HARSH RAJ Email: harshraj1695@example.com | Phone: +91 9XXXXXXXXX | LinkedIn: linkedin.com/in/harshraj Location: India

SUMMARY Highly motivated Linux Kernel Developer with 3+ years of experience in writing and debugging character drivers, kernel modules, and low-level system programming. Proficient in device driver development, embedded Linux systems, memory management, concurrency, and performance optimization. Strong understanding of Linux kernel subsystems including VFS, scheduling, interrupt handling, networking stack, and hardware interfaces.

TECHNICAL SKILLS - Programming Languages: C, C++, Python, Bash, Assembly (x86/ARM) - Kernel Development: Linux Kernel 2.6/5.x/6.x, Module Programming, Character Drivers, Block Drivers, PCI/USB Driver Development, Device Nodes, File Operations, Netfilter, NAPI, Tasklets, Soft IRQ - Tools & Utilities: GCC, GDB, Make, Git, QEMU, Valgrind, SystemTap, perf, ftrace, kgdb - Operating Systems: Linux (Ubuntu, Debian, Embedded Linux), Real-Time Linux - Other Skills: Synchronization (mutex, semaphore, spinlock, RCU), Interrupt handling, Memory allocation, Kernel preemption, Thread scheduling, Kernel debugging

EXPERIENCE

Linux Kernel Developer | Personal/Experimental Projects | India | 2024-Present - Developed multiple character and block device drivers for Linux 6.x kernel with dynamic major/minor allocation. - Implemented robust user-space to kernel-space communication using copy\_to\_user and copy\_from\_user, supporting blocking and non-blocking I/O. - Utilized wait queues, mutexes, semaphores, and tasklets to synchronize producer-consumer flows in kernel modules. - Debugged kernel panics, memory leaks, and race conditions using GDB, kmem, ftrace, and dmesg. - Optimized kernel module performance by reducing memory footprint, minimizing context switches, and ensuring safe concurrent access.

Embedded Systems Intern | XYZ Company | India | Jan 2023 – Dec 2023 - Assisted in porting Linux drivers for ARM-based embedded platforms. - Wrote unit tests for device drivers, ensuring robust read/write operations and proper interrupt handling. - Collaborated with senior engineers to resolve kernel preemption and scheduling issues in multi-threaded environments. - Implemented NAPI-based networking modules for improved throughput and CPU efficiency.

EDUCATION Bachelor of Technology in Computer Science | ABC University | 2020 – 2024 - Relevant Coursework: Operating Systems, Data Structures, Embedded Systems, Computer Architecture, Networking, Real-Time Systems

CERTIFICATIONS - Linux Kernel Development Training (LFD420) – The Linux Foundation - C Programming & Embedded Linux Development – Udemy - Advanced Linux Device Driver Development – Coursera

PROJECTS - Linux Character Driver with dynamic device node creation, supporting non-blocking read/write and fasync signaling. - Kernel module simulating producer-consumer using tasklets, soft IRQs, and wait queues. - Embedded Linux system setup with cross-compilation, kernel module debugging, and performance analysis via perf and SystemTap. - Implemented PCI/USB driver modules with interrupt handling, DMA transfers, and memory-mapped I/O.

ADDITIONAL INFORMATION - GitHub: github.com/harshraj1695 - Languages: English (Fluent), Hindi (Native) - Availability: Immediate