**ትረስትድ ኮምፒውቲንግ ቤዝ ፕላትፎርም ዲቨሎፕመንት**

**መግቢያ**

በትረስትድ ኮምፒውቲንግ ምርምር ቡድን ተልእኮ ላይ ሰፍሮ እንደሚታየው፣ ሁለት ን/ቡድን ምርምሮች ያሉት ሲሆን በስም ለመጥቀስም ያክል

1. ትረስትድ ኮምፒዩቲንግ ቤዝ ምርምር ን/ቡድን (Trusted Computing Base Platform Development)
2. መደበኛ ስርዓቶች ማረጋገጫ ምርምር ን/ቡድን (Formal Specification and Verification of Systems)

ከነዚህ የስራ ዘርፎች ውስጥም በኔ እውቀት ሊመራ የሚችለው በተራ ቁጥር 1 ላይ የተገለፀው (TCB Platform development) ን/ቡድን ሲሆን የስራ ሂደቱንም ለማሳለጥ ይረዳል ብዬ ያስቀመጥኩት (roadmap) እንደሚከተለው ይሆናል።

**ተልእኮ**

አለም የደረሰባቸውን ደህንነቱ የተረጋገጠ Trustworthy Systems ላይ ያሉ እውቀቶችን ባለቤት መሆንና መጠቀም።

**የስራ አካሄድ**

አዳዲስ ልጆች ወደ ቡድኑ ቢቀላቀሉ ምን አይነት እውቀትና መስፈርቶችን ሟሟላት እንዳለባቸው እንዲሁም ቡድኑ ወጥ የሆነ የስራ ሂደትና ትንታኔ እንዲኖረው ለማስቻል ይረዳ ዘንድ ከታች የተቀመጡትን እውቀትና ክህሎቶች እንዲሁም አካሄዶች መዳሰስ አስፈላጊ ነው።

**ስራ ---› TCB Platform Development Plan**

አንድ ሰው ትረስትድ ኮምፒውቲንግ ፕላትፎርም ውስጥ ገብቶ ውጤታማ ለመሆን ሲያስብ በ “conceptual, logical and practical” የmini-os ማበልፀግ ልምድ ሊኖረው ይገባል.

በእውቀት ደረጃም መከተል ያለበት ቅደም ተከተል እንደሚከተለው ይቀርባል።

**TCB Platform Development Plan**

1. **Covering basic OS concept in the following articles**.
   * + 1. [Linux kernel introduction](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html)

* [Basic operating systems terms and concepts](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#basic-operating-systems-terms-and-concepts)
  + [User vs Kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#user-vs-kernel)
  + [Typical operating system architecture](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#typical-operating-system-architecture)
  + [Monolithic kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#monolithic-kernel)
  + [Micro kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#micro-kernel)
  + [Micro-kernels vs monolithic kernels](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#micro-kernels-vs-monolithic-kernels)
  + [Address space](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#address-space)
  + [User and kernel sharing the virtual address space](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#user-and-kernel-sharing-the-virtual-address-space)
  + [Execution contexts](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#execution-contexts)
  + [Multi-tasking](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#multi-tasking)
  + [Preemptive kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#preemptive-kernel)
  + [Pageable kernel memory](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#pageable-kernel-memory)
  + [Kernel stack](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#kernel-stack)
  + [Portability](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#portability)
  + [Asymmetric MultiProcessing (ASMP)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#asymmetric-multiprocessing-asmp)
  + [Symmetric MultiProcessing (SMP)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#symmetric-multiprocessing-smp) Advanced concept
* [Overview of the Linux kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#overview-of-the-linux-kernel)
* [Linux development model](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#linux-development-model)
* [Linux source code layout](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#linux-source-code-layout)
* [Linux kernel architecture](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#linux-kernel-architecture)
* [arch](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#arch)
* [Device drivers](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#device-drivers)
* [Process management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#process-management)
* [Memory management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#memory-management)
* [Block I/O management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#block-i-o-management)
* [Virtual Filesystem Switch](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#virtual-filesystem-switch)
* [Networking stack](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#networking-stack)
* [Linux Security Modules](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec1-intro.html#linux-security-modules)
  + - 1. [System calls](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html)
* [System](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html#lecture-objectives) call
* [Linux system calls implementation](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html#linux-system-calls-implementation)
* [System call table](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html#system-call-table)
* [System call parameters handling](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html#system-call-parameters-handling)
* [Virtual Dynamic Shared Object (VDSO)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec2-syscalls.html#virtual-dynamic-shared-object-vdso)
  + - 1. [Processes](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html)
* [Processes and threads](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#processes-and-threads)
* [Overview of process resources](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#overview-of-process-resources)
* [struct task\_struct](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#struct-task-struct)
* [Inspecting task\_struct](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#inspecting-task-struct)
* [Threads](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#threads)
* [The clone system call](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#the-clone-system-call)
* [Namespaces and "containers"](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#namespaces-and-containers)
* [Accessing the current process](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#accessing-the-current-process)
* [Context switching](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#context-switching)
* [Quiz: context switch](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#quiz-context-switch)
  + [Blocking and waking up tasks](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#blocking-and-waking-up-tasks)
* [Task states](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#task-states)
* [Blocking the current thread](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#blocking-the-current-thread)
* [Waking up a task](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#waking-up-a-task)
  + [Preempting tasks](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#preempting-tasks)
* [Non preemptive kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#non-preemptive-kernel)
* [Preemptive kernel](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#preemptive-kernel)
  + [Process context](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#process-context)
* [Kernel threads](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec3-processes.html#kernel-threads)
  + - 1. [Interrupts](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html)
  + [What is an interrupt?](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#what-is-an-interrupt)
* [Exceptions](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#exceptions)
  + [Hardware Concepts](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#hardware-concepts)
* [Programmable Interrupt Controller](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#programmable-interrupt-controller)
* [Interrupt controllers in SMP systems](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-controllers-in-smp-systems)
* [Interrupt Control](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-control)
* [Interrupt priorities](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-priorities)
  + [Interrupt handling on the x86 architecture](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-handling-on-the-x86-architecture)
* [Interrupt Descriptor Table](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-descriptor-table)
* [Interrupt handler address](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-handler-address)
* [Stack of interrupt handler](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#stack-of-interrupt-handler)
* [Handling an interrupt request](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#handling-an-interrupt-request)
* [Returning from an interrupt handler](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#returning-from-an-interrupt-handler)
* [Inspecting the x86 interrupt handling](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#inspecting-the-x86-interrupt-handling)
  + [Interrupt handling in Linux](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-handling-in-linux)
* [Nested interrupts and exceptions](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#nested-interrupts-and-exceptions)
* [Interrupt context](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#interrupt-context)
* [Deferrable actions](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#deferrable-actions)
* [Soft IRQs](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#soft-irqs)
* [Packet flood example](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#packet-flood-example)
* [Tasklets](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#tasklets)
* [Timers](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec4-interrupts.html#timers)
  + - 1. [Symmetric Multi-Processing](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html)
  + [Synchronization basics](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#synchronization-basics)
  + [Linux kernel concurrency sources](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#linux-kernel-concurrency-sources)
  + [Atomic operations](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#atomic-operations)
  + [Disabling preemption (interrupts)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#disabling-preemption-interrupts)
  + [Spin Locks](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#spin-locks)
  + [Cache coherency in multi-processor systems](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#cache-coherency-in-multi-processor-systems)
  + [Optimized spin locks](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#optimized-spin-locks)
  + [Process and Interrupt Context Synchronization](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#process-and-interrupt-context-synchronization)
  + [Mutexes](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#mutexes)
  + [Per CPU data](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#per-cpu-data)
  + [Memory Ordering and Barriers](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#memory-ordering-and-barriers)
  + [Read Copy Update (RCU)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec5-smp.html#read-copy-update-rcu)
    - 1. [Address Space](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html)
  + [x86 MMU](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#x86-mmu)
* [Selectors](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#selectors)
* [Segment descriptor](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#segment-descriptor)
* [Segmentation in Linux](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#segmentation-in-linux)
* [Inspecting selectors and segments](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#inspecting-selectors-and-segments)
* [x86 Paging](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#x86-paging)
* [Page tables](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#page-tables)
* [Linux paging](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#linux-paging)
* [Translation Look-aside Buffer](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#translation-look-aside-buffer)
  + [Linux address space](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#linux-address-space)
* [Address space options for 32bit systems](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#address-space-options-for-32bit-systems)
* [Linear mappings](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#linear-mappings)
* [Highmem](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#highmem)
* [Fixed-mapped linear addresses](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#fixed-mapped-linear-addresses)
* [Temporary mappings](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#temporary-mappings)
* [Permanent mappings](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec6-address-space.html#permanent-mappings)
  + - 1. [Memory Management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html)
  + [Physical Memory Management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#physical-memory-management)
* [Memory zones](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#memory-zones)
* [Non-Uniform Memory Access](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#non-uniform-memory-access)
* [Page allocation](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#page-allocation)
* [Small allocations](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#small-allocations)
  + [Virtual memory management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#virtual-memory-management)
  + [Fault page handling](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec7-memory-management.html#fault-page-handling)
    - 1. [Filesystem Management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html)
  + [Lecture objectives:](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#lecture-objectives)
  + [Filesystem Abstractions](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#filesystem-abstractions)
  + [Filesystem Operations](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#filesystem-operations)
* [Mounting a filesystem](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#mounting-a-filesystem)
* [Opening a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#opening-a-file)
* [Querying file attributes](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#querying-file-attributes)
* [Reading data from a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#reading-data-from-a-file)
* [Writing data to a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#writing-data-to-a-file)
* [Closing a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#closing-a-file)
* [Directories](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#directories)
* [Creating a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#creating-a-file)
* [Deleting a file](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#deleting-a-file)
  + [Linux Virtual File System](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#linux-virtual-file-system)
* [Superblock Operations](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#superblock-operations)
* [Inode Operations](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#inode-operations)
* [The Inode Cache](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#the-inode-cache)
* [The Dentry Cache](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#the-dentry-cache)
* [The Page Cache](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec8-filesystems.html#the-page-cache)
  + - 1. [Networking](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html)
  + [Network Management Overview](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#network-management-overview)
  + [Sockets Implementation Overview](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#sockets-implementation-overview)
  + [Sockets Families and Protocols](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#sockets-families-and-protocols)
* [Example: UDP send](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#example-udp-send)
  + [Network processing phases](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#network-processing-phases)
  + [Packet Routing](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#packet-routing)
* [Routing Table(s)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#routing-table-s)
* [Routing Policy Database](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#routing-policy-database)
* [Routing table processing](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#routing-table-processing)
* [Forwarding Information Database](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#forwarding-information-database)
  + [Netfilter](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#netfilter)
  + [Network packets / skbs (struct sk\_buff)](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#network-packets-skbs-struct-sk-buff)
  + [Network Device](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#network-device)
  + [Hardware and Software Acceleration Techniques](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec10-networking.html#hardware-and-software-acceleration-techniques)
    - 1. [Architecture Layer](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html)
  + [Overview of the arch layer](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#overview-of-the-arch-layer)
* [Boot strap](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#id1)
* [Memory setup](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#memory-setup)
* [MMU management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#mmu-management)
* [Thread Management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#thread-management)
* [Time Management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#time-management)
* [IRQs and exception management](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#irqs-and-exception-management)
* [System calls](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#system-calls)
* [Platform Drivers](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#platform-drivers)
* [Machine specific code](https://linux-kernel-labs.github.io/refs/heads/master/so2/lec11-arch.html#machine-specific-code)

1. **Applying major OS concepts with ‘C’ and ‘Assembly’ languages**.

* Create a workable mini-OS image that can be installed in empty computer or
* That can be used as a live usb.

Talking about major OS concepts, it includes the following points.

* The Boot Process
* BIOS, Boot Blocks, and the Magic Number
* CPU Emulation (Exercising with Bochs or QEmu)
* Boot Sector Programming with 16-bit Real Mode
* Writing mini OS that displays “hello world” message
* Interrupts
* CPU Registers
* Memory, Addresses, and Labels
* Using the Stack
* Reading the Disk
* Extended Memory Access Using Segments
* Using BIOS to Read the Disk
* Entering 32-bit Protected Mode
* Adapting to Life Without BIOS
* Understanding the Global Descriptor Table
* Defining the GDT in Assembly
* Making the Switch
* Putting it all Together
* Writing, Building, and Loading Your Kernel
* Understanding C Compilation
* Generating Raw Machine Code
* Pointers, Addresses, and Data
* Creating a Boot Sector to Bootstrap our Kernel
* Finding Our Way into the Kernel
* Automating Builds with Make
* Organizing Our Operating System's Code Base
* Developing Essential Device Drivers and a Filesystem
* Hardware Input/Output
* I/O Buses
* I/O Programming
* Direct Memory Access
* Screen Driver
* Understanding the Display Device
* Basic Screen Driver Implementation
* Scrolling the Screen
* Handling Interrupts
* Keyboard Driver
* File System
* Implementing Processes management
* Single Processing
* Multi-processing

1. **Understand and work on sel4 secure microkernel**

* This will be covered in the future.

**ከላይ የተዘረዘሩ አካሄዶች የሚከተለውን ጥቅም ያመጣሉ**

1. አዲስ የቡድን አባል ባነሰ ግዜ ካሉት የስራ አባላት ጋር በእውቀትና ክህሎት እንዲመጣጠን ይረዳዋል
2. ሁሉም የስራ አባላት ያልተናነሰ እውቀትና ክህሎት እንዲኖራቸው ይረዳል
3. ሰራተኞች ብዙ ግዜ ሳያባክኑ ለቡድኑ ስራ ያላቸውን ፍላጎት በቶሎ እንዲለዩ ያደርጋል
4. ቡድኑም የተሟላ የአካሄድና የእውቀት መረጃ ቋት ይኖረዋል።
5. ይህ አካሄድ የመረጃና የስራ ውጤቶችን መመዝግብና ማየት ስለሚያስችል ለወደፊቱም መሰረት ይጥላል