Cai Jiechao

PERSONAL PROFILE

- Email: 2651159710@qq.com
- Tel.: (86) 13602705148
- Blog: http://www.caijiechao.com/

COMPUTER SKILL

- C/C++
- Java
- Web front end
- Python
- MS Office
- Linux
- Android Kotlin

LANGUAGE SKILL

- Chinese
- Cantonese
- English

IELTS 6.5 (Speaking 7.5)

CET-4 & CET-6

AWARD

- The Second-class Scholarship for Excellent Academic Performance in Academic Year 2023-2024
- The Winning Award of Guangdong Province in CUMCM
- Successful Participant in MCM/ICM

HOBBY

- E-Sports
- Sports

EDUCATION

09/2022-06/2026 **Beijing Normal - Hong Kong Baptist University**

Computer Science & Technology | Bachelor of Science (Honours)

12/2024-06/2025 **Universiti Malaya**

Computer Science & Technology | Exchange Program

INTERNSHIP

09/2025-01/2026 **BNBU Student Teaching Assistant Program**

Teaching Helper in the course *Data Structures and Algorithms*

- Assisted in preparing lessons, explaining coursework, correcting students' assignments and exam papers, etc.;
- Helped students solve code-related problems and answered inquiries in the lab classes.

06/2024-07/2024 **Zhuhai Young Friend Software Co., Ltd.**

Software Implementation Engineer

- Participated in the modular work of developing OA systems for client companies;
- Handled daily operation requests such as such as system account permission management for employees and form submission anomalies.

ACADEMIC PROJECT

10/2024-12/2024 **Compiler Development Project**

Project Overview

- **Goal:** Built a compiler capable of recognizing a custom-defined language from scratch using C language
- **Modular Architecture:** Independent components for Lexer, Parser, Evaluator, and Symbol Table
- **Functionality:** Character validation; Syntax validation; Semantic validation
- **Output:** Execution results formatted in JSON

Technical Implementation

- **Lexer (Lexical Analyzer):** Stated machine-based keyword recognition (e.g., let, int, set); Regex-based matching for identifiers (lowercase letters only) and integer constants (with max length restriction); Symbol table management using a chained hash table for variable storage and retrieval
- **Parser (Syntax Analyzer):** Constructed an SLR(1) parsing table with 28 states → Implemented LR Parsing Algorithm (Shift-Reduce Parsing) to supports 34 production rules (including left-recursion elimination) → Built and stores an Abstract Syntax Tree (AST) → Made grammatical rule judgment
- **Evaluator (Execution Engine):** Used LR Parsing Algorithm to execute source code → Processed semantic rules and outputted results
- **Output Generation:** Recursively traversed the AST and exported results in JSON format

Key Challenges

- **Manual SLR Table Adaptation:** Handled complex grammar rules (e.g., D → let T id be E, C → show A)
- **Dynamic AST Construction & Memory Management:** Multi-level node relationships; Sibling node pointer maintenance; Dynamic memory allocation based on grammar rules and source input
- **Token-to-Symbol Table Mapping:** Ensured proper variable resolution (e.g., id + id in grammar vs. a + b in input), and maintained consistency in evaluation and JSON output
- **Syntax Rule Conflict Resolution & Type Checking:** Enforced type rules within the LR Parsing Algorithm

04/2024-05/2024 Instant Messaging System based on TCP Protocol

Project Overview

- **Goal:** Developed a Winsock-based C/S (Client-Server) architecture network communication program supporting multi-user login, private chat, group chat, and user management functions
- **Core Functionality**

  Client-side: Supported functions including login, registration, private chat, group chat, and information modification

  Server-side:

  • Adopted a multi-thread model (dedicated thread for each client)

  •Maintained user and group information (full lifecycle management of registration/login/deletion)

  • Data persistence: Stored user and group data in files

Technical Implementation

- **Network Communication Architecture**

  Protocol: TCP persistent connection based on Winsock

  I/O Multiplexing: Monitored multiple socket connection states to improve concurrent performance

  Message format: Customized command prefix + data payload

  Group chat implementation:

  • Private chats → single-user message (username → Socket A)

  • Group chats → iterated through group members to send private messages (including group ID & senders)
- **Session Management:** Used mapping tables to maintain the relationship between the Socket and the user name. For private chats: User name → Socket A → Send information to Socket A
- **Data Management:** Periodically wrote user/group data to files to prevent data loss

Project Details

- **Atomic registration process:** Checked for username conflicts to prevent duplicate registration
- **Command framework:** Supported 10+ commands (e.g., REGISTER, LOGIN, CREATE_GROUP)
- **Thread safety:** Mutex protection for critical data

EXTRACURRICULAR ACTIVITY

09/2022-07/2023 UIC 3D Innovation and Development Lab

Member of the Technology Department

- Acquainted myself with 3D printing, including how to model 3D objects, debug and use 3D printers, etc.;
- Learned to design and sell related 3D products.

AWARD

- Participation in YOLO Overboard Model Detection Project
- Successful Participant in 2024 MCM/ICM
- Participant in The 5th Guangdong-Hong Kong-Macao Youth Robot Competition
- Honorable Mention of Guangdong Province, 2023 China Undergraduate Mathematical Contest in Modeling