



Fase 1: Fundação Estrutural

Esta fase cria o corpo do shell, capaz de entender a linguagem mas ainda sem executar ações complexas.

- **Estrutura do Projeto:** Criação do `Makefile`, `minishell.h` e loop principal (`main`).
- **Sinais Interativos:** Configuração de `Ctrl-C`, `Ctrl-D` e `Ctrl-\` no prompt.
- **Análise Léxica (Lexer):** Transformar a string bruta do usuário em tokens (`WORD`, `PIPE`, `REDIR`).
- **Análise Sintática (Parser):** Organizar tokens em estruturas de comando (`t_cmd`), separando argumentos de redirecionamentos.

Fase 2: Sistema e Execução Simples

Esta fase dá vida ao shell, permitindo que ele execute programas simples e tenha memória.

- **Gerenciamento de Ambiente:** Capturar o `envp` do sistema e convertê-lo em uma lista encadeada manipulável (`t_env`).
- **Builtins Informativos:** Implementação de `env`, `pwd` e `exit`.
- **Resolução de Caminhos (PATH):** Lógica para encontrar onde os binários estão instalados no sistema.
- **Executor Simples:** Implementação do núcleo de execução com `fork`, `execve` e `waitpid` para um único comando.

Fase 3: Inteligência e Manipulação de Dados

Esta fase foca em processar o texto antes da execução e alterar o estado do shell.

- **Expansão de Variáveis:** Substituição de `$VAR` e `$?` pelos seus valores reais.
- **Tratamento de Aspas:** Lógica para remover aspas simples e duplas pós-expansão, preservando strings únicas.
- **Builtins de Manipulação:** Implementação de `export` (criar variáveis), `unset` (apagar variáveis) e `cd` (mudar diretório de trabalho).
- **Redirecionamentos de Arquivo:** Implementação de `<`, `>` e `>>` usando manipulação de File Descriptors (`dup2`).
- **Heredoc:** Implementação da entrada especial `<<` com delimitador.

Fase 4: Conexões e Polimento Final

Esta fase conecta múltiplos processos e garante a estabilidade do sistema.

- **Pipelines (|):** Lógica complexa para conectar a saída padrão (`STDOUT`) de um processo na entrada padrão (`STDIN`) do próximo.
- **Gerenciamento de Múltiplos Processos:** Sincronização de múltiplos filhos e gerenciamento de seus file descriptors.
- **Sinais em Processos Filhos:** Garantir que comandos como `cat` ou `grep` reajam corretamente ao `Ctrl-C` sem matar o shell pai.
- **Auditoria de Memória:** Verificação final com Valgrind para garantir zero vazamentos de memória (leaks).
- **Norminette:** Revisão final de estilo de código e conformidade com a Norma v4.