

TP0: Contador de palabras

Ezequiel Zbogar - 102216

Taller de Programación I

El presente trabajo hace hincapié en el correcto entendimiento y uso de las herramientas de desarrollo utilizadas en el curso. El trabajo está dividido en pasos, los cuales pueden representar estados o versiones de un programa en pleno desarrollo. En cada uno de estos se tratará de identificar el motivo de diferentes errores con la ayuda de Valgrind, GDB y el SERCOM para su posterior corrección.

Paso 0: Entorno de Trabajo

Una vez listo el ambiente de trabajo local se compila y ejecuta una aplicación simple ISO C que imprime por pantalla el mensaje "Hola Mundo" y finalice retornando 0 (cero). Luego se ejecuta dicha aplicación utilizando Valgrind.

```
(base) ezezbogar@Ezequiel:~/Desktop$ gcc main.c -o tp
(base) ezezbogar@Ezequiel:~/Desktop$ ./tp
Hola Mundo!
(base) ezezbogar@Ezequiel:~/Desktop$ valgrind ./tp
==74199== Memcheck, a memory error detector
==74199== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==74199== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==74199== Command: ./tp
==74199==
Hola Mundo!
==74199==
==74199== HEAP SUMMARY:
==74199==      in use at exit: 0 bytes in 0 blocks
==74199==    total heap usage: 1 allocs, 1 frees, 1,024 bytes allocated
==74199==
==74199== All heap blocks were freed -- no leaks are possible
==74199==
==74199== For lists of detected and suppressed errors, rerun with: -s
==74199== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Valgrind es una herramienta que tiene como objetivo ayudar al programador a identificar errores de memoria, en muchas situaciones puede suceder que se reserva una determinada cantidad de memoria con un `malloc` pero que luego nunca es liberada con un `free` y ese sector de la memoria queda inutilizable. Valgrind lo que hace es identificar problemas de ese tipo para su posterior corrección.

`sizeof()` es un operador del lenguaje C que recibe como parámetro una variable, o el nombre de un tipo de dato y este retorna su correspondiente tamaño en bytes. `sizeof(char)` devuelve el valor `1`, mientras que `sizeof(int)` devuelve el valor de `4`. Este último puede variar según la arquitectura y/o sistema donde está siendo ejecutado.

El `sizeof()` de una struct de C no necesariamente es igual a la suma del `sizeof()` de cada uno sus elementos. Esto depende del alineamiento y el padding. El compilador puede hacer que se guarden las variables en posiciones de memoria múltiplos de 4, si este es el caso las variables están alineadas y su acceso es mas rápido, pero se desperdicia memoria. A su vez, las variables se pueden guardar consecutivamente, lo cual hace mas lento su acceso pero consume menos memoria

```
#include <stdio.h>

struct Dato {
    int numero1;
    char caracter1;
    int numero2;
};

int main() {
    printf("int: %d\n", (int)sizeof(int));
    printf("char: %d\n", (int)sizeof(char));
    printf("Dato: %d\n", (int)sizeof(struct Dato));
    return 0;
}

(base) ezezbogar@Ezequiel:~/Desktop$ gcc main.c -o tp
(base) ezezbogar@Ezequiel:~/Desktop$ ./tp
int: 4
char: 1
Dato: 12
```

Si bien el tipo de dato `int` ocupa 4 bytes y el `char` ocupa 1 byte el `struct Dato` ocupa 12 bytes porque las variables están alineadas, del bloque de 4 bytes donde se encuentra el `char` solo se esta utilizando 1 byte y los otros 3 se ignoran

STDIN, STDOUT, STDERR son archivos estándar característicos de los sistemas Unix. Todos son tratados como un flujo continuo de caracteres y pueden ser manipulados como cualquier `FILE`. STDIN es la entrada estándar al programa, donde los datos pueden ser ingresados por un teclado, por ejemplo. STDOUT es la salida estándar, vía por la cual el programa devuelve información durante su ejecución. STDERR es la vía por la cual se transmite cualquier mensaje de error que emita el programa.

Para redirigir estos flujos en caso de ser necesario se pueden utilizar los caracteres `>` si lo que queremos es que la salida (`STDOUT`) vaya a algún archivo en particular, `<` si queremos que la entrada a nuestro programa (`STDIN`) sea mediante un archivo en particular o `|` si lo que buscamos es que la salida de un determinado programa sea la entrada de otro

Paso 1: SERCOM - Errores de generación y normas de programación

Todos los errores de compilación se deben a que nunca se incluyo la biblioteca donde están definidas las funciones correspondientes a `wordscouter`.

El sistema no reporto ningún warning porque se esta compilando con el flag `-werror`, el cual los trata a todos como errores.

```
Descomprimiendo el codigo 'source_unsafe.zip'...
```

```

Archive:  source_unsafe.zip
5d0c636e63aba08d756e746f4352e3bb888bc527
  inflating: source_unsafe/paso1_main.c
  inflating: source_unsafe/paso1_wordscounter.c
  inflating: source_unsafe/paso1_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el codigo...
  CC  paso1_main.o
paso1_main.c: In function 'main':
paso1_main.c:22:9: error: unknown type name 'wordscounter_t'
    22 |         wordscounter_t counter;
        |         ^~~~~~
paso1_main.c:23:9: error: implicit declaration of function 'wordscounter_create'
[-Wimplicit-function-declaration]
    23 |         wordscounter_create(&counter);
        |         ^~~~~~
paso1_main.c:24:9: error: implicit declaration of function 'wordscounter_process'
[-Wimplicit-function-declaration]
    24 |         wordscounter_process(&counter, input);
        |         ^~~~~~
paso1_main.c:25:24: error: implicit declaration of function
'wordscounter_get_words' [-Wimplicit-function-declaration]
    25 |         size_t words = wordscounter_get_words(&counter);
        |                                ^~~~~~
paso1_main.c:27:9: error: implicit declaration of function 'wordscounter_destroy'
[-Wimplicit-function-declaration]
    27 |         wordscounter_destroy(&counter);
        |         ^~~~~~
make: *** [<builtin>: paso1_main.o] Error 1

real    0m0.025s
user    0m0.020s
sys      0m0.004s
[Error] Fallo la compilacion del codigo en 'source_unsafe.zip'. Código de error 2

```

Los archivos no cumplen con las normas de codificación de CPPLint.

```

/task/student//source_unsafe/paso1_wordscounter.c:27: Missing space before ( in
while( [whitespace/parens] [5] //Falta un espacio despues del while
/task/student//source_unsafe/paso1_wordscounter.c:41: Mismatching spaces inside
() in if [whitespace/parens] [5] //De un lado de la condicion hay un espacio
pero del otro lado no
/task/student//source_unsafe/paso1_wordscounter.c:41: Should have zero or one
spaces inside ( and ) in if [whitespace/parens] [5]
/task/student//source_unsafe/paso1_wordscounter.c:47: An else should appear on
the same line as the preceding } [whitespace/newline] [4] // El else if deberia
estar en la linea de arriba a continuacion de la llave
/task/student//source_unsafe/paso1_wordscounter.c:47: If an else has a brace on
one side, it should have it on both [readability/braces] [5]
/task/student//source_unsafe/paso1_wordscounter.c:48: Missing space before ( in
if( [whitespace/parens] [5] //Falta un espacio despues del if
/task/student//source_unsafe/paso1_wordscounter.c:53: Extra space before last
semicolon. If this should be an empty statement, use {} instead.
[whitespace/semicolon] [5] //Hay un espacio entre next_state y el ;
/task/student//source_unsafe/paso1_wordscounter.h:5: Lines should be <= 80
characters long [whitespace/line_length] [2] //La linea de texto es muy larga
(Mayor a 80 caracteres)

```

```
/task/student//source_unsafe/paso1_main.c:12: Almost always, snprintf is better
than strcpy [runtime/printf] [4] //Recomienda usar snprintf en lugar de strcpy
/task/student//source_unsafe/paso1_main.c:15: An else should appear on the same
line as the preceding } [whitespace/newline] [4] //El else deberia estar en la
linea de arriba a continuacion de la llave
/task/student//source_unsafe/paso1_main.c:15: If an else has a brace on one
side, it should have it on both [readability/braces] [5]
Done processing /task/student//source_unsafe/paso1_wordscounter.c
Done processing /task/student//source_unsafe/paso1_wordscounter.h
Done processing /task/student//source_unsafe/paso1_main.c
Total errors found: 11
```

Paso 2: SERCOM - Errores de generación 2

Los cambios realizados entre las entregas son:

- Se incluye la librería `wordscounter`
- En lugar de usar `strcpy` se utiliza `memcpy`, la principal diferencia es que se le pasa por parámetro la cantidad máxima de caracteres a copiar
- Cambia la descripción de `wordscounter_t`
- Se modifica ligeramente el código conforme las normas de codificación de CPPLint.

```
$ diff paso1_main.c paso2_main.c
3a4
> #include "paso2_wordscounter.h"
12c13
<         strcpy(filepath, argv[1]);
---
>         memcpy(filepath, argv[1], strlen(argv[1]) + 1);
14,15c15
<     }
<     else {
---
>     } else {
```

```
$ diff paso1_wordscounter.c paso2_wordscounter.c
0a1
> #include "paso2_wordscounter.h"
4d4
< #include "paso1_wordscounter.h"
13,14c13
< void wordscounter_create(wordscounter_t *self)
< {
---
> void wordscounter_create(wordscounter_t *self) {
27c26
<     } while(state != STATE_FINISHED);
---
>     } while (state != STATE_FINISHED);
41c40
<     if ( c == EOF) {
---
>     if (c == EOF) {
46,48c45,46
```

```

<     }
<     else if (state == STATE_IN_WORD) {
<         if(strchr(delim_words, c) != NULL) {
---
>     } else if (state == STATE_IN_WORD) {
>         if (strchr(delim_words, c) != NULL) {
53c51
<     return next_state ;
---
>     return next_state;

```

```

$ diff paso1_wordscounter.h paso2_wordscounter.h
5c5
< // Tipo wordscounter_t: almacena la cantidad de palabras procesadas de un
archivo.
---
> // Tipo wordscounter_t: procesa cantidad de palabras dentro de un archivo.

```

Se producen errores en la compilación

```

Descomprimiendo el código 'source_unsafe.zip'...
Archive:  source_unsafe.zip
ca832a167f59b0683b0328bba0d020eda7eae60a
  inflating: source_unsafe/paso2_main.c
  inflating: source_unsafe/paso2_wordscounter.c
  inflating: source_unsafe/paso2_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el código...
CC  paso2_wordscounter.o

//No se incluye la biblioteca <stddef.h>, la cual contiene la definición de
size_t
In file included from paso2_wordscounter.c:1:
paso2_wordscounter.h:7:5: error: unknown type name 'size_t'
    7 |     size_t words;
      |     ^~~~~~
paso2_wordscounter.h:20:1: error: unknown type name 'size_t'
   20 | size_t wordscounter_get_words(wordscounter_t *self);
      | ^~~~~~
paso2_wordscounter.h:1:1: note: 'size_t' is defined in header '<stddef.h>'; did
you forget to '#include <stddef.h>'?
+++ |+#include <stddef.h>
    1 | #ifndef __WORDSCOUNTER_H__

//No se incluye la biblioteca estandar, la cual contiene la definición de FILE
paso2_wordscounter.h:25:49: error: unknown type name 'FILE'
   25 | void wordscounter_process(wordscounter_t *self, FILE *text_file);
      |                                                    ^~~~
paso2_wordscounter.h:1:1: note: 'FILE' is defined in header '<stdio.h>'; did you
forget to '#include <stdio.h>'?
+++ |+#include <stdio.h>
    1 | #ifndef __WORDSCOUNTER_H__
paso2_wordscounter.c:17:8: error: conflicting types for 'wordscounter_get_words'
   17 | size_t wordscounter_get_words(wordscounter_t *self) {
      | ^~~~~~
In file included from paso2_wordscounter.c:1:

```

```
paso2_wordscounter.h:20:8: note: previous declaration of 'wordscounter_get_words'
was here
    20 | size_t wordscounter_get_words(wordscounter_t *self);
        |           ^~~~~~

//No se incluye la biblioteca estandar, la cual contiene la definicion de malloc
paso2_wordscounter.c: In function 'wordscounter_next_state':
paso2_wordscounter.c:30:25: error: implicit declaration of function 'malloc' [-
Wimplicit-function-declaration]
    30 |     char* delim_words = malloc(7 * sizeof(char));
        |                               ^~~~~~
paso2_wordscounter.c:30:25: error: incompatible implicit declaration of built-in
function 'malloc' [-Werror]
paso2_wordscounter.c:5:1: note: include '<stdlib.h>' or provide a declaration of
'malloc'
    4 | #include <stdbool.h>
+++ |+#include <stdlib.h>
    5 |
cc1: all warnings being treated as errors
make: *** [<built-in>: paso2_wordscounter.o] Error 1

real    0m0.032s
user    0m0.022s
sys      0m0.008s
[Error] Fallo la compilacion del codigo en 'source_unsafe.zip'.Codigo de error 2
```

Todos estos errores son de compilación, en ningún momento se llega a crear el archivo objeto (.o). Por lo tanto, nunca se llega a la fase de linkeo

Todos los archivos cumplen con las normas de codificación de CPPLint

```
Done processing /task/student//source_unsafe/paso2_wordscounter.c
Done processing /task/student//source_unsafe/paso2_main.c
Done processing /task/student//source_unsafe/paso2_wordscounter.h
```

Paso 3: SERCOM - Errores de generación 3

Los cambios realizados entre las entregas son:

- Incluir las bibliotecas `string.h` y `stdio.h`

```
$ diff paso2_wordscounter.h paso3_wordscounter.h
3a4,5
> #include <string.h>
> #include <stdio.h>
```

Se producen errores en la compilación

```
Descomprimiendo el codigo 'source_unsafe.zip'...
Archive:  source_unsafe.zip
23da7ec7cf028683c0da436e3f8cb1b607245195
  inflating: source_unsafe/paso3_main.c
  inflating: source_unsafe/paso3_wordscounter.c
  inflating: source_unsafe/paso3_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el codigo...
```

```

CC  paso3_wordscounter.o
CC  paso3_main.o
LD  tp
/usr/bin/ld: paso3_main.o: in function `main':
/task/student/source_unsafe/paso3_main.c:27: undefined reference to
`wordscounter_destroy'
collect2: error: ld returned 1 exit status
make: *** [/task/student/MakefileTP0:142: tp] Error 1

real    0m0.374s
user    0m0.105s
sys     0m0.073s
[Error] Fallo la compilacion del codigo en 'source_unsafe.zip'. Codigo de error 2

```

Si bien `main` y `wordscounter` se compilan correctamente (se generan los `.o`), hay un error debido a que `wordscounter_destroy` no esta definida y por lo tanto no puede ser utilizada. Vale aclarar que la función si fue declarada. Este es un error del linker

Todos los archivos cumplen con las normas de codificación de CPPLint.

```

Done processing /task/student//source_unsafe/paso3_main.c
Done processing /task/student//source_unsafe/paso3_wordscounter.c
Done processing /task/student//source_unsafe/paso3_wordscounter.h

```

Paso 4: SERCOM - Memory Leaks y Buffer Overflows

Los cambios realizados entre las entregas son:

- Se define la función `wordscounter_destroy` vacía

```

$ diff paso3_wordscounter.c paso4_wordscounter.c
1c1
< #include "paso3_wordscounter.h"
---
> #include "paso4_wordscounter.h"
15a16,19
> }
>
> void wordscounter_destroy(wordscounter_t *self) {
>     //do nothing

```

El código compila sin problemas

```

Descomprimiendo el codigo 'source_unsafe.zip'...
Archive:  source_unsafe.zip
c34bcb3f0b7de0499591324acbe1c817ded46893
  inflating: source_unsafe/paso4_main.c
  inflating: source_unsafe/paso4_wordscounter.c
  inflating: source_unsafe/paso4_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el codigo...
CC  paso4_wordscounter.o
CC  paso4_main.o
LD  tp

real    0m0.395s

```

```
user    0m0.122s
sys     0m0.074s
```

Todos los archivos cumplen con las normas de codificación de CPPLint.

```
Done processing /task/student//source_unsafe/paso4_main.c
Done processing /task/student//source_unsafe/paso4_wordscounter.h
Done processing /task/student//source_unsafe/paso4_wordscounter.c
```

Se corren las pruebas

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (sin valgrind)...
[=>] Command line: ./tp invalid_file
```

```
real    0m0.009s
user    0m0.004s
sys     0m0.001s
```

```
[=>] Executando el caso '/task/student//cases/entrada_estandar' (sin valgrind)...
[=>] Command line: ./tp
```

```
real    0m0.004s
user    0m0.001s
sys     0m0.002s
```

```
[=>] Executando el caso '/task/student//cases/language_c' (sin valgrind)...
[=>] Command line: ./tp input_c.txt
```

```
real    0m0.003s
user    0m0.001s
sys     0m0.002s
```

```
[=>] Executando el caso '/task/student//cases/nombre_largo' (sin valgrind)...
[=>] Command line: ./tp input_extremely_long_filename.txt
/task/student//test_single_program.sh: line 61: 31 Aborted
timeout --kill-after $kill_timeout $timeout $cmd > __stdout__ 2> __stderr__ <
__stdin__
```

```
real    0m0.270s
user    0m0.001s
sys     0m0.002s
```

```
[=>] Executando el caso '/task/student//cases/tda' (sin valgrind)...
[=>] Command line: ./tp input_tda.txt
```

```
real    0m0.003s
user    0m0.003s
sys     0m0.000s
```

```
[=>] Executando el caso '/task/student//cases/una_palabra' (sin valgrind)...
[=>] Command line: ./tp input_single_word.txt
```

```
real    0m0.003s
user    0m0.001s
sys     0m0.002s
```


Las pruebas 'TDA', 'C Language' y 'STDIN' corrieron exitosamente (sin Valgrind) aunque no las pruebas 'Invalid File', 'Long Filename' y 'Single Word' (El valor de retorno no coincide con el esperado)

```
[=>] Comparando archivo_invalido/__return_code__...
1c1
< 255
---
> 1
```

```
[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando language_c/__return_code__...
[=>] Comparando language_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
1c1
< 134
---
> 0
```

```
[=>] Comparando nombre_largo/__stdout__...
0a1
> 2
```

```
[=>] Comparando tda/__return_code__...
[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
[=>] Comparando una_palabra/__stdout__...
1c1
< 0
---
> 1
```

[Error] Se encontraron diferencias entre las salidas obtenidas y las esperadas.

Sin embargo, Valgrind encuentra errores en las pruebas 'TDA', 'C Language' y 'STDIN' que antes habían corrido exitosamente

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
      --tool=memcheck
      --leak-check=full --leak-resolution=med --show-reachable=yes
      --trace-children=yes
      --track-fds=yes
      --track-origins=no
      --time-stamp=yes --num-callers=20
      --error-exitcode=42
      --log-file=__valgrind__
      --suppressions=/task/student/suppressions.txt ./tp invalid_file
```

```
real    0m1.070s
```

```
user    0m0.857s
sys     0m0.094s
```

Valgrind no encontro errores.

[=>] Ejecutando el caso '/task/student//cases/entrada_estandar' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp
```

```
real    0m1.014s
user    0m0.870s
sys     0m0.078s
```

```
==00:00:00:00.000 23== Memcheck, a memory error detector
==00:00:00:00.000 23== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward
et al.
==00:00:00:00.000 23== Using Valgrind-3.15.0 and LibVEX; rerun with -h for
copyright info
==00:00:00:00.000 23== Command: ./tp
==00:00:00:00.000 23== Parent PID: 22
==00:00:00:00.000 23==
==00:00:00:00.977 23==
==00:00:00:00.977 23== FILE DESCRIPTORS: 4 open at exit.
==00:00:00:00.977 23== Open file descriptor 3:
/task/student/cases/entrada_estandar/__valgrind__
==00:00:00:00.977 23==      <inherited from parent>
==00:00:00:00.977 23==
==00:00:00:00.977 23== Open file descriptor 2:
/task/student/cases/entrada_estandar/__stderr__
==00:00:00:00.977 23==      <inherited from parent>
==00:00:00:00.977 23==
==00:00:00:00.977 23== Open file descriptor 1:
/task/student/cases/entrada_estandar/__stdout__
==00:00:00:00.977 23==      <inherited from parent>
==00:00:00:00.977 23==
==00:00:00:00.977 23== Open file descriptor 0:
/task/student/cases/entrada_estandar/__stdin__
==00:00:00:00.977 23==      <inherited from parent>
==00:00:00:00.977 23==
==00:00:00:00.977 23==
==00:00:00:00.977 23== HEAP SUMMARY:
==00:00:00:00.977 23==      in use at exit: 3,402 bytes in 486 blocks
==00:00:00:00.977 23==      total heap usage: 488 allocs, 2 frees, 11,594 bytes
allocated
==00:00:00:00.977 23==
==00:00:00:00.978 23== 3,402 bytes in 486 blocks are definitely lost in loss
record 1 of 1
==00:00:00:00.978 23==      at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.978 23==      by 0x109301: wordscounter_next_state
(paso4_wordscounter.c:35)
```

```

==00:00:00:00.978 23== by 0x1093B5: wordscounter_process
(paso4_wordscounter.c:30)
==00:00:00:00.978 23== by 0x109197: main (paso4_main.c:24)
==00:00:00:00.978 23==
==00:00:00:00.978 23== LEAK SUMMARY:
==00:00:00:00.978 23== definitely lost: 3,402 bytes in 486 blocks
==00:00:00:00.978 23== indirectly lost: 0 bytes in 0 blocks
==00:00:00:00.978 23== possibly lost: 0 bytes in 0 blocks
==00:00:00:00.978 23== still reachable: 0 bytes in 0 blocks
==00:00:00:00.978 23== suppressed: 0 bytes in 0 blocks
==00:00:00:00.978 23==
==00:00:00:00.978 23== For lists of detected and suppressed errors, rerun with: -
s
==00:00:00:00.978 23== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0
from 0)

[=>] Ejecutando el caso '/task/student//cases/language_c' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_c.txt

real    0m0.971s
user    0m0.857s
sys     0m0.081s

==00:00:00:00.000 35== Memcheck, a memory error detector
==00:00:00:00.000 35== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward
et al.
==00:00:00:00.000 35== Using Valgrind-3.15.0 and LibVEX; rerun with -h for
copyright info
==00:00:00:00.000 35== Command: ./tp input_c.txt
==00:00:00:00.000 35== Parent PID: 34
==00:00:00:00.000 35==
==00:00:00:00.940 35==
==00:00:00:00.940 35== FILE DESCRIPTORS: 5 open at exit.
==00:00:00:00.940 35== Open file descriptor 4: input_c.txt
==00:00:00:00.940 35== at 0x495FEAB: open (open64.c:48)
==00:00:00:00.940 35== by 0x48E2195: _IO_file_open (fileops.c:189)
==00:00:00:00.940 35== by 0x48E2459: _IO_file_fopen@@GLIBC_2.2.5
(fileops.c:281)
==00:00:00:00.940 35== by 0x48D4B0D: __fopen_internal (iofopen.c:75)
==00:00:00:00.940 35== by 0x48D4B0D: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:00.940 35== by 0x109177: main (paso4_main.c:14)
==00:00:00:00.940 35==
==00:00:00:00.940 35== Open file descriptor 3:
/task/student/cases/language_c/__valgrind__
==00:00:00:00.940 35== <inherited from parent>
==00:00:00:00.940 35==
==00:00:00:00.940 35== Open file descriptor 2:
/task/student/cases/language_c/__stderr__
==00:00:00:00.940 35== <inherited from parent>

```

```

==00:00:00:00.940 35==
==00:00:00:00.940 35== Open file descriptor 1:
/task/student/cases/language_c/__stdout__
==00:00:00:00.940 35==      <inherited from parent>
==00:00:00:00.940 35==
==00:00:00:00.940 35== Open file descriptor 0:
/task/student/cases/language_c/__stdin__
==00:00:00:00.940 35==      <inherited from parent>
==00:00:00:00.940 35==
==00:00:00:00.940 35== HEAP SUMMARY:
==00:00:00:00.940 35==      in use at exit: 3,874 bytes in 487 blocks
==00:00:00:00.940 35==      total heap usage: 489 allocs, 2 frees, 12,066 bytes
allocated
==00:00:00:00.940 35==
==00:00:00:00.941 35== 472 bytes in 1 blocks are still reachable in loss record 1
of 2
==00:00:00:00.941 35==      at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.941 35==      by 0x48D4AAD: __fopen_internal (iofopen.c:65)
==00:00:00:00.941 35==      by 0x48D4AAD: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:00.941 35==      by 0x109177: main (paso4_main.c:14)
==00:00:00:00.941 35==
==00:00:00:00.941 35== 3,402 bytes in 486 blocks are definitely lost in loss
record 2 of 2
==00:00:00:00.941 35==      at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.941 35==      by 0x109301: wordscounter_next_state
(paso4_wordscounter.c:35)
==00:00:00:00.941 35==      by 0x1093B5: wordscounter_process
(paso4_wordscounter.c:30)
==00:00:00:00.941 35==      by 0x109197: main (paso4_main.c:24)
==00:00:00:00.941 35==
==00:00:00:00.941 35== LEAK SUMMARY:
==00:00:00:00.941 35==      definitely lost: 3,402 bytes in 486 blocks
==00:00:00:00.941 35==      indirectly lost: 0 bytes in 0 blocks
==00:00:00:00.941 35==      possibly lost: 0 bytes in 0 blocks
==00:00:00:00.941 35==      still reachable: 472 bytes in 1 blocks
==00:00:00:00.941 35==      suppressed: 0 bytes in 0 blocks
==00:00:00:00.941 35==
==00:00:00:00.941 35== For lists of detected and suppressed errors, rerun with: -
s
==00:00:00:00.941 35== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0
from 0)

[=>] Ejecutando el caso '/task/student//cases/nombre_largo' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
      --tool=memcheck
      --leak-check=full --leak-resolution=med --show-reachable=yes
      --trace-children=yes
      --track-fds=yes
      --track-origins=no
      --time-stamp=yes --num-callers=20
      --error-exitcode=42
      --log-file=__valgrind__
      --suppressions=/task/student/suppressions.txt ./tp
input_extremely_long_filename.txt

```

```
real    0m0.898s
user    0m0.759s
sys     0m0.098s
```

```
**00:00:00:00.851 47** *** memcpy_chk: buffer overflow detected ***: program
terminated
==00:00:00:00.000 47== Memcheck, a memory error detector
==00:00:00:00.000 47== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward
et al.
==00:00:00:00.000 47== Using Valgrind-3.15.0 and LibVEX; rerun with -h for
copyright info
==00:00:00:00.000 47== Command: ./tp input_extremely_long_filename.txt
==00:00:00:00.000 47== Parent PID: 46
==00:00:00:00.000 47==
**00:00:00:00.851 47** *** memcpy_chk: buffer overflow detected ***: program
terminated
==00:00:00:00.851 47==      at 0x483E9CC: ??? (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.851 47==      by 0x4843C0A: __memcpy_chk (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.851 47==      by 0x109168: memcpy (string_fortified.h:34)
==00:00:00:00.851 47==      by 0x109168: main (paso4_main.c:13)
==00:00:00:00.867 47==
==00:00:00:00.867 47== FILE DESCRIPTORS: 4 open at exit.
==00:00:00:00.867 47== Open file descriptor 3:
/task/student/cases/nombre_largo/__valgrind__
==00:00:00:00.867 47==      <inherited from parent>
==00:00:00:00.867 47==
==00:00:00:00.867 47== Open file descriptor 2:
/task/student/cases/nombre_largo/__stderr__
==00:00:00:00.867 47==      <inherited from parent>
==00:00:00:00.867 47==
==00:00:00:00.867 47== Open file descriptor 1:
/task/student/cases/nombre_largo/__stdout__
==00:00:00:00.867 47==      <inherited from parent>
==00:00:00:00.867 47==
==00:00:00:00.867 47== Open file descriptor 0:
/task/student/cases/nombre_largo/__stdin__
==00:00:00:00.867 47==      <inherited from parent>
==00:00:00:00.867 47==
==00:00:00:00.867 47== HEAP SUMMARY:
==00:00:00:00.867 47==      in use at exit: 0 bytes in 0 blocks
==00:00:00:00.867 47==      total heap usage: 0 allocs, 0 frees, 0 bytes allocated
==00:00:00:00.867 47==
==00:00:00:00.867 47== All heap blocks were freed -- no leaks are possible
==00:00:00:00.867 47==
==00:00:00:00.867 47== For lists of detected and suppressed errors, rerun with: -
s
==00:00:00:00.867 47== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)

[=>] Executando el caso '/task/student//cases/tda' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
      --tool=memcheck
      --leak-check=full --leak-resolution=med --show-reachable=yes
      --trace-children=yes
      --track-fds=yes
```

```
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_tda.txt
```

```
real    0m0.914s
user    0m0.778s
sys     0m0.091s
```

```
==00:00:00:00.000 59== Memcheck, a memory error detector
==00:00:00:00.000 59== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward
et al.
==00:00:00:00.000 59== Using Valgrind-3.15.0 and LibVEX; rerun with -h for
copyright info
==00:00:00:00.000 59== Command: ./tp input_tda.txt
==00:00:00:00.000 59== Parent PID: 58
==00:00:00:00.000 59==
==00:00:00:00.881 59==
==00:00:00:00.881 59== FILE DESCRIPTORS: 5 open at exit.
==00:00:00:00.881 59== Open file descriptor 4: input_tda.txt
==00:00:00:00.882 59==   at 0x495FEAB: open (open64.c:48)
==00:00:00:00.882 59==   by 0x48E2195: _IO_file_open (fileops.c:189)
==00:00:00:00.882 59==   by 0x48E2459: _IO_file_fopen@@GLIBC_2.2.5
(fileops.c:281)
==00:00:00:00.882 59==   by 0x48D4B0D: __fopen_internal (iofopen.c:75)
==00:00:00:00.882 59==   by 0x48D4B0D: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:00.882 59==   by 0x109177: main (paso4_main.c:14)
==00:00:00:00.882 59==
==00:00:00:00.882 59== Open file descriptor 3:
/task/student/cases/tda/__valgrind__
==00:00:00:00.882 59==   <inherited from parent>
==00:00:00:00.882 59==
==00:00:00:00.882 59== Open file descriptor 2: /task/student/cases/tda/__stderr__
==00:00:00:00.882 59==   <inherited from parent>
==00:00:00:00.882 59==
==00:00:00:00.882 59== Open file descriptor 1: /task/student/cases/tda/__stdout__
==00:00:00:00.882 59==   <inherited from parent>
==00:00:00:00.882 59==
==00:00:00:00.882 59== Open file descriptor 0: /task/student/cases/tda/__stdin__
==00:00:00:00.882 59==   <inherited from parent>
==00:00:00:00.882 59==
==00:00:00:00.882 59==
==00:00:00:00.882 59== HEAP SUMMARY:
==00:00:00:00.882 59==   in use at exit: 1,977 bytes in 216 blocks
==00:00:00:00.882 59==   total heap usage: 218 allocs, 2 frees, 10,169 bytes
allocated
==00:00:00:00.882 59==
==00:00:00:00.882 59== 472 bytes in 1 blocks are still reachable in loss record 1
of 2
==00:00:00:00.882 59==   at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.882 59==   by 0x48D4AAD: __fopen_internal (iofopen.c:65)
==00:00:00:00.882 59==   by 0x48D4AAD: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:00.882 59==   by 0x109177: main (paso4_main.c:14)
==00:00:00:00.882 59==
==00:00:00:00.882 59== 1,505 bytes in 215 blocks are definitely lost in loss
record 2 of 2
```

```

==00:00:00:00.882 59==      at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:00.882 59==      by 0x109301: wordscounter_next_state
(paso4_wordscounter.c:35)
==00:00:00:00.882 59==      by 0x1093B5: wordscounter_process
(paso4_wordscounter.c:30)
==00:00:00:00.882 59==      by 0x109197: main (paso4_main.c:24)
==00:00:00:00.882 59==
==00:00:00:00.882 59== LEAK SUMMARY:
==00:00:00:00.883 59==      definitely lost: 1,505 bytes in 215 blocks
==00:00:00:00.883 59==      indirectly lost: 0 bytes in 0 blocks
==00:00:00:00.883 59==      possibly lost: 0 bytes in 0 blocks
==00:00:00:00.883 59==      still reachable: 472 bytes in 1 blocks
==00:00:00:00.883 59==      suppressed: 0 bytes in 0 blocks
==00:00:00:00.883 59==
==00:00:00:00.883 59== For lists of detected and suppressed errors, rerun with: -
s
==00:00:00:00.883 59== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0
from 0)

```

[=>] Ejecutando el caso '/task/student//cases/una_palabra' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```

--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp

```

input_single_word.txt

```

real    0m1.067s
user    0m0.929s
sys     0m0.083s

```

```

==00:00:00:00.000 71== Memcheck, a memory error detector
==00:00:00:00.000 71== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward
et al.
==00:00:00:00.000 71== Using Valgrind-3.15.0 and LibVEX; rerun with -h for
copyright info
==00:00:00:00.000 71== Command: ./tp input_single_word.txt
==00:00:00:00.000 71== Parent PID: 70
==00:00:00:00.000 71==
==00:00:00:01.034 71==
==00:00:00:01.034 71== FILE DESCRIPTORS: 5 open at exit.
==00:00:00:01.034 71== Open file descriptor 4: input_single_word.txt
==00:00:00:01.034 71==      at 0x495FEAB: open (open64.c:48)
==00:00:00:01.034 71==      by 0x48E2195: _IO_file_open (fileops.c:189)
==00:00:00:01.034 71==      by 0x48E2459: _IO_file_fopen@@GLIBC_2.2.5
(fileops.c:281)
==00:00:00:01.034 71==      by 0x48D4B0D: __fopen_internal (iofopen.c:75)
==00:00:00:01.034 71==      by 0x48D4B0D: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:01.034 71==      by 0x109177: main (paso4_main.c:14)
==00:00:00:01.034 71==
==00:00:00:01.034 71== Open file descriptor 3:
/task/student/cases/una_palabra/__valgrind__

```

```

==00:00:00:01.034 71== <inherited from parent>
==00:00:00:01.034 71==
==00:00:00:01.034 71== Open file descriptor 2:
/task/student/cases/una_palabra/__stderr__
==00:00:00:01.034 71== <inherited from parent>
==00:00:00:01.034 71==
==00:00:00:01.034 71== Open file descriptor 1:
/task/student/cases/una_palabra/__stdout__
==00:00:00:01.034 71== <inherited from parent>
==00:00:00:01.034 71==
==00:00:00:01.034 71== Open file descriptor 0:
/task/student/cases/una_palabra/__stdin__
==00:00:00:01.034 71== <inherited from parent>
==00:00:00:01.034 71==
==00:00:00:01.034 71== HEAP SUMMARY:
==00:00:00:01.034 71==     in use at exit: 507 bytes in 6 blocks
==00:00:00:01.034 71==     total heap usage: 8 allocs, 2 frees, 8,699 bytes
allocated
==00:00:00:01.034 71==
==00:00:00:01.035 71== 35 bytes in 5 blocks are definitely lost in loss record 1
of 2
==00:00:00:01.035 71==     at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:01.035 71==     by 0x109301: wordscounter_next_state
(paso4_wordscounter.c:35)
==00:00:00:01.035 71==     by 0x1093B5: wordscounter_process
(paso4_wordscounter.c:30)
==00:00:00:01.035 71==     by 0x109197: main (paso4_main.c:24)
==00:00:00:01.035 71==
==00:00:00:01.035 71== 472 bytes in 1 blocks are still reachable in loss record 2
of 2
==00:00:00:01.035 71==     at 0x483B7F3: malloc (in /usr/lib/x86_64-linux-
gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==00:00:00:01.035 71==     by 0x48D4AAD: __fopen_internal (iofopen.c:65)
==00:00:00:01.035 71==     by 0x48D4AAD: fopen@@GLIBC_2.2.5 (iofopen.c:86)
==00:00:00:01.035 71==     by 0x109177: main (paso4_main.c:14)
==00:00:00:01.035 71==
==00:00:00:01.035 71== LEAK SUMMARY:
==00:00:00:01.035 71==     definitely lost: 35 bytes in 5 blocks
==00:00:00:01.035 71==     indirectly lost: 0 bytes in 0 blocks
==00:00:00:01.035 71==     possibly lost: 0 bytes in 0 blocks
==00:00:00:01.035 71==     still reachable: 472 bytes in 1 blocks
==00:00:00:01.035 71==     suppressed: 0 bytes in 0 blocks
==00:00:00:01.035 71==
==00:00:00:01.035 71== For lists of detected and suppressed errors, rerun with: -
s
==00:00:00:01.035 71== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0
from 0)

```

```

[=>] Comparando archivo_invalido/__return_code__...
1c1
< 255
---
> 1

```



```

[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
1c1
< 42 (from valgrind)
---
> 0

[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando language_c/__return_code__...
1c1
< 42 (from valgrind)
---
> 0

[=>] Comparando language_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
1c1
< 42 (from valgrind)
---
> 0

[=>] Comparando nombre_largo/__stdout__...
0a1
> 2

[=>] Comparando tda/__return_code__...
1c1
< 42 (from valgrind)
---
> 0

[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
1c1
< 42 (from valgrind)
---
> 0

[=>] Comparando una_palabra/__stdout__...
1c1
< 0
---
> 1

[Error] Se encontraron diferencias entre las salidas obtenidas y las esperadas.

```

En la prueba "Long Filename" lo que se produce es un "buffer overflow" ya que se esta tratando de cargar en un array de tamaño igual a 30 una cadena de caracteres de tamaño mayor, entonces cuando se va a escribir el primer caracter fuera del bloque de memoria que comprende al array se produce un error ya que quiere acceder a una zona de memoria restringida

En la prueba "TDA" lo que sucede es que se reserva memoria con un `malloc` la cual nunca es liberada

Además, en ninguno de los 2 casos se cierra el archivo leído con `fclose`

Por otro lado, un "segmentation fault" se produce cuando un programa intenta acceder a un sector de memoria restringido, es decir, un sector al que no tiene acceso. También puede suceder, por ejemplo, si se trata de escribir una zona de solo lectura.

Paso 5: SERCOM - Código de retorno y salida estándar

Los cambios realizados entre las entregas son:

- En la implementación de `wordscounter_next_state` antes se reservaba memoria con `malloc` para `delim_words`, la cual nunca se liberaba con un `free`, razón por la cual las pruebas fallaban con Valgrind. Ahora `delim_words` se define como un `const char *`, es decir, el array de caracteres delimitadores se almacena en el code segment al inicio del programa, por lo tanto nunca se reserva memoria en el transcurso de la ejecución
- Cambia ligeramente la forma en que se lee el archivo de texto para evitar el "buffer overflow"
- Se cierra el archivo con `fclose`

```
$ diff paso4_main.c paso5_main.c
4c4
< #include "paso4_wordscounter.h"
---
> #include "paso5_wordscounter.h"
12,14c12
<     char filepath[30];
<     memcpy(filepath, argv[1], strlen(argv[1]) + 1);
<     input = fopen(filepath, "r");
---
>     input = fopen(argv[1], "r");
27a26,27
>     if (input != stdin)
>         fclose(input);
```

```
$ diff paso4_wordscounter.c paso5_wordscounter.c
1c1
< #include "paso4_wordscounter.h"
---
> #include "paso5_wordscounter.h"
35,42c35
<     char* delim_words = malloc(7 * sizeof(char));
<     delim_words[0] = ' ';
<     delim_words[1] = ',';
<     delim_words[2] = '.';
<     delim_words[3] = ';';
<     delim_words[4] = ':';
<     delim_words[5] = '\n';
<     delim_words[6] = '\0';
---
>     const char* delim_words = " ,.;:\n";
```

El código compila sin problemas

```
Descomprimiendo el codigo 'source_unsafe.zip'...
Archive:  source_unsafe.zip
4a3c15d1a87ea43e2e87e04aeaaeb21e0dd26684
  inflating: source_unsafe/paso5_main.c
  inflating: source_unsafe/paso5_wordscounter.c
  inflating: source_unsafe/paso5_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el codigo...
  CC  paso5_main.o
  CC  paso5_wordscounter.o
  LD  tp

real    0m0.442s
user    0m0.130s
sys     0m0.081s
```

Todos los archivos cumplen con las normas de codificacion de CPPLint.

```
Done processing /task/student//source_unsafe/paso5_wordscounter.h
Done processing /task/student//source_unsafe/paso5_main.c
Done processing /task/student//source_unsafe/paso5_wordscounter.c
```

Se corren las pruebas

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (sin valgrind)...
[=>] Command line: ./tp invalid_file

real    0m0.005s
user    0m0.001s
sys     0m0.002s

[=>] Executando el caso '/task/student//cases/entrada_estandar' (sin valgrind)...
[=>] Command line: ./tp

real    0m0.002s
user    0m0.002s
sys     0m0.000s

[=>] Executando el caso '/task/student//cases/language_c' (sin valgrind)...
[=>] Command line: ./tp input_c.txt

real    0m0.002s
user    0m0.001s
sys     0m0.001s

[=>] Executando el caso '/task/student//cases/nombre_largo' (sin valgrind)...
[=>] Command line: ./tp input_extremely_long_filename.txt

real    0m0.002s
user    0m0.002s
sys     0m0.000s

[=>] Executando el caso '/task/student//cases/tda' (sin valgrind)...
[=>] Command line: ./tp input_tda.txt

real    0m0.003s
```

```
user    0m0.002s
sys     0m0.001s
```

```
[=>] Executando el caso '/task/student//cases/una_palabra' (sin valgrind)...
[=>] Command line: ./tp input_single_word.txt
```

```
real    0m0.002s
user    0m0.002s
sys     0m0.000s
```

```
[=>] Comparando archivo_invalido/__return_code__...
1c1
< 255
---
> 1
```

```
[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando language_c/__return_code__...
[=>] Comparando language_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
[=>] Comparando nombre_largo/__stdout__...
[=>] Comparando tda/__return_code__...
[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
[=>] Comparando una_palabra/__stdout__...
```

```
1c1
< 0
---
> 1
```

[Error] Se encontraron diferencias entre las salidas obtenidas y las esperadas.

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
```

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp invalid_file
```

```
real    0m1.055s
user    0m0.810s
sys     0m0.148s
```

Valgrind no encontro errores.

```
[=>] Executando el caso '/task/student//cases/entrada_estandar' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
--tool=memcheck
```

```
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp
```

```
real    0m1.078s
user    0m0.918s
sys     0m0.091s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/language_c' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_c.txt
```

```
real    0m1.082s
user    0m0.952s
sys     0m0.085s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/nombre_largo' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp
```

input_extremely_long_filename.txt

```
real    0m0.986s
user    0m0.850s
sys     0m0.085s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/tda' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
```

```

--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_tda.txt

real    0m1.075s
user    0m0.931s
sys     0m0.090s

Valgrind no encontro errores.
[=>] Ejecutando el caso '/task/student//cases/una_palabra' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
        --tool=memcheck
        --leak-check=full --leak-resolution=med --show-reachable=yes
        --trace-children=yes
        --track-fds=yes
        --track-origins=no
        --time-stamp=yes --num-callers=20
        --error-exitcode=42
        --log-file=__valgrind__
        --suppressions=/task/student/suppressions.txt ./tp
input_single_word.txt

real    0m1.087s
user    0m0.913s
sys     0m0.095s

Valgrind no encontro errores.

```

```

[=>] Comparando archivo_invalido/__return_code__...
1c1
< 255
---
> 1

[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando lenguaje_c/__return_code__...
[=>] Comparando lenguaje_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
[=>] Comparando nombre_largo/__stdout__...
[=>] Comparando tda/__return_code__...
[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
[=>] Comparando una_palabra/__stdout__...
1c1
< 0
---
> 1

[Error] Se encontraron diferencias entre las salidas obtenidas y las esperadas.

```

Ejecución con gdb

```
(base) ezezbogar@Ezequiel:~/Desktop/taller/paso5$ gdb ./tp
```

```
GNU gdb (Ubuntu 9.1-0ubuntu1) 9.1
Copyright (C) 2020 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:
<http://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 ./tp...
(gdb) info functions
All defined functions:
```

File paso5_main.c:

```
9:  int main(int, char **);
```

File paso5_wordscounter.c:

```
14: void wordscounter_create(wordscounter_t *);
18: void wordscounter_destroy(wordscounter_t *);
22: size_t wordscounter_get_words(wordscounter_t *);
26: void wordscounter_process(wordscounter_t *, FILE *);
34: static char wordscounter_next_state(wordscounter_t *, char, char);
```

Non-debugging symbols:

```
0x00000000000001000  _init
0x00000000000001090  __cxa_finalize@plt
0x000000000000010a0  fclose@plt
0x000000000000010b0  __stack_chk_fail@plt
0x000000000000010c0  strchr@plt
0x000000000000010d0  __printf_chk@plt
0x000000000000010e0  fopen@plt
0x000000000000010f0  getc@plt
0x000000000000011b0  _start
0x000000000000011e0  deregister_tm_clones
--Type <RET> for more, q to quit, c to continue without paging--c
0x00000000000001210  register_tm_clones
0x00000000000001250  __do_global_dtors_aux
0x00000000000001290  frame_dummy
0x00000000000001380  __libc_csu_init
0x000000000000013f0  __libc_csu_fini
0x000000000000013f8  _fini
```

(gdb) list wordscounter_next_state

```
33
34 static char wordscounter_next_state(wordscounter_t *self, char state, char c)
{
35     const char* delim_words = " ,.;:\n";
36
37     char next_state = state;
38     if (c == EOF) {
39         next_state = STATE_FINISHED;
40     } else if (state == STATE_WAITING_WORD) {
41         if (strchr(delim_words, c) == NULL)
```

```

42         next_state = STATE_IN_WORD;
(gdb) list
43     } else if (state == STATE_IN_WORD) {
44         if (strchr(delim_words, c) != NULL) {
45             self->words++;
46             next_state = STATE_WAITING_WORD;
47         }
48     }
49     return next_state;
50 }
51
(gdb) break 45
Breakpoint 1 at 0x1300: file paso5_wordscounter.c, line 45.
(gdb) run input_single_word.txt
Starting program: /home/ezezbogar/Desktop/taller/paso5/tp_input_single_word.txt
0
[Inferior 1 (process 64244) exited normally]
(gdb) quit

```

Como se puede observar, si bien creamos un breakpoint, al ejecutar el programa nunca llegamos a esa línea de código (La ejecución del programa debería pausarse hasta que nosotros le indiquemos continuar) ya que nunca se cumple la condición `strchr(delim_words, c) != NULL` en la línea 44 ya que el archivo (o la palabra) no termina con alguno de los caracteres delimitadores y no se incrementa el contador de palabras

Se ejecuta el comando hexdump para examinar el contenido del archivo 'input_single_word.txt'

```

(base) ezezbogar@Ezequiel:~/Desktop/taller/paso5$ hexdump -C
input_single_word.txt
00000000  77 6f 72 64                                |word|
00000004

```

Como se puede observar el archivo no termina con ninguno de los caracteres delimitadores. (El 64 en hexadecimal es la letra d)

Paso 6: SERCOM - Entrega exitosa

Los cambios realizados entre las entregas son:

- Se cambia el código de error de -1 a 1
- Se modifica el flujo de la función `wordscounter_next_state` para que contemple el caso de que luego de la última palabra del archivo puede no haber un carácter delimitador y en su lugar estar directamente el carácter EOF

```

$ diff paso5_main.c paso6_main.c
4c4
< #include "paso5_wordscounter.h"
---
> #include "paso6_wordscounter.h"
7c7
< #define ERROR -1
---
> #define ERROR 1

```

```

$ diff paso5_wordscounter.c paso6_wordscounter.c

```



```

1c1
< #include "paso5_wordscounter.h"
---
> #include "paso6_wordscounter.h"
9a10
> #define DELIM_WORDS " ,.:\n"
35,36d35
<     const char* delim_words = " ,.:\n";
<
38,41c37,41
<     if (c == EOF) {
<         next_state = STATE_FINISHED;
<     } else if (state == STATE_WAITING_WORD) {
<         if (strchr(delim_words, c) == NULL)
---
>
>     if (state == STATE_WAITING_WORD) {
>         if (c == EOF) {
>             next_state = STATE_FINISHED;
>         } else if (strchr(DELIM_WORDS, c) == NULL) {
42a43
>     }
44c45,48
<     if (strchr(delim_words, c) != NULL) {
---
>     if (c == EOF) {
>         next_state = STATE_FINISHED;
>         self->words++;
>     } else if (strchr(DELIM_WORDS, c) != NULL) {
48a53
>

```

El código compila sin problemas

```

Descomprimiendo el código 'source_unsafe.zip'...
Archive:  source_unsafe.zip
e5033d51c973cee10bbb3fd7cbf0afcb94332a43
  inflating: source_unsafe/paso6_main.c
  inflating: source_unsafe/paso6_wordscounter.c
  inflating: source_unsafe/paso6_wordscounter.h
  extracting: source_unsafe/README.md
Compilando el código...
  CC  paso6_wordscounter.o
  CC  paso6_main.o
  LD  tp

real    0m0.399s
user    0m0.139s
sys     0m0.063s

```

Todos los archivos cumplen con las normas de codificación de CPPLint.

```

Done processing /task/student//source_unsafe/paso6_main.c
Done processing /task/student//source_unsafe/paso6_wordscounter.c
Done processing /task/student//source_unsafe/paso6_wordscounter.h

```

Se corren las pruebas exitosamente

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (sin valgrind)...
```

```
[=>] Command line: ./tp invalid_file
```

```
real    0m0.005s
```

```
user    0m0.003s
```

```
sys     0m0.000s
```

```
[=>] Executando el caso '/task/student//cases/entrada_estandar' (sin valgrind)...
```

```
[=>] Command line: ./tp
```

```
real    0m0.003s
```

```
user    0m0.003s
```

```
sys     0m0.000s
```

```
[=>] Executando el caso '/task/student//cases/language_c' (sin valgrind)...
```

```
[=>] Command line: ./tp input_c.txt
```

```
real    0m0.003s
```

```
user    0m0.001s
```

```
sys     0m0.002s
```

```
[=>] Executando el caso '/task/student//cases/nombre_largo' (sin valgrind)...
```

```
[=>] Command line: ./tp input_extremely_long_filename.txt
```

```
real    0m0.003s
```

```
user    0m0.001s
```

```
sys     0m0.002s
```

```
[=>] Executando el caso '/task/student//cases/tda' (sin valgrind)...
```

```
[=>] Command line: ./tp input_tda.txt
```

```
real    0m0.003s
```

```
user    0m0.002s
```

```
sys     0m0.000s
```

```
[=>] Executando el caso '/task/student//cases/una_palabra' (sin valgrind)...
```

```
[=>] Command line: ./tp input_single_word.txt
```

```
real    0m0.002s
```

```
user    0m0.002s
```

```
sys     0m0.000s
```

```
[=>] Comparando archivo_invalido/__return_code__...
[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando language_c/__return_code__...
[=>] Comparando language_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
[=>] Comparando nombre_largo/__stdout__...
[=>] Comparando tda/__return_code__...
[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
[=>] Comparando una_palabra/__stdout__...
No se encontraron diferencias.
```

Se corren las pruebas con Valgrind exitosamente

```
[=>] Executando el caso '/task/student//cases/archivo_invalido' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
                --tool=memcheck
                --leak-check=full --leak-resolution=med --show-reachable=yes
                --trace-children=yes
                --track-fds=yes
                --track-origins=no
                --time-stamp=yes --num-callers=20
                --error-exitcode=42
                --log-file=__valgrind__
                --suppressions=/task/student/suppressions.txt ./tp invalid_file

real    0m0.993s
user    0m0.806s
sys     0m0.122s
```

Valgrind no encontro errores.

```
[=>] Executando el caso '/task/student//cases/entrada_estandar' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
                --tool=memcheck
                --leak-check=full --leak-resolution=med --show-reachable=yes
                --trace-children=yes
                --track-fds=yes
                --track-origins=no
                --time-stamp=yes --num-callers=20
                --error-exitcode=42
                --log-file=__valgrind__
                --suppressions=/task/student/suppressions.txt ./tp

real    0m1.096s
user    0m0.842s
sys     0m0.118s
```

Valgrind no encontro errores.

```
[=>] Executando el caso '/task/student//cases/language_c' (con valgrind)...
[=>] Command line: /usr/bin/valgrind
                --tool=memcheck
                --leak-check=full --leak-resolution=med --show-reachable=yes
                --trace-children=yes
                --track-fds=yes
```

```
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_c.txt
```

```
real    0m0.944s
user    0m0.802s
sys     0m0.087s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/nombre_largo' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp
```

input_extremely_long_filename.txt

```
real    0m0.989s
user    0m0.836s
sys     0m0.077s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/tda' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp input_tda.txt
```

```
real    0m1.048s
user    0m0.890s
sys     0m0.105s
```

Valgrind no encontro errores.

[=>] Executando el caso '/task/student//cases/una_palabra' (con valgrind)...

[=>] Command line: /usr/bin/valgrind

```
--tool=memcheck
--leak-check=full --leak-resolution=med --show-reachable=yes
--trace-children=yes
--track-fds=yes
--track-origins=no
--time-stamp=yes --num-callers=20
--error-exitcode=42
--log-file=__valgrind__
--suppressions=/task/student/suppressions.txt ./tp
```

input_single_word.txt

```
real    0m0.972s
user    0m0.817s
sys     0m0.086s
```

Valgrind no encontro errores.

```
[=>] Comparando archivo_invalido/__return_code__...
[=>] Comparando archivo_invalido/__stdout__...
[=>] Comparando entrada_estandar/__return_code__...
[=>] Comparando entrada_estandar/__stdout__...
[=>] Comparando lenguaje_c/__return_code__...
[=>] Comparando lenguaje_c/__stdout__...
[=>] Comparando nombre_largo/__return_code__...
[=>] Comparando nombre_largo/__stdout__...
[=>] Comparando tda/__return_code__...
[=>] Comparando tda/__stdout__...
[=>] Comparando una_palabra/__return_code__...
[=>] Comparando una_palabra/__stdout__...
No se encontraron diferencias.
```

Excelente. Todos los casos de pruebas, publicos y privados, fueron superados con exito.

```
(base) ezezbogar@Ezequiel:~/Desktop/taller/paso6$ make
CC  paso6_wordscounter.o
CC  paso6_main.o
LD  tp
(base) ezezbogar@Ezequiel:~/Desktop/taller/paso6$ ./tp input_single_word.txt
1
(base) ezezbogar@Ezequiel:~/Desktop/taller/paso6$ ./tp <input_single_word.txt
1
(base) ezezbogar@Ezequiel:~/Desktop/taller/paso6$ ./tp <input_single_word.txt
>output_single_word.txt
//En este ultimo la salida se guarda en output_single_word.txt
```

Paso 7: SERCOM - Revisión de la entrega

Como se pudo ver en el paso anterior, esta última entrega pasa correctamente todas las pruebas ejecutadas por el SERCOM

- Link al repositorio de GitHub: <https://github.com/ezezbogar/TP0-Taller>