1. save context, handle exception/interrupt, restore context and return.
2. When a trap is taken into S-mode, sepc is written with the virtual address of the instruction that was interrupted or that encountered the exception. When return from trap, xRET sets the pc to the value stored in the xepc register.
3.

```
PMP0    : 0x0000000080000000-0x000000008001ffff (A)
PMP1    : 0x0000000000000000-0xffffffffffffffff (A,R,W,X)
os is loading ...

ebreak caught at 0x000000008020003a
start testing illegal instructions
illegal instruction at 0x0000000080200048, instruction 0x30200073end testing illegal instructions
++ setup timer interrupts
100 ticks
100 ticks
100 ticks
100 ticks
100 ticks
100 ticks
```

```
case CAUSE_ILLEGAL_INSTRUCTION:
    cprintf("illegal instruction at 0x%016llx, instruction 0x%08x", tf->epc, *(uint32_t *)tf->epc);
    tf->epc += 4;
    break;
case CAUSE_BREAKPOINT:
```

```
cputs("start testing illegal instructions");
asm volatile("mret"::);
cputs("end testing illegal instructions");
```