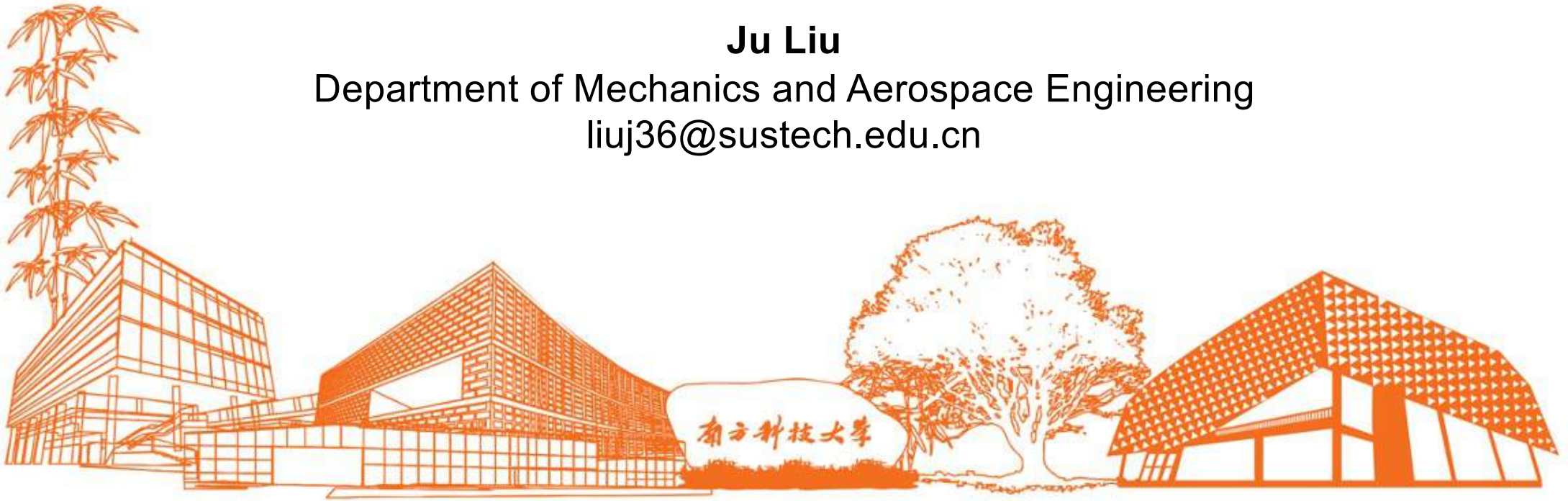
MAE 5032 High Performance Computing: Methods and Applications

Lab 4: Compile & Link

Ju Liu
Department of Mechanics and Aerospace Engineering
liuj36@sustech.edu.cn



Objective

- You will experiment two .c files and one .h file
  - experience the invocation of compiler
  - experience the build process for static and dynamic libraries
  - experience the linking process

Task 1: Obtain the code

- Go to https://github.com/ju-liu/MAE-5032-2025S/tree/main/week-05/ex-3-multiple-files

- Download the source code

```c
#include "hello.h"

int main()
{
  hello("world");
  return 0;
}
```

main.c

```c
void hello( const char * name );
```

hello.h

```c
#include <stdio.h>
#include "hello.h"

void hello( const char * name )
{
  printf("Hello %s!\n", name);
}
```

hello.c

Task 1: simple compiling

- Creating executables from source files

```
gcc hello.c main.c -o hello
```

Task 1: simple compiling

- Compile without linking

```
gcc -c hello.c
gcc -c main.c
```

- Creating executables from object files

```
gcc hello.o main.o -o hello
```

Task 2: Static library

- Compile without linking

```
gcc -c hello.c
gcc -c main.c
```

- Creating a static library libhello.a from object files

```
ar rs libhello.a hello.o
```

- Link the library to generate the executable

```
gcc main.o -o hello -L/PATH -lhello
```

Task 2: Dynamic library

- Compile with position independent code

```
gcc -c -Wall -fpic hello.c
```

- Creating a dynamic library libhello.so from object files

```
gcc -shared -o libhello.so hello.o
```

- Move libhello.so to a lib folder and hello.h to an include folder

- Regenerate the main.o

```
gcc -c main.c -I/INCLUDE-PATH
```

Task 2: Dynamic library

- Link the dynamic lib

```
gcc main.o -o hello -L/PATH-TO-LIB -lhello
```

- Run the code

- You may want to set the LD_LIBRARY_PATH (Linux) or DYLD_LIBRARY_PATH (Mac)

- Link the dynamic lib with –rpath

```
gcc main.o -o hello -L/PATH-TO-LIB -lhello -Wl,-rpath
PATH-TO-LIB
```

Task 3: optional

- What will happen if you do not have #include “hello.h” in the main.c?
- What will happen if you do not have –lhello during linking?
- How to maintain/add more functions for the library?

Task 4: install a lib to user-specified location

- Sometimes you do not have **sudo** permission
- You need to install the lib to your own folder
- We will go over this using fftw as an example

Download GitHub Mailing List Benchmark Features Documentation FAQ Links Feedback

www.fftw.org

Task 4: install a lib to user-specified location

Do the following and explain their purpose

1. wget http://www.fftw.org/fftw-3.3.10.tar.gz
2. tar –zxvf fftw-3.3.10.tar.gz
3. mv fftw-3.3.10 fftw-3.3.10-src
4. cd fftw-3.3.10-src

Task 4: install a lib to user-specified location

The following are three standard procedures for library install

./configure --prefix=$HOME/lib/fftw-3.3.10 --enable-shared

make –j6

make install

Node: some make use other tools to customize the build process. You may need to pay some attention for the prefix setting.

Task 5: link the following code to fftw

```c
#include <stdio.h>
#include <fftw3.h>

int main() {
    int N = 8;
    fftw_complex in[N], out[N], inv[N];
    fftw_plan p_forward, p_backward;

    // Fill input with real values, imag = 0
    for (int i = 0; i < N; ++i) {
        in[i][0] = i + 1;  // real part
        in[i][1] = 0.0;    // imag part
    }

    // Create plans
    p_forward = fftw_plan_dft_1d(N, in, out, FFTW_FORWARD, FFTW_ESTIMATE);
    p_backward = fftw_plan_dft_1d(N, out, inv, FFTW_BACKWARD, FFTW_ESTIMATE);

    // Execute forward FFT
    fftw_execute(p_forward);

    printf("FFT result:\n");
    for (int i = 0; i < N; ++i)
        printf("out[%d] = %.2f + %.2fi\n", i, out[i][0], out[i][1]);

    // Execute inverse FFT
    fftw_execute(p_backward);

    printf("\nInverse FFT (should recover input):\n");
    for (int i = 0; i < N; ++i)
        printf("inv[%d] = %.2f + %.2fi\n", i, inv[i][0] / N, inv[i][1] / N);

    // Cleanup
    fftw_destroy_plan(p_forward);
    fftw_destroy_plan(p_backward);
    fftw_cleanup();

    return 0;
}
```

This code performs discrete Fourier transform for a real array [1,2,3,4,5,6,7,8], and then perform its inverse transform.

Compile this code and have it correctly linked to the library you just installed?