Creating and Using a Custom Static Library in C

A Beginner's Tutorial

Date: December 16, 2025

This guide walks you through creating a simple custom library in C, promoting code modularity and reuse. We'll build a library with one function and link it to a main program.

Step 1: The Header File (`mylib.h`)

Headers declare functions and types.

```c
/* mylib.h */
#ifndef MYLIB_H
#define MYLIB_H

void hello(void);

#endif
```

- **Include guards** prevent duplicate inclusions.
- Save as `mylib.h`.

Step 2: Library Implementation (`mylib.c`)

```c
/* mylib.c */
#include <stdio.h>
#include "mylib.h"

void hello(void)
{
    printf("HELLO WORLD! with my personal library\n");
}
```

- Use quotes `""` for custom headers.
- Save as `mylib.c`.

Step 3: Main Program (`main.c`)

```c
/* main.c */
#include <stdio.h>
```

```c
#include "mylib.h"

int main(void)
{
    hello();

    printf("HELLO PROGRAMMER! This greeting does not use the library.\n");

    return 0;
}
```

- Save as `main.c`.

Step 4: Compiling and Linking

Simple One-Line Compilation

```
gcc main.c mylib.c -o myprogram
```

Creating a Static Library

```
gcc -c mylib.c
ar rcs libmylib.a mylib.o
gcc main.c -L. -lmylib -o myprogram
```

Expected Program Output

```
HELLO WORLD! with my personal library
HELLO PROGRAMMER! This greeting does not use the library.
```