

CURSO DE INFORMÁTICA



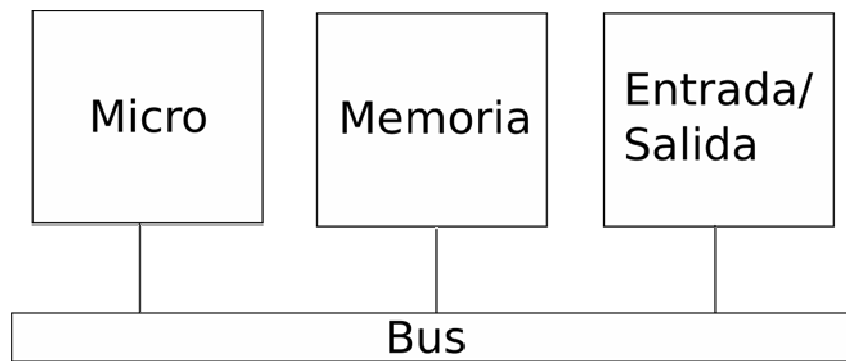
**Unidades Penales
Provincia de Buenos Aires
2009**

Konrad Bucheli

Curso Informática

de Konrad Bucheli

Modelo de una computadora



El elemento más importante de una computadora es el micro. Se llama también procesador o micro procesador. En inglés se usa processor, micro-processor o la abreviación CPU (Central Processing Unit: unidad central de procesamiento). Es la cabeza de la computadora y hace todos los cálculos.

La memoria contiene los datos y las instrucciones (programa) que el micro procesa.

Sobre el I/O (abreviación inglés para Input/Output: entrada/salida) la computadora habla con el ambiente. Todos los periféricos, los discos, audio, vídeo, comunicación, etc. se resumen en el término I/O.

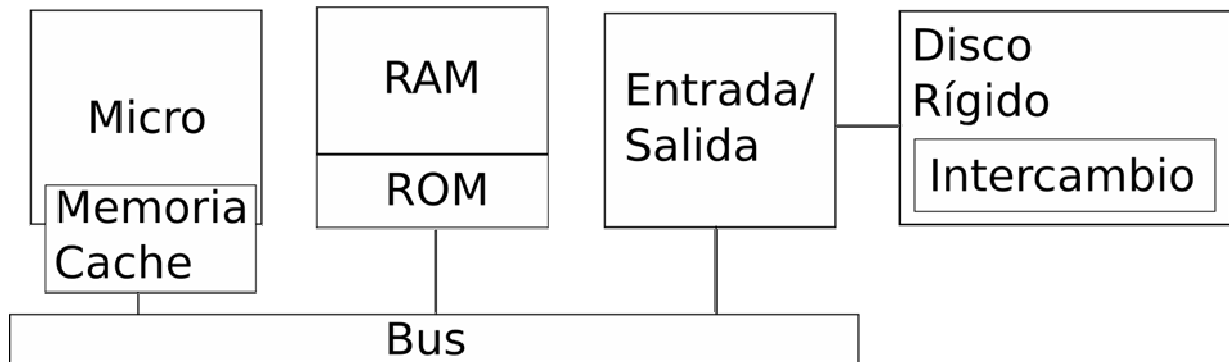
La conexión entre todas esas partes se llama **bus**. El micro mediante el bus comunica con la memoria y con el I/O. En el PC, se usa también la abreviación inglés FSB (Front Side Bus: Bus de la parte frontal).

Un programa de ejemplo

Para hacer un cálculo, por ejemplo $3+5$, una computadora hace así:

Primero el micro lee una instrucción de la memoria. Esta instrucción dice que lo tiene que leer la memoria a la dirección X que contiene el 3. La próxima instrucción comanda al micro leer la memoria a la dirección Y que contiene el 5. La tercera instrucción es sumar estos dos datos y la última es escribir el resultado a la memoria a la posición Z.

Diferentes Maneras de Memoria



La memoria tiene dos partes: el RAM y el ROM. Normalmente el micro trabaja con el RAM (ingles: Random Access Memory: memoria de acceso arbitrario). Él puede leer y escribir en el RAM. El RAM es como la memoria a corto plazo: se borra cuando apagas la corriente. Al contrario, en el ROM (ingles: Read Only Memory: memoria solo para leer) queda el contenido y el micro solo puede leerlo. En el ROM se encuentra el código para inicializar y arrancar la computadora. En el PC se llama BIOS (ingles para Basic Input/Output System: sistema básico de entrada/salida).

Porque el RAM es más lento que el micro hay una pequeña pero rápida memoria que se llama memoria cache. El micro carga la memoria cache con datos del RAM, trabaja con estos datos y cuando está listo, escribe los cambios de vuelta al RAM. En el Pentium 1 la memoria cache está en el mother, en el Pentium 2 en el módulo al lado del micro y en micros más nuevos está dentro del chip.

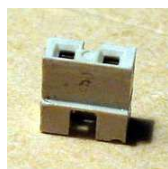
Si los programas necesitan más memoria que el tamaño del RAM, el sistema operativo externaliza datos al espacio de intercambio en el disco rígido. Cuando un programa necesita estos datos el sistema operativo los descarga del disco a la memoria. En Windows este espacio de intercambio es un archivo con el nombre pagefile.sys. En Linux se usa una partición particular para eso, la partición de intercambio (swap partition en ingles).

Configuración de Mother o Placa Base

Qué configuraciones una mother tiene y con cuales micros y plaqueta de memoria funciona es una cosa muy individual y diferente de mother a mother. El manual de la mother contiene esas informaciones. Muchas veces se puede bajar el manual de la página de Web de la empresa que vende la mother. Pero casi siempre esos manuales **son** en ingles.

A veces la evolución técnica hace que puedas poner una parte más nueva en una mother más vieja, pero no anda. Por ejemplo, un Pentium 1 MMX tiene el mismo zócalo que un Pentium 1 normal, pero necesita una mother más nueva que conoce las necesidades del modelo más nuevo.

Para hacer la configuración del mother hay jumpers o interruptores DIP.



Jumper §



Interruptores DIP

Casi cada mother hasta ahora tiene una configuración que se llama “CMOS Clear” o “CMOS CLR” o “Clear Setup” o “Clear Config” o algo así. Eso sirve para borrar el setup del BIOS. Eso también desactiva una contraseña del BIOS. Para hacer eso, apagar la computadora, poner el jumper o el interruptor en “Clear” u “On”. Después encender por unos segundos y apagar otra vez. Ahora poner

el jumper o el interruptor a “Normal” u “Off”.



Las mother modernas muchas veces ya no tienen jumpers para configurar. Todo es automático. Pero en el Pentium 1, Pentium 2 y Pentium 3 tienes que configurar la frecuencia del bus (FSB). Es importante poner la frecuencia que el micro requiere. En el ejemplo a la izquierda el micro de Pentium 3 dice: “900/256/100/1.7V”. Eso significa que el micro tiene una frecuencia de 900 MHz y 256 kB de memoria cache. Después requiere una frecuencia del bus de 100 MHz y una tensión de 1.7V.

En el Pentium 1 es necesario configurar la frecuencia del micro también. Eso pasa con un multiplicador (en inglés ratio, multiplier o factor). La frecuencia del bus se multiplica con este multiplicador y el resultado es la frecuencia del bus. Unos ejemplos:

$$66 \text{ MHz} \times 1.5 = 100 \text{ MHz}$$

$$60 \text{ MHz} \times 2 = 120 \text{ MHz}$$

$$60 \text{ MHz} \times 2.5 = 150 \text{ MHz}$$

$$66 \text{ MHz} \times 3 = 200 \text{ MHz}$$

¿Que pasa si la frecuencia del micro es más baja de lo que la especificación dice? Todo anda bien, solo el micro es más lento. ¿Y qué pasa si la frecuencia del micro es más alta? Eso no se sabe: puede funcionar bien, puede funcionar cuando no haga calor, puede funcionar sólo cuando no hay mucho trabajo para el micro, posiblemente no funciona o se funde el chip del micro.

En los Pentium 1 (especialmente por el Pentium MMX) tienes que configurar la tensión también. Cuando la tensión es demasiado baja, no anda o anda inestable. Cuando la tensión es demasiado alta quema el micro.

Monitor

En una pantalla CRT (gruesa) se ve sólo un punto a la vez. La pantalla pinta la imagen muy rápido así el ojo humano lento ve una imagen. La televisión muestra 50 imágenes por segundo (frecuencia de imágenes de 50 Hz). Eso sirve cuando se mira de lejos y no es necesario concentrarse en detalles. De cerca puedes ver un centelleo. Cuando hay más de 70 Hz ya no ves el centelleo. Pero el monitor tiene un límite de frecuencia. Este límite depende de la resolución de la imagen. Con una resolución grande el límite de la frecuencia de imágenes es más bajo y con una resolución baja, el límite es más alto. El valor del límite también depende del modelo.

En las pantallas LCD (delgadas) esto es diferente. Estas nunca tienen un centelleo porque los muestra todo el tiempo en la imagen. Funcionan con una frecuencia de 60 Hz y normalmente no funcionan con una frecuencia más alta.

Representación de Información

Bit

La base de la informática es, como el nombre dice, la información, o más precisamente la representación y el procesamiento de la información. La representación de la información más básica es el bit. El bit tiene dos estados: 0 y 1. La representación física del bit puede ser de

diferentes maneras: hay tensión o no hay tensión (como en el RAM o en el micro), hay luz o no hay luz, la dirección de un campo magnético (como en un disco rígido), etc.

Se puede hacer cálculos con bits: con un bit o con dos bits:

Negación

La puerta lógica NO (NOT en inglés) realiza la función booleana de inversión o negación de un bit.

Entrada	Salida
0	1
1	0

Y

La puerta lógica Y, más conocida por su nombre en inglés AND, realiza la función booleana de producto lógico: el resultado es 1 cuando todas las entradas son 1:

Entrada 1	Entrada 2	Salida
0	0	0
0	1	0
1	0	0
1	1	1

O

La puerta lógica O, más conocida por su nombre en inglés OR, realiza la operación de suma lógica: el resultado es 1 si hay un 1 en las entradas.

Entrada 1	Entrada 2	Salida
0	0	0
0	1	1
1	0	1
1	1	1

Estas operaciones o puertas lógicas son el fundamento de la computación. Con ellos puedes construir todas las funciones. El micro trabaja sólo con estas puertas lógicas y contiene millones de ellas.

Byte

Porque el bit es muy pequeño, en computación se usa normalmente el byte, que se compone de ocho bits.

1 byte	1 B = 8 bit
1 kilobyte	1 kB = 1024 B = 2^{10} Byte
1 megabyte	1 MB = 1024 kB = 2^{20} Byte

1 gigabyte	1 GB = 1024 MB = 2^{30} Byte
1 terabyte	1 TB = 1024 GB = 2^{40} Byte
1 petabyte	1 PB = 1024 TB = 2^{50} Byte
1 exabyte	1 EB = 1024 PB = 2^{60} Byte
1 zetabyte	1 ZB = 1024 EB = 2^{70} Byte

Para tamaños grandes se usa el mismo prefijo que se usa en las unidades físicas, pero el factor no es 1000, es 1024 que es 2 elevado a 10.

Atención: en otras partes de la informática (frecuencias, ancho de banda) se usa el factor 1000 (y no el factor 1024)

Ejemplos de representaciones

Arriba se describe como se puede representar la información en una secuencia de bits o bytes. Pero eso no dice nada sobre la significación de la información. Para interpretar la información tenemos que conocer la convención que se usa para entender esta información. En los archivos, para decidir de qué formato (de qué convención) se trata, se usa la última parte del nombre que viene después del último punto. Por ejemplo .doc son documentos de Word, .jpg son imágenes, .html son páginas de web, etc.

Atención: normalmente Windows no lo muestra. Este comportamiento se puede configurar.

Ahora seguimos con unas convenciones básicas.

Números

Para representar números, se usa el sistema binario que tiene el numero dos por base y por eso sólo tiene las cifras 0 y 1.

En la sistema decimal, la última cifra es parte del uno (10^0), la penúltima es parte del diez (10^1), la tercera del cien (10^2) ...

En el sistema binario se hace lo mismo, pero con el 2: la ultima cifra es parte del uno (2^0), la penúltima es parte del 2 (2^1), la tercera es parte del 4 (2^2), después es 8 (2^3) ...

decimal	binario	hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A

11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

En un byte se puede guardar los números entre 0 y 255 (2^8-1), en total 256 (2^8) diferentes números. Para números más grandes se usa más bytes. En dos bytes (16 bit) se guarda hasta 65.535, en 4 bytes (32 bit) se guarda hasta 4.294.967.295 y en 8 bytes (64 bit) se guarda hasta 18.446.744.073.709.551.615.

A veces se usa el sistema hexadecimal para expresar el contenido de un byte o varios bytes. Este sistema utiliza el dieciséis como base. Se necesita nuevas cifras, desde A que representa diez hasta F que representa quince. Eso sirve para presentar el contenido de un byte en dos letras: 00 cuando todos los bits son ceros (0) hasta FF cuando todos los bits son unos (1).

Texto

Para guardar textos normalmente se usa un byte por letra. El código ASCII es el más importante y define qué valor significa cada carácter (letra, cifra, símbolo, carácter de control).

Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	
0	00	NUL	16	10	DLE	32	20	48	30	0	64	40	@	80	50	P
1	01	SOH	17	11	DC1	33	21	49	31	1	65	41	A	81	51	Q
2	02	STX	18	12	DC2	34	22	50	32	2	66	42	B	82	52	R
3	03	ETX	19	13	DC3	35	23	51	33	3	67	43	C	83	53	S
4	04	EOT	20	14	DC4	36	24	52	34	4	68	44	D	84	54	T
5	05	ENQ	21	15	NAK	37	25	53	35	5	69	45	E	85	55	U
6	06	ACK	22	16	SYN	38	26	54	36	6	70	46	F	86	56	V
7	07	BEL	23	17	ETB	39	27	55	37	7	71	47	G	87	57	W
8	08	BS	24	18	CAN	40	28	56	38	8	72	48	H	88	58	X
9	09	HT	25	19	EM	41	29	57	39	9	73	49	I	89	59	Y
10	0A	LF	26	1A	SUB	42	2A	58	3A	:	74	4A	J	90	5A	Z
11	0B	VT	27	1B	ESC	43	2B	59	3B	;	75	4B	K	91	5B	[
12	0C	FF	28	1C	FS	44	2C	60	3C	<	76	4C	L	92	5C	\
13	0D	CR	29	1D	GS	45	2D	61	3D	=	77	4D	M	93	5D]
14	0E	SO	30	1E	RS	46	2E	62	3E	>	78	4E	N	94	5E	^
15	0F	SI	31	1F	US	47	2F	63	3F	?	79	4F	O	95	5F	_
														111	6F	o
														127	7F	DEL

El código ASCII sólo usa 7 bit de los 8 bit en un byte y sólo cubre las necesidades de las letras del inglés. Por eso existen otros códigos para otros idiomas que extienden el ASCII. Esos usan los valores de 128 a 255 para otros caracteres. Por ejemplo el Latin1 o ISO 8859-1 contiene todas las letras que se usa en idiomas con escritura latina (caracteres con tilde o diéresis, etc.). Como así hay otros códigos para el alfabeto griego, el alfabeto cirílico, etc.

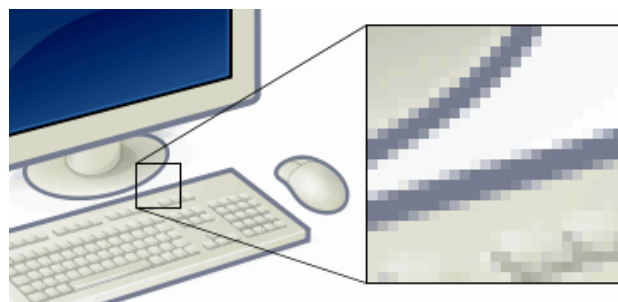
Pero eso no sirve para todos los idiomas del mundo por que unas escrituras asiáticas tienen miles de caracteres y también a veces se quiere mezclar caracteres de diferentes sistemas de escritura. Por eso se hizo el estándar Unicode. Unicode contiene todos los caracteres conocido en el mundo. En la codificación UTF-8 para caracteres de ASCII no cambia nada. Otros caracteres como el ñ se guarda en 2 bytes, otros más exóticos en 3 o 4 bytes.

Imágenes

Para guardar imágenes, se divide la imagen en cuadrados que se llaman “píxel”. Un píxel es la parte más pequeña de la imagen y tiene un color.

Para una imagen en blanco y negro sólo necesitamos 2 estatus: uno para el blanco y uno para el negro. Entonces se necesita un bit por píxel, por ejemplo 0 para negro y 1 para blanco.

Las imágenes con gris usan un byte por píxel, desde el 0 para el negro, y sigue con diferentes intensidades de gris hasta llegar a 255 para el blanco.



píxel en un imagen

Para imágenes de color hay tres colores básicos: rojo, verde y azul (la abreviación inglesa RGB: red, green, blue). Cuando se mezclan la luz roja con la luz verde con la luz azul se obtiene una luz blanca. Al igual que en las imágenes con gris, se usa un byte por color por píxel. El 0 también significa negro y 255 es rojo, verde o azul intensos.

Guardar Datos

Un disco rígido es un espacio grande para guardar datos. Pero este espacio necesita organización para guardar, reencontrar y leer los datos de nuevo. Por eso se puede dividir un disco rígido en particiones. En dentro de una partición hay un sistema de archivo para organizar los directorios y los archivos.

Particiones

El primero sector de un disco rígido se llama Máster Boot Record (MBR) y contiene una tabla para cuatro particiones. Esta tabla guarda la información sobre dónde empieza y dónde termina una partición y que tipo de sistema de archivo contiene. Las particiones en la tabla de particiones en el MBR se llaman particiones primarias. Una de estas particiones puede ser una partición extendida. En su espacio interior se puede hacer más particiones, las particiones lógicas.

Para hacer particiones, se usa el comando “fdisk” en la línea de comandos de Windows o Linux. Más **fácil** es la “Administración de discos” de Windows NT, 2000, XP o más reciente, el programa “Partition Magic” de Norton o el “QParted” de Linux (software libre). Los últimos dos programas también pueden cambiar el tamaño de una partición sin perder los archivos.

Sistemas de Archivos

Para escribir y leer archivos de una partición, se usa un sistema de archivo. Para crear un sistema de archivo (formatear) hay diferentes programas. Se puede usar “Partition Magic” o “QParted”, la “Administración de discos” o los comandos “format” en Windows o “mkfs” en Linux.

FAT

El sistema de archivos FAT (File Allocation Table: Tabla de Asignación de Archivos) es el más usado y bastante viejo. Casi todos los sistemas operativos lo conocen. DOS (disquete de arranque), Windows 95, 98, y ME sólo conocen FAT. El máximo tamaño de un archivo es 4 GB. Se usa normalmente este sistema para memoria flash (pen drive, cámaras digitales, ...). Con FAT, todos los usuarios de una computadora pueden leer y escribir todo los archivos. No hay permisos de acceso a archivos.

NTFS

Para Windows NT Microsoft desarrolló el sistema de archivos NTFS (New Technology File System). Para Windows 2000, XP y los más nuevos se lo usa normalmente. El máximo tamaño de un archivo es 16 TB y conoce permisos de acceso a archivos.

ext3

El ext3 (Third Extended Filesystem: tercer sistema de archivos extendido) es el más usado para Linux. El máximo tamaño de un archivo es entre 16 GB y 2 TB (depende del tamaño de bloque) y conoce permisos de acceso a archivos.

ISO 9660

Se usa este sistema para CD y DVD. Todos los sistemas operativos lo conocen. El máximo tamaño de un archivo es 2 GB y no hay permisos de acceso a archivos.

UDF

El UDF (Universal Disk Format) se usa para DVD y Blu-ray. El máximo tamaño de un archivo es 16 EB y no hay permisos de acceso a archivos.

Tareas de un Sistema Operativo

- manejar procesos (ejecutar programas, terminar procesos)
- comunicación entre procesos (por ejemplo una diagrama de Excel contenido en un documento de Word)
- compartir el tiempo procesional del micro entre los procesos
- compartir el espacio de memoria entre los procesos
- comunicar con el hardware (red, pantalla, disco, impresor, ...) con el driver adecuado
- leer, escribir y manejar archivos
- sistemas operativos modernos conocen diferentes usuarios con derechos de acceso individuales. Eso permite esconder o proteger archivos de otros usuarios. Siempre hay un usuario especial que tiene todos los derechos. En "Windows" se llama Administrador y en Linux "root".

