Understanding GCC Commands for Static Libraries in C/C++

A Beginner-Friendly Guide (English Version)

This guide explains step by step how GCC commands work when creating and using static libraries in C/C++.

Step 1: Compile the Library Implementation

```
gcc -c mylib.c
```

- `gcc` = the compiler for C/C++.
- `-c` = compile the source file into an **object file (`.o`)** without linking.
- `mylib.c` = the file containing your function definitions.

💡 **Result:** - You get `mylib.o` which contains compiled code but is **not linked yet** to `main.c` or other libraries.

Step 2: Create a Static Library

```
ar rcs libmylib.a mylib.o
```

- `ar` = command to create an **archive/static library (`.a`)**.
- `rcs` = flags:
- `r` = insert/replace object file into the library
- `c` = create the library if it does not exist
- `s` = create an index for faster linking
- `libmylib.a` = name of the static library to create
- `mylib.o` = object file to include in the library

💡 **Result:** - You now have a **static library** containing all functions from `mylib.o`. - You can use it in any program without recompiling `mylib.c`.

Step 3: Link the Library to the Main Program

```
gcc main.c -L. -lmylib -o myprogram
```

- `main.c` = the main program using functions from the library

- `-L.` = tell the compiler to look in the **current folder** for libraries
- `-lmylib` = link with the library `libmylib.a` (note: you write `-l<name>` without `lib` or `.a`)
- `-o myprogram` = output executable file

💡 **Result:** - You get an executable `myprogram` ready to run. - Functions from `libmylib.a` are available without recompiling `mylib.c`.

Using the Library in a New File

Once `libmylib.a` is created, any new source file can use the functions from the library without recompiling `mylib.c`:

```
gcc newfile.c -L. -lmylib -o newprogram
```

- `newfile.c` = your new source file
- `-L.` = tells the compiler where to find the library
- `-lmylib` = link with `libmylib.a`
- `-o newprogram` = final executable

🔗 **Important:** - Every time you add a new file using the library, you just link it. No need to recompile the library. - This is the main advantage of static libraries.

Bonus Tip: Automate with Makefile

- You can create a **Makefile** to automate compiling and linking multiple files.
- This saves typing all the commands every time you add a new file or function.

This approach ensures **modularity, code reuse, and faster compilation** for larger projects.