

- Step 1
- Step 2: Relocation
 - `.elf` \rightarrow `.img`
 - use obcopy for bytes
 - re-organize to become `.img` file that can be loaded into machine

Kernel Mode

- execution w/ full privileges of the hardware
- read/write to memory, access I/O device, send/read network packet

User Mode

- limited privileges

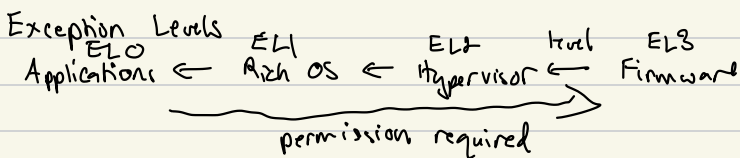
2 hats

- compilation does not care about privileges
- hardware that periodically interrupts processor
- hardware timer \Rightarrow controls interrupts
 - periodically interrupts the processor
- kernel must copy string from user program to screen memory
- interrupts: asynchronous "irq"
 - \hookrightarrow external event happened
- exception: synchronous
 - \hookrightarrow one-time events
 - \hookrightarrow programming errors

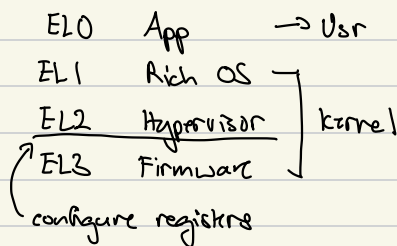
Hardware Support for OS

- RPI3: gpu focused
- 64-bit v. 32-bit
 - ↳ address
- represented in bytes

Exception Levels



- registers = 64-bit
- current EL \leftarrow only accessible EL +



- when exception happens (erret)
 - current processor state to SPSR
 - save return address to ELR
- $\text{ing} \Rightarrow$ exception handler; handles things like timers

b: short jump; relative address

br: long jump; address stored in register

PSState

- condition flags
- interrupt masks: D, A, I, F
- execution state

- Pstate \rightarrow SPSR
- SPSR_ELX \leftarrow saved processor state
- ELR_ELX: saved PC upon taking exception
 - \hookrightarrow nature of instruction determines if current or future address/instruction that is held
- boots from highest exception level