



# **DESARROLLO DE PROGRAMAS**

## **CAPITULO II DESARROLLO DE PROGRAMAS**

### **INTRODUCCION**

Cuando se desarrollo un programa para un sistema digital generalmente pocas veces se tiene un procedimiento bien implementado. Debido a que los programas generalmente son cortos: Inicialización del dispositivo a controlar, alguna rutina de entrada y salida, y la operación lógica sobre estas.

Para la programación de sistemas digitales generalmente se utiliza lenguaje ensamblador o algún lenguaje de bajo nivel como C.

Los problemas se complican cuando se desea realizar una aplicación más extensa en donde se tienen que diseñar e implantar grandes programas.

Para resolver este problema muchas compañías ofrecen una gran variedad de herramientas de desarrollo, pero en pocas plantean el fondo de la metodología real del desarrollo de programas para sistemas digitales.

A continuación se presentan los pasos para el desarrollo de programas utilizando las herramientas y utilería de Borland.

### **DESARROLLO DE PROGRAMAS**

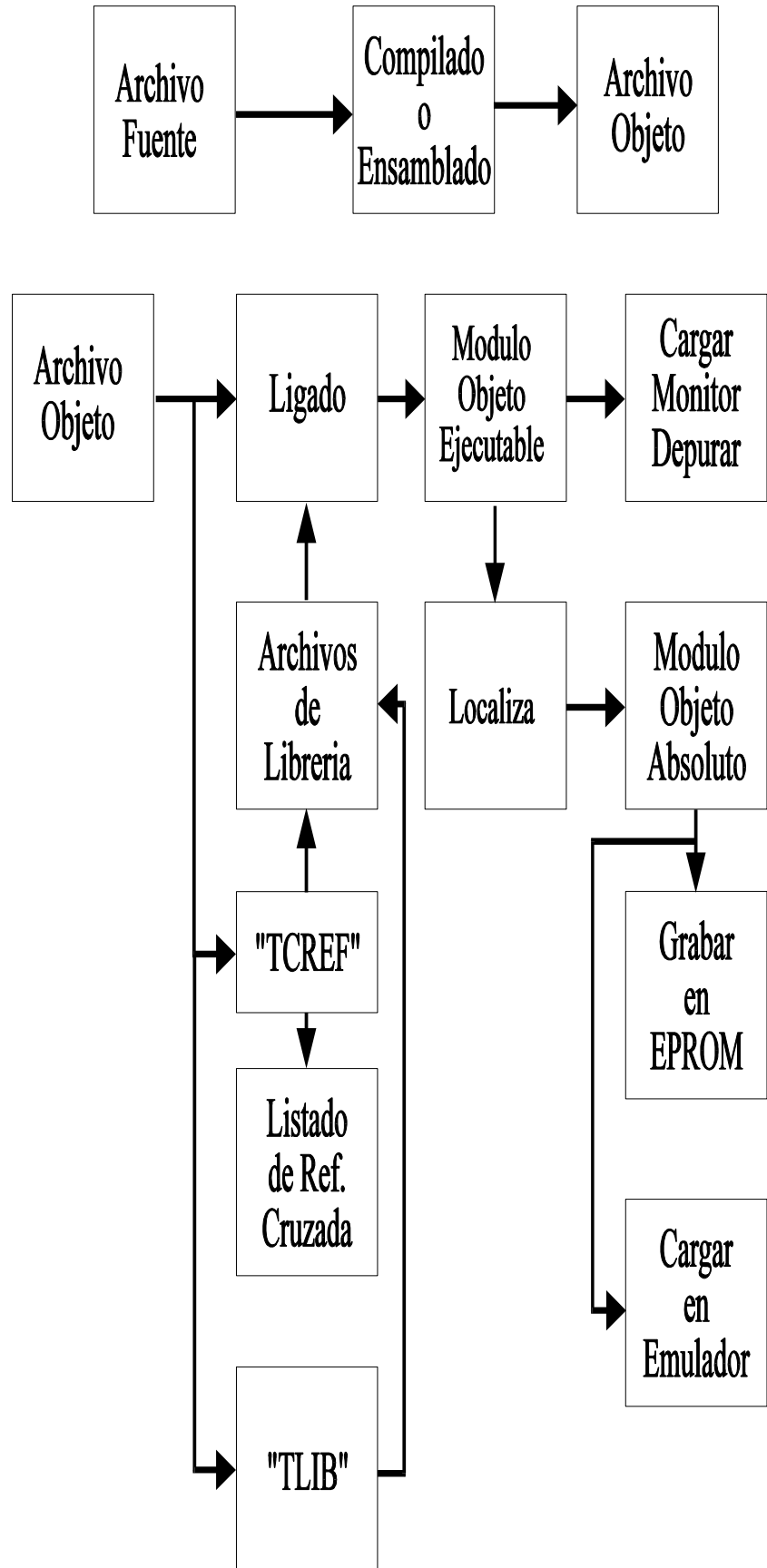
El desarrollo de programas para sistemas digitales es un proceso

## **DESARROLLO DE PROGRAMAS**

sencillo pero laborioso. Los pasos necesario depende: del lenguaje usado para obtener el código ejecutable, la complejidad del producto final y de las herramienta utilizadas. En la figura 2.1 se presenta en forma clara y breve los pasos necesarios para el desarrollo de un programa.

Se presentan los pasos y posibilidades que ofrecen las herramientas y utilerias de Borland. Los pasos clásicos son: editar el programa, ligarlo con librerías y otros programas, después localizarlo en una dirección específica. El programa resultante en esta etapa, puede se cargado a un monitor de depuración o bien grabado en una EPROM.

## DESARROLLO DE PROGRAMAS



## DESARROLLO DE PROGRAMAS

Fig. 2.1      Procedimiento en el desarrollo de programas con herramienta de programación de Borland.

En el NTRMX se pueden correr programas como en el monitor de depuración. Para ello, hay que crear un programa ejecutable para el ambiente del NTRMX. Los pasos que se siguen, son similares a la creación de un programa ejecutable en cualquier lenguaje de programación. Hay que crear el archivo fuente, compilarlo o ensamblarlo el programas fuente, dando como resultado un archivo objeto. Teniendo el arhivo objeto, se procede a ligarlo con otros archivos objeto y librerias para obtener un programa ejecutable -formato "EXE" o "COM".

## DESARROLLO DE PROGRAMAS

Por ejemplo, el ligado del archivo fuente prueba.c, despues de haber obtenidos los archivos objetos es:

```
>TLINK /T /m /s inicio.obj prueba.obj, prueba, ,  
..\lib\stdio.lib ..\lib\ntmx.lib c:\tc2\lib\cs.lib
```

Archivo de salida -> ***prueba.com***

Este comando, del turbo ligador(TLINK), liga los archivos objeto: inicio y prueba; con la librerías stdio.lib, ntrmx.lib y cs.lib. Del ligado se obtiene una archivo en formato "COM" con la opción /T. Las opciones /m y /s dan un listado del mapa de localización de todas la variables y funciones públicas.

El archivo de salida ***prueba.com*** se puede cargar en ambiente del NTRMX y crear un proceso. Utilizando los comandos "cargar" programa y "ejecuta" programa.

## HERRAMIENTAS UTILIZADAS:

TASM.EXE Turbo ensamblador. Ensamblador de programas. El resultado en un archivo objeto.

TLIB.EXE Turbo librerías. Permite la creación de librerías. Esta librería son generalmente funciones de uso frecuente.

TCC.EXE Compilador de C en línea. El resultado de este programa en un archivo objeto. Compilador en línea ofrece una gran variedad de opciones muy utiles para depurado de los programas.

## **DESARROLLO DE PROGRAMAS**

TC.EXE Turbo C. Compilador de C en ambiente de ventanas. La característica de ventanas ofrece rapidez y sencillez en compilado de los programas.

TLINK.EXE Turbo ligador. Permite el ligado de programas objetos y librerías para crear programas ejecutables. El resultado de este programas ejecutable y listados de los mapas que facilitan la depuración del programa.

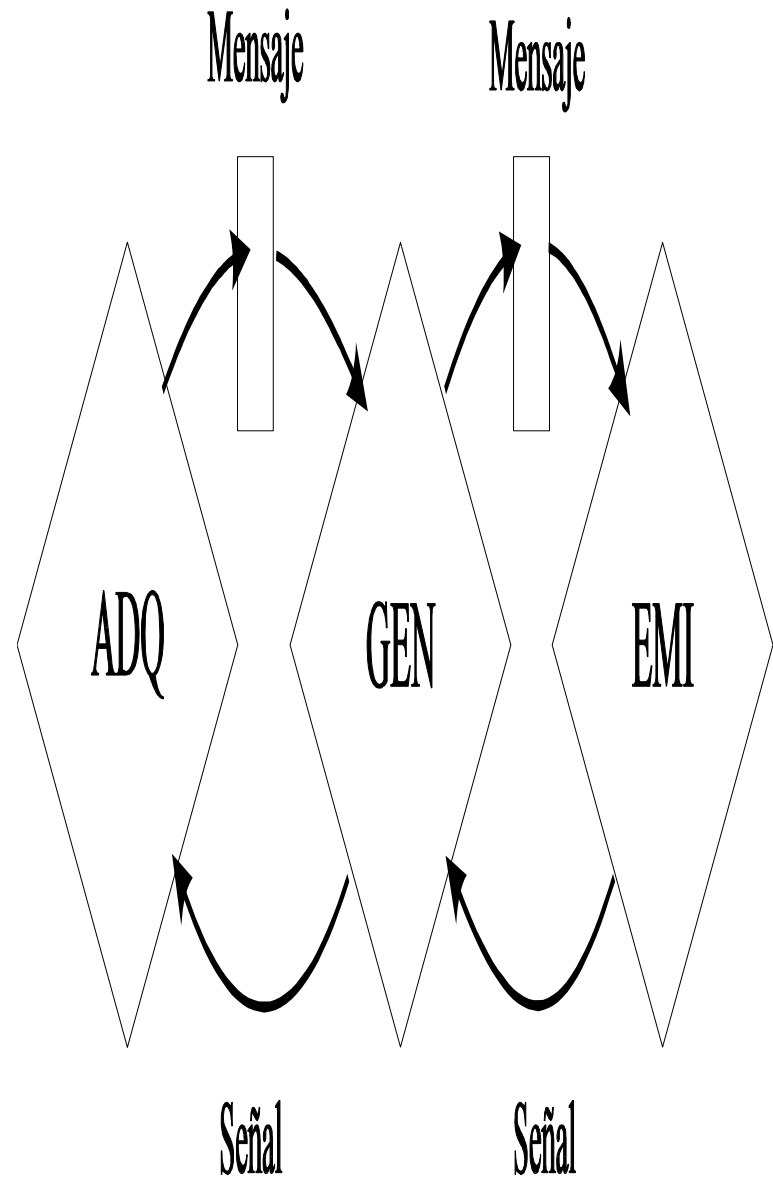
MAKE.EXE "Makefiles". Permite la realización de archivos de proyectos. En él cual se indica la dependencia de archivos y la compilación condicional con respecto a las última fecha de modificación de los archivos fuentes. Por lo que "make" solo realiza el ensamblado o compilación de los archivos que hallan sido modificados.

LINEA.EXE Monitor de depuración. Permite cargar y ejecutar programas en el sistema digital. Para mayores detalles consultar el manual de Monitor de Depuración.

## **REQUERIMIENTOS PARA APLICACIONES DE TIEMPO REAL.**

Una aplicación de tiempo real, comúnmente está constituida por varios procesos que cooperan entre si. Generalmente existe: 1) un manejador de entrada- proceso de adquisición de entrada. 2) Un proceso de transformación de la entrada Generación de Comandos. Y 3) Un proceso que maneja las salidas- Proceso emisor de Comandos. En la figura 2.2 se muestra como se comunican y sincronizan estos procesos.

## DESARROLLO DE PROGRAMAS



### Procesos

ADQ: Adquisición de información.

GEN: Generación de Comandos.

EMI: Emisión de Comando.



## DESARROLLO DE PROGRAMAS

Fig. 2.2      Coordinación de procesos en una aplicación de tiempo real.

Para crear cada uno de estos procesos se puede diseñar programas por separados; utilizando las herramientas y la metodología de desarrollo de programas antes expuesta. Donde cada programa ejecutable se convierte en un proceso utilizando la función de creación de tarea(TACHE). El envío y recepción de mensaje y señales lo facilitan las funciones del NTRMX. La creación de manejadores de bajo nivel se detalla en el capítulo de ejemplos de proyectos de este manual.

En resumen el desarrollo de una aplicación en tiempo real requiere:

- ▲ Creación de procesos.
- ▲ Creación de manejadores de E/S de bajo nivel.
- ▲ Mecanismo de sincronización y comunicación.
- ▲ Mecanismo de temporización.

## **DESARROLLO DE PROGRAMAS**

Todas estas funciones, que ofrece el NTRMX, pueden ser utilizadas dentro de los procesos de aplicación. Para ello, solo hay que hacer uso adecuado de las primitivas y funciones que ofrece el sistema. Para mayores detalles véase el capítulo de librerías y llamadas al sistema.

## **DESARROLLO DE PROGRAMAS**

### **BASURA**

Para crear uno proceso se muestra la imagen de memoria que guarda los procesos.