Missing Class Before College

Package Manger, Shell, Vscode and Markdown

Zhaojiacheng Zhou

September 6, 2025

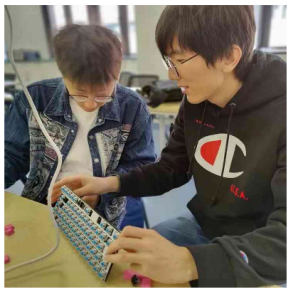
TechJI

Outline

TechJI Introduction

Who are we?

- Fans of Computer Science (but not focs)
- Host tech related workshop like linux install party, git, bash, reflow ...



机械键盘

DIY

Reflow

Create your own electrical cat

Git

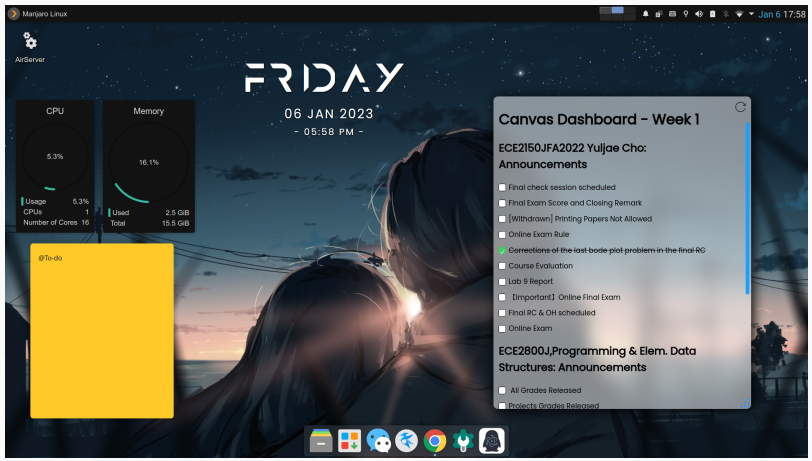
"Handle everything with speed and efficiency"

Bash

An SH-Compatible Shell

Who are we?

- Conduct a bunch of open source project development like course selection community, dancing party software, canvas helper...
- Post online tutorial like terminal beautify...



Package Manager

Introduction to package manager

- A package manager or package management system (PMS) is a collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs for a computer in a consistent manner.
- The core function of a package manager is to handle software dependencies and version conflicts to ensure that programs work correctly. It typically uses a local database to keep track of installed packages and their relationships. By working with software repositories or app stores, a package manager automates the entire software lifecycle, making it unnecessary to manually manage installations and updates.

Problem solved

- **Dependency Hell**: This is a common problem where different software packages require different, and sometimes conflicting, versions of the same shared libraries. Package managers solve this by managing and allowing for the coexistence of multiple library versions.
- **Manual Installation**: They eliminate the need for users to manually download, compile, and install software, which can be a complex and time-consuming process. This is particularly useful for large operating systems with hundreds or thousands of distinct packages.
- **Synchronization Issues**: They ensure that the list of installed software is always consistent and up-to-date with a central database, preventing conflicts and missing prerequisites that could arise from manual interventions.

How It Works

A package manager operates by interacting with three key components:

- **Packages**: These are the fundamental units of a package management system. A package is a file that contains the application, its necessary files, and metadata like the name, version, and dependencies.
- **Repositories**: These are centralized locations or servers where packages are stored. A package manager downloads packages from these repositories, eliminating the need for users to search for and download software from various websites.
- **Local Database**: The package manager maintains a local database on the user's system. This database keeps a record of all installed packages, their versions, and their dependencies, allowing the system to efficiently track and manage software and prevent conflicts.

Examples

- Windows: winget, scoop, chocolatey
- Macos: homebrew, macport
- Linux: pacman, apt, dnf ...

Mirror

- The mirror/source config of package defines where your package manager fetch remote packages
- It can be customized to improve download speed and availability
- There are many good quality source like tsinghua, ustc...

Exercise: vscode installation

- Windows users: use winget to install vscode
- Macos users: download homebrew and install vscode
- Linux users: you should now how to do so

You can install vscodium if you value your privacy since it is open source and no one will steal your data and code :)

Exercise: vscode installation (Windows)

1. Check USTC Mirror and change your source
2. Run the following command to install vscode

```
winget install --location
↪  <path-you-want-to-install>
↪  Microsoft.VisualStudioCode
```

Exercise: vscode installation (Macos)

1. Download homebrew from tsinghua mirror
2. Run the following script to install homebrew

```
xcode-select --install
export HOMEBREW_BREW_GIT_REMOTE="https://mirrors.t
 ↪  una.tsinghua.edu.cn/git/homebrew/brew.git"
export HOMEBREW_CORE_GIT_REMOTE="https://mirrors.t
 ↪  una.tsinghua.edu.cn/git/homebrew/homebrew-core
 ↪  .git"
export HOMEBREW_INSTALL_FROM_API=1
export HOMEBREW_API_DOMAIN="https://mirrors.tuna.t
 ↪  singhua.edu.cn/homebrew-bottles/api"
export HOMEBREW_BOTTLE_DOMAIN="https://mirrors.tun
 ↪  a.tsinghua.edu.cn/homebrew-bottles"
git clone --depth=1 https://mirrors.tuna.tsinghua.
 ↪  edu.cn/git/homebrew/install.git brew-install
/bin/bash brew-install/install.sh
rm -rf brew-install
```

Exercise: vscode installation (Macos)

For apple scilicon CPU user run following command

```
test -r ~/.bash_profile && echo 'eval
 ↪  "$(/opt/homebrew/bin/brew shellenv)"' >>
 ↪  ~/.bash_profile
test -r ~/.zprofile && echo 'eval
 ↪  "$(/opt/homebrew/bin/brew shellenv)"' >>
 ↪  ~/.zprofile
```

Exercise: vscode installation (Macos)

For long term substitution of mirror, run following command, also see this website

```
export HOMEBREW_CORE_GIT_REMOTE="https://mirrors.t
 ↪  una.tsinghua.edu.cn/git/homebrew/homebrew-core
 ↪  .git"
for tap in core cask command-not-found; do
brew tap --custom-remote "homebrew/${tap}"
 ↪  "https://mirrors.tuna.tsinghua.edu.cn/git/home
 ↪  brew/homebrew-${tap}.git"
done
brew update
```

Exercise: vscode installation (Linux)

1. Check vscode official website and download
2. For Ubuntu users, I don't recommend you to use snap

Thank You

Questions?