# （一）基于榫卯架构的操作系统内核开发

## 系统背景

随着操作系统技术的不断演进，虚拟化与嵌入式操作系统成为前沿的研究和开发热点。Tenon（榫卯），作为基于LibOS架构的轻量级操作系统，提供了灵活的微库设计和高效的操作系统实例生成机制。本实训项目基于开源项目 TenonOS-EDU（<https://gitee.com/tenon-os-edu/summer-internship>），通过一系列实际操作和代码提交，培养学生在内存管理、IO、网络、中断等核心模块方面的开发能力，并通过在线评测系统实现自动化测试，以确保学生提交的代码符合实验要求。

## 项目目标

 掌握操作系统核心模块的开发，包括内存管理、IO处理、网络和中断。

 学习如何在TenonOS平台上为ARM64架构开发和调试内核实例。

 掌握编程和调试工具的使用，通过在线评测系统验证代码功能和性能。

 锻炼学生的调试能力和解决实际问题的能力。

## 项目内容

**第一阶段：常用工具与hello world**

目标是掌握基础开发工具，并编译运行一个hello world程序。内容包括搭建开发环境（QEMU + Unikraft + KVM）、编译并运行hello world示例，以及修改程序输出。

**第二阶段：ARM64汇编与寄存器操作**

在此阶段，学生将学习ARM64汇编指令及其应用。主要内容包括ARM64汇编基础和系统调用与寄存器操作的实现。

**第三阶段：GDB调试与内存管理**

学生将在这一阶段学习如何使用GDB进行调试，并实现内存分配器。内容涉及使用GDB调试Unikraft中的代码和实现malloc与free功能。

**第四阶段：低级IO与串口**

目标是实现串口通信和低级IO操作。学生将学习串口通信的基础，并实现串口的输入输出功能。

**第五阶段：virtIO与网络通信**

在这一阶段，学生将理解virtIO的基本概念，并进行网络通信实验。内容包括virtIO驱动的开发以及实现网络请求处理（如ping）。

**第六阶段：中断处理与应用**

学生需编写中断处理程序来响应某个硬件设备（如键盘或定时器）的中断请求，确保中断处理程序的响应及时且不影响系统的正常运行。

**第七阶段：外部库集成与应用**

最后，学生将学习如何在TenonOS中集成和使用外部库。目标是理解第三方库的集成过程，并在操作系统中应用外部功能库。学生将编写一个第三交互方库集成到TenonOS。

## 项目评测

为确保代码的质量与协作开发的有效性，学生在完成各实验阶段的开发任务后，将经历以下代码提交流程：

1. **代码开发与调试**

学生需在本地完成相应实验阶段的开发任务，并在本地通过调试工具验证代码的正确性。每个阶段的开发内容会有相应的代码功能要求，如内存管理阶段要求正确实现malloc/free接口。

1. **Pull Request 提交**

当学生确认本地代码无误后，需将代码推送到指定的远程仓库（<https://gitee.com/tenon-os-edu/summer-internship>），并发起Pull Request (PR)。该PR需要详细说明所修改的内容及实验完成情况。每个PR都将经过代码审查，确保代码风格、逻辑正确性以及功能完整性。

1. **代码审查与反馈**

由课程助教或指导教师对提交的PR进行审查，确保代码符合实验要求。在审查过程中，助教会提出可能的改进建议或指出需要修正的问题。学生需根据反馈进一步完善代码，并再次提交修改。

1. **提交至在线评测系统**

PR不会被合并，学生需将代码提交至在线评测系统（Online Judge），系统将自动执行功能测试与性能测试。

1. **在线测试反馈**

在线评测系统将自动生成测试报告，反馈代码功能与性能的测试结果。学生根据测试结果进行最后的修改与优化，确保提交的代码无误并通过最终测试。

## 开发环境

操作系统: Ubuntu 18.04 LTS 或更高版本

开发工具: TenonOS 0.1.0, QEMU 2.11或更高, GDB 8.1或更高，Git

虚拟化平台: KVM, QEMU

目标架构: ARM64