QEMU和GDB使用总结

方法1（不推荐）

启动：创建两个终端，分别输入：

```
# 终端1，启动qemu
qemu-system-riscv64 \
    -machine virt \
    -nographic \
    -bios default \
    -device loader,file=bin/ucore.img,addr=0x80200000\
    -s -S

# 终端2，启动gdb
riscv64-unknown-elf-gdb \
-ex 'file bin/kernel' \
-ex 'set arch riscv:rv64' \
-ex 'target remote localhost:1234'
```

退出时：

```
# 退出qemu
使用 `Ctrl+a` 再按下 `x` 退出（注意要松开`Ctrl`再单独按`x`）

# 退出gdb
quit
```

- 注1： 在启动qemu时，直接在终端运行下面命令可能报错：

  ```
  qemu-system-riscv64 \
  -machine virt \
  -nographic \
  -bios default \
  -device loader,file=$(UCOREIMG),addr=0x80200000\
  -s -S
  ```

  这是因为

  ```
  file=$(UCOREIMG)
  ```

中，UCOREIMG是一个变量，该变量在Makefile中的定义为：

```
UCOREIMG  := $(call totarget,ucore.img)
```

同理，totarget是一个函数，其定义在tools/function.mk：

```
totarget = $(addprefix $(BINDIR)$(SLASH),$(1))
```

依次类推，将所有变量都替换掉后，命令改为

```
file=bin/ucore.img
```

- 注2： 如果bin/目录下不存在ucore.img文件，在Makefile目录下使用命令

```
make
```

方法二

启动：在Makefile文件目录下，创建两个终端，分别输入：

```
# 终端1，启动qemu
make debug

# 终端2，启动gdb
make gdb
```

退出： 同上

方法三（推荐）

配置方法

1.在Makefile文件中添加

```
test:
    $(V)$(QEMU) \
        -machine virt \
        -nographic \
        -bios default \
        -device loader,file=$(UCOREIMG),addr=0x80200000\
```

```
        -s -S &
    riscv64-unknown-elf-gdb \
    -x init.gdb
```

其中，&表示qemu在后台运行

2.在Makefile目录下创建init.gdb文件，其内容为：

```
# 在退出gdb的时候自动关闭QEMU
define hook-quit
    kill
end

# 与远程连接
target remote localhost:1234
# 加载bin/kernel文件
file bin/kernel
# 设置体系结构为RISC-V的64版本
set arch riscv:rv64
# 执行gdb命令
# 在0x80200000处设置断点
break *0x80200000
# 单步执行一条语句
si
```

其中，“执行gdb命令”下面的内容可根据自己需求进行增删，每次启动gdb的时候，都会自动运行添加的gdb命令，从而便于调试。

使用方法

启动：在Makefile目录下，创建终端，输入：

```
# 同时启动qemu和gdb
make test
```

退出：

```
# 退出gdb后，qemu会自动退出
quit
```