

# Minishell Projesi - İş Bölümü ve Planlama

## 1. Genel Strateji

Projeyi 4 ana aşamaya bölelim:

1. Altyapı ve Yapı (Setup)
2. Parsing (Komut satırı ayrıştırması)
3. Execution (Komutların çalıştırılması ve işlemler)
4. Signal yönetimi ve özel durumlar (örn: ctrl-C, heredoc)

Her aşamada görevler paylaşılabılır ve bu sırayla ilerlemek projeyi yönetilebilir kılar.

## 2. İş Bölümü Önerisi

**Alperen baba (Parser & Built-in):**

- Tokenizer, quote parsing, \$VAR parsing, redirection analizi
- Built-in komutlardan `export`, `unset`, `env`

**Sado (Executor & Signal):**

- `fork`, `execve`, `pipe`, `dup2`, `heredoc` uygulaması
- Built-in komutlardan `cd`, `echo`, `exit`, `pwd`, sinyal yönetimi

**İkiniz birlikte:**

- `readline` entegrasyonu, `Makefile`, `test`, `valgrind` leak kontrolü

## 3. Araştırılması Gerekenler

**Alperen baba :**

- Shell parsing mantığı
- Redirection türleri: `<`, `>`, `>>`, `<<`
- Environment değişkenleri ve `export`

**Sado:**

- Process yönetimi: `fork`, `execve`, `waitpid`
- `pipe` ve `dup2` kullanımı
- Signal handler yazımı ( `sigaction` )

**Ortak:**

- `readline` kullanımı, `exit` status yönetimi, `Makefile` hazırlama

## 5. To-Do List (Önerilen)

- ☐ `readline` prompt
- ☐ lexer/tokenizer
- ☐ quote parsing
- ☐ env variable expansion
- ☐ parser -> command struct
- ☐ redirection parse
- ☐ pipe parser
- ☐ executor ( `execve` , `fork` )
- ☐ redirection uygulaması
- ☐ heredoc
- ☐ signal handling
- ☐ built-in fonksiyonlar
- ☐ exit status yönetimi
- ☐ test script'leri