

WebSummerCamp2023后端第0期任务

欢迎来到联创web组夏令营！本次为第0期任务，任务难度不大，望大家认真完成。本期任务将在下次MeetUp进行检查。

Linux

安装、学习 Linux

建议后面的所有任务都在 Linux 环境下完成

发行版选择

发行版强烈推荐Arch Linux，安装 ArchLinux 真的能让你学到很多。Arch永远的神。

最好根据 [官方Wiki](#) 来进行安装，联创 Android 组 [吴迪学长的博客](#) 或者 [鲨鱼姐姐的文档](#) 也是很不错的教程

当然，使用你熟悉的发行版也是很好的选择

如果你的 Linux 跑在虚拟机里，推荐不使用图形界面。如果是实装，图形界面推荐 Wayland+Gnome / Xorg+KDE

Wayland生态目前不如Xorg

Linux基本概念和命令

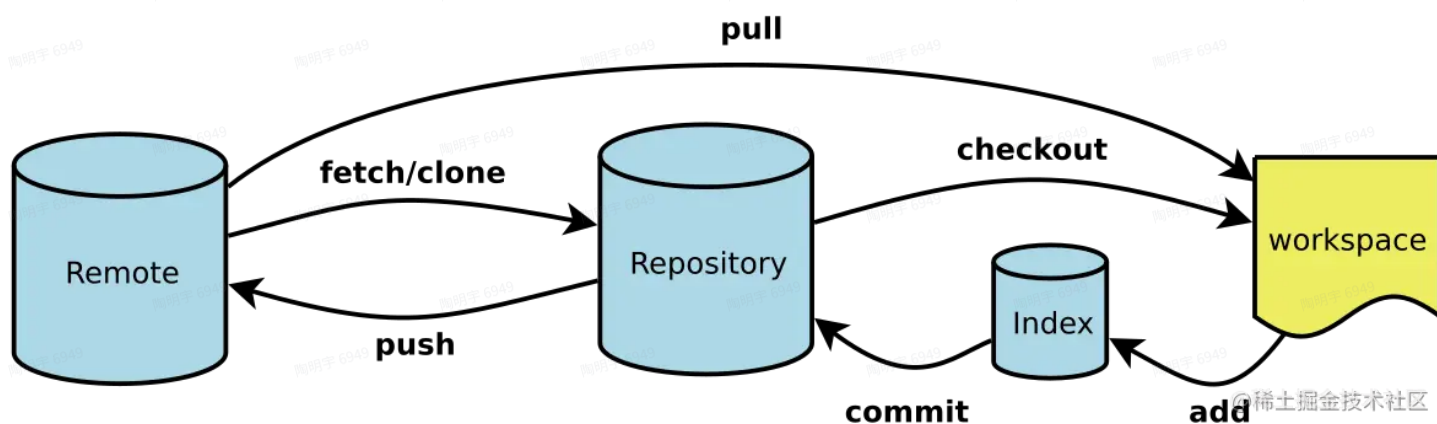
- 了解发行版的包管理器的使用（Arch是pacman，debian系是apt，centos系是yum / dnf）
- 能使用命令行进行对文件或文件夹的创建、复制、删除、搜索、移动、查看等操作（ls/cd/mkdir/touch/cp/rm/find/mv）
- 了解 Linux 文件权限，会修改文件权限、文件所有者(chmod/chown)
 - 注意粘滞位
 - 如果可以，希望你能简单了解隐藏权限
- 什么是进程？
- 会查看、结束进程(ps/top/kill/killall/pkill)
- 什么是端口？
- 如何查看系统中端口占用的情况(80端口是否被占用？若被占用，是哪个进程占用了？)
- 了解守护进程的概念及其管理方式（systemd/init.d/sysv，不同的发行版预置的守护进程会相应不同）

- 对于systemd，[阮一峰的教程](#)是一个好的入门
- 尝试配置一下你的Shell，让他变得更好看且powerful！尤其指自动补全
 - 思考一下terminal和shell的区别
 - [oh-my-zsh](#)
- 了解一下Linux的防火墙机制，避免你的机器惨遭“挖矿病毒”
 - archlinux和debian系是iptables，centos系是firewalld

文本处理

- 学习简单的标记语言：**markdown**
 - Daily report我们要求使用markdown
- 学习基本的文本处理命令（less/more/tail/cat）
- 一些高级的文本处理命令（grep/awk/sed）
- 更高级的文本处理Vim
 - Nano / Emacs也可以
 - [Learn-Vim](#)是一个不错的Vim入门教程
- For Vim（此时，请丢掉你的鼠标）
 - vim常见键位及其对应作用（如插入，删除，查找等）
 - vim多窗口 / 多文件
 - vim查找替换

版本控制Git



附上一张git不同space与命令之间的联系

- Git是什么，有什么用？
 - 拥有一个Github帐号
- 学习简单的git操作，如add, commit, branch, status等

- 如果没有 `git switch/ git restore` 命令，请更新你的git版本
- 学习如何回退版本（了解三种不同的回退模式）
- 如何查看git日志？
- 如何修改git commit信息？
- 学习如何合并分支
- 学习如何暂存工作区（stash）
- 考虑多个上游的管理（remote）
- 可以看看 [Pro Git](#) 深入了解一下git

Shell和构建工具

- 能熟练使用管道（pipe），I/O重定向（I/O redirection）等shell内置功能
 - 比如一个经典的组面题 - 如何输出一个文件夹的前 / 后 X 个文件的文件名。进一步地，如何输出最新的X个文件
- 常用构建工具的使用(Makefile/CMake)
 - 对于Makefile工具，可以考虑如何通过写Makefile启动你的项目。比如通过 ***make start*** 启动你的项目而不是使用裸命令（`go run main.go -arg1 -arg2 -arg3`）

ysy 必做部分

排序

- 插入排序
- 堆排序
- 快速排序
- 归并排序

树

构造树的数据结构，并实现基本的树操作：

- 树的先序、中序、后序、层级遍历, 前中后序遍历要求递归和非递归实现；
- *Optional*: Morris 遍历（线索二叉树）

实现中缀表达式转后缀表达式的算法：

- 计算的数据均为十进制正整数；
- 支持的四则运算符（加减乘除）；
- 可以使用栈实现，也可以尝试构建表达式树。

哈希表

构造哈希表的数据结构，并实现基本的表操作：

- 键值对的增删改查；
- 防碰撞。

图

基于邻接表实现有向无环图的数据结构：

- 实现图的 BFS 和 DFS；
- 实现图上的拓扑排序。

要求

- C/C++ 实现；
- 允许并推荐使用 C++ 的 STL；
- 代码存放在 Github 仓库中。

yj 必做部分

Go

Go 语言语法本身并不难，希望你能快速上手并能实现一些简单程序

- 《The Go Programming Language》
- 《Go入门指南》

Docker

docker

- 尝试了解一下docker
- 什么是镜像，什么是容器？
- 如何构建一个镜像？如何启动一个容器？
- 简单的Dockerfile的编写
- 如何查看现有容器的状态？
- 如何进入一个容器？
- 如何停止一个容器？
- 什么是docker-compose？为什么我们需要它？
- 简单的docker-compose.yml的编写

- 什么是docker network，有什么作用？

lgx 必做部分

Docker

docker

- 尝试了解一下docker
- 什么是镜像，什么是容器？
- 如何构建一个镜像？如何启动一个容器？
- 简单的Dockerfile的编写
- 如何查看现有容器的状态？
- 如何进入一个容器？
- 如何停止一个容器？
- 什么是docker-compose？为什么我们需要它？
- 简单的docker-compose.yml的编写，使用 docker compose 统一管理你的 java service, mysql, redis 等容器
- 了解 docker 容器卷的挂载及其作用
- 了解 docker network 及其作用

Traefik

Traefik文档

- 了解网关的作用
- 了解traefik的作用,特点
- 实践traefik

你需要完成以下几个任务

1. linux安装traefik (推荐docker安装)
2. 用docker部署一个应用,用traefik代理它.(用两种方式配置,docker label 和static file)
3. 配置https(lets-encrypt)
4. 部署traefik面板并配置http base auth,https,使用特点域名访问

思考一下 docker 和 traefik 对于服务的部署有什么作用？