1 Knowledge about system development on true middle developer position

Scope	Must do
Common courses	• `Operating systems basis` • `Linux programming basis` • `How does the Internet work` • `Multithreading C/C++`(optional)
Algorithms: list, trees, rbtrees, search	• `C++ Standard Containers` • `Data Structures` • What does mean shared memory and how it works in optee with tee driver?
Just for English	• `Embedded Systems Specialization` (optional)
Processes, threads & Memory Managment	• Andrew Tanenbaum - Modern operating systems • `Operating Systems Internals and Design Principles` • What Every Programmer Should Know About Memory - MMU, page tables, virtual address translations(ARM), TLB, TLB exeptions, CPU caches. How does it work in Linux? In OP-TEE? How can I align memory for cache line? • What exactly does mean memory alignment in Linux kernel and in userspace?
syscalls	• Robert Love - Linux System Programming. Learn all syscalls and see its handlers in Linux • Write syscall calling with ARM assembler
Linux Kernel	• Robert Love - Linux Kernel Development • See real interrupt handler vectors in Linux kernel, in OP-TEE • Learn device tree, task struct, struct page • Linux Device Drivers • Run rotary with interrupt handling in BeagleBoneBoard
Common qeustions	• When does userspace start? What does mean initramfs? Buildroot? Yocto? • Understand what is kprobes, SElinux, cgroups, ASLR • What mustn't you do in interrupt context? • Run Linux in qemu
ARM specific	• Introducing the Arm architecture • Armv8-A Exception model • Armv8-A Instruction Set Architecture • Armv8-A memory management • ARMv8-A Address Translation • Armv8-A memory model • TrustZone for Armv8-A • Generic Interrupt Controller • Generic Timer • Armv8-A Virtualization • Hardware and Software Support for Virtualization • ARM System Developer's Guide