Lab 6 PreLab

Main Steps/Commands Involved in Reading Data from a File

Opening a file: FILE *fp = fopen("filename.txt", "r");

Modes: "r" (read), "w" (write), "a" (append)

Reading from a file: Use file reading functions like fscanf(), fgets(), etc.

Checking for end of file (EOF): while (!feof(fp))

Closing a file: fclose(fp);

FILE Pointers and I/O Streams (stdin, stdout, stderr)

FILE Pointers: FILE *fp

Used to manage file streams

I/O Streams:

stdin: Standard input stream (usually keyboard input)

stdout: Standard output stream (usually the terminal)

stderr: Standard error stream (usually terminal, for error messages)

Input/Output Functions in C

scanf(): Reads formatted input from stdin

Example: scanf("%d", &var);

sscanf(): Reads formatted input from a string

Example: sscanf(buffer, "%d", &var);

fscanf(): Reads formatted input from a file

Example: fscanf(fp, "%d", &var);

fgets(): Reads a line from a file or stdin into a buffer

Example: fgets(buffer, 100, fp);

printf(): Prints formatted output to stdout

Example: printf("%d", var);

sprintf(): Writes formatted output to a string

Example: sprintf(buffer, "%d", var);

fprintf(): Prints formatted output to a file

Example: fprintf(fp, "%d", var);

fputs(): Writes a string to a file or stdout

Example: fputs(buffer, fp);

Important Commands for vi/vim

Opening a file: vi filename.c

Inserting text: Press i for insert mode

Saving and quitting: :wq

Exiting without saving: :q!

Copying text: yy (yank)

Pasting text: p (put)

Searching: /word (find word in file)

Strategies for Breaking a Program into Multiple Files

Separate code into header files (.h) and source files (.c)

Use header files to declare functions, and source files for implementation

Example:

main.c: Contains main() function and calls functions from other files

fileio.c: Implements file reading and writing functions

fileio.h: Declares functions implemented in fileio.c

Using Makefiles

Why Use a Makefile?

- Automates compilation of multi-file projects
- Tracks dependencies and only recompiles necessary files

Commands in Makefile

make: Compiles the program by running the rules in the Makefile

make clean: Removes compiled object files (.o) and the executable