

APENDICE C MENSAJES Y CODIGOS DE ERROR

MENSAJES DE ERROR DEL COMPILADOR

La siguiente es una lista de los posibles mensajes de error que se generan durante el proceso de compilación. Es posible, sin embargo, que el compilador complemente la información con identificadores y/o nombres de archivos, por ejemplo:

ERROR 15: FILE NOT FOUND [WINDOW.TPU]

Error 15: No se encontró archivo [WINDOW.TPU]

Cuando se detecta el error, Turbo Pascal carga el archivo fuente y coloca el cursor en donde se encontró. Nótese que algunos errores no se detectan hasta que se han compilado algunos estatutos más. Por ejemplo, un error de inconcordancia de tipo (type mismatch) en un estatuto de asignación no puede ser detectado hasta que se evaluó completamente la expresión después del símbolo "=". En tales casos el error está a la izquierda o arriba del cursor.

1) Out of memory (Memoria insuficiente).

Este error ocurre cuando el compilador no tiene memoria suficiente para ejecutar el programa. Algunas soluciones a este problema son:

- Compilar a disco, generando un archivo .EXE
- Encadenar (linkear) a disco, a través de la directiva del compilador {\$L-} al inicio del programa.
- Remover de memoria los programas residentes, como el SideKick o SuperKey.
- Utilizar el compilador TPC.EXE, en lugar del TURBO.EXE.

Si ninguna de estas sugerencias ayuda, significa que el programa o unidad es demasiado grande y que debe particionarse en dos o más unidades.

2) Identifier expected (Se espera un identificador).

En este punto se espera un identificador, o probablemente se está tratando de declarar una palabra reservada.

3) Unknown identifier (Identificador desconocido).

El identificador no ha sido declarado.

4) Duplicate identifier (Identificador duplicado).

El identificador ya ha sido utilizado en el bloque actual.

5) Syntax error (Error de sintaxis).

Se encontró un caracter ilegal en el texto fuente, o probablemente falta la comilla que encierra a un string.

6) Error in real constant (Error en la constante real).

Error en la sintaxis declarativa de la constante real.

7) Error in integer constant (Error en la constante entera).

Error en la sintaxis declarativa de la constante entera.

8) String constant exceeds line (La constante string excede la línea).

Probablemente se olvidó poner la comilla final en la constante string.

9) Too many nested files (Muchos archivos anidados).

El compilador permite no más de cinco archivos anidados.

10) Unexpected end of file (Fin de archivo inesperado).

Este error se genera cuando el archivo fuente termina antes del END principal. La mayor parte de las veces el número de BEGIN y END está desbalanceado.

11) Line too long (Línea muy larga).

El número máximo de una línea es de 126 caracteres.

12) Type identifier expected (Se espera un identificador de tipo).

El identificador no denota un tipo como debiera.

13) Too many open files (Demasiados archivos abiertos).

Cuando ocurre este error, significa que en el archivo CONFIG.SYS no se incuyó el comando FILES = XXX, o se especificó un número muy pequeño de archivos. Incremente el valor a un número mayor, por ejemplo 20.

14) Invalid file name (Nombre de archivo inválido).

El nombre del archivo es inválido, o no existe el subdirectorio especificado.

15) File not found (No se encontró el archivo).

No se pudo encontrar el archivo en el directorio actual, o en cualquiera de los subdirectorios de búsqueda.

16) Disk full (Disco lleno).

El disco está lleno. Borre algunos archivos o utilice un disco nuevo.

17) Invalid compiler directive (Directiva del compilador inválida).

La letra que especifica la directiva del compilador no pudo ser reconocida, o alguno de los parámetros no es válido, o se está usando una directiva de compilador global cuando ya ha empezado la compilación del bloque principal.

18) Too many files (Demasiados archivos).

Hay muchos archivos involucrados en el proceso de compilación del programa o de la unidad. Trate de utilizar menos, uniéndolos y utilizando nombres cortos para ellos.

19) Undefined type in pointer definition (Tipo indefinido en la definición del apuntador).

El tipo fue referenciado en la declaración de tipos, pero nunca fue declarado.

20) Variable identifier expected (Se espera identificador de variable).

El identificador no denota una variable.

21) Error in type (Error en tipo).

Este símbolo no puede empezar la definición de un tipo.

22) Structure too large (Estructura demasiado grande).

El tamaño de memoria máximo es de 45520 bytes.

23) Set base type out of range (El tipo set está fuera de rango).

La base del tipo set debe tener su rango en los límites de 0 a 255, o un tipo enumerado que no contenga más de ese número de elementos.

24) File componentes may not be files (Los campos de un archivo no pueden ser archivos).

La estructura de un archivo no puede contener archivos.

25) Invalid string length (Longitud del string inválida).

La declaración de la longitud máxima de un string no debe entrar fuera del rango de 1..255.

26) Type mismatch (Inconcordancia de tipo).

Esto puede deberse a lo siguiente:

- Tipos incompatibles de la variable y la expresión en el estatuto de asignación.
- Tipos incompatibles en la llamada a una función o procedimiento, entre los parámetros formales y los actuales.

- El tipo de la expresión y el tipo índice de un arreglo no concuerdan.
- Tipos incompatibles entre la expresión y los operandos.

27) Invalid subrange base type (Tipo base de subrango inválido).

Todos los tipos ordinales son tipos base válidos.

28) Lower bound greater than upper bound (El límite inferior es más grande que el superior).

La declaración de un tipo subrango especifica un límite inferior mayor al superior.

29) Ordinal type expected (Se espera tipo ordinal).

No se permiten aquí los tipos reales, strings, estructurados y apuntadores.

30) Integer constant expected (Se espera constante del tipo integer).

31) Constant expected (Se espera constante).

32) Integer o real constant expected (Se espera constante entera o real).

33) Type identifier expected (Se espera el identificador de un tipo).

El identificador no denota un tipo como debiera.

34) Invalid function result type (Tipo inválido en el resultado de la función).

Los tipos de resultados válidos para una función son los tipos simples, string y apuntadores.

35) Label identifier expected (Se espera un identificador de etiqueta).

El identificador no denota una etiqueta como debiera.

36) BEGIN expected (Se espera la palabra BEGIN).

37) END expected (Se espera la palabra END).

38) Integer expression expected (Se espera una expresión entera).

La expresión anterior debe ser de tipo INTEGER.

39) Ordinal expression expected (Se espera una expresión ordinal).

La expresión anterior debe ser de tipo ordinal.

40) Boolean expression expected (Se espera una expresión booleana).

La expresión anterior debe ser de tipo BOOLEAN.

41) Operand types do not match operator (Los tipos de los operandos no concuerdan con el operador).

El operador no puede aplicarse a operandos de este tipo, por ejemplo: 'A' div '2'.

42) Error in expression (Expresión con error).

Este símbolo no puede participar en la expresión de la forma en que lo hace. Es posible que se haya olvidado escribir un operador entre dos operandos.

43) Illegal assignment (Asignación ilegal).

Esto puede deberse a que:

- No se les puede asignar valor a los archivos y a las variables sin tipo.
- A un identificador de una función solamente se le pueden asignar valores dentro del estatuto que es parte de la función.

44) Field identifier expected (Se espera identificador de campo).

El identificador no denota un campo en la variable anterior de tipo registro.

45) Object field too large (El archivo objeto es demasiado grande).

Turbo Pascal no puede encadenar archivos .OBJ más grandes de 64K.

46) Undefined external (External no definido).

Una función o procedimiento del tipo EXTERNAL no tiene una definición de tipo PUBLIC que concuerde con el archivo objeto.

Asegúrese de que están especificados todos los archivos objeto con la directiva {\$ L nomarch}, y revise los identificadores en los archivos .ASM.

47) Invalid object file record (Registro del archivo objeto inválido).

El archivo .OBJ contiene un registro inválido; asegúrese de que dicho archivo es realmente un archivo .OBJ.

48) Code segment too large (Segmento de código demasiado grande).

El tamaño máximo del código de un programa o una unidad es de 65520 bytes. Si está compilando un programa, cambie algunos procedimientos o funciones en una unidad. Si está compilando una unidad, particiónela en dos o más unidades.

49) Data segment too large (Segmento de datos demasiado grande).

El tamaño máximo del segmento de datos de un programa es de 65520 bytes, incluyendo los datos declarados en las unidades a usar. Si necesita más datos globales, declare las estructuras grandes como apuntadores y almacénelas dinámicamente usando el procedimiento NEW.

50) DO expected (Se espera la palabra DO).

51) Invalid PUBLIC definition (Definición del tipo PUBLIC inválida).

Esto puede deberse a que:

- El identificador fue hecho público por medio de la directiva PUBLIC en lenguaje ensamblador, pero no concuerda con la declaración externa del programa o unidad de Pascal.
- Dos o más directivas PUBLIC en lenguaje ensamblador definen al mismo identificador.
- El archivo .OBJ define símbolos PUBLIC que no residen en el segmento CODE.

52) Invalid EXTRN definition (Definición EXTRN inválida).

Esto puede deberse a que:

- Se referenció el identificador por medio de la directiva EXTRN en lenguaje ensamblador, pero no está declarada en el programa o unidad de Pascal, ni en la parte de interfase de cualquiera de las unidades que se usan.
- El identificador denota una variable absoluta.
- El identificador denota un procedimiento o función del tipo INLINE.

53) Too many EXTRN definitions (Demasiadas definiciones de tipo EXTRN).

Turbo Pascal no puede manejar archivos .OBJ con más de 256 definiciones EXTRN.

54) OF expected (Se espera la palabra OF).

55) INTERFACE expected (Se espera la palabra INTERFACE).

56) Invalid relocatable reference (Referencia relocatable inválida).

Esto puede deberse a que:

- El archivo .OBJ contiene datos y referencias relocatables en segmentos diferentes al CODE. Por ejemplo, se está intentando declarar variables inicializadas en el segmento DATA.
- El archivo .OBJ contiene referencias medidas en bytes para símbolos relocatables. Este error ocurre cuando se usan los operadores HIGH y LOW con símbolos relocatables, o cuando se refieren los símbolos relocatables en directivas.
- Un operando refiere a un símbolo relocatable que no fue definido en el segmento CODE o en el segmento DATA.
- Un operando refiere a un procedimiento o función EXTRN con un offset, por ejemplo, CALL MiFuncion + 8.

CAPILLA ALFONSO

57) THEN expected (Se espera la palabra THEN).

58) TO or DOWNT0 expected (Se espera la palabra TO o DOWNT0).

59) Undefined FORWARD (FORWARD indefinido).

Esto puede deberse a que:

- El procedimiento o función fue declarado en la parte INTERFACE de una unidad, pero su definición nunca ocurrió en la parte de la implementación.
- El procedimiento o función fue declarado con un FORWARD, pero su definición nunca se encontró.

60) Too many procedures (Demasiados procedimientos).

Turbo Pascal no permite más de 512 procedimientos o funciones por módulo. Si está compilando un programa, cambie algunos procedimientos o funciones en una unidad. Si está compilando una unidad, particiónela en dos o más unidades.

61) Invalid typecast (Tipo forzado inválido).

Esto puede deberse a que:

- Los tamaños de la variable referencial y el tipo destino difieren en la variable de tipo forzado.
- Está intentando forzar el tipo de una expresión, cuando solamente se permite referenciar a una variable.

62) Division by zero (División entre cero).

63) Invalid file type (El tipo del archivo es inválido).

El procedimiento que maneja el archivo no soporta archivos de este tipo. Por ejemplo, no se puede utilizar READLN en un archivo con registros o SEEK en un archivo de texto.

64) Cannot READ or WRITE variables of this type (No se puede utilizar un READ o un WRITE con variables de este tipo).

Esto puede deberse a que:

- READ y READLN pueden aceptar variables de tipo CHAR, INTEGER, REAL y STRING.
- WRITE y WRITELN pueden aceptar variables de tipo CHAR, INTEGER, REAL, STRING y BOOLEAN.

65) Pointer variable expected (Se espera una variable de tipo apuntador).

La variable anterior debe ser de tipo apuntador.

66) String variable expected (Se espera variable de tipo STRING).

La variable anterior debe ser de tipo STRING.

67) String expression expected (Se espera una expresión STRING).

La expresión anterior debe ser de tipo STRING.

68) Circular unit reference (Referencia circular en unidad).

Dos unidades no pueden usarse la una a la otra.

69) Unit name mismatch (Inconcordancia en el nombre de la unidad).

El nombre de la unidad en el archivo .TPU no concuerda con el nombre que se usó en la cláusula USES.

70) Unit version mismatch (Versión incorrecta de unidad).

Una o más unidades se han modificado desde que la unidad fue compilada. Utilice la opción COMPILE/MAKE o COMPILE/BUILD para compilar automáticamente aquellas unidades que lo necesiten.

71) Duplicate unit name (Nombre de unidad duplicado).

El nombre de la unidad ya se está usando en otra parte.

72) Unit file format error (Error en el formato del archivo de la unidad).

El archivo .TPU contiene un error en alguna de sus partes. Revise las unidades involucradas.

73) Implementation expected (Se espera la palabra IMPLEMENTATION).

La palabra reservada IMPLEMENTATION es parte del desarrollo de unidades en Turbo Pascal.

74) Constant and case do not match (La constante y el estatuto case no concuerdan).

75) Record variable expected (Se espera una variable de tipo registro).

76) Constant out of range (La constante está fuera de rango).

77) File variable expected (Se espera una variable de tipo FILE).

78) Pointer expression expected (Se espera una expresión de tipo POINTER).

79) Integer or real expression expected (Se espera una expresión de tipo INTEGER o REAL).

80) Label not within current block (La etiqueta no está en el bloque actual).

Un estatuto goto no puede referenciar a una etiqueta fuera del bloque actual.