Name:Kusuma

Capstone Project: Kernel-Level Multi-System Game Interface Using UART, I2C OLED, and ALSA Audio

This project involves the design and implementation of a kernel-level game interface that communicates with multiple hardware and software subsystems using UART, I2C OLED. The project demonstrates how embedded systems can interact at a low level to create a functional, real-time game interface with sound, display, and serial communication.

Tools & Technologies Used:

Languages: C (Kernel-space programming)

Kernel Subsystems: Input, UART (TTY), I²C, ALSA Audio, Networking (TCP Sockets)

Hardware: Raspberry Pi, I²C OLED Display.

1. Platform Driver Development

Developed and tested drivers for UART, GPIO, I2C, and RTC using the Linux platform driver framework. Gained a strong understanding of how hardware devices interact with the kernel through the platform bus. Explored the complete flow of driver initialization, probing, and device binding during system boot.

Tools used:dmesg, insmod, rmmod, ftrace.

2. Crashdump Analysis

Studied Oops vs Kernel Panic to understand different types of kernel crashes. Used tools like kdump and kexec to analyze the root cause.

Tools used:dmesg, journalctl, crash, addr2line, kdump, kexec.

3. Linux Performance Optimization

Worked on reducing boot time and optimizing kernel memory usage. Used meminfo and analyzed different memory segments to improve performance.

Tools used:meminfo, rootfs.

4. Memory Leak Detection

Simulated memory leaks in the Linux kernel and identified them using kmemleak, which is designed specifically for detecting kernel-space memory issues.

Used valgrind to detect and analyze memory leaks in user-space applications, ensuring efficient memory management outside the kernel.

Tools used: Kernel space tool: kmemleak, User space tool: Valgrind

5. PCIe Driver Development

Did R&D on PCIe architecture and understood how devices are detected and initialized. Wrote a small test driver to interact with a PCIe device.

Tools used: lspci, /sys/bus/pci/devices, /proc/interrupts

Name:Kusuma

6. U-Boot Environment Variables for Kernel and DTB Loading

Studied how U-Boot environment variables define the load addresses and sizes for the Linux kernel and DTB. Learned how boot arguments are passed and how kernel/DTB loading is handled during boot.

Tools used: minicom, cross compiler toolchain, SD card, dmesg, strace.

7. Kernel Bring-up Using Yocto on Raspberry Pi

Configured a minimal Linux image using the Yocto Project and successfully booted it on Raspberry Pi hardware for embedded bring-up.

Integrated and tested a custom kernel driver within the Yocto build, validating functionality in a controlled embedded environment.

Tools used: BitBake, Poky, Git, cross compiler toolchain.

8.Device Tree Enhancement and Kernel Log Analysis for Linux BSP

Enhanced a Linux BSP by updating device tree files and integrating a custom kernel driver to enable persistent crash log capture, real-time system log analysis, and automated error detection using pstore, dmesg, and journalctl.

Tools used:pstore, journalctl, dmesg.

9.Watchdog Timer Driver Development

Working on the implementation of a watchdog timer to monitor system and prevent software hangs.

Additional Practice:

10.Explored embedded debugging tools and gained knowledge on interfaces like JTAG and SWD used for low-level system debugging and development.

11. Practiced Linux Device Driver (LDD) MCQs to strengthen understanding of kernel driver development and internal Linux mechanisms.

12. Practiced logic-based exercises and puzzles to enhance critical thinking and problem-solving abilities.