CPU - > registers(if you need more data to calculate -> RAM(Random Access Memory)

RAM -> Bottom part(Physical) of memory - > Zero Page

ZP-> are attached to I/O devices, not working for computation

ROM/BIOS -> running program

Interrupt vector->indexes of Interrupt Vector are responding to I/O devices ordering of interrupt vector is hardware wise ordered. (interrupt service routines)

I/O(Moving data from outside world to Computers or vice versa)

Program counter -> points at the next instruction to be executed.

Stack pointer(Base pointer)->

Frame pointer-> allows to ->

A stack pointer (sp) defines the end of the current frame(top of the stack),

while a frame pointer (fp) defines the end of the last frame

* Both are for making it possible to computing in cpu and navigating

Processor status –

1.

Kernel mode : Kernel bit is set - > PC knows that we are super users

User mode: If it’s not set -> PC knows that we are regular users (no special instructions, I/O, memory access)

Processes(User Space) 🡨-------Kernel Space(CPU):Bridge ------🡪Hardware

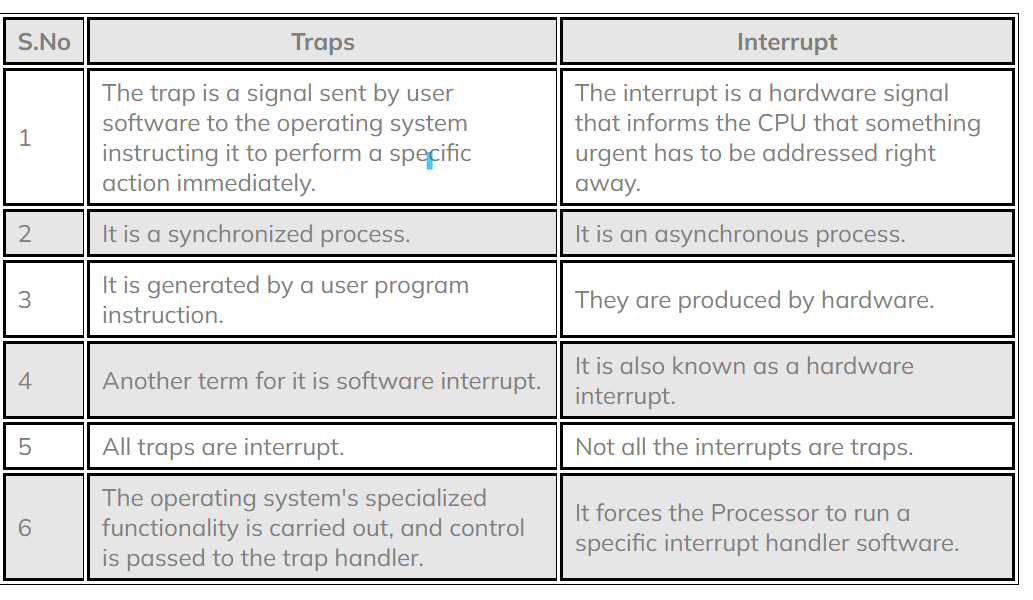
(Trap or Interrupt)

* **Interrupt**

A hardware or software signal that demands immediate attention from the OS. Interrupts are asynchronous events that occur independently of instructions. They are usually triggered by hardware devices, such as a keyboard or timer.

* **Trap**

A type of exception that is triggered by software or hardware. Traps are generally software-initiated, such as system calls or errors. They are synchronous events that can be raised by an instruction



Program on a machine - > Atomic (Either it happened or it didn’t happen. There’s no in between)

3 types of device drivers

BLOCK

->read/write blocks(buffer)

NETWORK

->  
CHARACTER

->

/dev/null -> bit bucket (garbage bin)

/dev/random -> read random numbers