

CS302: Assignment2 Answers

Name: 陆荻芸 SID: 12011537

Answer 1

qemu-system-riscv64: 启动 QEMU 命令并创建 RISC-V64 操作系统的虚拟机。-machine virt: 虚拟机在机器类型 Virt 上运行。-nographic: 不启用图形化界面。-bios default: 指定使用默认的 BIOS。-device loader,file=bin/ucore.bin,addr=0x80200000: 使用 loader 设备, 加载硬盘文件 ucore.bin 至内存地址 0x80200000 处, 使虚拟机得以启动。

Answer 2

这个片段是链接器脚本文件的一部分, 主要目的是描述如何将输入文件中的部分映射到输出文件中。SECTIONS 关键字后面跟着一系列符号赋值和用花括号括起来的输出节描述。第一行指定内核将被加载到 BASE_ADDRESS 的内存地址中。第二行的 .text 指定了可执行文件加载的地方。* 是通配符, 满足后面指定模式的文件会被包括进这一输出节。第三行的 PROVIDE 将当前地址赋值给了定义的符号 etext, 用来记录 .text 这一节的结束。第四行 .rodata 指定的是只读文件对应的映射。符合后面通配符指定格式的文件将会被包括进这一节的输出。第五行在调整下一页数据节的地址, 将下一节的内存地址设置为与下一页对齐。

Answer 3

memset() 是一个 C/C++ 中标准库中的函数, 用来将某一块内存中的内容全部设置为指定的值, 通常为新申请的内存做初始化工作。第一个参数 edata 是指向内存块开始处的指针, 是 kernel.ld 中 data 分区结束位置的内存地址。第二个参数 0 是这个内存块中每个字节的大小。0 指这个内存块将被清空为 0 了。第三个参数 end - edata 指这个内存块中将被替换成大小为 0 (即清除) 的数据大小。这意味着该函数将清除从数据到 end 但不包括 end 的所有内存。

Answer 4

我们需要使用 OpenSBI 提供的接口, 让运行在 m 模式的 OpenSBI 去实现输出。sbi_call 函数实现了使用 ecall 指令发起 sbi 调用请求, 它定义了不同的 sbi_type 可以调用不同功能的 OpenSBI 接口并将 sbi_type 放到寄存器 a7 中。OpenSBI 按照 RISC-V 的 calling convention 会把返回值放到 x10 寄存器里。然后我们再通过内联汇编拿到返回值存入变量中。我们需要向 sbi_call 输入指定字符的 ascii 码作为参数。它会将这个参数放入 a0 寄存器中。sbi_console_putchar 调用了 sbi_call 实现对一个字符的打印, cputs 再通过一层一层调用封装了 sbi_call 的函数来实现字符串的输出。

Answer 5

在 `sbi_shutdown` 定义 `sbi.h`, 然后在 `sbi.c` 中调用 `sbi_call` 实现 shutdown 功能, 并在 `init.c` 中 `includesbi.h` 后直接调用 shutdown 函数。

```

7  void sbi_shutdown();
3

```

(a) `sbi.h`

```

42 void sbi_shutdown(){
43     sbi_call(SBI_SHUTDOWN,0,0,0);
44 }

```

(b) `sbi.c`

```

82 void shutdown(){
83     sbi_shutdown();
84 }

```

(c) `init.c`

图 1: Modified Codes

```

MIDELEG : 0x00000000000000222
MEDELEG : 0x0000000000000b109
PMP0     : 0x0000000080000000-0x000000008001ffff (A)
PMP1     : 0x0000000000000000-0xffffffffffff (A,R,W,X)
os is loading

```

The system will close

○ (base) `ldy12011537@ludiyun-R0G:~/Desktop/OSlab/lab3/lab$` █

图 2: Result