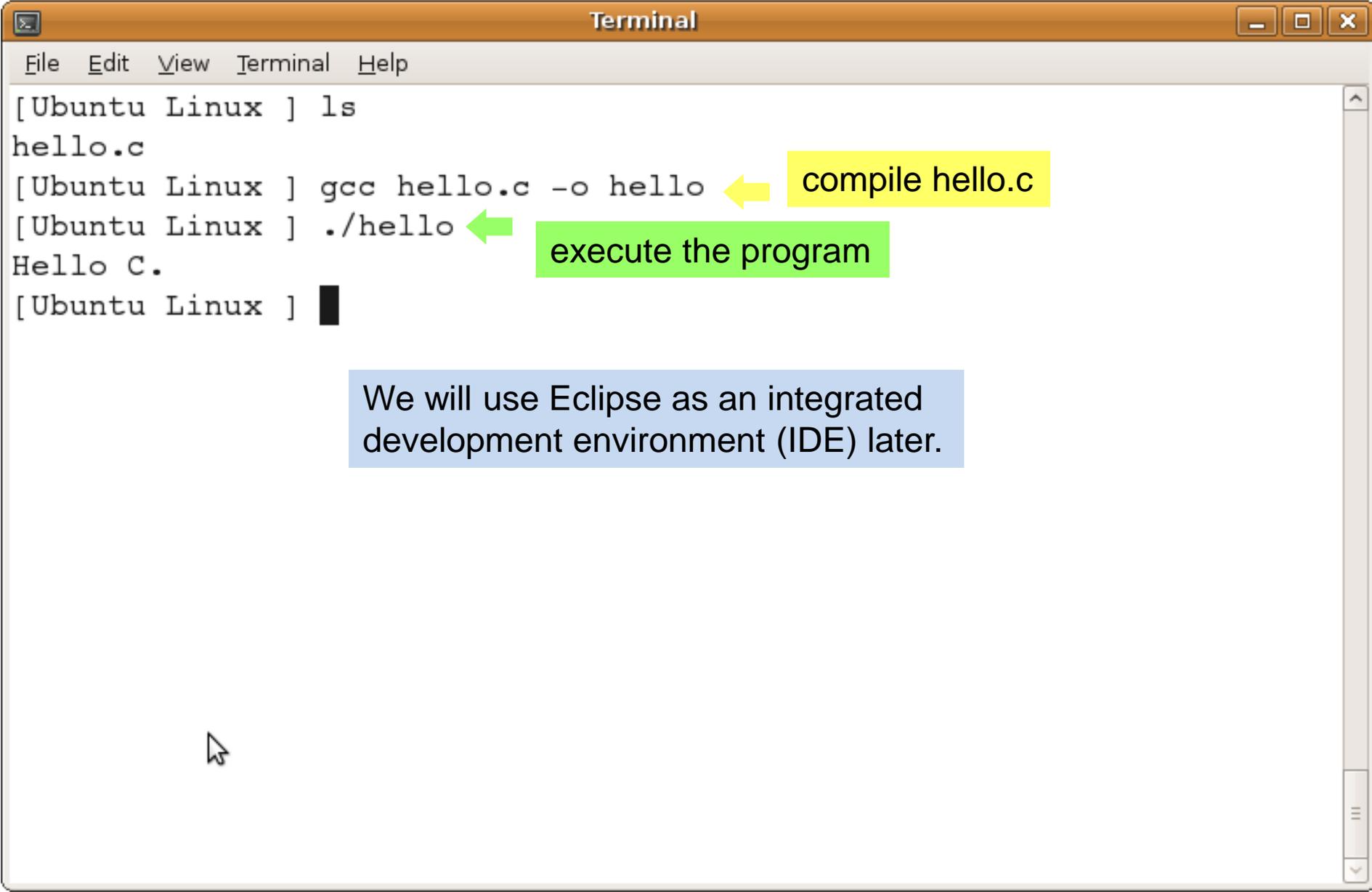


Simple Examples of C Programs

Yung-Hsiang Lu

hello.c



We will use Eclipse as an integrated development environment (IDE) later.

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    printf("Hello C.\n");
    return 0;
}
```

print "Hello C."
\n means new line
printf is a library function

C programs are case-sensitive.
Printf, printf, printF are **different**.

"hello.c" 7 lines, 91 characters

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    printf("Hello C.\n");
    return 0;
}
```

return 0 if the program ends normally
return -1 if the program ends abnormally

```
"hello.c" 7 lines, 91 characters
```

Terminal

File Edit View Terminal Help

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    printf("Hello C.\n");
    return 0;
}
```

“main” is a special function and
it is the starting point of the program

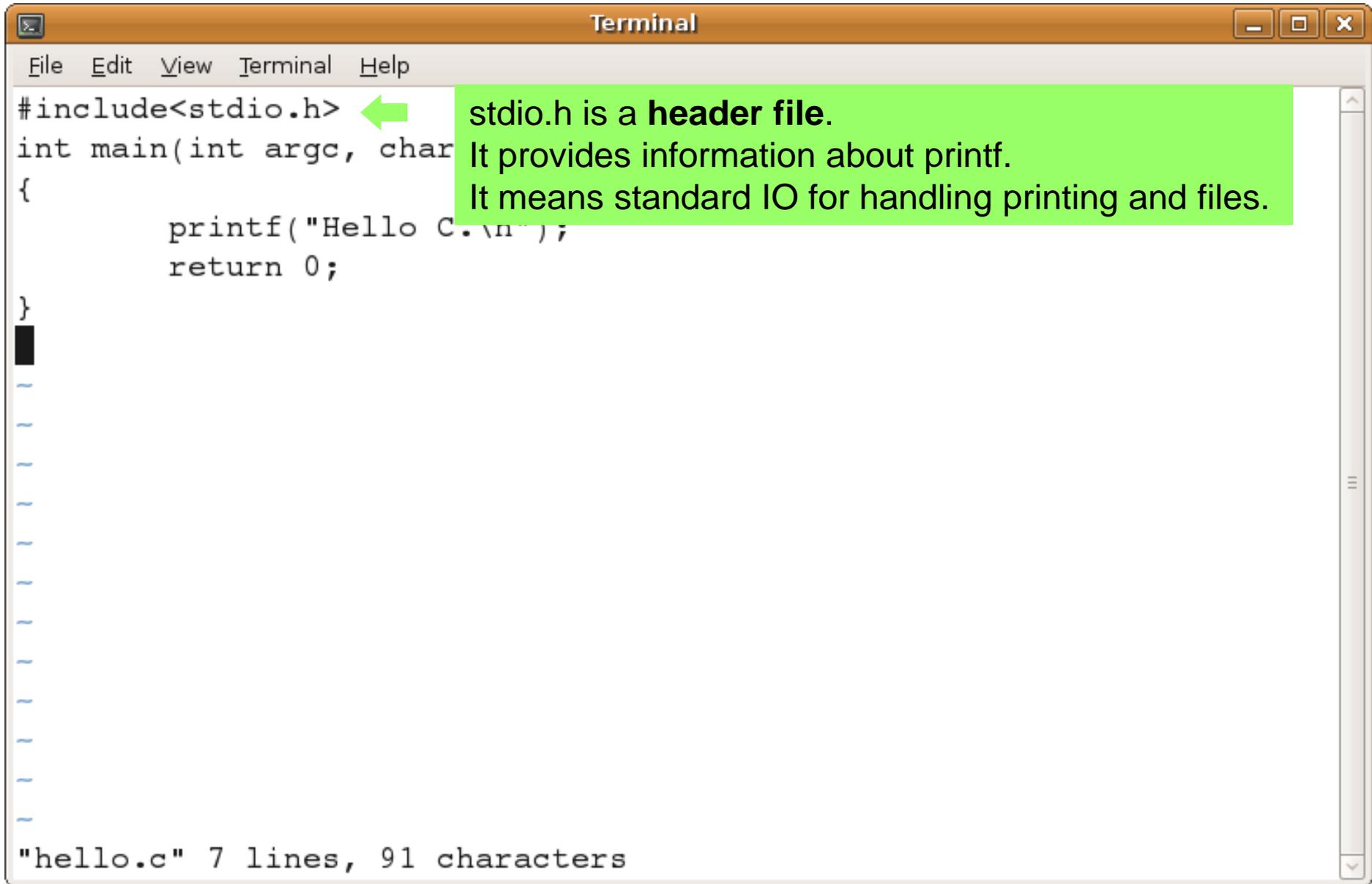
"hello.c" 7 lines, 91 characters

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    printf("Hello C.\n");
    return 0;
}
```

We will explain argv later.

argc is an integer.
In C, "int" means integer.

"hello.c" 7 lines, 91 characters



The image shows a terminal window titled "Terminal" with a menu bar containing "File", "Edit", "View", "Terminal", and "Help". The terminal displays the following C code:

```
#include<stdio.h>
int main(int argc, char
{
    printf("Hello C.\n");
    return 0;
}
```

A green callout box with a left-pointing arrow highlights the `#include<stdio.h>` line. The text in the callout box reads:

stdio.h is a header file.
It provides information about printf.
It means standard IO for handling printing and files.

At the bottom of the terminal, the command prompt shows the result of the `cat` command: `"hello.c" 7 lines, 91 characters`.

How to Write (Simple) C Programs?

- Use the main function as the starting point.
- Select appropriate library functions (such as printf).
 - ⇒ How to know which library functions are available?
It is similar to learning vocabulary in English.
The more you know, the better.
- Include necessary header files (such as stdio.h).
 - ⇒ This can be found by using manual (man) pages.
- Return 0 if the program ends normally.

man printf(3) - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.manpagez.com/man/3/printf/ Google

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <stdio.h> ←
int
asprintf(char **ret, const char *format, ...);
int
fprintf(FILE *restrict stream, const char *restrict format, ...);
int
printf(const char *restrict format, ...);
int
snprintf(char *restrict s, size_t n, const char *restrict format, ...);
int
sprintf(char *restrict s, const char *restrict format, ...);
#include <stdarg.h>
#include <stdio.h>
int
```

args.c

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    int cnt;
    printf("There are %d arguments\n", argc);
    for (cnt = 0; cnt < argc; cnt ++)
        {
            printf("%s\n", argv[cnt]);
        }
    return 0;
}
```

```
"args.c" 12 lines, 205 characters
```

```
Terminal
File Edit View Terminal Help
[Ubuntu Linux ] gcc args.c -o args
[Ubuntu Linux ] ./args one two three four five arguments
There are 7 arguments
./args
one
two
three
four
five
arguments
[Ubuntu Linux ] ./args ThisIsALongArgument short 1 2
There are 5 arguments
./args
ThisIsALongArgument
short
1
2
[Ubuntu Linux ] █
```

The program prints the arguments one by one.

Terminal

File Edit View Terminal Help

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    int cnt;
    printf("There are %d arguments in ", argc);
    for (cnt = 0; cnt < argc; cnt++)
    {
        printf("%s\n", argv[cnt]);
    }
    return 0;
}
```

↑ argc is an integer and it reports the number of arguments in the command line.

"args.c" 12 lines, 205 characters

```
#include<stdio.h>
int main(int argc, char * argv[])
{
    int cnt;
    printf("There are %d arguments\n", argc);
    for (cnt = 0; cnt < argc; cnt++)
    {
        printf("%s\n", argv[cnt]);
    }
    return 0;
}
```

A red curved arrow originates from the `%d` format specifier in the `printf("There are %d arguments\n", argc);` line and points to the highlighted text `printf ... %d ... prints an integer.` A yellow arrow points from the highlighted text to the `cnt` variable in the `for` loop.

```
"args.c" 12 lines, 205 characters
```

```
Terminal
File Edit View Terminal Help
[Ubuntu Linux ] gc 1 2 3 4 5 6 7
[Ubuntu Linux ] ./args one two three four five arguments
There are 7 arguments
./args
one
two
three
four
five
arguments 1 2 3 4 5
[Ubuntu Linux ] ./args ThisIsALongArgument short 1 2
There are 5 arguments
./args
ThisIsALongArgument
short
1
2
[Ubuntu Linux ]
```



```
#include<stdio.h>
int main(int argc, char * argv[])
{
    int cnt;
    printf("There are %d arguments\n", argc);
    for (cnt = 0; cnt < argc; cnt ++)
    {
        printf("%s\n", argv[cnt]);
    }
    return 0;
}
```

← an integer counter

← cnt starts at zero,
increases by one each time
until it is no longer smaller than argc

↑ printf ... %s ... prints a string.

```
"args.c" 12 lines, 205 characters
```

```
#include <stdio.h> argc and argv are the conventional names. It is better to keep them.
```

```
int main(int argc, char * argv[])
```

```
{
```

```
    int cnt; ←
```

```
    printf("There are %d arguments\n", argc);
```

```
    for (cnt = 0; cnt < argc; cnt ++)
```

```
    {
```

```
        printf("%s\n", argv[cnt]);
```

```
    }
```

```
    return 0;
```

```
}
```

You can change the names of these variables.

```
"args.c" 12 lines, 205 characters
```

Congratulations.

What flag is used to specify the name of gcc's output file?

- A) -a
- B) -c
- C) -o
- D) -g

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

Starting point of a program?

The function is the starting point of a program.

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

In C programs, indexes starting from

- A) -1.
- B) 0.
- C) 1.
- D) Specified by programmers.

Correct - Click anywhere to continue

Incorrect - Click anywhere to continue

Your answer:

You did not answer this question

You must answer the question before continuing

Submit

Clear

Simple C Examples

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz