

Managing Linux with the Embedded Perspective Exercise 2

Created necessary files.

```
mark@fernando:~/metropolia/3rd_year/embedded_linux/ex_2
> touch main.c random.c random.h
~/m/3/embedded_linux/ex_2
```

Testing the code

```
mark@fernando:~/metropolia/3rd_year/embedded_linux/ex_2
}
> nvim random.c
> gcc main.c random.c random.h -o program
> ./program
Enter the range that you wish to get the random no
Enter the range low: ff
Enter valid no: fff
Enter valid no: fff
Enter valid no: fff
Enter valid no: fff
Enter valid no: fff
Enter valid no: fff
Enter valid no: 23
> nvim main.c
> gcc main.c random.c random.h -o program
> ./program
Enter the range that you wish to get the random no
Enter the range low: ff
Enter valid no: g
Enter valid no:
Enter valid no:
Enter valid no:
Enter valid no: 2
Enter range high: gg
Enter valid no: gg
Enter valid no: g
Enter valid no: 6
Low: 2 High: 6%
> nvim main.c
> gcc main.c random.c random.h -o program
> ./program
Enter the range that you wish to get the random no
Enter the range low: 2
Enter range high: 8
Low: 2 High: 8
> nvim random.c
> nvim random.h
> nvim random.c
> gcc main.c random.c random.h -o program
> ./program
Enter the range that you wish to get the random no
Enter the range low: 2
Enter range high: 6
Low: 2 High: 6
~/m/3/embedded_linux/ex_2
```

I took out all errors before doing the preprocessor. Now I have only the warnings.

```
mark@fernando:~/metropolia/3rd_year/embedded_linux/ex_2
> gcc -E main.c > premain.c
> gcc -E random.c > prerandom.c
> gcc -E random.h > prerandomh.h
~/m/3/embedded_linux/ex_2
```

Preprocessor

Premain.c

```
nvim premain.c
1 1 # 0 "main.c"
2 1 # 0 "<built-in>"
3 2 # 0 "<command-line>"
4 3 # 1 "/usr/include/stdc-predef.h" 1 3 4
5 4 # 0 "<command-line>" 2
W 5 5 # 1 "main.c"      ■ unused variable 'buffer' [-Wunused-variable]
6 6 # 1 "/usr/include/stdio.h" 1 3 4
7 7 # 28 "/usr/include/stdio.h" 3 4
8 8 # 1 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 1 3 4
W 9 9 # 33 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 3 4      ■ unused variable
W 10 1 # 1 "/usr/include/features.h" 1 3 4      ■ unused variable 'high_state' [-Wunused-variable]
11 11 # 394 "/usr/include/features.h" 3 4
12 12 # 1 "/usr/include/features-time64.h" 1 3 4
13 13 # 20 "/usr/include/features-time64.h" 3 4
14 14 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
15 15 # 21 "/usr/include/features-time64.h" 2 3 4
16 16 # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
17 17 # 19 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 3 4
18 18 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
19 19 # 20 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 2 3 4
20 20 # 22 "/usr/include/features-time64.h" 2 3 4
21 21 # 395 "/usr/include/features.h" 2 3 4
22 22 # 502 "/usr/include/features.h" 3 4
23 23 # 1 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 1 3 4
24 24 # 576 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 3 4
25 25 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
26 26 # 577 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
27 27 # 1 "/usr/include/x86_64-linux-gnu/bits/long-double.h" 1 3 4
28 28 # 578 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
29 29 # 503 "/usr/include/features.h" 2 3 4
30 30 # 526 "/usr/include/features.h" 3 4
31 31 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 1 3 4
32 32 # 10 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 3 4
33 33 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs-64.h" 1 3 4
34 34 # 11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
35 35 # 527 "/usr/include/features.h" 2 3 4
36 36 # 34 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 2 3 4
37 37 # 29 "/usr/include/stdio.h" 2 3 4
38
39
40
41
42
premain.c 1,1 Top
```

Prerandom.c

```
12 # 0 "random.c"
11 # 0 "<built-in>"
10 # 0 "<command-line>"
9 # 1 "/usr/include/stdc-predef.h" 1 3 4
8 # 0 "<command-line>" 2
7 # 1 "random.c"
6 # 1 "/usr/include/stdio.h" 1 3 4
5 # 28 "/usr/include/stdio.h" 3 4
4 # 1 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 1 3 4
3 # 33 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 3 4
2 # 1 "/usr/include/features.h" 1 3 4
1 # 394 "/usr/include/features.h" 3 4
13 # 1 "/usr/include/features-time64.h" 1 3 4
1 # 20 "/usr/include/features-time64.h" 3 4
2 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
3 # 21 "/usr/include/features-time64.h" 2 3 4
4 # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
5 # 19 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 3 4
6 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
7 # 20 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 2 3 4
8 # 22 "/usr/include/features-time64.h" 2 3 4
9 # 395 "/usr/include/features.h" 2 3 4
10 # 502 "/usr/include/features.h" 3 4
11 # 1 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 1 3 4
12 # 576 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 3 4
13 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
14 # 577 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
15 # 1 "/usr/include/x86_64-linux-gnu/bits/long-double.h" 1 3 4
16 # 578 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
17 # 503 "/usr/include/features.h" 2 3 4
18 # 526 "/usr/include/features.h" 3 4
19 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 1 3 4
20 # 10 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 3 4
21 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs-64.h" 1 3 4
22 # 11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
23 # 527 "/usr/include/features.h" 2 3 4
24 # 34 "/usr/include/x86_64-linux-gnu/bits/libc-header-start.h" 2 3 4
25 # 29 "/usr/include/stdio.h" 2 3 4
26
27
28
W 29 ■ unused variable 'input_is_charactor' [-Wunused-variable]
30
```

In the preprocessing stage all `#include` are replaced with full header files and Macros are expanded. Also, comments are removed. As a result, the preprocessor file is larger than the original c file.

I could see the warnings, but I didn't see any errors since I had already cleared them before running the preprocessor step. So, I intentionally broke the code to generate errors, so I could study them.

I broke the random.c file.

```
40 # 0 "random.c"
41 # 0 "<built-in>"
42 # 0 "<command-line>"
43 # 1 "/usr/include/stdc-predef.h" 1 3 4
44 # 0 "<command-line>" 2
45 # 1 "random.c"
46
47 # 1 "random.h" 1
48
49 # 33 "length" undeclared (first use in this function)
50
51 void random_num(int low, int high);
52 int input();
53 # 3 "random.c" 2
54 # 1 "/usr/lib/gcc/x86_64-linux-gnu/13/include/stdbool.h" 1 3 4
55 # 4 "random.c" 2
56 # 1 "/usr/include/time.h" 1 3 4
57 # 25 "/usr/include/time.h" 3 4
58 # 1 "/usr/include/features.h" 1 3 4
59 # 394 "/usr/include/features.h" 3 4
60 # 1 "/usr/include/features-time64.h" 1 3 4
61 # 20 "/usr/include/features-time64.h" 3 4
62 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
63 # 21 "/usr/include/features-time64.h" 2 3 4
64 # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
65 # 19 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 3 4
66 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
67 # 20 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 2 3 4
68 # 22 "/usr/include/features-time64.h" 2 3 4
69 # 395 "/usr/include/features.h" 2 3 4
70 # 582 "/usr/include/features.h" 3 4
71 # 1 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 1 3 4
72 # 576 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 3 4      incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
73 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
74 # 577 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4      incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
75 # 1 "/usr/include/x86_64-linux-gnu/bits/long-double.h" 1 3 4
76 # 578 "/usr/include/x86_64-linux-gnu/sys/cdefs.h" 2 3 4
77 # 583 "/usr/include/features.h" 2 3 4
78 # 526 "/usr/include/features.h" 3 4
79 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 1 3 4
80 # 10 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 3 4
81 # 1 "/usr/include/x86_64-linux-gnu/gnu/stubs-64.h" 1 3 4      unused variable 'input_is_character' [-Wunused-variable]
82 # 11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
```

```
11 "/usr/include/x86_64-linux-gnu/gnu/stubs.h" 2 3 4
12 # 527 "/usr/include/features.h" 2 3 4
13 # 26 "/usr/include/time.h" 2 3 4
14
15
16 # 1 "/usr/lib/gcc/x86_64-linux-gnu/13/include/stddef.h" 1 3 4      incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
17 # 214 "/usr/lib/gcc/x86_64-linux-gnu/13/include/stddef.h" 3 4
18
19 # 214 "/usr/lib/gcc/x86_64-linux-gnu/13/include/stddef.h" 3 4      'stdin' undeclared (first use in this function)
20 typedef long unsigned int size_t;      incompatible implicit declaration of built-in function 'scanf' [-Wbuiltin-declaration-mismatch]
21 # 30 "/usr/include/time.h" 2 3 4
22
23
24      incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
25 # 1 "/usr/include/x86_64-linux-gnu/bits/time.h" 1 3 4
26 # 26 "/usr/include/x86_64-linux-gnu/bits/time.h" 3 4
27 # 1 "/usr/include/x86_64-linux-gnu/bits/types.h" 1 3 4
28 # 27 "/usr/include/x86_64-linux-gnu/bits/types.h" 3 4
29 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
30 # 28 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
31 # 1 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 1 3 4
32 # 19 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 3 4
33 # 1 "/usr/include/x86_64-linux-gnu/bits/wordsize.h" 1 3 4
34 # 20 "/usr/include/x86_64-linux-gnu/bits/timesize.h" 2 3 4
35 # 29 "/usr/include/x86_64-linux-gnu/bits/types.h" 2 3 4
36
37
38 typedef unsigned char __u_char;
39 typedef unsigned short int __u_short;
40 typedef unsigned int __u_int;
41 typedef unsigned long int __u_long;
```

Here the error says, “**Implicit declaration of function ‘fgets’ [-Wimplicit-function-declaration]**”. It means compiler saying you called fgets but it didn’t declare. This happens because it couldn’t find **stdio.h**

Compilation

Then I compiled both main .c and random.c

```
mark@fernando:~/metropolia/3rd_year/embedded_linux/ex_2
> gcc -Wall -Wextra -Wpedantic -c random.c -o random.o

random.c: In function 'random_num':
random.c:10:17: error: 'length' undeclared (first use in this function)
  10 |     int numbers[length];
      |                  ^~~~~~
random.c:10:17: note: each undeclared identifier is reported only once for each function it appears in
random.c:33:9: warning: implicit declaration of function 'printf' [-Wimplicit-function-declaration]
  33 |     printf("%d ", numbers[j]);
      |     ^~~~~~
random.c:6:1: note: include '<stdio.h>' or provide a declaration of 'printf'
   5 | #include <stdlib.h>
+++ |+#include <stdio.h>
   6 |
random.c:33:9: warning: incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
  33 |     printf("%d ", numbers[j]);
      |     ^~~~~~
random.c:33:9: note: include '<stdio.h>' or provide a declaration of 'printf'
random.c:35:5: warning: incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
  35 |     printf("\n");
      |     ^~~~~~
random.c:35:5: note: include '<stdio.h>' or provide a declaration of 'printf'
random.c:10:9: warning: unused variable 'numbers' [-Wunused-variable]
  10 |     int numbers[length];
      |     ^~~~~~
random.c:7:30: warning: unused parameter 'high' [-Wunused-parameter]
   7 | void random_num(int low, int high){
      |                      ~~~~~^~~~~~
random.c: In function 'input':
random.c:50:13: warning: incompatible implicit declaration of built-in function 'printf' [-Wbuiltin-declaration-mismatch]
  50 |     printf("Enter valid no: ");
      |     ^~~~~~
random.c:50:13: note: include '<stdio.h>' or provide a declaration of 'printf'
random.c:53:12: warning: implicit declaration of function 'fgets' [-Wimplicit-function-declaration]
  53 |     if(fgets(buffer, sizeof(buffer), stdin) != NULL){
      |     ^~~~~~
random.c:53:42: error: 'stdin' undeclared (first use in this function)
  53 |     if(fgets(buffer, sizeof(buffer), stdin) != NULL){
      |                                  ^~~~~~
random.c:53:42: note: 'stdin' is defined in header '<stdio.h>'; did you forget to '#include <stdio.h>'?
```

The major issue was that it threw a bunch of errors related to everything connected to stdio.h. It also showed an error saying it couldn't find the length variable.

I have noticed an interesting thing here.

```
> gcc -Wall -Wextra -Wpedantic -c main.c -o main.o

~/m/3/embedded_linux/ex_2 [ ] ok 20:41:46
```

When I tried to compile main.c without fixing the errors in random.c, it didn't give any errors. What I understood is that the compiler compiles files individually, and each .c file is handled independently. Since main.c had everything, it needed, it compiled correctly. However, issues came up during linking. Even though main.c compiled successfully, random.c didn't. So, when I tried to link, I got output as follows.

```
/usr/bin/ld: cannot find random.o: No such file or directory
collect2: error: ld returned 1 exit status
~/m3/embedded_linux/ex_2 20:53:24
```

We don't have the random.o file because the compiler terminated while compiling random.c.

Flags

-Wall --> Enables common warnings

-Wextra --> Enables additional warnings not included in -wall (e.g. Unused variables)

-Wfatal-errors --> Stops compilation immediately after the first error.

-Wpedantic --> Warns about non-standard practices. It helps developers to write clean code and standard-compliant code.

Other flags

-g, -pg --> debugging and information

-o, -Os --> Optimization

-std=c99, -std=c11, -ansi --> language standards

-l<library> --> Linking and libraries

-Wshadow, -Wconversion, -Wunreachable-code --> warnings and errors

-o <file>, -c, -s, -E --> Output control

Linking

`gcc main.o random.o -o program`

Once we fix all the errors, it compiles and links properly.

```
> gcc -Wall -Wextra -Wpedantic -g -c main.c -o main.o
gcc -Wall -Wextra -Wpedantic -g -c random.c -o random.o
gcc main.o random.o -o program

> ./program 5 12 5

Generating 5 unique random numbers between 5 and 12:
12 9 7 6 8
> ./program

Enter the range that you wish to get random numbers
Enter low: 5
Enter high: 12
Enter how many unique random numbers you want: 5
Generating 5 unique random numbers between 5 and 12:
11 10 7 6 9
~/m/3/e/ex_2  ok 27s 15:11:05
```

Debugging

I used debugger `-g` flag

`gcc -Wall -Wextra -Wpedantic -g -c main.c -o main.o`

`gcc -Wall -Wextra -Wpedantic -g -c random.c -o random.o`

`gcc main.o random.o -o program`

Commands that I used to debug

(gdb) `break main` --> goes to the main

(gdb) `run` --> run the debugger

(gdb) `next` --> next line

(gdb) `step` --> goes into the function

```
gdb ./program
Enter how many unique random numbers you want: 4
Generating 4 unique random numbers between 9 and 15:
13 9 12 15
> gdb ./program
GNU gdb (Ubuntu 15.0.50.20240403-0ubuntu1) 15.0.50.20240403-git
Copyright (C) 2024 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./program...
(gdb) break main
Breakpoint 1 at 0x127c: file main.c, line 16.
(gdb) run
Starting program: /home/mark/metropolia/3rd_year/embedded_linux/ex_2/program
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main (argc=1, argv=0x7fffffffde18) at main.c:16
16      if (argc == 4) {
(gdb) n
33      } else if(argc == 1){
(gdb) n
35      printf("Enter the range that you wish to get random numbers\n");
(gdb) n
Enter the range that you wish to get random numbers
37      printf("Enter low: ");
(gdb) n
38      low = input();
(gdb) n
Enter low: rrr
Invalid input. Enter again: rrr
Invalid input. Enter again: -9
Enter a positive no. Enter again: 1
41      printf("Enter high: ");
(gdb) 
```

I played around with debugger with different kind of inputs.

I tested out all scenarios.

- 1) Characters
- 2) negative no
- 3) For the range, the high number and no of random n must not be less than the low number
- 4) No of random numbers should be less than the range

Then I tested the program with all possible inputs.

```
> ./program

Enter the range that you wish to get random numbers
Enter low: ggggg
Invalid input. Enter again: -9
Enter a positive no. Enter again: 5
Enter high: ggg
Invalid input. Enter again: -9
Enter a positive no. Enter again: 4
High must be greater than low.
Enter high: 10
Enter how many unique random numbers you want: -9
Enter a positive no. Enter again: 10
Error: Count cannot exceed range size.
Enter how many unique random numbers you want: 3
Generating 3 unique random numbers between 5 and 10:
8 9 6

~/m/3/e/ex_2
```

```
> ./program 2 10 12

Error! Count cannot exceed range size.
> ./program 2 gg 52

Error! last three arguments must be numbers and positive
> ./program 10 8 12

Error! High should be greater than low.
> ./program 10 15 6

Generating 6 unique random numbers between 10 and 15:
13 15 12 14 11 10

~/m/3/embedded_linux/ex_2
```

Input validation is fine, and the output is as expected. Then, I also generated the assembly output to see what's inside.

```
nvim main.s
42 .file "main.c"
41 .text
40 .section .rodata
39 .align 8
38 .LC0:
37 .string "Enter the range that you wish to get the random no"
36 .LC1:
35 .string "Enter the range low: "
34 .LC2:
33 .string "Enter the range high: "
32 .LC3:
31 .string "Enter high again: "
30 .LC4:
29 .string "Low: %d High: %d\n"
28 .text
27 .globl main
26 .type main, @function
25 main:
24 .LFB0:
23 .cfi_startproc
22 endbr64
21 pushq %rbp
20 .cfi_def_cfa_offset 16
19 .cfi_offset 6, -16
18 movq %rsp, %rbp
17 .cfi_def_cfa_register 6
16 subq $16, %rsp
15 movb $0, -13(%rbp)
14 movl $0, -12(%rbp)
13 leaq .LC0(%rip), %rax
12 movq %rax, %rdi
11 call puts@PLT
10 leaq .LC1(%rip), %rax
9 movq %rax, %rdi
8 movl $0, %eax
7 call printf@PLT
6 movl $0, %eax
5 call input@PLT
4 movl %eax, -8(%rbp)
3 .L6:
2 cmpl $0, -12(%rbp)
1 jne .L2
43 leaq .LC2(%rip), %rax
main.s 43,2-5 Top
```

Then I looked at the .o file. Since it's not a human readable, file I didn't pay much attention to it.

[illegible]