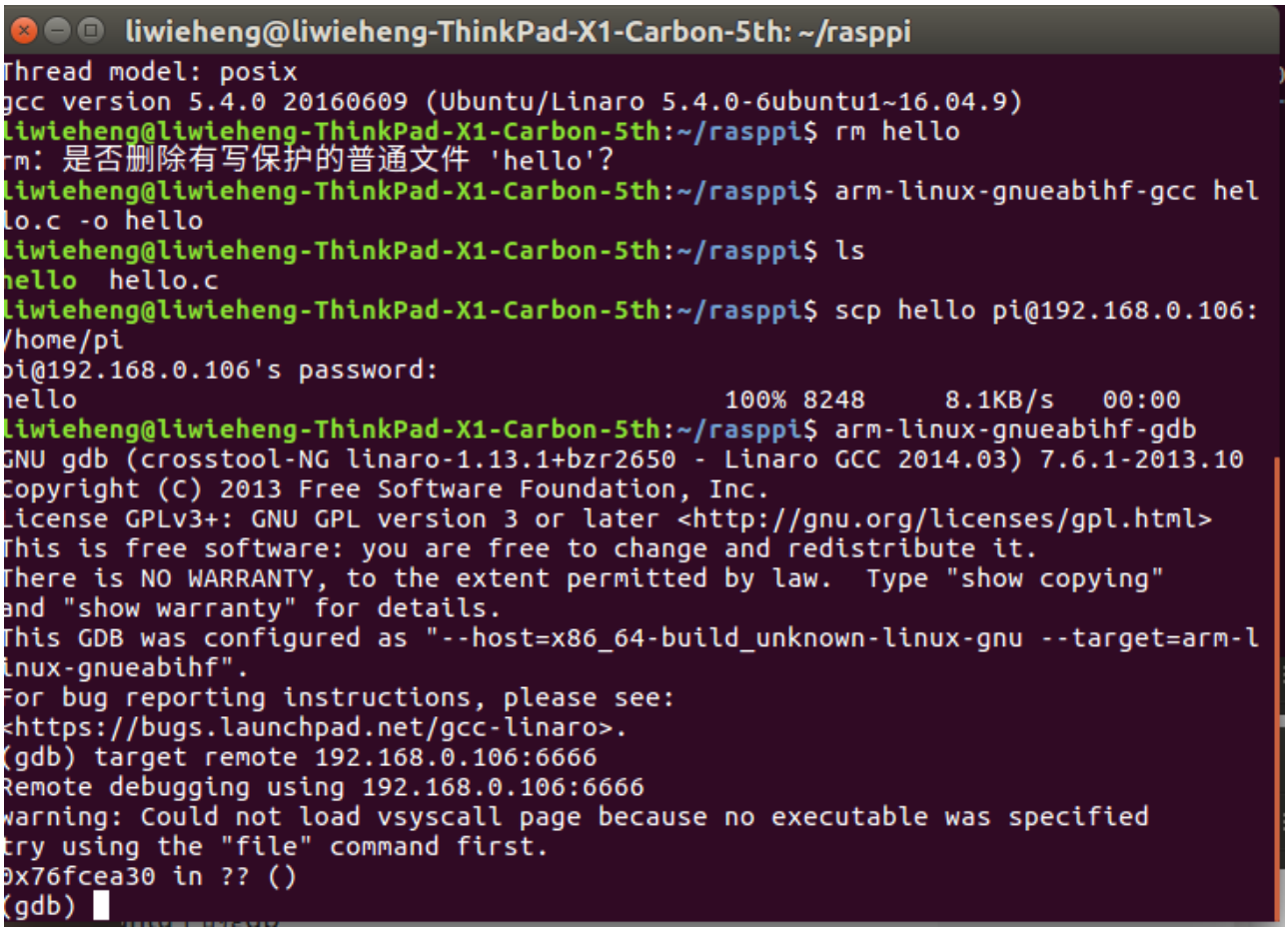
- 树莓派上启动gdbserver，打开6666端口作为调试使用

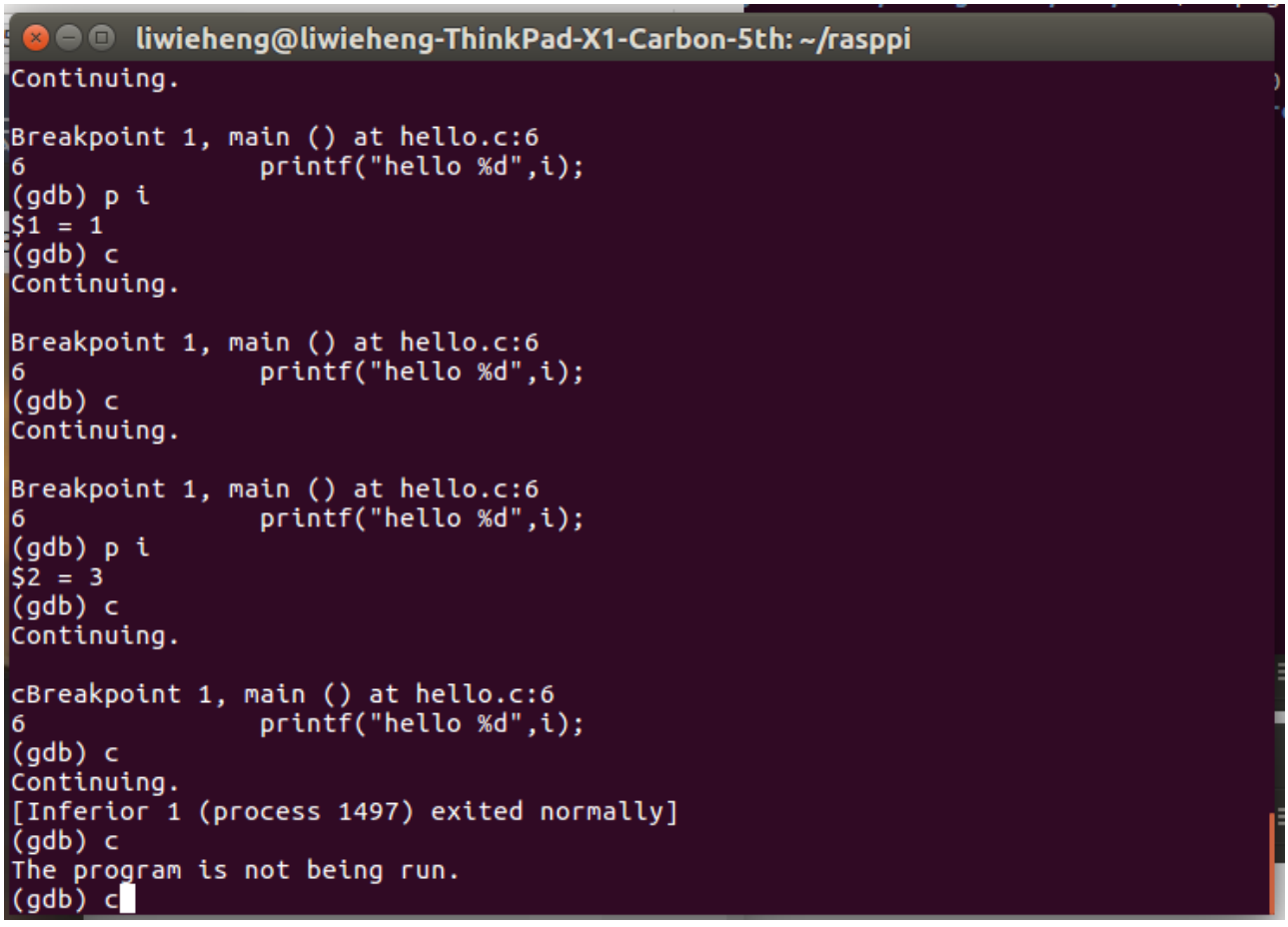


- 主机端Ubuntu使用-g关键字重新编译hello.c，支持gdb调试

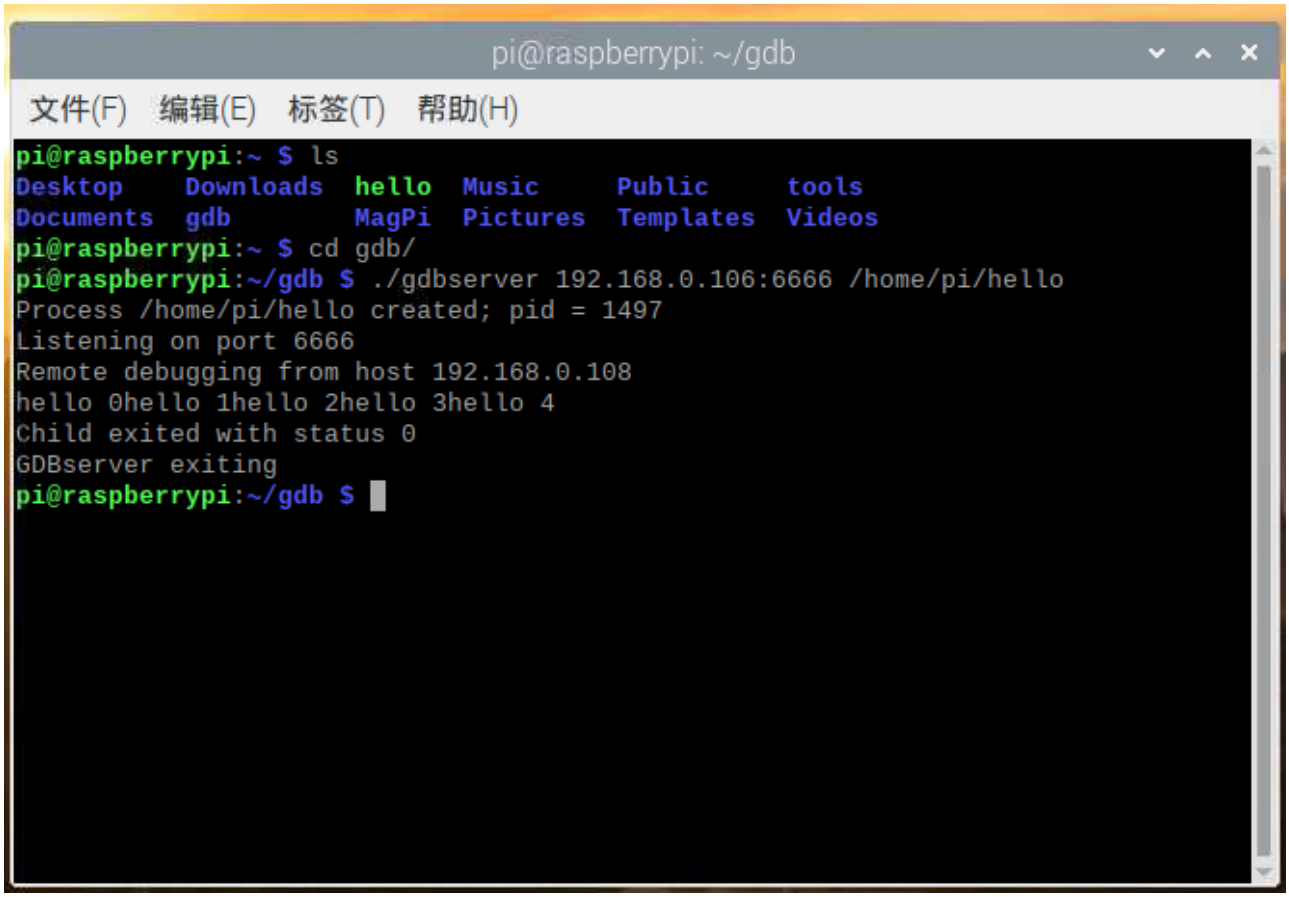


- 主机端Ubuntu启动gdb，调试hello。使用`target remote`命令连接至目标端，`b 4`在第四行加入断点，`c`执行





- 在目标端树莓派gdbserver观察到程序执行成功



- 结论
  使用gdb-gdbserver进行远程调试可以非常方便的在主机端调试目标端程序，并可加入断点，观察变量等，极大的方便了程序的开发。
  需要注意的是在交叉编译时需要加入-g关键字使程序支持gdb调试，否则在gdb中进行调试将无法被识别。

四、实验总结

本次实验我掌握了在树莓派上安装了Linux系统，并实现了主机端与目标端的交叉编译、交叉调试环境，熟练了Linux下的编程。