

项目难度的设置应结合项目要求、项目规模、技术难度、项目开发预估工作量、资源限制等因素设置。

基础类项目示例

1. 为 openEuler 的 UKUI 桌面增加或适配日程管理应用

- 项目描述：

目前在 UKUI 桌面任务栏的右下角，提供了日历，但是却无法管理日程，未完善整体生态，希望能够针对 UKUI 桌面研发或者适配一款日程管理的相关软件。

- 产出标准：

1. 能够增加、删除、编辑日程
2. 能够多种视图来显示日程，如月、周、日的模式显示日程
3. 支持任务栏托盘
4. 最好能够同步 outlook 或是其它账户建立的日程

- 技术要求：

熟悉 linux 操作系统，具备在 Linux 系统下的研发能力

2. 在 openEuler 系统中提供可扩展的应用可视化配置软件

- 项目描述：

linux 长期以来需要专业的技术人员进行操作及管理，技术门槛较高，对于初学者使用不够友好。linux 环境已经拥有优秀的软件包管理机制，但是在软件使用上，基于文件的配置也带来手动编辑的繁琐。对比 windows 应用生态，所有的软件都具有良好的配置可视化功能。我们的目标是：降低 linux 环境的使用门槛，使初学者能快速上手使用 linux 环境。

- 产出标准：

1. 完成项目的设计、开发、测试并提供架构和使用以及插件二次开发的文档
2. 前端提供尽可能详细的配置描述
3. 优先考虑将核心功能设计成独立的库，如插件库等
4. 支持至少 5 个常用应用程序的配置（如 mysql 、redis 、httpd 等），并考虑不同版本之间的
5. 配置差异，也可考虑扩展系统内置的应用配置（如网络、防火墙、crontab 等），提供重启服务生效功能
6. 在 openEuler 社区中开源

- 技术要求：

1. 使用 go 语言进行开发
2. 基于 web 的前端可视化及配置
3. 插件架构，支持扩展任意软件的配置

进阶类项目示例

1. 为 openEuler 提供 OpenStack 容器化部署能力

- 项目描述：

OpenStack 是一个开源的云计算管理平台项目，是一系列软件开源项目的组合。由 NASA（美国国家航空航天局）和 Rackspace 合作研发并发起，以 Apache 许可证（Apache 软件基金会发布的一个自由软件许可证）授权的开源代码项目。OpenStack 为私有云和公有云提供可扩展的弹性的云计算服务。项目目标是提供实施简单、可大规模扩展、丰富、标准统一的云计算管理平台。整个 OpenStack 框架由众多组件组成，这些组件为云计算平台提供不同的服务。

OpenStack 包含组件繁多，其安装部署是非常令人头疼的问题。使用容器化技术，可以显著降低 OpenStack 的部署难度。本项目使用 OpenStack+Kubernetes+Isula 技术栈，实现 OpenStack 在 openEuler 上的容器化快速部署能力。

- 产出标准：

1. 完成 OpenStack 对应的 kubernetes 部署文件编写
2. 部署、测试并验证 OpenStack+kubernetes+isula 集群环境，输出相关测试报告

- 技术要求：

1. 熟练使用 Linux 操作系统
2. 了解基本的云计算概念
3. 了解并会使用 docker 或 isula 等容器服务
4. 了解并会使用 kubernetes 容器编排服务
5. 对虚拟化或容器化网络有一定了解

2. StratoVirt 虚拟化平台启动时间的优化

- 项目描述：

轻量虚拟机（Mirto VM）是近几年新兴的一项虚拟化技术，StratoVirt 是华为 openEuler 开源的轻量虚拟化软件，使用内存安全语言 Rust 编写，基于 Rust 语言编写，具备功能完善、启动快速、内存开销小等特点，在容器和 serverless 场景有广阔的应用空间。本实验的目标是 基于 StratoVirt 现有的能力，进一步提升其的轻量低噪的能力。

- 产出标准：

1. 在 x86_64 平台上达到或接近 kunpeng920 平台的启动时间
2. 总结整个开发流程和遇到的问题，并形成文档

- 技术要求：

1. 熟悉 Linux 内核启动流程
2. 对虚拟化技术有一定了解
3. 熟悉 Rust 语言

本项目要求对轻量虚拟机在 x86_64 平台上的启动时长进行优化，可以尝试并不限于以下方案：

1. 优化 StratoVirt 的 bootloader 模块
2. 裁剪与 MicroVm 适配的 guest kernel 内核
3. 实现以 PVH 的方式启动

The difficulty level of a project should be determined based on factors such as project requirements, project scale, technical complexity, estimated workload for project development, and resource limitations.

Example for Basic Project

1. Add or adapt schedule management application for UKUI desktop of openEuler

- **Project Description:**

At present, a calendar is provided in the lower right corner of the UKUI desktop taskbar, but it is unable to manage the schedule and does not improve the overall ecology. We hope to develop or adapt a schedule management related software for the UKUI desktop.

- **Project Output Standard**

1. Be able to add, delete and edit the schedule.
2. It can display the schedule in a variety of views, such as month, week and day.
3. Support taskbar tray.
4. It's better to synchronize the schedule established by outlook or other accounts.

- **Technical Requirements:**

Familiar with Linux operating system, with research and development ability under Linux system.

2. Provide scalable application visualization configuration software in openEuler system.

- **Project Description**

For a long time, Linux needs professional technicians to operate and manage, the technical threshold is high, and it is not friendly for beginners. Linux environment has excellent software package management mechanism, but in the use of software, File-based configuration also brings the tediousness of manual editing. Compared with windows application ecology, all software has good configuration visualization function. Our goal is to reduce the threshold of using Linux environment and enable beginners to quickly start using Linux environment.

- **Project Output Standard**

1. Complete the design, development and testing of the project, and provide the documents of architecture, use and plug-in secondary development.
2. The front end provides as detailed a configuration description as possible.
3. Give priority to designing core functions as independent libraries, such as plug-in libraries, etc.
4. Support the configuration of at least five common application software (such as mysql, red is, httpd, etc.), and consider the configuration differences between different versions, the built-in application configuration of the system (such as network, firewall, crontab, etc.) can also be extended to provide the effective function of restart service.
5. Open source in openEuler community.

- **Technical Requirements**

1. Use go language for development.
2. Front-end visualization and configuration based on Web.
3. Requires a plug-in architecture to support the configuration of any extended software

Example for Advanced Project

1. Provide OpenStack containerized deployment capabilities for openEuler

- Project Description

OpenStack is an open source cloud computing management platform project, which is a combination of a series of open source software projects. It is also an open source code project authorized by the Apache License (a free software license issued by the Apache Software Foundation), which was jointly developed and initiated by NASA (National Aeronautics and Space Administration) and Rackspace. OpenStack provides scalable, flexible cloud computing services for both private and public clouds. The goal of the project is to provide a cloud computing management platform that is simple to implement, scalable, rich and standardized. The entire OpenStack framework consists of a number of components that provide different services for cloud computing platforms. OpenStack contains many components, and its installation deployment is a very headache problem. Using containerization technology can significantly reduce the difficulty of deploying OpenStack.

This topic uses the "OpenStack+Kubernetes+Isula" technology stack to realize the rapid containerized deployment capability of OpenStack on openEuler.

- Project Output Standard

1. Complete the kubernetes deployment file writing corresponding to OpenStack;
2. Deploy, test and verify the "OpenStack+kubernetes+isula" cluster environment, and output relevant test reports.

- Technical Requirements

1. Familiar with Linux operating system.
2. Understand basic cloud computing concepts.
3. Understand and use container services such as docker or isula.
4. Understand and use the kubernetes container orchestration service.
5. Have some knowledge of virtualized or containerized networking.

2. Optimization of Startup Times of StratoVirt Virtualization Platform

- Project Description

Lightweight Virtual Machine (Mirto VM) is a new virtualization technology in recent years. StratoVirt is open source lightweight virtualization software for Huawei openEuler. It is written in Rust, a memory security language. It is based on Rust, and has the characteristics of complete functions, fast startup, and low memory overhead. StratoVirt has a wide application space in containers and serverless scenarios. The aim of this experiment is to further improve StratoVirt's lightweight and low noise capabilities based on its existing capabilities.

- Project Output Standard

1. Startup time to reach or close to Kunpeng 920 platforms on the X86_64 platform.
2. Summarize the entire development process and problems encountered, and rm a document.

- Technical Requirements

1. Familiar with the Linux kernel boot process.
2. Have a certain understanding of virtualization technology.

3. Familiar with Rust.

This topic requires optimization of the startup time of lightweight virtual machines on the x86_64 platform. You can try and are not limited to the following solutions:

1. Optimize the bootloader module for StratoVirt.
2. Clip guest kernel adapted to MicroVm.
3. Startup by way of PVH.