



# PRESENTACION DEL NTRMX

## CAPITULO I

La tarea fundamental de cualquier sistema operativo es la de supervisión de los recursos y funciones del equipo de computo. Dentro de Estas funciones se incluye: proporcionar una interfase entre la computadora y el usuario, la administración del sistema de E/S, creación y manejo de archivos para almacenar y operar datos, manejo de la memoria del usuario y del sistema y proporcionar un método para la ejecución de programas.

### COMO INICIAR LA UTILIZACION DEL NUCLEO NTRMX.

El primer paso en la utilización de NTRMX, es familiarizarse con su estructura y los distintos archivos que lo conformas.

La estructura del NTRMX tiene tres partes básicas:

***Núcleo del sistema.*** La estructura standard comprende: Reloj de tiempo real, Manejador de procesos e interrupciones, Manejo de Memoria y comunicación.

***Los manejadores de Entradas y Salidas.*** El cual depende de la circuitería donde se va implementar el sistema. Tiene manejadores de caracteres para puertos serie y manejadores para puertos de entrada y salida paralelo.

## **PRESENTACION DEL NTRMX**

## **PRESENTACION DEL NTRMX**

## PRESENTACION DEL NTRMX

Fig. 1.1 Estructura del NTRMX.

***El interpretador de comandos.*** El cuál facilita el desarrollo y monitoreo de los programas de aplicación. La aplicación no se considera como una parte adicional porque es el problema específico que se desea resolver, depende de la capacidad de uso de la estructura base del núcleo y de los manejadores de entrada y salida. Aunque se da ejemplos base que serán útiles para desarrollar aplicaciones más complejas.

## **PRESENTACION DEL NTRMX**

Para cada una de estas partes se tiene un conjunto de archivos que nos permiten diferenciar a que parte pertenece. Y están agrupados en subdirectorios.

### **VIAJE A TRAVES DE LOS ARCHIVOS DEL NTRMX.**

Archivos de librerías del Núcleo.

CON825X.H      Definiciones de Datos y Tipos para inicialización de periféricos.

INICIALI.H      Definición de las estructuras del bloque de creación de tarea y del descripción de Código.

PRIMITIV.H      Define las funciones prototipo para las llamadas al sistema o primitivas que utiliza el núcleo. Constantes utilizadas en las llamada de las primitivas y el valor de los saltos indirecto.

PROCESOS.H Define las estructuras de datos que utiliza el Núcleo de Tiempo Real MX (NTRMX).

NTRMX.LIB      Contiene la librería, en ensamblador, de todas las llamadas del NTRMX. Las cuales pueden ser utilizadas en lenguaje C. (librería de llamadas).

## **PRESENTACION DEL NTRMX**

Archivos de Entrada y Salida

STDIO.H Define las funciones prototipo de Entrada y Salida.

STDIO.LIB Contiene la librería, en ensamblador, de todas las funciones de llamadas al sistema de Entradas y Salidas.

Archivos de Ejemplos de Aplicación.

HOLA.C Ejemplo del clásico programa "hola" que envía un letrero a la salida estandard, -puerto serie.

ENVIA.C Envía un mensaje en forma periódica a la tarea *espera.c*.

ESPERA.C Recibe el mensaje que envía la tarea *envia.c* y lo despliega a la pantalla -salida estandard.

LIGA.BAT Archivo "batch" que hace el ligado de los archivos: fuente.obj, inicio.obj y las librerías ntrmx.lib, cs.lib y stdio.lib.

Archivos de herramientas o utilerías

## **PRESENTACION DEL NTRMX**

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

TASM.EXE Turbo ensamblador. Ensamblador de programas.

TLIB.EXE Turbo librerías. Permite la creación de librerías.

TCC.EXE Compilador de C en línea versión de Turbo C.

TC.EXE Turbo C. Compilador de C en ambiente de ventanas.

TLINK.EXE Ligador de programas Turbo. Permite el ligado de programas objetos y librerías para crear programas ejecutables.

MAKE.EXE Utilería de "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.

GREP.EXE Utilería que imprime líneas que coincidan con un *patrón* en un archivo. Permite la localización de patrones de palabras en un archivo.

PP.EXE Impresión Formateada. Permite la impresión de archivos con el encabezado de fecha y nombre de archivo, también numera las páginas.



## PRESENTACION DEL NTRMX

Para familiarizarse con el NTRMX se tiene una configuración de ejemplo con una aplicación. Los pasos para correr este ejemplo son:

HOLA.C

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

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

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

El programa *hola.com* puede ser cargado y ejecutado en el ambiente del NTRMX utilizando los siguientes comandos:

```
&carga  
Carga el programa desde la PC  
<ESC>  
c
```

```
>DAME EL NOMBRE DEL ARCHIVO EN BINARIO  
hola.com  
TRANSMITIENDO EL ARCHIVO hola.com  
LA LONGITUD DEL ARCHIVO ES 344  
ARCHIVO TRASMITIDO TOTALMENTE  
&ejecuta
```

Al archivo hola.c al momento de su ejecución manda un mensaje a la pantalla en forma periódica.

## PRESENTACION DE NTRMX

## PRESENTACION DEL NTRMX

### Estructura del Núcleo de tiempo Real NTRMX

El kernel del NTRMX esta conformado de cuatro partes fundamentales:

- 1) Control y planificación de tareas.
- 2) El manejador del reloj de tiempo real e interrupciones.
- 3) El mecanismo de mensajes y señales.
- 4) La administración de memoria.

El núcleo es completado con otros mecanismos que nos facilitan los servicios que ofrece el núcleo, estos son:

- ▲ Nivel bajo de entrada y salidas.
- ▲ Un interpretador de comandos("shell").

La estructura del sistema NTRMX se observa en la figura 1.1.

### SERVICIOS QUE OFRECEN EL NTRMX.

El núcleo de tiempo real(NTRMX) es un sistema operativo que nos permite ejecutar multitareas en el ambiente del sistema digital. En él se facilita el desarrollo de la programación de tiempo real. Los principales servicios que ofrece el NTRMX son:

- a) **EL control de tareas** que se ejecuten en paralelo

## PRESENTACION DEL NTRMX

es suministrada por planificador. Ejemplo: la función `creaTarea()`.

b) ***El control de tiempos y alarmas.*** Ofrecido por servicios de manejador de reloj. Ejemplo la función `duerme()`.

c) ***Los servicios de sincronización*** se dan a través de un manejador de señales. El intercambio de información entre las tareas es a través de mensajes que se depositan en buzones. Ejemplos de funciones: `enviaSenal()`, `esperaSenal()`, `enviaMensaje()`, `esperaMensaje()`.

d) ***El manejo de ocupación y disponibilidad de la memoria*** se realiza por un mecanismo de control de "pozos" de memoria y se ofrecen tres funciones básicas: Inicialización de memoria `inPool()`, petición de memoria `asignaMem()` y liberación de memoria `libMem()`.

e) ***Para la comunicación externa de control y monitoreo.*** El sistema de E/S ofrece los servicios de: lectura de un carácter `getch()`, escritura de un carácter `putchar()`, lectura de una cadena. El sistema de lectura es buffereado.

## CARACTERISTICAS DEL NTRMX

Las características técnicas del sistema operativo de tiempo real NTRMX son:

▲ El sistema operativo está destinado a las familias de los microprocesadores INTEL 80X86.

## **PRESENTACION DEL NTRMX**

- ▲ Ocupa un espacio de memoria de aproximadamente 10K octetos.
- ▲ Maneja de 1 a 255 tareas, señales, buzones, y "pozos" de memoria.
- ▲ El código del sistema operativo es independiente de su posición en memoria.

## COMANDOS DEL "SHELL"

### COMANDOS DEL "SHELL".

Para tener el sistema funcionando en un forma mínima y comunicándose con una computador IBM-PC. Donde el usuario puede cargar y depurar sus programas. El NTRMX cuenta con un intepretador de comandos("shell"). Los servicios que ofrece esta tarea son: Inicialización y prueba del sistema, lectura de memoria, cargar el código de una tarea desde una PC., ejecutar la tarea cargada en memoria. Ver el estado de las tareas y otros servicios que se explicará su uso en esta sección.

En la sintaxis las palabras y símbolos que se imprimen en tipo normal son argumentos y se deben sustituir por nombres y/o argumentos numéricos. En los ejemplos, toda la información que va después del signo de número(#) son comentario. La diagonal "/" que se puede utilizar uno de las dos formas del comando.

<b>Comando:</b>	<b><i>ayuda</i></b>	<b><i>--Ayuda del "shell"</i></b>
<b>Sintaxis:</b>	<b><i>ayuda</i></b>	
<b>Ejemplos:</b>	<b><i>ayuda</i></b>	<i># Lista los comandos del # "shell" del NTRMX.</i>

Lista todos los comandos del interpretador de comandos ("shell") del NTRMX.

<b>Comando:</b>	<b><i>carga</i></b>	<b><i>--Carga un archivo ejecutable</i></b>
<b>Sintaxis:</b>	<b><i>c/carga&lt;cr&gt; archivo</i></b>	

## COMANDOS DEL "SHELL"

### Ejemplos:

A.com

**carga**<cr>**A.com**      # Carga el archivo

# a la memoria del sistema

# digital.

Este comando es un cargador de programas en formato "COM". Todos los archivos en formato "COM" inician en la dirección 0100h. El archivo es cargado en la dirección de la memoria RAM *SEG:0100h*, donde el valor de *SEG* es el valor que se asigna cuando se solicita memoria para este proceso.

Se envía un mensaje de error cuando el archivo que se va a cargar no se encuentra.

Durante la ejecución del comando se despliega en pantalla el porcentaje del programa que ha sido cargado en memoria.

**Comando:**      **duerme**      *--Duerme el "shell"*

**Sintaxis:**      **duerme [tics]**

**Ejemplos:**      **duerme 10**      # Para la ejecución del  
# "shell" durante 10 tic's  
# del reloj del sistema.

El comando duerme se utiliza para bloquear por un tiempo determinado la operación del

## COMANDOS DEL "SHELL"

"shell", después el shell continua su operación normal. Equivale a la función "sleep" de Unix.

<b>Comando:</b>	<b><i>ejecuta</i></b>	<b><i>--Ejecuta una tarea</i></b>
<b>Sintaxis:</b>	<b><i>x/ejecuta</i></b>	
<b>Ejemplos:</b>	<b><i>ejecuta</i></b>	<i># Ejecuta la última tarea</i> <i># carga en memoria.</i>

Ejecuta la última tarea cargada en memoria, si no existe ninguna tarea en memoria el comando envía un mensaje de error. Para saber cuantas tareas se encuentra en memoria y su estado consultar el comando "tareas".

<b>Comando:</b>	<b><i>tareas</i></b>	<b><i>--Tareas en memoria</i></b>
<b>Sintaxis:</b>	<b><i>t/tareas</i></b>	
<b>Ejemplos:</b>	<b><i>tareas</i></b>	<i># Despliega el estado de</i> <i># las tareas que guardan en</i> <i># memoria.</i>

El comando *tareas* nos despliega en pantalla el estado que guardan todos los tareas activas en memoria.

## COMANDOS DEL "SHELL"

**Comando:** *lista --Listar memoria*

**Sintaxis:** *l/lista [seg][offset]*

**Ejemplos:** *lista 100* # Lista memoria a partir de  
# la dirección 0100h del  
# segmento actual.

*lista 40:100* # Lista memoria a partir de  
# la dirección 40:0100h.

*lista* # Lista memoria a partir de  
# la última dirección  
# listada.

**Comando:** *memoria --Ocupación de la memoria*

**Sintaxis:** *memoria*

**Ejemplos:** *memoria* # Despliega el estado de  
# que guardan la ocupación  
# de memoria en NTRMX.

Despliega el estado de memoria libre y  
ocupada en general y particular para las tareas  
del NTRMX.



## **COMANDOS DEL "SHELL"**

Basura:

- Control de procesos

  - Creación de tareas

  - Temporización y alarmas

- Sincronización y Mensajes

  - Envío y recepción de señales

  - Envío y recepción de mensajes

    - ▲ La identificación de los objetos del sistema: tareas, señales, buzones, y zonas de memoria; será través de números de identificación para hacerlos independientes de su dirección en memoria. Todo esto hace el código fácilmente relocalizable.