

YEARLY PROJECT

By @raptazure - Haoran Liu

[Source Code of the Slide](#)



I. CHOOSE YOUR FIELDS

- Machine Learning
- Embedded Systems
- Web Applications
- Computer Graphics
- Operating Systems
- Distributed Systems
- Databases and Storage System
- Programming Language Theory
- ...

MACHINE LEARNING

- Video: [Andrew Ng Machine Learning](#)
- Frameworks: [tensorflow](#), [pytorch](#)
- Computer vision: [OpenCV\(cv2\)](#), [YOLOv3](#)
- Not only Python (Maybe JS or Rust?)

EMBEDDED SYSTEMS

- Raspberry Pi: <https://www.raspberrypi.org/>
- Linux and Open Source
- The Rust Programming Language
 - Rust is blazingly fast and memory-efficient: with no runtime or garbage collector, it can power performance-critical services, run on embedded devices, and easily integrate with other languages.
 - Rust's rich type system and ownership model guarantee memory-safety and thread-safety – enabling you to eliminate many classes of bugs at compile-time.
 - Rust has great documentation, a friendly compiler with useful error messages, and top-notch tooling – an integrated package manager and build tool, smart multi-editor support with auto-completion and type inspections, an auto-formatter, and more.
 - [The Embedded Rust Book | Chinese version](#)

WEB APPLICATIONS

- Goals are important: what do people need?
- Front-end:
 - HTML, CSS, JavaScript → TypeScript
 - Framework/Library: Vue, React, Angular
 - Reactive, Functional: rxjs, elm, OCaml...
- Back-end:
 - RESTful API design, MVC
 - Frameworks: Node.js(express, nest), Java(Spring), Python(flask, django), Ruby(rails), elixir(phoenix)...
 - Databases: MySQL, PostgreSQL, MongoDB...
 - Container: Docker, Kubernetes...
- Mobile Apps:
 - Native: Java(Kotlin) for Android, Objective-C(Swift) for IOS
 - Hybrid: React Native, Flutter (go back to front-end)
- Desktop Apps: Cross-platform solution - QT, Electron...

COMPUTER GRAPHICS

- Rust implementation of "Ray Tracing in One Weekend"
- Games: Unity, UE4...



OPERATING SYSTEMS

- [Tutorial for rCore OS step by step \(2nd edition\)](#)
- [Writing an OS in Rust - Philipp Oppermann's blog](#)

DISTRIBUTED STORAGE SYSTEM

- MIT 6.824: Distributed Systems
- PingCAP Talent Plan: TiDB and TiKV

如何参与 Talent Plan 课程的学习?

Step 1

结合个人兴趣爱好及知识背景，选择适合自己的学习路径



路径（一）

实现一个 Mini 版本的分布式关系型数据库



路径（二）

实现一个 Mini 版本的分布式 Key-value 数据库



路径（三）

参与工业级开源分布式关系型数据库 TiDB 的开发实践



路径（四）

参与工业级开源分布式 Key-value 数据库 TiKV 的开发实践



路径（五）

Rust 编程原理与实践

PROGRAMMING LANGUAGE THEORY

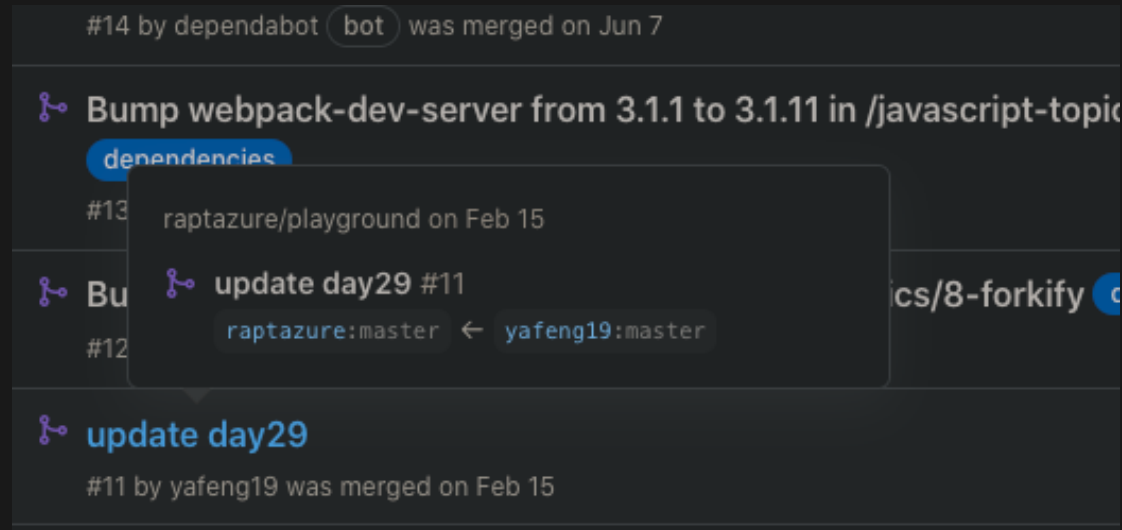
- SICP (MIT 6.001) and Computer Magic
- Functional programming: Haskell, Elixir, λ -calculus...
- Formal Proof: proof assistant \rightarrow Agda, Arend...
- Type Theory: HoTT, CuTT, Dependent type \rightarrow Agda, Idris...
- Implement a lisp/scheme interpreter using Haskell...
- Write a C compiler using Rust \rightarrow ref: rcc
- Design your own programming language

II. ORGANIZE YOUR TEAM

- Open Source Workflow
- GitHub Project

OPEN SOURCE WORKFLOW

- Push your code to GitHub (using git)
- Pull Requests / Code Review
- Merge PR and work together
- Assistant: QQ Group



GITHUB PROJECT

- Manage your progress
- Assign tasks to your team members

The image shows a GitHub Project board with four columns: "To do", "In progress", "Done", and "Noticed". Each column contains a list of tasks or projects, some with checkboxes indicating progress. The "To do" column has three items, "In progress" has two, "Done" has one, and "Noticed" has one. Each item is a card with a title, a list of sub-tasks, and a "Manage" button at the bottom. The "Noticed" column also shows a progress bar and a status label.

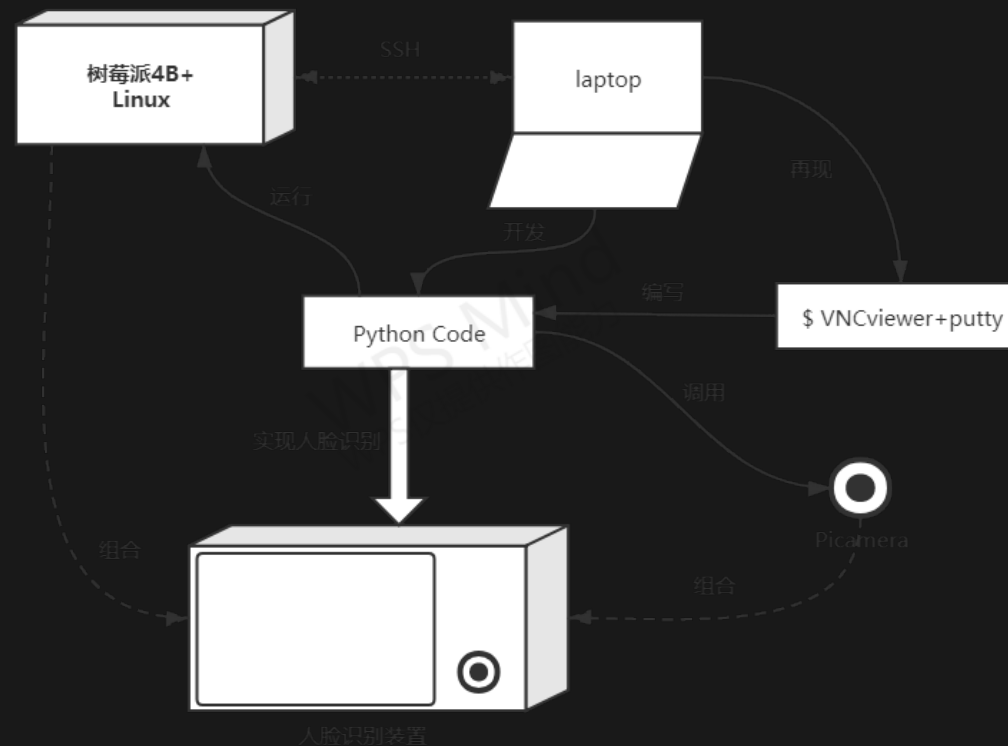
Column	Count	Item	Sub-tasks	Automated as	Manage
To do	4	Daily Project: key-value store	<input checked="" type="checkbox"/> The Rust toolbox <input checked="" type="checkbox"/> Log-structured file I/O <input type="checkbox"/> Synchronous client-server networking <input type="checkbox"/> Concurrency and Parallelism <input type="checkbox"/> Asynchronous programming in Rust	To do	Manage
		Daily Project: raft k-v store	<input type="checkbox"/> The Percolator lab <input type="checkbox"/> The Raft lab		
		Daily Project: simple redis	<input type="checkbox"/> Spawning <input type="checkbox"/> Shared state <input type="checkbox"/> Channels <input type="checkbox"/> I/O <input type="checkbox"/> Framing <input type="checkbox"/> Async in depth <input type="checkbox"/> Select <input type="checkbox"/> Streams		
In progress	3	Daily Project: many linked lists	<input checked="" type="checkbox"/> A bad safe deque <input type="checkbox"/> An unsafe queue <input type="checkbox"/> The double single list	In progress	Manage
		Daily Lesson: 6.824	<input checked="" type="checkbox"/> RPC and Threads <input type="checkbox"/> GFS <input type="checkbox"/> Primary-backup replication <input type="checkbox"/> Threads and Raft <input type="checkbox"/> Fault Tolerance 1 <input type="checkbox"/> Fault Tolerance 2 <input type="checkbox"/> Zookeeper <input type="checkbox"/> More replication, CRAQ <input type="checkbox"/> Cloud replication DB <input type="checkbox"/> Cache consistency <input type="checkbox"/> Distributed Transactions <input type="checkbox"/> Spanner <input type="checkbox"/> Optimistic Concurrency Control <input type="checkbox"/> Big Data: Spark <input type="checkbox"/> Memcached <input type="checkbox"/> COPS, Causal Consistency <input type="checkbox"/> Fork Consistency		
Done	1	Daily Project: browser engine	<input checked="" type="checkbox"/> Build the DOM and the HTML Parser. <input checked="" type="checkbox"/> Add a CSS engine and CSS parsing. <input checked="" type="checkbox"/> Add a Style Tree. <input checked="" type="checkbox"/> Boxes and the Layout Tree. <input checked="" type="checkbox"/> Commands and Rendering in OpenGL. <input checked="" type="checkbox"/> Finish up.	Done	Manage
Noticed	1	Added by raptazure	coprocessor/expression: push down scalar functions 333 of 469 #3275 opened by AndreMouche in tikv/tikv difficulty/easy sig/copyleft status/help-wanted		

III. PRESENT YOUR IDEAS

- Creation and Presentation

CREATION AND PRESENTATION

- Everything creative: from your thoughts to the implementation...
- Show it to others: module graph & flowchart



THANK YOU!

- QQ: 1051276278
- <https://github.com/raptazure>