

Lab0.5实验报告

一、实验目的

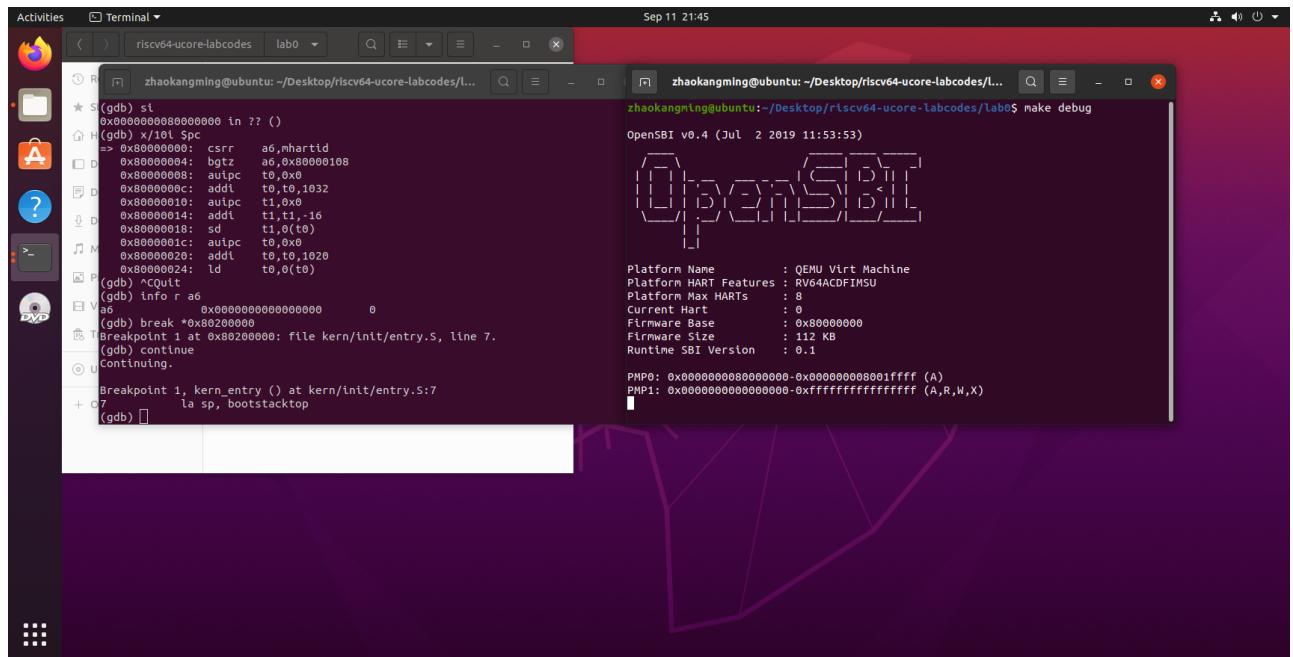
- 练习1: 使用GDB验证启动流程为了熟悉使用qemu和gdb进行调试工作,使用gdb调试QEMU模拟的RISC-V计算机加电开始运行到执行应用程序的第一条指令（即跳转到0x80200000）这个阶段的执行过程，说明RISC-V硬件加电后的几条指令在哪里？完成了哪些功能？要求在报告中简要写出练习过程和回答。

二、实验过程

1.make debug和make gdb后启动调试，首先先观察0x1000处开始后的十条指令

```
(gdb) x/10i 0x1000
=> 0x1000: auipc    t0,0x0
0x1004: addi      a1,t0,32
0x1008: csrr     a0,mhartid
0x100c: ld       t0,24(t0)
0x1010: jr      t0
0x1014: unimp
0x1016: unimp
0x1018: unimp
0x101a: 0x8000
0x101c: unimp
```

- 为什么是1000：resetvec的宏定义的初始值为0x1000然后将resetvec的值赋给了pc 所以pc的初始值为0x1000
- 执行下一条指令：PC为0x1004，info r t0发现t0的值为0x0000000000001000，AUIPC指令的操作是将一个符号扩展的立即数（通常是一个符号偏移量或全局地址）左移12位（即乘以 2^{12} ），然后将当前PC的值加到这个结果上，最终将结果存储到目标寄存器中。因为此时的立即数为0x0，所以将原本的0x1000左边移做符号扩展12位得到这个结果。
- 执行下一条指令：此时PC为0x1008，查看a1的寄存器的值为0x0000000000001020，即t1+0x20.
- 执行下一条指令：PC为0x100C，csrr指令的作用是将当前硬件的id加载到a0的寄存器当中，查看a0寄存器的内容，发现为0，那就是用了0号硬件！
- 执行下一条指令：PC为PC0x1010，ld（Load Doubleword）指令，用于从寄存器t0指向的内存地址偏移24处加载一个双字（64位数据）到寄存器t0中。查看此时t0寄存器的值，为0x0000000080000000
- 执行下一条指令：跳转指令，跳转到内存地址为0x0000000080000000处，而QEMU的复位代码指定加载Bootloader的位置为0x80000000，此时Bootloader将加载操作系统内核并启动操作系统的执行。
- 接下来我们在0x80200000处打断点，然后continue执行到断点处，然后就启动了QEMU预先加载好的内核镜像。



The screenshot shows a terminal window with two panes. The left pane displays a GDB session for a riscv64-ucore-labcodes project. The right pane shows the output of the 'make debug' command, which prints the OpenSBI version and platform information.

```
(gdb) si
0x0000000000000000 in ?? ()
(gdb) x/10i $pc
=> 0x80000000: csrr    a6,mhartid
0x80000004: bgtz    a6,0x80000108
0x80000008: auipc    t0,0x0
0x8000000c: addi     t0,t0,1032
0x80000010: auipc    t1,0x0
0x80000014: addi     t1,t1,-16
0x80000018: sd       t1,0(t0)
0x8000001c: auipc    t0,0x0
0x80000020: addi     t0,t0,1020
0x80000024: ld       t0,0(t0)
(gdb) ^Cquit
(gdb) info r a6
a6             0x0000000000000000      0
(gdb) break *0x80200000
Breakpoint 1 at 0x80200000: file kern/init/entry.S, line 7.
(gdb) continue
Continuing.
Breakpoint 1, kern_entry () at kern/init/entry.S:7
+ C 7
+ la sp, bootstacktop
(gdb) [
```

```
zhaokangming@ubuntu: ~/Desktop/riscv64-ucore-labcodes/Lab0$ make debug
OpenSBI v0.4 (Jul  2 2019 11:53:53)

  OpenSBI
  =====

Platform Name       : QEMU Virt Machine
Platform HARTI Features : RV64ACDFIMSU
Platform Max HARTs   : 8
Current Hart        : 0
Firmware Base       : 0x80000000
Firmware Size       : 112 KB
Runtime SBI Version  : 0.1

PHP0: 0x0000000000000000-0x0000000000000001ffff (A)
PHP1: 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
```