Intensive LibFT Learning and Implementation Schedule

Optimized for Maximum Progress

Daily Fixed Schedule

- □ 07:00 - 08:00: Morning routine
- □ 08:00 - 09:30: Gym session (essential for mental clarity)
- □ 10:00 - 14:00: Main LibFT work block
- □ 14:00 - 15:00: Lunch break and short rest
- □ 15:00 - 19:00: Second LibFT work block
- □ 19:00 - 20:00: Review and planning for next day

Intensive 10-Day Plan

Week 1: Foundations and Core Functions

Day 1: Setup and Memory Functions

- **Morning Block (10:00 - 14:00):**
  - Development environment setup (30min)
  - Study memory management concepts (1hr)
  - Implement and test:
    - ft_memset
    - ft_bzero
    - ft_memcpy
- **Evening Block (15:00 - 19:00):**
  - Study pointer arithmetic (1hr)
  - Implement and test:
    - ft_memmove
    - ft_memchr
    - ft_memcmp

Day 2: String Essentials

- **Morning Block:**
  - Study string concepts (1hr)
  - Implement and test:

        - ft_strlen
        - ft_strlcpy
        - ft_strlcat
        - ft_strchr
- **Evening Block:**
    - Implement and test:
        - ft_strrchr
        - ft_strncmp
        - ft_strnstr
        - ft_strdup

Day 3: Character Functions and Type Conversion

- **Morning Block:**
    - Study ASCII table and character types (1hr)
    - Implement and test:
        - ft_isalpha
        - ft_isdigit
        - ft_isalnum
        - ft_isascii
        - ft_isprint
- **Evening Block:**
    - Study number conversion (1hr)
    - Implement and test:
        - ft_toupper
        - ft_tolower
        - ft_atoi
        - ft_calloc

Day 4: Advanced String Operations Part 1

- **Morning Block:**
    - Study string manipulation and memory allocation (1hr)
    - Implement and test:
        - ft_substr
        - ft_strjoin
- **Evening Block:**
    - Implement and test:
        - ft_strtrim
        - ft_split (first part)

Day 5: Advanced String Operations Part 2

- **Morning Block:**
    - Complete and optimize ft_split
    - Study number-string conversion

    - Start ft_itoa
- **Evening Block:**
    - Complete ft_itoa
    - Extensive testing of all string operations
    - Fix any issues found

Week 2: Advanced Functions and Bonus

Day 6: Function Pointers and File Operations

- **Morning Block:**
    - Study function pointers (1hr)
    - Implement and test:
        - ft_strmapi
        - ft_striteri
- **Evening Block:**
    - Study file descriptors (1hr)
    - Implement and test:
        - ft_putchar_fd
        - ft_putstr_fd
        - ft_putendl_fd
        - ft_putnbr_fd

Day 7: Linked Lists Foundations

- **Morning Block:**
    - Study linked list concepts (1.5hrs)
    - Implement and test:
        - ft_lstnew
        - ft_lstadd_front
        - ft_lstsize
- **Evening Block:**
    - Implement and test:
        - ft_lstlast
        - ft_lstadd_back
        - ft_lstdelone

Day 8: Advanced List Operations

- **Morning Block:**
    - Implement and test:
        - ft_lstclear
        - ft_lstiter
- **Evening Block:**
    - Implement and test:

        - ft_lstmap
    - Thorough testing of all list functions

Day 9: Comprehensive Testing

- **Morning Block:**
    - Create comprehensive test suite
    - Test all mandatory functions
    - Memory leak checks
- **Evening Block:**
    - Test all bonus functions
    - Performance optimization
    - Fix any issues found

Day 10: Final Review and Preparation

- **Morning Block:**
    - Final round of testing
    - Memory leak checks with Valgrind
    - Code optimization
- **Evening Block:**
    - Documentation review
    - Prepare for evaluation
    - Final checks and fixes

Success Tips for Intensive Schedule:

1. **Stay Focused:**
    - Use Pomodoro technique (45min work, 15min break)
    - Minimize distractions during work blocks
2. **Health Maintenance:**
    - Don't skip gym - it's crucial for mental performance
    - Take short walks during breaks
    - Maintain good nutrition and hydration
3. **Learning Approach:**
    - Understand concepts before implementation
    - Write tests before functions (Test-Driven Development)
    - Keep man pages open while working
4. **Progress Tracking:**
    - Mark completed functions

    - Note challenging areas for review
    - Keep a bug list for systematic fixing

5. **Efficiency Techniques:**

    - Use a debugger consistently
    - Keep reference materials organized
    - Review previous day's work each morning

Emergency Buffer:

- Days 9-10 can be used to catch up if needed
- Morning study time can be shortened if concepts are well understood
- Break times can be adjusted based on progress

Daily Review Checklist:

- [ ] All implemented functions work correctly
- [ ] Memory leaks checked
- [ ] Tests written and passing
- [ ] Concepts understood
- [ ] Notes taken for next day