***TP N°1: Compilación y Linkedición***

La siguiente guía cubre los contenidos vistos en las clases teóricas:

**1. Introducción a la materia**

**2. Introducción al lenguaje C**

Antes de comenzar con la guía, se recomienda la lectura del apunte **Tutorial de Compilación en Linux** para comprender el uso del compilador gcc. Disponible en Campus ITBA.

Se recomienda para todo el transcurso de la materia **utilizar los siguientes flags de gcc**.

**-Wall** Lista más advertencias del código

**-pedantic** Realiza las validaciones con el estándar de C

**-std=c99** Para forzar el estándar ANSI C ISO C-99 recomendado por la cátedra

### ***Ejercicio 1***

Escribir y compilar el siguiente programa. Anotar los errores de compilación encontrados.

**int**

**main(void)**

**{**

**/\* Esto es un primer comentario**

**\*/  
 \\* Que buen comentario! \*\**

**/\* Es valido /\* o no \*/ este comentario? \*/**

**/ \* Esto es un ultimo comentario \*/**

**return 0**

**}**

**error: void function 'main' should not return a value [-Wreturn-type]**

**error: expected expression**

**warning: backslash and newline separated by space [-Wbackslash-newline-escape]**

**warning: '/\*' within block comment [-Wcomment]**

***Ejercicio 2***

Idem al ejercicio 1.

**int**

**main( void /\* programa que no hace nada \*/)**

**{**

**return 0**

**}**

**error: expected ';' after return statement**

### ***Ejercicio 3***

Idem al ejercicio 1. Corregir los errores y correr el programa.

**#include <stdio.h>**

**int**

**Main(void)**

**{**

**printf ("Estamos escribiendo un mensaje \n")**

**Return = 0**

**}**

**error: expected ';' after expression**

**error: use of undeclared identifier 'Return'**

**—---- corregido**

**#include <stdio.h>**

**int main(void)**

**{**

**printf ("Estamos escribiendo un mensaje \n");**

**return 0;**

**}**

### ***Ejercicio 4***

Idem al ejercicio 3.

**#include <stdioh>**

**int**

**main[void]**

**{**

**int a, b = c = 5**

**a = b + c**

**return: 0**

**}**

**fatal error: 'stdioh' file not found. Tengo que poner stdio.h**

**error: expected expression debe ir main(void)**

**warning: variable named 'main' with external linkage has undefined behavior [-Wmain]**

**error: expected ';' after top level declarator**

**—---- corregido**

**#include <stdio.h>**

**int main(void)**

**{**

**int c;**

**int a, b = c = 5;**

**a = b + c;**

**return 0;**

**}**

### ***Ejercicio 5***

Idem al ejercicio 3.

**# include <stdio.h>**

**int**

**main(void)**

**{**

**int i, j, max;**

**i = j = 2;**

**max = (i>j? I : j ) ;**

**end**

**}**

**error: use of undeclared identifier 'I'**

**error: use of undeclared identifier 'end'**

**—--- corregido**

**# include <stdio.h>**

**int**

**main(void)**

**{**

**int i, j, max;**

**i = j = 2;**

**max = (i>j? i : j );**

**return 0;**

**}**

***Ejercicio 6***

Compilar pero no linkeditar el archivo **tp1\_06.c**  ¿Con qué atributos (permisos) se crea el archivo resultante? Intentar ejecutarlo. el archivo objeto resulto ser solo de lectura.

Recordá que si sólo se desea compilar (obtener el código objeto), es necesario utilizar el flag -c. Se recomienda utilizar esta opción para diferenciar claramente los errores de compilación y linkedición.

$> gcc -c tp1ej6.c -Wall -pedantic -std=c99

### ***Ejercicio 7***

Compilar y linkeditar (en pasos separados) el archivo **tp1\_07.c**.Indicar en qué paso falla y por qué.

falla en la linkedicion, porque no encuentra el main. hay que agregarle el main.

**error: linker command failed with exit code 1 (use -v to see invocation)**

### ***Ejercicio 8***

Idem ejercicio anterior con archivo **tp1\_08.c**

Falla en la compilacion pq redefine el main.

### ***Ejercicio 9***

Compilar los archivos **tp1\_09a.c** y **tp1\_09b.c,** y linkeditarlos juntos. ¿En qué paso falla la traducción y por qué?

**Falla en la linkedicion pq hay dos main**

Para compilar y linkeditar juntos a dos archivos fuentes los vamos a mandar a ambos como parámetros de entrada del gcc así:

$> gcc tp1\_09a.c tp1\_09b.c -o tp1\_09 -Wall -pedantic -std=c99

### ***Ejercicio 10***

Compilar y linkeditar el archivo **tp1\_10.c** con la opción **-o tp1\_10** y probar los siguientes comandos:

1. **tp1\_10**
2. **tp1\_10 | cat**
3. **tp1\_10 > out.txt**

Explicar el por qué de cada salida.

1. Funciona perfectamente
2. No funciona pues el codigo pide una letra, si lo concatenar con un cat, nunca va a recibir el input que pide. No ejecuta pero concatena.
3. idem anterior.

## Ejercicio 11

Compilar el programa **tp1\_11.c** . Si se produce un error corregirlo y luego probar los siguientes comandos:

1. **tp1\_11 < tp1\_11.txt**
2. **tp1\_11 < tp1\_11.txt > out.txt**

Error fatal, se incluye a si mismo.

Explicar el por qué de cada salida.

1. Ejecuta el ejecutable con la primer letra del .txt como input
2. idem anterior, pero ademas con el > redirigimos el output a un .txt

### ***Ejercicio 12***

Preprocesar el programa **tp1\_12.c** y analizar la salida.

Con la opción -E, gcc se detiene luego del preprocesamiento.

$> gcc -E tp1ej12.c

### ***Ejercicio 13***

Compilar, linkeditar y ejecutar el programa **tp1\_13.c**. Hacer lo mismo con el programa **tp1\_13b.c**. Analizar las diferencias entre ambos.

***en el a define una macro, pero no la usa, en cambio en el b define y la usa.***

***Ejercicio 14***

Compilar y linkeditar el programa **tp1\_14.c**. Hacer lo mismo con el programa **tp1\_14b.c**. Analizar diferencias con el ejercicio anterior.

Este es un ejemplo similar al anterior. Se define una constante octal (porque comienza con 0) y uno de sus dígitos es nueve, que es un error. En el primer caso, al no usarla, no provoca error.

En el segundo caso, a diferencia del ejercicio anterior, se genera un **error de compilación** ya que no puede interpretar el valor de la constante.

***Ejercicio 15***

¿Por qué el siguiente código falla en la compilación?

**#include <stdio.h>**

**int**

**main(void)**

**{**

**int manual = 7;**

**int auto = 9;**

**/\* Se calcula el promedio \*/**

**return 0;**

**}**

Revisá las primeras diapositivas de la clase teórica 2. Introducción al lenguaje C

***auto es una palabra reserva, no se puede usar para una variable.***

***Ejercicio 16***

Indicar, para el escenario listado, si gcc falla o no en la construcción de un programa ejecutable. Si falla, decir si lo hace en la etapa de preprocesamiento, compilación o linkedición:

1. Archivo de encabezado .h inexistente
2. Más de una función main() en un único archivo fuente .c
3. Más de una función main() en un conjunto de archivos fuente .c
4. Ninguna función main() presente en un conjunto de archivos fuente .c
5. Uso de una palabra reservada del lenguaje como nombre de una variable
6. Redefinición de una variable dentro de un bloque o función (en el siguiente ejemplo, la variable x está definida dos veces):

**#include <stdio.h>**

**int**

**main(void)**

**{**

**int x;**

**int x;**

**return 0;**

**}**

1. La función main() no tiene la línea return 0; o similar al final del cuerpo de la misma
2. Dos funciones con el mismo nombre en el mismo archivo fuente .c

Cuando veas que gcc lista un error con el término ld, significa que falló la linkedición. Si el error es de linkedición entonces el preprocesamiento y la compilación de todos los archivos fuentes involucrados fue exitosa.