

1) What is Linux IIO subsystem? Please explain IIO in terms of its purpose and user interface provision.

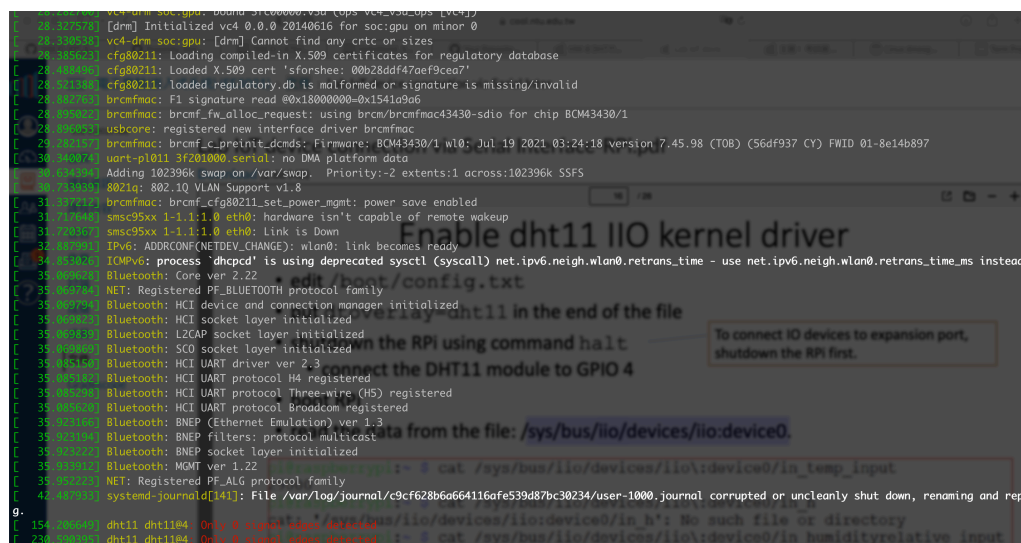
The Linux IIO subsystem is a part of the Linux kernel that provides support for devices that perform I/O operations in an industrial context. This includes a wide range of devices like sensors, ADCs (Analog to Digital Converters), DACs (Digital to Analog Converters), and other similar types of hardware.

The user interface provision in the IIO subsystem is likely through the sysfs (system file system) interface, as indicated by references to "IIO buffer sysfs interface" and "IIO trigger sysfs interface". Sysfs in Linux is a virtual file system that provides a window into the kernel, allowing user-space programs to interact with kernel components.

This subsystem is crucial for applications that require precise and reliable data acquisition and control in industrial settings, making it a key component for developers working with such hardware on Linux systems.

2) How is the efficiency difference when compared between interrupt-driven I/O and programming I/O (polling I/O)?

In programmed I/O, the CPU actively checks (polls) the status of the peripheral device to determine if it's ready for an I/O operation. This involves repeatedly reading the device's status register. On the other hand, In interrupt-driven I/O, the CPU is freed from the task of continuously checking the device status. Instead, the I/O device sends an interrupt signal to the CPU when it's ready for an I/O operation. Therefore, the interrupt-driven I/O allows for better CPU utilization and typically results in more responsive systems.



```

24.177774 [drm] Initialized vc4 0.0.0 20140616 for soc:gpu on minor 0
24.180348 vc4-drm soc:gpu: [drm] Cannot find any crtc or sizes
24.185611 cfg80211: Loading compiled-in X.509 certificates for regulatory database
24.188180 cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
24.191180 cfg80211: loaded regulatory.db is malformed or signature is missing/invalid
24.193747 brcmfmac: F1 signature read @0x18000000=0x1541a9a6
24.196007 brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac43430-sdio for chip BCM43430/1
24.198421 usbcore: registered new interface driver brcmfmac
24.200849 brcmfmac: brcmf_c_preinit_firmware: Firmware: BCM43430/1 wl0: Jul 19 2021 03:24:18 version 7.45.98 (T08) (56df937 CY) FWID 01-8e14b897
24.203274 vport-p1011 3f201000.serial: no DMA platform data
24.205699 Adding 102396k swap on /var/swap. Priority:-2 extents:1 across:102396k SSFS
24.208128 8021q: 802.1Q VLAN Support v1.8
31.179112 brcmfmac: brcmf_cfg80211_set_power_mgmt: power save enabled
31.181648 smsc95xx 1-1.1.1.0 eth0: hardware isn't capable of remote wakeup
31.184176 smsc95xx 1-1.1.1.0 eth0: Link is Down
31.186704 IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
34.170000 ICMPv6: process `dhcpcd' is using deprecated sysctl (sysctl) net.ipv6.neigh.wlan0.retrans_time - use net.ipv6.neigh.wlan0.retrans_time_ms instead
35.169000 Bluetooth: Core ver 2.22
35.169724 NET: Registered PF_BLUETOOTH protocol family
35.169724 Bluetooth: HCI device and connection manager initialized
35.169724 Bluetooth: HCI socket layer initialized
35.169724 Bluetooth: L2CAP socket layer initialized
35.169724 Bluetooth: SCO socket layer initialized
35.169724 Bluetooth: HCI UART driver ver 2.3
35.169724 Bluetooth: HCI UART protocol H4 registered
35.169724 Bluetooth: HCI UART protocol Three-wire (H5) registered
35.169724 Bluetooth: HCI UART protocol Broadcom registered
35.169724 Bluetooth: BNEP (Ethernet Emulation) ver 1.3
35.169724 Bluetooth: BNEP filters: protocol multicast
35.169724 Bluetooth: BNEP socket layer initialized
35.169724 Bluetooth: MGMT ver 1.22
35.169724 NET: Registered PF_ALG protocol family
41.169724 systemd-journald[14]: File /var/log/journal/c9cf628b6a664116afe539d87bc30234/user-1000.journal corrupted or uncleanly shut down, renaming and replacing
154.169724 dht11 dht1104 Only 4 samples taken (4000000)
230.169724 dht11 dht1104 Only 4 samples taken (4000000)

```

We thought something went wrong with either Rpi or DHT11. However, by tracing the source codes, we get to see the difference between interrupt-driven I/O and programming I/O. For interrupt-driven method, when an edge (falling and rising) is detected, a ISR is triggered to store the time stamp and the value. The data is later decoded to obtain the sensor value. On the other hand, the polling method continuously checks for the updates from pin, rendering CPU idle and busy waiting.