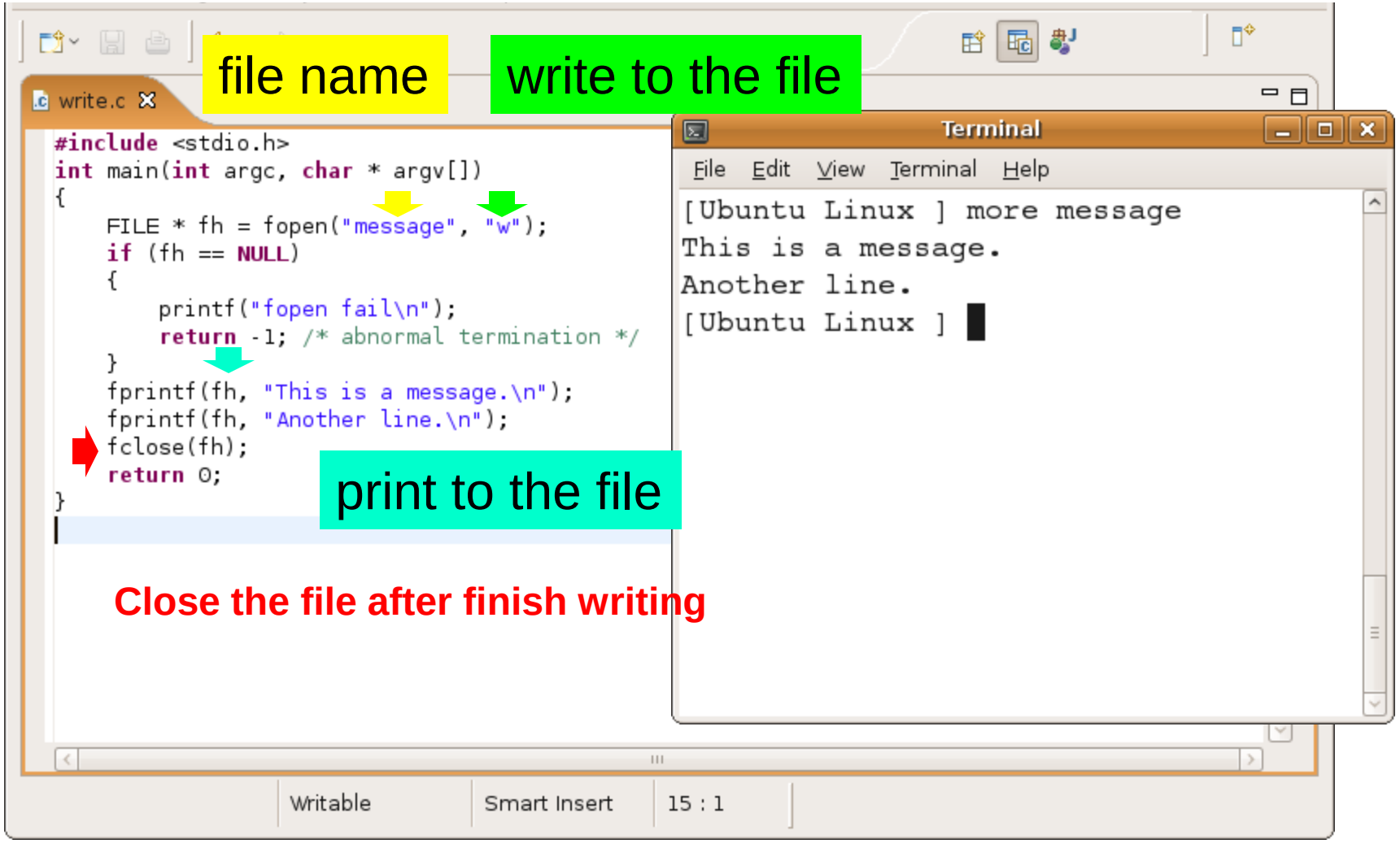
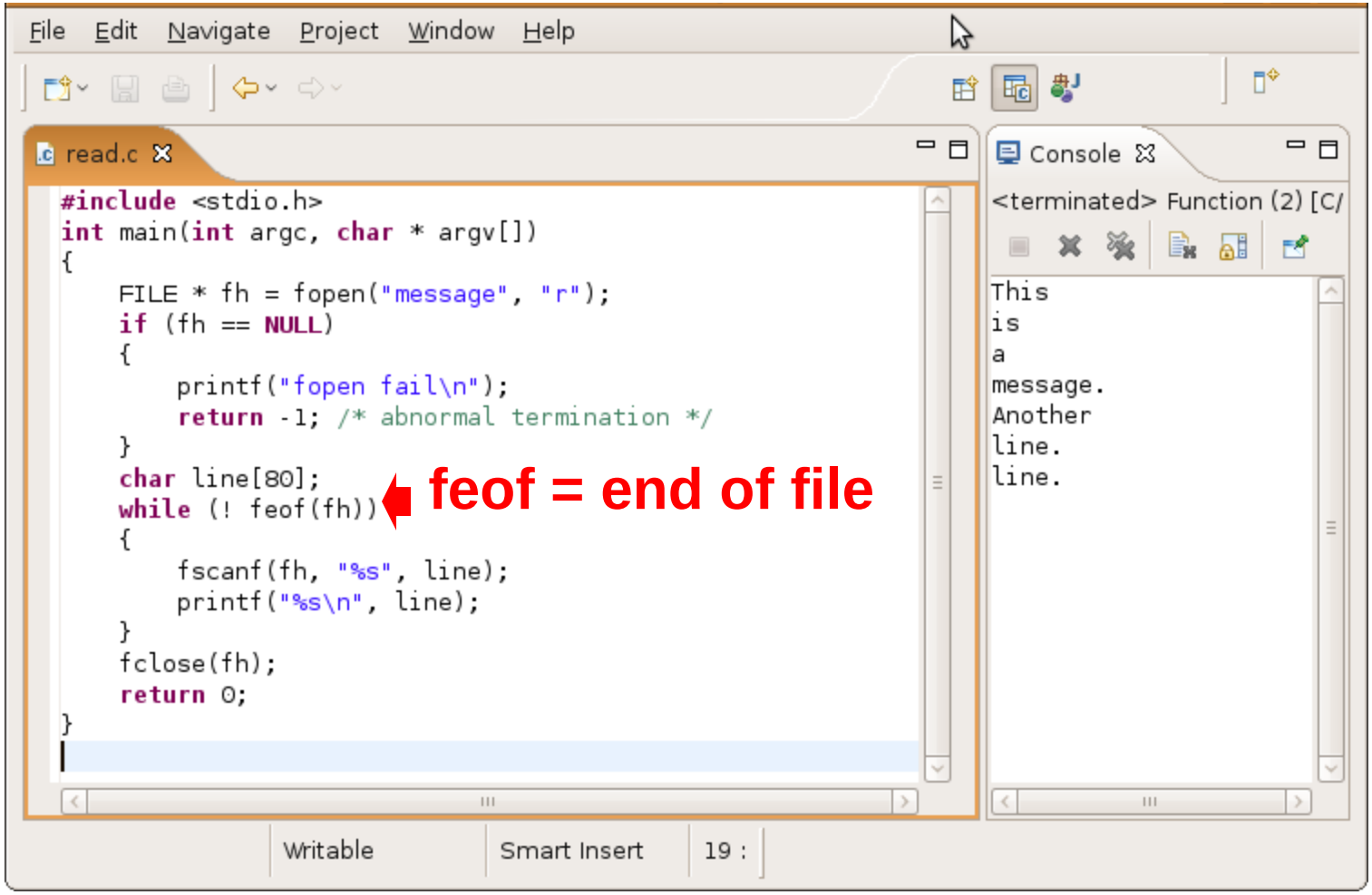
File Write and Read

Yung-Hsiang Lu

fopen, fprintf, fclose



fscanf


File Edit Navigate Project Window Help
read.c
#include <stdio.h>
int main(int argc, char * argv[])
{
FILE * fh = fopen("message", "r");
if (fh == NULL)
{
printf("fopen fail\n");
return -1; /* abnormal termination */
}
char line[80];
while (! feof(fh))
{
fscanf(fh, "%s", line);
printf("%s\n", line);
}
fclose(fh);
return 0;
}
feof = end of file
Console
<terminated> Function (2) [C/
This
is
a
message.
Another
line.
line.
Writable
Smart Insert
19 :

man fscanf(3) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.manpagez.com/man/3/fscanf/

manpage = manual page

NAME

fscanf, scanf, sscanf, vfscanf, vscanf, vsscanf -- input format conversion

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <stdio.h>

int
fscanf(FILE *restrict stream, const char *restrict format, ...);

int
scanf(const char *restrict format, ...);

int
sscanf(const char *restrict s, const char *restrict format, ...);

#include <stdarg.h>
#include <stdio.h>

int
vfscanf(FILE *restrict stream, const char *restrict format, va_list arg);
```

Done

be a pointer to *unsigned int*.

x, **X** Matches an optionally signed hexadecimal integer; the next pointer must be a pointer to *unsigned int*.

a, **A**, **e**, **E**, **f**, **F**, **g**, **G**
Matches a floating-point number in the style of **strtod(3)**. The next pointer must be a pointer to *float* (unless **l** or **L** is specified.)

s Matches a sequence of non-white-space characters; the next pointer must be a pointer to *char*, and the array must be large enough to accept all the sequence and the terminating NUL character. The input string stops at white space or at the maximum field width, whichever occurs first.

If an **l** qualifier is present, the next pointer must be a pointer to *wchar_t*, into which the input will be placed after conversion by **mbrtowc(3)**.

S The same as **ls**.

c Matches a sequence of *width* count characters (default 1); the next pointer must be a pointer to *char*, and there must be enough room for all the characters (no terminating NUL is added). The usual skip of leading white space is suppressed. To skip white space first, use an explicit space in the format.

If an **l** qualifier is present, the next pointer must be a pointer to *wchar_t*, into which the input will be placed after conversion by **mbrtowc(3)**.

C The same as **lc**.

DESCRIPTION

The **fopen**() function opens the file whose name is the string pointed to by *filename* and associates a stream with it.

The argument *mode* points to a string beginning with one of the following sequences (Additional characters may follow these sequences.):

``r'' Open text file for reading. The stream is positioned at the beginning of the file.

``r+'' Open for reading and writing. The stream is positioned at the beginning of the file.

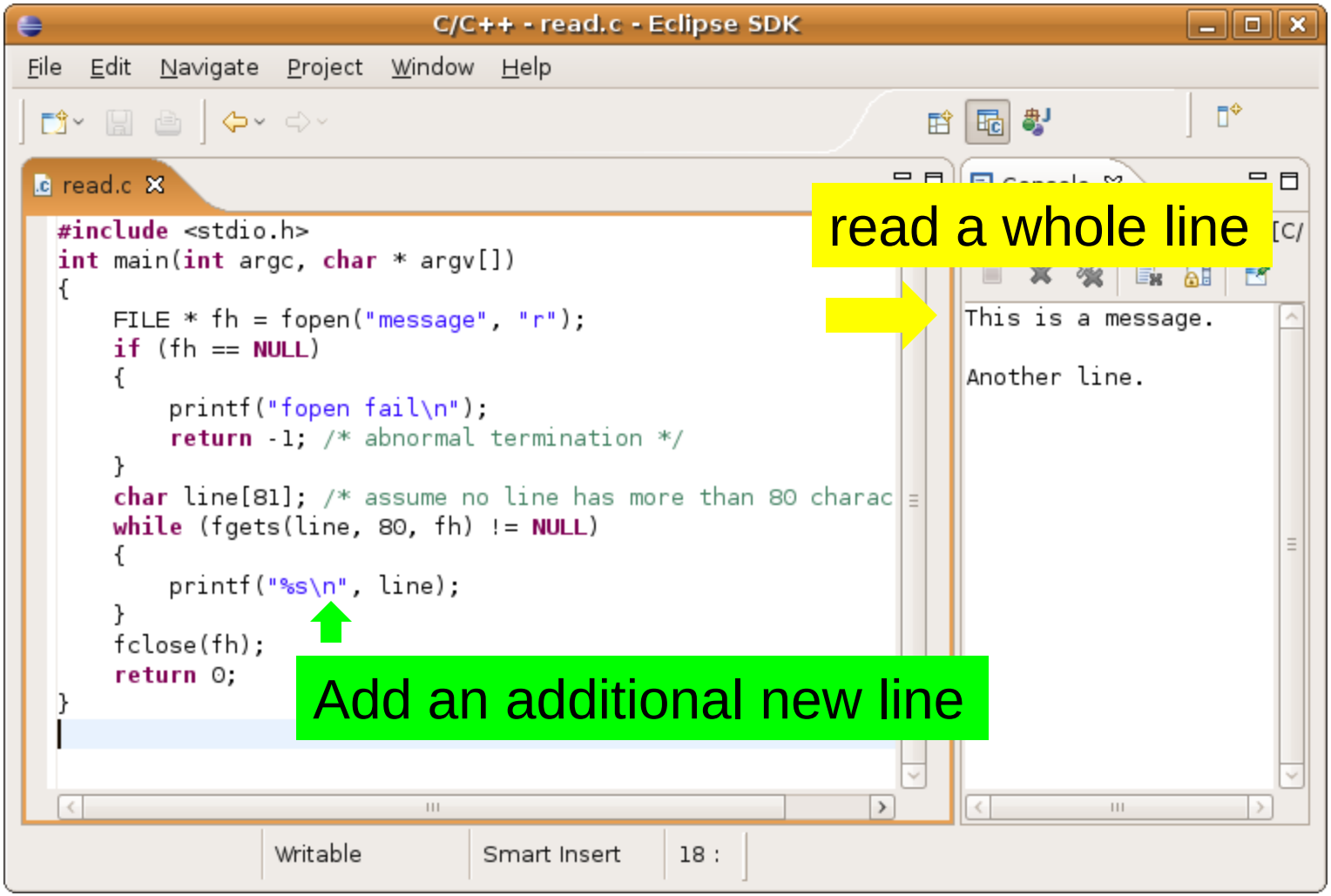
``w'' Truncate file to zero length or create text file for writing. The stream is positioned at the beginning of the file.

``w+'' Open for reading and writing. The file is created if it does not exist, otherwise it is truncated. The stream is positioned at the beginning of the file.

``a'' Open for writing. The file is created if it does not exist. The stream is positioned at the end of the file. Subsequent writes to the file will always end up at the then current end of file, irrespective of any intervening **fseek(3)** or similar.

``a+'' Open for reading and writing. The file is created if it does not exist. The stream is positioned at the end of the file. Subsequent writes to the file will always end up at the then current end of file, irrespective of any intervening **fseek(3)** or similar

fgets


C/C++ - read.c - Eclipse SDK
File Edit Navigate Project Window Help
read.c
#include <stdio.h>
int main(int argc, char * argv[])
{
FILE * fh = fopen("message", "r");
if (fh == NULL)
{
printf("fopen fail\n");
return -1; /* abnormal termination */
}
char line[81]; /* assume no line has more than 80 charac
while (fgets(line, 80, fh) != NULL)
{
printf("%s\n", line);
}
fclose(fh);
return 0;
}
read a whole line
This is a message.
Another line.
Add an additional new line
Writable
Smart Insert
18 :

man fgets(3) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.manpagez.com/man/3/fgets/

Google

NAME

fgets, **gets** -- get a line from a stream

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <stdio.h>

char *
fgets(char *restrict s, int n, FILE *restrict stream);

char *
gets(char *s);
```

DESCRIPTION

Done

man fgets(3) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

http://www.manpagez.com/man/3/fgets/

Google

DESCRIPTION

The **fgets**() function reads at most one less than the number of characters specified by *n* from the given *stream* and stores them in the string s. Reading stops when a newline character is found, at end-of-file or error. The newline, if any, is retained. If any characters are read and there is no error, a `\0' character is appended to end the string.

The **gets**() function is equivalent to **fgets**() with an infinite *n* and a *stream* of stdin, except that the newline character (if any) is not stored in the string. It is the caller's responsibility to ensure that the input line, if any, is sufficiently short to fit in the string.

RETURN VALUES

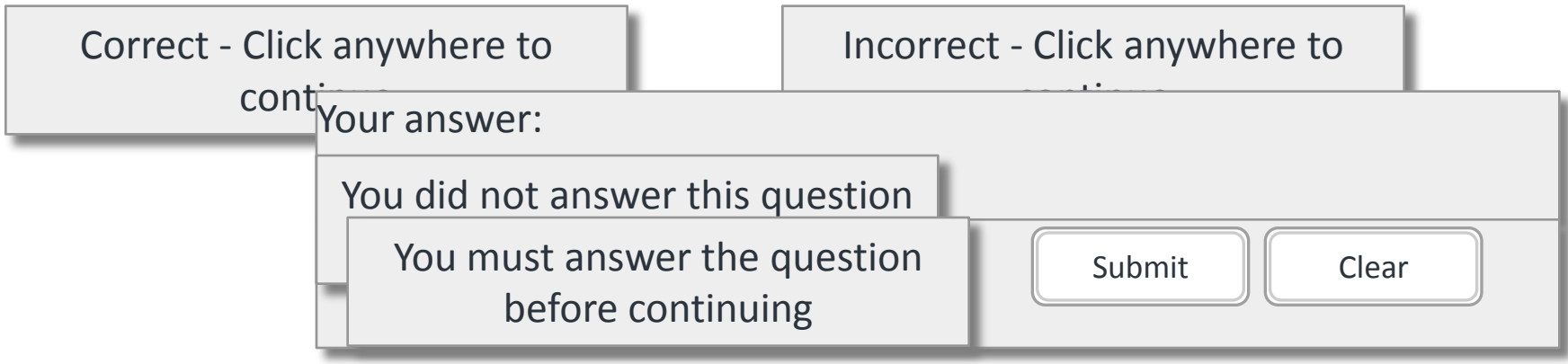
Upon successful completion, **fgets**() and **gets**() return a pointer to the string. If end-of-file occurs before any characters are read, they return NULL and the buffer contents remain unchanged. If an error occurs, they return NULL and the buffer contents are indeterminate. The **fgets**() and **gets**() functions do not distinguish between end-of-file and error; callers must use **feof(3)** and **ferror(3)** to determine which occurred.

ERRORS

Done

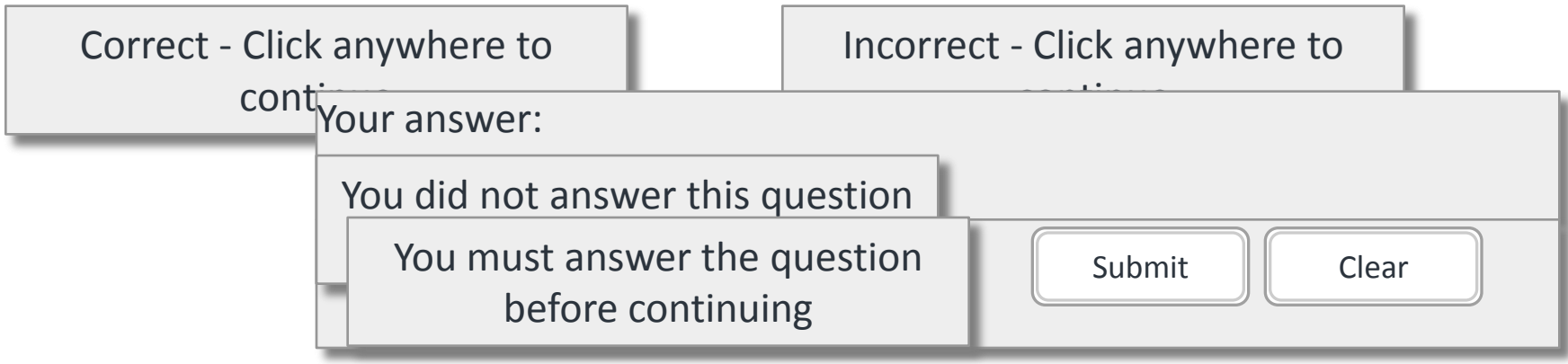
FILE * fh = fopen("filename", "r"); which is correct?

- A) If fh is NULL, fopen fails.
- B) The file is open for writes.
- C) If fh is zero, fopen succeeds.
- D) feof(fh) is false if the end of file has been reached.



Close File

Which function is used to close an opened file?



File

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

Question Feedback/Review Information Will Appear Here

Continue | Review Quiz