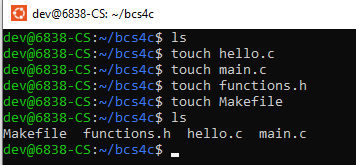
1. **Step 1**: Creating multiple (dependent) files



1. **Step 2**: Write Code in .C files
   1. hello.c file

| #include <stdio.h>   *//definition of hello function*   void hello() {    *// printf() displays the string inside quotation*    printf("Hello, World!");   } |
| --- |

* 1. main.c file

| #include <stdio.h>   #include "functions.h"   int main(){    hello();    return 0;   } |
| --- |

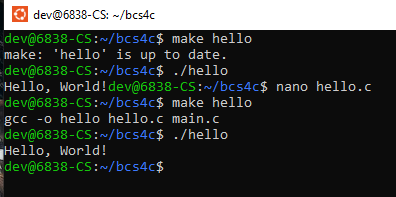
* 1. function.h file

| *// function declaration here*   void hello(); |
| --- |

1. **Step 3**: Write execution steps in Makefile (use tab before gcc, extra spaces can cause error)

| hello: hello.c main.c    gcc -o hello hello.c main.c -I. |
| --- |

1. **Step 4:** Execute files with Makefile



* **Updating Code to work with command line arguments**

1. Hello.c

| void print\_arguments(int argc, char \*argv[]) {  printf("Number of arguments: %d\n", argc - 1);  for (int i = 1; i < argc; i++) {  printf("Argument %d: %s\n", i, argv[i]);  } } |
| --- |

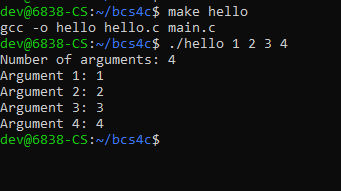
1. Main.c

| int main(int argc, char \*argv[]) {  if (argc < 2) {  printf("Usage: %s <argument1> <argument2> ...\n", argv[0]);  return 1;  }   print\_arguments(argc, argv);   return 0; } |
| --- |

1. Function.h

| void print\_arguments(int argc, char \*argv[]); |
| --- |

1. Makefile (same as step3 will work fine)



Makefile example using object files:

