

# Design Notes for Journal CLI

Manjunatha Sarma Majety

November 4, 2025

**Abstract**

**Contents**

<b>1</b>	<b>Motivation</b>	<b>2</b>
<b>2</b>	<b>Design Evolution</b>	<b>2</b>
2.1	Phase 1: Structure before clarity . . . . .	2
2.2	Phase 2: OOP refactor . . . . .	2

## 1 Motivation

## 2 Design Evolution

### 2.1 Phase 1: Structure before clarity

Initially, the entirety of the program was completely written in Procedural oriented architecture. It comprised of two functions, display and add entry for the display and addition of new entries respectively, with a helper function for acquiring timestamp and UNIX-to-human time conversion. Modularity gave it structure but the core was fragile, any edit to the `jrnl` file would break `getline()`. More importantly, the inherent procedural design of the program, made it difficult to add new functionality, always requiring either partial or complete structural rework. It functionally justified *my idea*, but was far too fragile for me to accept as a tool even for my own personal use.

Regardless, it was an important milestone for me, due to two primary reasons. It was my first coherent, complete, working :) program. Secondly, arriving to this point itself had much learning associated with it. I had understood `cmake`, how time libraries work(superfially, at the least), ANSI color codes and more importantly, a good practise of header-implementation file structure. All of this helped streamline my work, down-the-line.

### 2.2 Phase 2: OOP refactor