

100% Money-Back Guarantee

Advanced Kernel & Embedded Systems Fellowship

A five-month deep immersion in Linux internals, device drivers and real hardware — underwritten by a written refund commitment.

5^{mo}

ACADEMIC SPRINT

12^{mo}

PLACEMENT WINDOW

74

LAB-BACKED TOPICS

100%

TUITION REFUND



01

Program Overview & Guarantee Structure

WHAT THE GUARANTEE IS, AND WHO IT APPLIES TO

COMPONENT	POLICY BREAKDOWN & DEFINITION
Program Title	Advanced Kernel & Embedded Systems Fellowship 5-MONTH DEEP IMMERSION
Target Audience	ECE, EEE and CS graduates, or software engineers transitioning into core engineering roles.
The Core Guarantee	100% tuition fee refund if the student does not receive a qualifying job offer matching the baseline Cost to Company (CTC) within 12 months of the placement window opening.
Contract Inclusions	Covers all sourcing channels. If a student secures a job off-campus or self-sourced — via LinkedIn, a job portal or any other channel — the guarantee is considered fulfilled and no refund is due.
Internal Offers	A direct hire onto Kernel Foundation's own internal engineering projects (e.g. Junior Software Engineer) counts as a qualifying offer where it meets the baseline CTC, and discharges the guarantee in the same way as an external offer.

In one line: clear the evaluation, meet the participation metrics, keep applying — and if no qualifying offer arrives inside twelve months, the tuition comes back in full.

02

Implementation Procedure

ADMISSIONS → ACADEMIC SPRINT → PLACEMENT CLOCK



PHASE 1 · MONTH 0

Selective Admissions

- **Technical filter.** Raw beginners are not accepted. A mandatory entrance test assesses basic C programming, pointers and digital electronics fundamentals.
- **Hardware kit sign-off.** Students must own or acquire the mandatory target hardware (Raspberry Pi 4/5 or BeagleBone Black) so that lab execution is uniform across the cohort.



PHASE 2 · MONTHS 1–5

The Academic Sprint

- **Months 1–2 — Linux Architecture & Core Systems.** Toolchain and ELF internals, x86 and ARM architecture, process creation, memory management, scheduling algorithms, IPC and POSIX multithreading.
- **Months 3–4 — Kernel Internals, Device Drivers & Hardware Interfacing.** Kernel build and boot, init systems, custom character drivers, concurrency primitives, device tree, platform and PCI drivers, and bus subsystems (I2C, SPI, UART).
- **Month 5 — Networking, Debugging & Profiling Capstone.** Network device drivers and socket programming, production-grade driver development, and edge-workload optimisation using tracing and debugging suites such as ftrace, eBPF, KASAN and crash.

Detailed Curriculum — 5 Modules, 74 Topics

Every module is lab-backed. Topic completion and code review are the metrics used for the 85% threshold in Section 3.

1

MONTHS 1–2

Linux System Programming

19 TOPICS

- GCC compilation steps, optimisation, bin utilities
- Compilation stages; static vs shared libraries; environment variables
- Profile-guided optimisation; compiler utilities; ldd, objdump, objcopy
- How programs are launched
- readelf, LLVM; introduction to Linux components
- Linux filesystem hierarchy; mdev
- x86 architecture; big vs little endian; x86 modes and registers
- x86 registers; calling convention; instruction set
- x86 instructions; calling convention; addressing modes; C to assembly
- x86 instruction classes
- GCC assembler directives; C to assembly; inline assembly
- ARM; RISC design; topologies; ARM7TDMI; pipelining
- ARM program counter; exception handling; conditional execution
- Branch instructions; LDM and STM operation; stack; Thumb state
- Filesystem operations and layers
- File operations; profiling with gprof; debugging with strace and GDB
- Process management — fork, exec, system, wait
- IPC — message queues, shared memory, semaphores
- Synchronisation primitives — mutexes, condition variables

2

MONTHS 2–3

Linux Kernel Internals

17 TOPICS

- Kernel versioning and source tree structure
- Kernel configuration and build process
- Boot sequence — BIOS and bootloader stage
- Storage strategy — NOR, NAND, SSD, MMC
- Linux boot sequence and virtual console
- Startup init program — BusyBox
- System V init.d implementation
- systemd architecture and services
- cgroups fundamentals
- cgroups — systemd integration
- D-Bus — signals and policy implementation
- Building UML and QEMU kernels
- Building and running a QEMU ARM kernel image
- Building QEMU ARM BusyBox
- Building and setting up QEMU ARM U-Boot
- Building the Raspberry Pi kernel

3

MONTHS 3–4

Linux Device Drivers

20 TOPICS

- Module programming basics and error handling
- Module dependency and parameters
- Character device drivers — old and new registration
- Device tree — discoverable and non-discoverable devices
- Device tree overlays and platform drivers
- Platform driver implementation
- PCI driver implementation and addressing
- Concurrency primitives — spinlocks, mutexes, atomics
- Interrupt handling — top and bottom halves
- Bus subsystems — I2C, SPI, UART

4

MONTHS 4–5

Network Programming & Network Device Drivers

10 TOPICS

- Network interfaces and Ethernet basics
- Network driver fundamentals
- Network queuing disciplines
- Network driver SPI implementation
- TCP and UDP client-server programming
- Network transmission paths
- Name and address conversion
- Socket timeout implementation
- Unix domain protocol
- ARP operation and tcpdump

5

MONTH 5 • CAPSTONE

Kernel Debugging Techniques

8 TOPICS

- Memory debugging — SLAB, KASAN
- Kernel debugging tools — kmemleak, lockdep
- Kernel patching techniques
- kdump and the crash tool
- Kernel panic debugging
- KGDB and KDB implementation
- eBPF and bpftrace



PHASE 3 • THE 12-MONTH CLOCK

Placement Window

- **The gateway trigger.** On completing the course, students must clear a Comprehensive Technical Evaluation (CTE). Passing this panel officially starts the 12-month placement window.
- **Dual pipelines.** The placement wing actively markets student profiles to semiconductor design houses, automotive Tier-1 suppliers and firmware companies, while students simultaneously pursue off-campus opportunities.

03

Eligibility Conditions

WHAT THE STUDENT MUST MAINTAIN TO KEEP THE GUARANTEE LIVE

The refund guarantee is conditioned on the following metrics being met and documented in the signed Student Agreement. These protect the academy from liability arising from non-participation or lack of effort.

90%

Academic metrics

Minimum 90% live attendance, and 85% timely completion of kernel compilation labs and code reviews.



First-time-right compilation

Submitting non-compiling or plagiarised kernel modules on two consecutive occasions immediately revokes MBG eligibility.

3/wk

Active job hunt

At least three applications per week to relevant firmware or embedded roles, logged on a shared tracker. Declining a bona fide offer at baseline CTC forfeits the refund.

04

Refund Disqualification Summary

WHEN NO REFUND IS PAYABLE

Read this section before enrolling. The Money-Back Guarantee is void, and no refund is payable, under any one of the following conditions.

- ✗ **A qualifying offer is received** — and accepted or declined — meeting the baseline CTC within the 12-month window, whether on-campus, off-campus or internal to Kernel Foundation.
- ✗ **Attendance falls below 90%**, or lab and code-review completion falls below 85%.
- ✗ **Two consecutive instances** of non-compiling or plagiarised module submissions.
- ✗ **Failure to maintain** the minimum weekly job-application activity documented on the tracker.
- ✗ **Failure to clear the Comprehensive Technical Evaluation** required to activate the 12-month window.

05

Governing Terms

PRECEDENCE AND STATUS OF THIS DOCUMENT

This policy summary is governed in full by the executed Student Agreement, which is the binding legal document between the academy and the student. In the event of any conflict between this summary and the Student Agreement, the terms of the Student Agreement shall prevail.

This document is a policy summary for institutional and marketing use. It is not a substitute for the legally binding Student Agreement and does not constitute legal advice.

STUDENT ACKNOWLEDGEMENT	KERNEL FOUNDATION
Name: _____	Authorised signatory: _____
Signature: _____	Signature: _____
Date: _____	Date: _____