

MODULO N° 3**CLASE N° 13****CONCEPTO DE SISTEMA OPERATIVO :**

Un **Sistema Operativo** (en sentido amplio) es un programa que controla y organiza la utilización de los recursos Hardware de una computadora digital. Ahora bien, la anterior definición no muestra con exactitud lo que realmente sucede en la práctica.

Para entender la función que realiza un sistema operativo, es útil recurrir a una sencilla comparación. Un SO es similar al gobierno de un País. Algunos gobiernos son eficientes en el control de un determinado país, de tal manera que su población vive en forma organizada. Algunos otros no poseen la capacidad de gobernar al pueblo y por eso el pueblo dicta sus propias reglas. De la misma manera algunos SO son suficientemente robustos como para controlar todas las funciones de la computadora y permitir que los programas corran organizadamente, mientras que otros dejan librado al criterio de cada programa la utilización, correcta o no, de los recursos de la máquina.

Existen S.Operativos **PROCEDIMENTALES** y **NO PROCEDIMENTALES**. Los Procedimentales, como su nombre lo indica, establecen un procedimiento específico (con comandos u órdenes especiales) para interactuar con la computadora. Los No Procedimentales no utilizan este recurso (comandos), sino que el usuario puede relacionarse con la computadora de una manera mucho más libre, utilizando lenguaje humano. Estos últimos todavía se encuentran en su fase de desarrollo y por eso no son de uso masivo.

SISTEMA OPERATIVO DE DISCO - D.O.S :

Hasta la aparición de la IBM-PC (1980), los SO de las HOME COMPUTERS, como la Commodore por ej., se encontraban escritos en BASIC y grabados en la ROM principal del equipo.

IBM, contrata a una pequeña empresa, por aquellos años, de nombre **MICROSOFT** para diseñar un SO para su primer modelo de PC. Este nuevo SO es más voluminoso que los anteriores y por tanto es necesario no almacenarlo ya en una memoria ROM, sino en un disco flexible. Es por eso que pasa a llamarse (en oposición a su predecesores) "DISK" Operative System (D.O.S.). Así, MS-DOS comenzó siendo vendido por IBM como PC-DOS para acompañar a su flamante modelo de computadora personal.

MS-DOS, fue diseñado como un SO muy compacto, capaz de funcionar bajo condiciones de memoria muy limitada (la cant. standard era 64Kb), dedicándose casi exclusivamente al manejo de archivos en disco y proveyendo escasas funciones en lo que a interfaces de usuario se refiere. En consecuencia, cada programa desarrollado para este sistema provee su propia interfaces de usuario. Para paliar este inconveniente, además de cubrir las falencias en cuanto a manejo de memoria, capacidad de multitasking (multitarea),etc., Microsoft desarrolló el entorno operativo **WINDOWS**.

COMPONENTES DE UN S.O. :

Todo sistema operativo se compone de dos partes fundamentales:

- **KERNELL (Núcleo):** Es el núcleo del SO. Pueden conformarlo varios archivos. Es la parte que más se relaciona con el hardware.

En el caso de MS-DOS está formado por dos archivos, que son: el MSDOS.SYS y el IO.SYS.

El primero Realiza todas las opciones básicas de administración de los elementos hardware del sistema (comunicación entre el micro, la memoria, y los buses). El segundo es un DRIVER (manejador) elemental, que rige y organiza las Entradas y Salidas básicas del sistema (I / O), como la comunicación con consola (teclado y monitor), las unidades de disco e impresora.

- **SHELL (CASCARA):** Es la Interfaces (intermediario) con el Usuario, y por eso la parte más visible del SO . Puede estar formada por un solo archivo o bien consistir en todo un ENTORNO GRAFICO.

Permite utilizar la computadora, bien a través de comandos que se ingresan por la consola, o bien mediante eventos gráficos como la pulsación de un botón o el desplazamiento de un cursor.

En el caso de **MS-DOS**, es el archivo **COMMAND.COM** y también se lo conoce como **INTERPRETE DE COMANDOS**.

TIPOS DE ARCHIVOS DOS:

Si bien es posible que conozcamos ya bastante sobre archivos, es importante tener claro lo siguiente. Existen básicamente 2 (dos) tipos de archivos:

- **Los PROGRAMAS:** Son los archivos **.EXE y .COM**. Este tipo de archivos guardan información que la PC puede EJECUTAR en forma directa, es decir apenas los invocamos.
- **Los Archivos de INFORMACION:** Son archivos “no ejecutables”, que contienen información ya sea para la PC o bien para ser leída en algún idioma cualquiera. Es importante aclarar que los archivos **.BAT**, si bien son archivos de información (de texto), tienen calidad de ejecutables para la PC. Es por eso que pertenecen a un tipo híbrido ya que tienen características de archivos de info. y ejecutables.

PROCESO DE BOOTEO:

El proceso de BOOT, consiste solamente en la carga de los archivos que conforman el SO (I/O.SYS, MSDOS.SYS y COMMAND.COM) en las primeras posiciones de la memoria RAM principal. Esta copia de los archivos de sistema, obviamente se pierde cada vez que se apaga el equipo.

SECUENCIA DE ARRANQUE:

Cuando encendemos la máquina por lo general decimos que la “**BOOTEAMOS**”, aunque ya sabemos que esto es incorrecto. Al encender la PC, en realidad, comienza un “Secuencia de Arranque” que culmina con el verdadero booteo. Esta es como sigue:

- 1°) Al encender el interruptor de POWER, llega la tensión a la Fuente de Alimentación, y ésta demora una pequeña cantidad de tiempo hasta poder entregar las tensiones correctas a la salida (+5v, -5v, +12v y -12v). Cuando esto sucede, la señal de POWER GOOD , el cable naranja, se activa indicando al chip controlador de teclado, que genere una señal de RESET para que el micro vaya a leer la primera instrucción del POST en la memoria ROM-BIOS.
- 2°) Se ejecuta el P.O.S.T , o Autochequeo en el Arranque, y si este encuentra todos los componentes OK, responde con un BEEP por el Speaker (parlante).
- 3°) Se realiza una lectura del B.I.O.S , Sistema Básico de Entradas y Salidas, que “ enseña “ al micro a controlar los periféricos básicos (de I/O) del sistema.
- 4°) Se lee de la memoria CMOS - RAM la “Configuración Física”, es decir por qué periféricos está formado el equipo. En ése momento aparecerá por pantalla un cuadro de configuración.
- 5°) Se da lugar al proceso de BOOTEO, es decir a la carga del SO en memoria RAM principal. En ese momento se exhibe por pantalla un mensaje como: “**Iniciando MS-DOS...**” Luego se leen los archivos de configuración de SO : CONFIG.SYS y AUTOEXEC.BAT, en ese orden. Finalmente se exhibe por pantalla el “PROMPT” o SIMBOLO de SISTEMA.

Con el quinto paso culmina la Secuencia de Arranque, quedando la máquina en estado operativo. Cualquier situación de falla o error en alguno de estos pasos impedirá el arranque.

ARRANQUE FRIO, CALIENTE Y POR SOFT:

Existen 3 (tres) formas de llegar al proceso de Boot. Ellas son:

- **COLD BOOT (Arranque Frío):** Se da cuando encendemos el switch de **POWER** de ordenador. El proceso es el descrito en el ítem anterior. Se le llama **FRIO**, ya que el sistema no está previamente alimentado con tensión.
Existe otra manera de realizar un arranque frío que es ligeramente diferente. Cuando la Fuente de Alimentación ya se encuentra encendida, y la PC funcionando, suele darse alguna situación de error (sistema colgado u otro). Es entonces que el usuario decide recuperar el estado operativo de su PC. Esto se lleva a cabo mediante la pulsación del Switch de **RESET**, el cual fuerza al Controlador de Teclado a enviar la señal de **RESET** al microprocesador ppal. El micro, de ésta manera Secuencia de Arranque con la lectura y ejecución del POST. De ésta manera se evita tener que apagar el ordenador y volver a encenderlo.

WARM BOOT (Arranque Caliente): Se da cuando se mantienen pulsadas las teclas **Ctrl y Alt**, y a continuación la tecla **Del**. De ésta manera se genera (vía soft) una **INTERRUPCION NO ENMASCARABLE, NMI**, que resetea el sistema. El POST se ejecuta, pero no el test de memoria RAM. Este modo de lograr un reset del sistema no siempre funciona ya que a veces la PC se encuentra tan “colgada” que ni siquiera es capaz de recibir datos del teclado.

CLASE N° 14**DIRECCIONES DE MEMORIA (RAMppal)**

Para comprender efectivamente el modo de direccionamiento de la memoria en una PC , debemos conocer como direccionaba la primer PC : la XT (μ P 8088).

Recordemos que el micro 8088 tenía un bus de datos externo de 8 bit, aunque internamente trabajaba a 16 bit. Esto quiere decir que todos sus registros, y sus instrucciones eran de 16 bit. Recordemos también que poseía un Bus de Direcciones de 20 hilos, por lo que se deduce que podía direccionar (encontrar) una cantidad máxima de 2^{20} posiciones de memoria de 8 bit cada una, esto es decir 1Mb.

Ahora bien, consideremos que cada una de esas posiciones de memoria en binario tiene 20 bits, y aquí se presenta el problema. Una dirección de memoria no deja de ser un DATO, que debe ser manejado por el μ P, pero éste solo puede almacenar en sus registros de datos un máximo de 16 bit . Por lo tanto es necesario "traducir" esta toda dirección de memoria de 20 bit a un formato de 16 bit, posible de manejar por el micro.

Direcciones de Memoria y Sistema Hexadecimal:

Cuando hablamos de direcciones de memoria, y en cantidades como 1 millón o más , se hace engorroso representarlas en binario. Imaginemos 20 unos y ceros, y nos daremos cuenta de lo complicado que puede ser leerlo.

Para simplificar el problema, haremos uso del sistema HEXADECIMAL, recordando que :

1 (un) dígito Hexa -----> equivale a -----> 4 (cuatro) dígitos Binarios

Ej:

Hexadecimal (Decimal) Binario

A (10) 1010

(Nota: Esto fue explicado con anterioridad en la Clase N° , de Sistemas Numéricos.)

De esta manera, cuando necesitemos referirnos a 8 bits (en cuanto direcciones de memoria) lo haremos mediante 2 dígitos hexadecimales. Ej:
la posición de memoria 160 (en Decimal) , sería....

en Binario -----> 1010 0000
y en Hexadecimal -----> A 0 h

ya que A h = 1010 y que 0 h = 0000

Paralelamente cuando necesitemos citar una dirección de 16 bits lo haremos mediante 4 dígitos hexa, ya que 4 dígitos hexa = 16 bits

Direcciones DIRECTA y SEGMENTADA:

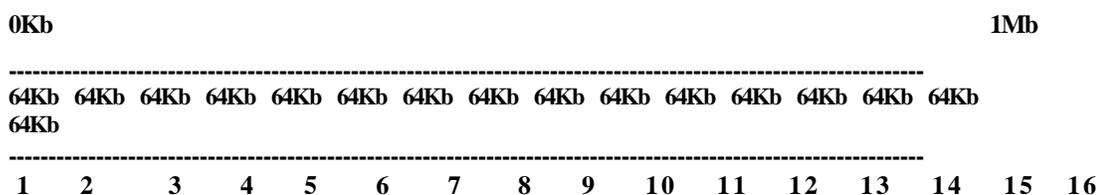
Retomemos ahora el problema central de las direcciones de memoria. Como ya dijimos, éstas, en una PC XT, son de 20 bits, o lo que es lo mismo decir de 5 dígitos hexa (cuatro bit por cada uno). A ésta forma de direccionar se le llama **DIRECTA**.

Ej: Tengamos en cuenta que siempre se cuenta la posición cero.

<u>Decimal</u>	<u>Hexadecimal</u>
dirección 0	00000 h
dirección1023 (1 Kb)	003FF h
dirección65535 (64 Kb)	0FFFF h
dirección655.359 (640 Kb)	9FFFF h
dirección ...1.048.576 (1024 Kb o 1 Mb)	FFFFFF h

El micro no puede trabajar con estas direcciones ya que sus registros pueden almacenar sólo 16 bits (o 4 dígitos hexa).

La solución al problema es utilizar la técnica de MEMORIA SEGMENTADA. Esta consiste en dividir el área total de memoria de 1 Mb en 16 segmentos de 64 Kb (65536 bytes o 2¹⁶), como muestra la siguiente figura:



Cada uno de los 16 segmentos se direcciona mediante la dirección hexa correspondiente a su posición inicial (directa) a la cual se le quita el último dígito, para que quede formada por sólo cuatro dígitos. Veamos un ejemplo:

DIRECCION DIRECTA

COMIENZA _____ TERMINA

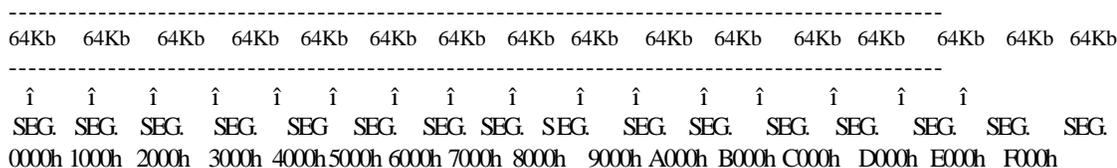
primer segmento de 64 Kb -----> 00000 h 0FFFF h (65535)

DIRECCION SEGMENTADA

COMIENZA

primer segmento de 64 Kb -----> 0000 h

De este modo el espacio total de 1 Mb quedaría dividido en los siguientes SEGMENTOS mostrados en la figura:



Es importante entender que esta Dirección de Segmento no identifica a una, y sólo una, posición de memoria (de 8 bit) sino a un bloque de 64 K . Para direccionar exactamente una posición dentro de ese bloque hace falta agregarle el DESPLAZAMIENTO o OFFSET.

Recordemos que cada segmento posee internamente 65536 posiciones de memoria, numeradas de 0 a 65535. En hexa éstas direcciones se numeran, equivalentemente, desde 0000 h hasta FFFF h

(con 4 dígitos), y sirven para indicar con exactitud, cuál de las 65536 posiciones del segmento es la seleccionada.

De esta manera una dirección SEGMENTADA queda conformada por su segmento y seguida por su desplazamiento, del cual se separa por dos puntos. EJ

<u>DIRECTA</u>	<u>SEGMENTADA</u>
dirección N° 20 de memoria ----> 000C8 h ram.	segmento desplazam. 0000 : 0200 (dos puntos)

Si en una dirección segmentada sumamos segmento y offset, en forma “desplazada” (corrida), recuperaremos la dirección directa original. Siguiendo el caso del ejemplo anterior, esto sería.....

segmento	0 0 0 0
+	+
<u>desplazamiento</u>	<u>0 2 0 0</u>
dir. Directa	0 0 2 0 0 h

División de la Memoria , desde un punto de vista hardware:

Desde un punto de vista físico, el primer Mb de memoria de la PC se encuentra dividido en dos partes : MEMORIA BASE (de 0 kb a 640 kb) y la MEMORIA RESERVADA (de 640 a 1 Mb)

- **Memoria BASE:** Es también llamada BAJA, CONVENCIONAL y DE USUARIO.

Consta de 10 segmentos de 64Kb. Sus direcciones de inicio y de fin son 0000:0000 y 9000:f f f f respectivamente. El siguiente gráfico muestra en detalle las dir. de inicio y fin de cada segmento:

	Primer segmento de 64 Kb	inicio ----> 0000:0000	----> 0 Kb	inicio de mem. base
		fin ----> 0000 : FFFF	----> 64 Kb	
M	-----			
	Segundo segmento de 64 Kb	inicio ----> 1000 : 0000		
E		fin ----> 1000 : FFFF	--> 128 Kb	
M	-----			
	Tercer segmento de 64 Kb	inicio ----> 2000 : 0000		
0		fin ----> 2000 : FFFF	--> 192 Kb	
R	-----			
	Cuarto segmento de 64 Kb	inicio ----> 3000 : 0000		
I		fin ----> 3000 : FFFF	----> 256 Kb	
A	-----			

	-----	inicio ----> 4000 : 0000	
	Quinto segmento de 64 Kb	fin ----> 4000 : FFFF	----> 320 Kb
B	-----	inicio ----> 5000 : 0000	
A	Sexto segmento de 64 Kb	fin ----> 5000 : FFFF	----> 384 Kb
S	-----	inicio ----> 6000 : 0000	
E	Séptimo segmento de 64 Kb	fin ----> 6000 : FFFF	----> 448 Kb
	-----	inicio ----> 7000 : 0000	
	Octavo segmento de 64 Kb	fin ----> 7000 : FFFF	----> 512 Kb
	-----	inicio ----> 8000 : 0000	
	Noveno segmento de 64 Kb	in ----> 8000 : FFFF	----> 576 Kb
	-----	inicio ----> 9000 : 0000	
	Décimo segmento de 64 Kb	fin ----> 9000:FFFF	----> 640 Kb fin de mem. base.

La memoria Base , principalmente se usa para dos funciones fundamentales : *Cargar el Sistema Operativo (además sus drivers)* y *Ejecutar los programas.*

La memoria RESERVADA, no se utiliza para ejecutar programas, sino que se encuentra “reservada” para copiar el BIOS de Sistema, el BIOS de la placa de Video y cualquier otro BIOS de alguna placa interfase que en el futuro se agregue al sistema.

Memoria Extendida:

Se denomina memoria Extendida a la memoria que comienza más allá de 1 Mb y que termina en el máximo expandible para una determinada motherboard (es decir hasta el máximo que se puede poner en los bancos de mem.)

Tomemos por ejemplo una placa con 4Mb, expandibles hasta 64 Mb. Esta tiene 3Mb de memoria extendida, y podría llegar a tener un máximo de 63 Mb de mem. Extendida.

Esta posibilidad de poner memoria en cantidad mayor a un Mb, surgió con el 286 gracias a su bus de direcc. de 20 hilos, con el que podía direccionar un maximo de 16Mb.

Memoria Expandida:

Se denomina memoria Expandida a la memoria que se agregaba montada sobre una placa interfase en un slot de expansión (de ahí su nombre). Este tipo de memoria, tal como se diseñó en un principio, ya no se utiliza ni se fabrica más.

Se trataba de una placa de memoria muy cara, y de muy baja performance ya que el sistema trabajaba con ella a la velocidad del bus ISA (8 MHz).

Manejo de Memoria por parte del D.O.S:

El SO MS-DOS fue diseñado originalmente para trabajar con el límite máximo de 1Mb de memoria ram ppal. De esa cantidad de memoria sólo los primeros 640 Kb eran posibles de usar para la ejecución de programas (que para la época se consideraba una cantidad inagotable).

Ya sabemos que consta de 10 segmentos de 64 KB y que sus direcciones límites son:

0000:0000 hasta 9000:FFFF .

En la memoria BASE, en el momento del Boot se cargan en orden...

Los Vectores de Interrupción

El EBDA (Extended Bios Data Area) o Stack del Bios

Los Archivos de Sistema

El Entorno (Enviroment)

Los Drivers

Los Programas Residentes o TSR (Terminate & Stay Resident)

Como resultado de la carga de todos estos programas, se reduce drásticamente el espacio de memoria Base. Por tanto los programas escritos para utilizar este tipo de memoria (y no otra) suelen verse impedidos de correr, presentando típicamente el mensaje: “ NOT ENOUGH MEMORY “ (memoria insuficiente).

Esta forma de utilización de la memoria hasta 1 MB, donde los programas corren de a uno , tomando y liberando sucesivamente el espacio de memoria libre hasta los 640 Kb se denomina : MODO REAL.

MEMORIA SUPERIOR (UPPER) y SHADOW - RAM:

La SHADOW RAM ,es la copia del BIOS , que reside en una lenta memoria ROM (500 ns de acceso), en la rápida memoria RAM (60 ns de acceso), para levantar la performance del Sistema. Recordemos que el BIOS es leído permanentemente mientras se utiliza la máquina, así que será de gran utilidad poder acceder a leerlo en forma rápida mediante éste sistema.

Hay un área espacial de memoria RAM ppal destinada a tal efecto:La Memoria RESERVADA o SUPERIOR. Esta se encuentra entre los 640 Kb y 1 MB, y mide 384 KB (6 bloques de 64 Kb).

Hay fundamentalmente dos BIOS que son necesarios copiar a RAM: el BIOS de SISTEMA (de la rom de equipo) y el BIOS de VIDEO (de la placa de video). La opción de copiarlos o no , se maneja desde el SETUP . Desde allí se nos permite copiar uno, los dos o ninguno de ellos activando el área correcta de la mem. Reservada o Superior .

Estos son los segmentos que forman la memoria Reservada y su utilización como lugar de almacenamiento de posibles copias SHADOW (sombra).

Segmento	Medida	Destino
inicio A000:0000		
A000	64 Kb	Se utiliza para la creación de gráficos Interactúa con la Placa de Video (buffers de video).
fin A000:FFFF		
inicio B000:0000		
B000	64 Kb	Se utiliza para la creación de gráficos. Interactúa con la Placa de Video (buffers de video).
fin B000:FFFF		
inicio C000:0000		
C000	32 Kb	Reservado para el Firmware del Video BIOS.
fin C000:FFFF	32 Kb	Reservado para el BIOS de placas propietarias
inicio D000:0000		
D000	64 Kb	Reservado para el Firmware de placas propietarias.
fin D000:FFFF		
inicio E000:0000		
E000	64 Kb	Reservado para el BIOS de Placas Propietarias.
fin: E000:FFFF		
inicio F000:0000		
F000	64 Kb	Reservado para el BIOS de Sistema.
fin F000:FFFF		

Las áreas no utilizadas para copiar shadows, no se pueden utilizar por programas diseñados para utilizar memoria BASE.

Memoria Extendida (LIM - XMS Lotus Intel Microsoft Extended Memory Specification):

Ya sabemos que la memoria Extendida es aquella que se encuentra por encima del límite de 1 Mb . Esta memoria es inaccesible para el SO DOS (no así para el procesador) por no poder manejar direcciones de más de 16 bit (en forma segmentada).

De ésta manera, la mem. extendida es un dispositivo no controlable desde BIOS y por lo tanto será necesario para manejarla un DRIVER (un programa de control).

El DRIVER “ HIMEM.SYS “ , de MS-DOS, es un XMM (Extended Memory Manager o Administrador de Memoria Extendida). Cumple básicamente dos funciones:

- Permite reconocer y trabajar con memoria Extendida. Esto lo hace lo hace dividiendo la mem. ext. en páginas (bloques) de 64 Kb y generando uno o más Marcos de Página ubicados en la memoria SUPERIOR, donde sí puede ser direccionados (o localizados) por el sistema operativo. En los mencionados marcos de página se copian de a una por vez las diversas páginas, para su uso. Este driver debe cargarse en el archivo de configuración CONFIG.SYS, mediante el comando DEVICE y debe situarse en la primera línea.Ej:

DEVICE=C:\HIMEM.SYS

- Permite reconocer y activar los primeros 64 Kb de memoria extendida llamados HMA (High Memory Area) para ejecutar un programa.
Esta área es independiente del resto de la memoria extendida y sólo sirve para ejecutar UN programa (de hasta 64 Kb). Generalmente se utiliza para cargar el SO (o sea los archivos de sistema), liberando así memoria CONVENCIONAL.

Para cargar el SO en la memoria alta, debemos agregar (después de de haber cargado HIMEM.SYS) el comando:

DOS=HIGH

Cabe aclarar que la mayoría de los programas actuales utilizan memoria EXTENDIDA (XMS) y por tanto necesitan de la carga de este driver.

También es importante destacar que no se puede instalar este driver en una máquina que posea solo 1 mb de mem RAM instalado.

CLASE N° 15

**COMENTARIOS ACERCA DE LA INSTALACION DEL SISTEMA OPERATIVO
Y DE WINDOWS 3.1 O 3.11:**

Al instalar una nueva versión de SO en una PC, deberíamos tener en cuenta algunos requisitos:

- 1° - Instalar versiones ORIGINALES o copias exactas de originales.**
- 2° - Leer las Instrucciones y requisitos previos a la instalación.**
- 3° - Hacer una copia de Resguardo de nuestros archivos.**
- 4° - Guardar una copia de los Archivos de Configuración de nuestro SO y de WIN (IO.SYS, MSDOS.SYS, COMMAND.COM, WIN.INI y SYSTEM.INI .**
- 5° - Verificar luego si corren todas las Aplicaciones preexistentes.**
- 6° - Revisar y optimizar la nueva configuración del SO o de Win.**
- 7° - Si todo funciona en forma correcta, guardar una copia de los nuevos archivos de configuración en un directorio creado a tal efecto.**
- 8° - Conservar una copia de los diskettes de instalación de (sobre todo WINDOWS) , ya que ante cualquier modificación serán necesarios.**

Es importante entender que instalar el SO o Win. en una máquina por medio de una copia directa de los archivos de otra, no es un procedimiento correcto. De esta manera, siempre observaremos problemas ya que tendremos elementos de más y de menos, pero no ajustados a la necesidad de nuestra PC.

CLASE N° 16**CONFIGURACION DEL SISTEMA OPERATIVO:****Archivo CONFIG.SYS:**

El archivo config.sys, se utiliza básicamente para cargar DRIVERS en memoria. Recordemos que los drivers son rutinas de control de dispositivos físicos que al no estar presentes en el BIOS, es necesario cargarlas vía soft.

Otra función del archivo config.sys es definir ciertos parámetros del sistema. Ellos son:

FILES : Define el número de archivos que DOS puede tener abiertos a la vez. Utiliza una pequeña cantidad de mem. (53 bytes) por cada archivo abierto.

BUFFERS: Genera buffers de lectura y escritura para dispositivos de disco.

FCBS: Es un viejo sistema de control de archivos abiertos por DOS. Sólo se utiliza por motivos de compatibilidad, con programas viejos.

TACKS: Genera pilas de almacenamiento de direcciones de inicio de rutinas de interrupción.

LASTDRIVE: Define la última letra de unidad a utilizar por el SO. Utiliza 80 bytes por cada letra no definida y no utilizada.

SHELL : Define la ubicación y el nombre del intérprete de comandos.

Archivo AUTOEXEC.BAT

Este archivo, que reside en la raíz del disco de arranque, define acciones automáticas en el momento del arranque.

El archivo autoexec.bat, cumple dos funciones. Como todo archivo BATCH, hace una llamada a ejecutables. Los archivos RESIDENTES (TSR) se encuentran entre ellos. Define ciertos parámetros del sistema mediante comandos específicos. Ellos son:

PROMPT: Cambia el indicador de la línea de comandos de DOS. No conviene hacerlo muy grande, ya que es una de las variables de entorno (environment).

PATH: Señala rutas alternativas para la búsqueda de archivos. Es una de las variables de ENTORNO. No debe ser extenso.

SET: Define “variables” o etiquetas para su uso por los programas. No conviene tampoco tener muchas variables ya que aumenta el espacio utilizado por el entorno (environment).

COMSPEC: Especifica (spec)cuál es el Intérprete de Comandos (com). Estas variables de entorno, deben ser respetadas por todos los programas que se cargan en la memoria. Debido a que estos usualmente las desplazan del espacio de mem, es necesario que mantengan una copia para poder devolverlas en el momento en que finalizan su tarea. Si el espacio de ENVIRONMENT (entorno) es muy grande, también lo será su copia, y se reducirá el espacio de mem convencional para la ejecución de programas.

Omisión de Archivos de Configuración:

A efectos de modificar la secuencia de Booteo, omitiendo los archivos de arranque, MS-DOS prevee dos métodos:

1.- Presionar la tecla F5 en el momento exacto en que comienza el BOOTEEO. Esto sucede cuando aparece el mensaje “Iniciando MS-DOS.....”. Por pantalla aparecerá un mensaje indicando la omisión de Config y Autoexec.

2.- Presionando la tecla SHIFT de la izquierda cuando aparece el mensaje “Iniciando MS - DOS.....”. Esta opción es más rápida que la anterior y no emite mensaje alguno por pantalla.

Existe también la opción de omitir sólo alguna parte (o línea) de los arch. de configuración.

Presionando la tecla F8 en el momento en que aparece el mensaje “Iniciando MS-DOS”, se nos da la posibilidad de seleccionar una a una las líneas del Config.sys y el Autoexec.bat, que deseamos cargar . Por pantalla aparecerá un mensaje indicando ésta opción.

CLASE N° 17**TRABAJO PRACTICO N° 3:****TEMA: CREACION DE AUTOEXEC Y CONFIG CON “COPY CON”****MODO REAL, PROTEGIDO Y VIRTUAL 386.**

“ Es necesario disponer para realizar este trabajo práctico con máquinas PC 386, o superior , con 2MB de RAM o más “.

1.- Modo Real.

a.- Desde el prompt, Cree un archivo de nombre config.sys, con los siguientes parámetros:

```
files = 10
bufferss = 10
device = ansi.sys
```

Grabe las modificaciones.

b.- Desde el prompt, cree un archivo de nombre autoexec.bat, con los siguientes parámetros:

```
@echo off
prompt $p$g
path C : \ DOS
```

Grabe las modificaciones.

c.- Arranque la máquina y verifique la cantidad de memoria disponible de cada tipo mediante el comando: mem /c/p .

d.- Anote aquí, por tipo, las cantidades de memoria que encuentre:

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

2.- Modo Protegido.

a.- Desde el prompt cree un archivo de nombre config.sys, con los siguientes parámetros:

```
device = HIMEM.SYS /v
dos = high
files = 10
bufferss = 10
device = C : \ dos \ ansi.sys
```

Grabe las modificaciones.

b.- Desde el prompt, cree un archivo de nombre autoexec.bat, con los siguientes parámetros:

```
@echo off
prompt $p$g
path C : \ DOS
doskey
```

Grabe las modificaciones.

c.- Arranque la máquina y verifique la cantidad de memoria disponible de cada tipo mediante el comando: `mem /c/p` .

d.- Anote aquí ,organizadas por tipo, las cantidades de memoria que encuentre:

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

3.- Modo Virtual 386.

a.- Desde el prompt cree un archivo de nombre `config.sys`, con los siguientes parámetros:

```
device = HIMEM.SYS /v
device = EMM386.EXE NOEMS /v
dos = high, umb
files = 10
bufferss = 10
devicehigh = C : \ dos \ ansi.sys
```

Grabe las modificaciones.

b.- Desde el prompt, cree un archivo de nombre `autoexec.bat`, con los siguientes parámetros:

```
@echo off
prompt $p$g
path C : \ DOS
loadhigh doskey
```

Grabe las modificaciones.

c.- Arranque la máquina y verifique la cantidad de memoria disponible de cada tipo mediante el comando: `mem /c/p` .

d.- Anote aquí, organizadas por tipo, las cantidades de memoria que encuentre:

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

e.- Entre a Windows-Principal-Panel de Control y active el Archivo de Intercambio `386spart.par` en el ícono de Modo Virtual del386 , de modo que éste archivo (memoria virtual) sea permanente en la raíz del disco C:

3.- Utilización de la mem EMS (expandida)

a.- Agregue al `config.sys` del caso anterior el siguiente parámetro:

```
device = EMM386.EXE 1024 RAM /v
```

Grabe los cambios.

c.- Arranque la máquina y verifique la cantidad de memoria disponible de cada tipo mediante el comando: `mem /c/p` .

d.- Anote aquí, organizadas por tipo, las cantidades de memoria que encuentre:

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

e.- Luego compare los totales de memoria obtenidos en los casos anteriores.

CLASE N° 18**CONFIGURACIONES MULTIPLES**

El software actual es cada vez más exigente con el hardware, y esa exigencia consiste habitualmente en una configuración de SO particular para cada caso.

A partir de la versión 6.0 de MS-DOS, se incluyen comandos especiales para generar múltiples juegos de Config.sys y Autoexec.bat que trabajen juntos. Estos juegos de archivos de configuración se seleccionan en el arranque por medio de un menú.

ARCHIVO CONFIG.SYS

Para lograr un config múltiple, dentro de un único archivo config.sys, se crean distintos bloques de configuración mediante comandos espaciales creados a tal efecto. Ellos son:

[MENU] : Encabeza el bloque de Items del Menu de Inicio.

[MENUITEM]: Define cada item en particular del Menú de Inicio.

[MENUDEFAULT]: Define cuál será la opción del Menú de Inicio, que será tomada por defecto.

[MENUCOLOR]: Fija el color del texto y del fondo para el Menú de Inicio.

Tomemos por ejemplo:

[MENU]

MENUITEM=OPCION1, Modo Real

MENUITEM=OPCION2, Modo Protegido sin EMS

MENUITEM=OPCION3, Modo Protegido con EMS

[OPCION1]

FILES=15

BUFFERS=15

[OPCION2]

DEVICE=C:\DOS\HIMEM.SYS /V

DEVICE=C:\DOS\EMM386.EXE NOEMS

DOS=HIGH, UMB

FILES=25

BUFFERS=25

[OPCION3]

DEVICE=C:\DOS\HIMEM.SYS /V

DEVICE=C:\DOS\EMM386.EXE 2048 RAM

DOS=HIGH, UMB

FILES=30

BUFFERS=30

[COMMON]

SHELL=C:\DOS\COMMAND.COM /P

ARCHIVO AUTOEXEC.BAT

Si se desea tener una configuración distinta para cada AUTOEXEC.BAT, se debe tener en cuenta que la opción elegida en el config queda asignada a una variable de entorno llamada CONFIG, y por tanto puede ser aprovechada de la siguiente manera:

```
@ECHO OFF
PROMPT $P$G -----> parte común
PATH C:\;C:\DOS
GOTO %CONFIG%

:OPCION1
GOTO FIN

:OPCION1
GOTO FIN

:OPCION1
GOTO FIN

:FIN
..... -----> parte común
```

CLASE N° 19

MEMMAKER de MS DOS :

Memmaker.exe es un gestor u optimizador de memoria de MS-DOS. Un optimizador de memoria es un programa que organiza los archivos de configuración (config y autoexec) de una forma “óptima” en cuanto a carga de residentes y drivers.

No debemos creer que este programa realiza una optimización general para nuestra PC. Esto quiere decir que este programa no modifica los parámetros propios del config y el autoexec, tarea que queda a cargo del usuario.

MemMaker, básicamente, que optimiza la memoria de su PC moviendo los controladores de dispositivos y los programas residentes a la memoria Convencional a la memoria Superior. Se debe disponer de un procesador 386, o superior, y de memoria Extendida para poder usar MemMaker.

Para correr este programa , debemos invocarlo desde el prompt, y no desde algún administrador de archivos (como Norton Comander). Tampoco se debe utilizar mientras se esté ejecutando Windows.

MEMORIA VIRTUAL y SWAP FILE 386SPART.PAR en WINDOWS:

Recordemos que en Modo Virtual 386 , a cada programa que corre en multitarea se le asigna un espacio de memoria de 1Mb de manera virtual.

Cuando no se dispone de una gran cantidad de memoria RAM, es necesario para correr en modo virtual del 386, crear un Archivo de Intercambio (Swap File).

Este archivo se llama 386spart.par ,es oculto y almacena las áreas que ocupan las aplicaciones inactivas en multitarea. De esta manera proporciona memoria “virtual” (no real).

El swapping con disco, consume demasiado tiempo de acceso y transferencia, por eso debe ser utilizado sólo cuando sea necesario.

La forma de activar la Memoria Virtual , es desde el grupo PRINCIPAL, Panel de Control, ícono de 386 Enhanced (Extendido).

El archivo puede ser Permanente o Temporal. Si es temporal, ocupa el 50% del espacio libre de disco, y se corre el peligro de que al no verlo nunca (por ser temporal) se vaya ocupando paulatinamente casi todo el espacio de disco , quedando insuficiente espacio libre para su creación.

Si el swap file es permanente, nos aseguramos de que siempre se cuente con espacio para su creación y funcionamiento.

En cuanto a su tamaño, debe tener una relación directa con la cantidad de aplicaciones que se abrirán al mismo tiempo. De todos modos lo aconsejable es aceptar un tamaño entre el máximo y el mínimo sugerido por Windows .

OTROS OPTIMIZADORES OEM:

Existen también algunos productos que aumentan y extienden la capacidad de gestión de memoria que ofrece la versión 6.22 del MS-DOS. Los líderes en el campo de gestión de memoria son: QEMM386 de Quarterdeck Office Systems , 386MAX de Qualitas y Memory Comander de V Communications .

En general ofrecen las siguientes características:

- **Un Driver de mem. XMS (extendida).**
- **Un Driver de mem. EMS (expandida).**
- **Drivers de mem. EMS para micros 8086/88 y 80286.**
- **Capacidad de cargar Drivers y TSR en UMB's.**
- **Capacidad de convertir los Buffers de mem. (segmentos A000 y B000) en mem. Convencional.**

CONSIDERACIONES ACERCA DE LA INSTALACION AUTOMATICA DE PROGRAMAS:

Es importante llevar un control acerca de las modificaciones que los programas de aplicación introducen en los archivos de configuración de SO.

Tengamos en cuenta que generalmente todos los programas que generen cambios en Config.sys y Autoexec.bat, crearán además una copia de los originales con la extensión cambiada.

Los programas antiguos, generalmente no informan acerca de todas las modificaciones que realizan, tanto en los archivos de configuración, como en el disco rígido. Esto hace que sea difícil eliminar el programa, volviendo a la situación previa a su instalación.

Los programas más nuevos, en cambio, vienen provistos de utilidades de desinstalación (uninstall.exe) e informe completo de modificaciones (Install.log), por lo que la tarea se simplifica notablemente.

En función de prevenir problemas de configuración, y poder revertir siempre cualquier cambio en ella, es de suma utilidad guardar los archivos de configuración del SO y de Windows (Win.ini y system.ini) en un directorio a tal efecto en un lugar seguro del disco rígido.

CLASE N° 20**TRABAJO PRACTICO N° 4:****TEMA: OPTIMIZACION DE MEMORIA Y CONFIGURACIONES MULTIPLES.**

“ Es necesario disponer para realizar este trabajo práctico con máquinas PC 386, o superior , con 2MB de RAM o más “.

1.- OPTIMIZACION DE MEMORIA MEDIANTE MEMMAKER DE MS -DOS

a.- Cree el archivo CONFIG.SYS , de modo que la PC puede trabajar en modo PROTEGIDO. Para ello será necesario cargar:

..... ?

Cargue además los dos siguientes Drivers:

ANSI.SYS (controlador para consola)
SETVER.EXE (compatibilizador de versiones)

b.- Cree el archivo AUTOEXEC.BAT , de modo que se carguen los dos siguientes programas residentes:

DOSKEY (histórico de teclado de MS -DOS)
GMOUSE (interfase soft para mouse)

c.- Resetee la PC y luego ejecute , desde el prompt el comando mem /c/p . Después anote aquí la cantidad de memoria disponible por tipos.

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

d.- Ejecute MEMMAKER (de MS-DOS) .

e.- Compare los cambios entre los nuevos y los anteriores archivos de arranque.

Recuerde que los archivos modificados se guardarán con la extensión .UMB en el directorio DOS

2.- CREACION DE CONFIG Y AUTOEXEC MULTIPLES

a.- Desde el prompt, cree, mediante un editor ASCII cualquiera, un archivo Config.sys múltiple para arrancar en los siguientes modos:

Modo REAL: No debiendo cargar ningún controlador de memoria XMS o EMS.

Modo PROTEGIDO: Debiendo cargar el controlador de Memoria Expandida (XMS) HIMEM.SYS. También deberemos cargar el SO en memoria ALTA(*HMA*).

Modo VIRTUAL 386: Debiendo cargar ambos controladores, de Memoria Expandida "HIMEM.SYS" (XMS) y el de Memoria Expandida (EMS) EMM386.EXE.

Grabe las modificaciones.

b.- Desde el prompt, cree un archivo autoexec.bat múltiple para combinar con el múltiple config.sys. Este deberá cargar para...

Modo REAL: *DOSKEY* (*histórico de teclado de MS-DOS*)

Modo PROTEGIDO: *SMARTDRV.EXE* (*caché de disco*)

Modo VIRTUAL 386: *SMARTDRV.EXE* (*caché de disco*)
GMOUSE (*interfase soft para mouse*)

Coloque comandos comunes a los tres autoexec, tanto al inicio como al final, y luego grabe las modificaciones.

c.- Arranque la máquina con cada una de las tres opciones y verifique la cantidad de memoria disponible de cada tipo mediante el comando: `mem /c/p` desde el prompt. Anote luego las cantidades de memoria que encuentre en cada opción de arranque en las siguientes tablas:

En MODO REAL.....

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

En MODO PROTEGIDO

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :

En MODO VIRTUAL 386

	TOTAL	UTILIZADA	LIBRE
Convencional :
Superior :
Extendida :
Expandida :