

Lab2

Preprocessor

- There are certain line markers which indicate the file name and line numbers. Helps the compiler and debugging tools to track the original source code locations.
- Header file full contents like `stdio.h` and its dependencies.
- The comments are taken out.

Compilation

I compiled using “gcc -Wall -Wextra -Wfatal-errors -Wpedantic -o program main.c numbers.c”

- Wall = Enables “all” compiler warning messages. Well not all anymore atleast.. Nowadays enables a large selection of useful flags! From GCC’s own manual I found the line “enables all the warnings about constructions that some users consider questionable, and that are easy to avoid”.
- Wextra = Enables some extra warning flags that are not enabled by wall.
- Wfatal-errors aborts compilation on the first error occurred.
- Wpedantic warns user if using non-standard extensions or write code that violates the strict C standard.

Execution

```
[sulo@sulo-arch src]$ gcc -Wall -Wextra -Wfatal-errors -Wpedantic -o program main.c numbers.c
```

```
[sulo@sulo-arch src]$ ./program 2 10 7
```

```
Generated number: 5
```

```
Generated number: 4
```

```
Generated number: 10
```

```
Generated number: 2
```

```
Generated number: 7
```

```
Generated number: 8
```

```
Generated number: 6
```

Now it is time to compile our files. Please make sure to compile with the -Wall, -Wextra, -Wfatal-errors, and -Wpedantic active. Please identify what all of these do and make sure that you program compiles with no errors or warnings. Please define what all of these compiler flags do.

Look for other compiler flags and experiment.

```
[sulo@sulo-arch src]$ ./program 2 5 9
```

```
Invalid range!
```

```
[sulo@sulo-arch src]$ ./program a 3 22
```

```
Invalid arguments given! Run the program again >:(
```

```
[sulo@sulo-arch src]$ ./program a 2 fefef4 2
```

```
Invalid arguments given! Run the program again >:(
```

```
[sulo@sulo-arch src]$ ./program a 2
```

```
Enter start of range:
```

```
2
```

```
Enter end of range:
```

```
5
```

```
Enter unique number count:
```

```
3
```

```
Generated number: 4
```

```
Generated number: 5
```

```
Generated number: 2
```

```
[sulo@sulo-arch src]$ ./program
```

```
Enter start of range:
```

```
2
```

```
Enter end of range:
```

```
3
```

```
Enter unique number count:
```

```
55
```

```
Invalid range!
```

```
[sulo@sulo-arch src]$ Execution and debugging
```