CS302 Lab4 Report

> 何泽安 12011323
>
> 2023.3.8

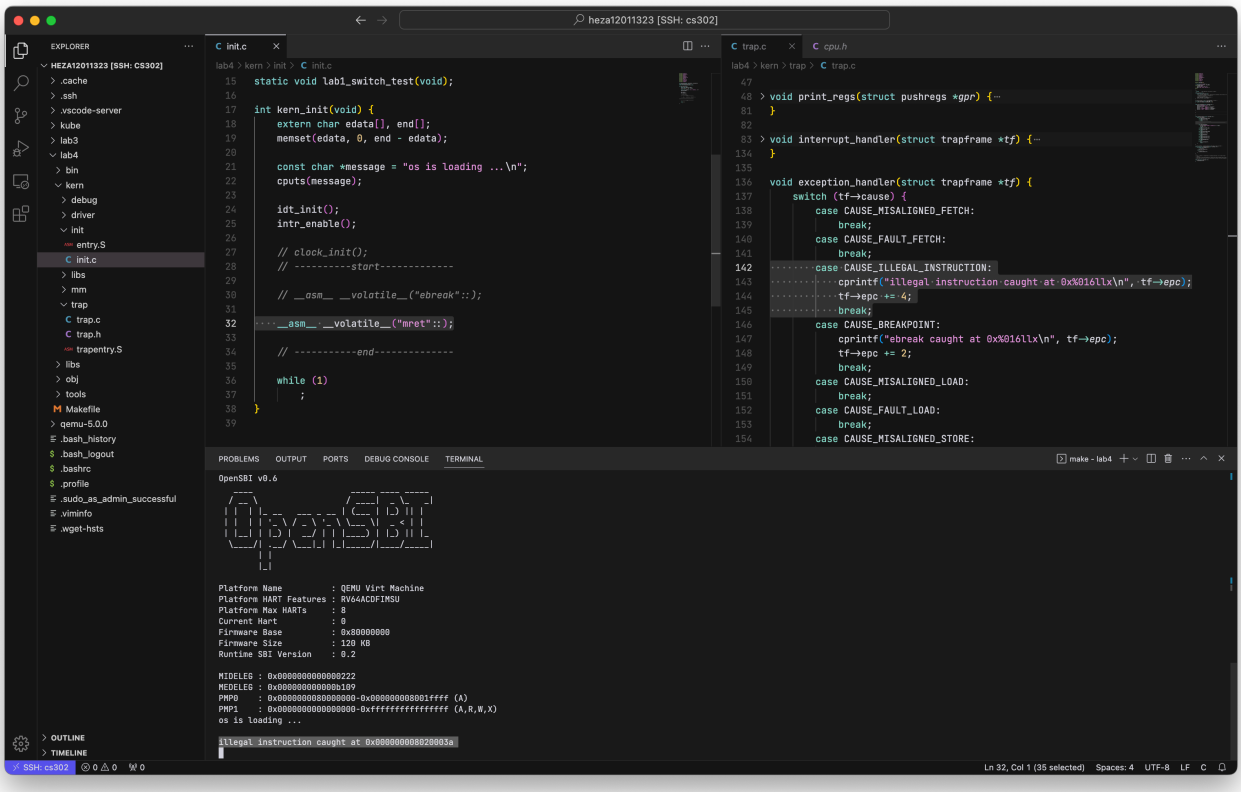
1. **How does the OS handle the breakpoint interrupt**

   In the illustration, we used a `asm volatile("ebreak"::);` to make a breakpoint signal, when catching an interrupt, OS first save the context, then find the approprate interruption handler from the `stvec` and call (jump) the handler method. After the method returns, jump back to the place where interrupted, reload the saved context, and use a `sret` to set PC as *spec*, and go back to the program before entering S mode.

2. **The functionality of `epc` register**

   The `epc` (stands for exception program counter) register stores the address of the instruction that caused the exception. The OS can there by set the PC as the value in epc, such that it can jump back to the program after handling the exception.

3. **Intentially cause a `ILLEGAL_INSTRUCTION` exception, and handle it**



We call `mret` in U mode, witch will cause an illegal instruction exception, and caught by the handler. The handler use the correct hander by switching the cause. We first print a prompt, then increase the epc by 4, since the `mret` instruction is 32 bits long.