

CSE 536: Advanced Operating Systems

Name: MOHITH SADASHIVA REDDY

ASU ID: 1233347260

Question 1:

- a) Boot ROM is loaded at the address 0x1000
- b) The steps taken by the Boot ROM are as follows:
 - Initializes registers like the stack pointer by pointing it to the base address
 - Loads the necessary registers and configures memory mapping
 - Locates and loads the bootloader
 - Jumps and calls start present in start.c
- c) Boot ROM jumps to the address 0x80000000.

Question 2:

- a) `_entry` is the specified entry function of the bootloader in the linker descriptor.
- b) The `_entry` function can be checked by
 - Establishing breakpoints at the start and end of the Boot ROM
 - We use GDB to run QEMU and single step the instructions and check the pc values
 - We can be certain that the switch from the Boot ROM to the bootloader was successful when you use GDB to get to the bootloader entry point.

Question 3:

- a) If we jump to the C code without setting up the stack the program would most likely fail. The stack is used to store local variables, stack pointer and other functions and if the stack is not present it could disrupt the program functionalities.
- b) We could create 2 stacks, one for each core giving each core its own stack space and point the stack pointer to the appropriate stack region. One stack pointer will

point to the start address of `bl_stack + 4KB` and the other point to the start address of `bl_stack + 8KB`.

Question 4:

- a) Since there is no status initially, we read the value of `mstatus` and then perform a bitwise-and with value of `mstatus` and the bitwise-not of `MSTATUS_MPP_MASK` to set the bit corresponding to M-mode as 0. Then we perform a bitwise-or operation to the above result and `MSTATUS_MPP_S` to set the bit corresponding to S-mode as 1. So, the MPP field in the register gets updated from 0 to 1.

Question 5:

- a) The following is a quick explanation of how `kernel_copy` functions:
- Checks if there is sufficient space in the filesystem to load the kernel.
 - It calculates the `diskaddr`. Then, it calculates the `addr` based on the `diskaddr`.
 - `b->data` is used as the source address. The `addr` is used as the destination address. It retrieves the `BSIZE` value.
 - Then, it calls the `memmove` function, which essentially is `memcpy` to copy the kernel.