



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Ingeniería en Computación

Sistemas Operativos

Profesor: DR. GUNNAR EYAL WOLF ISZAEVICH

Práctica 1. Revisión de MiComputer

Grupo: 8

Integrantes:

Reyes Herrera Janeth Randy 320338841

Rivas Gil María Lucía 320492057

Fecha: 18/09/2025

Fascículo

$m = m1 + m2 = 320338841 + 320492057$

$f = [(m \% 100) / 4] = [(640830898 \% 100) / 4] = 24.5$

Fascículo 24: "Curso práctico del ordenador personal, al micro y el miniordenador"

Research Machines 380Z

Un sistema físico sólido y unos soberbios gráficos en alta resolución han popularizado este microordenador en las escuelas y entre los militares.

Por los años 70's y 80's los productos creados por Research Machines Limited eran famosos por ser de los más duraderos, en este caso el Research Machines 380Z era el microordenador más popular ya que era una de las máquinas más solicitadas por los profesores y pedagogos pues era una máquina robusta con mucha solidez y extraordinariamente fiables, además permitía tener gráficos de "alta resolución" para la época, se recomendaba que para aprovecharla al máximo el monitor contará con una interface RGB. Los gráficos que mostraba eran de 512 x 255 píxeles, una diferencia abismal con los que vemos hoy en las computadoras que soportan resoluciones de 4K u 8K, con millones de colores y gráficos 3D fluidos.

El 380Z utilizaba un procesador Z80 de 8 bits, esto significaba que su manejo de memoria era mucho más reducido al que tenemos actualmente, ya que contaba con tan solo 56 KB de RAM y 6 KB de ROM (se podía agregar más RAM hasta 64 K, una cantidad muy pequeña comparada con los gigabytes de memoria que tienen los equipos hoy en día, lo mismo sucede con los ciclos de reloj que tenía la computadora ya que contaba con 4 MHz , esto significa que el procesamiento que tenía era muy lento si lo comparamos con la velocidad que manejan los procesadores actuales de GHz (miles de millones de hertz).

Era un sistema monotarea, capaz de ejecutar solo una tarea a la vez, actualmente los procesadores pueden ejecutar miles de hilos en paralelo gracias a sus múltiples núcleos y de acuerdo con el fascículo, el 380Z tenía únicamente un núcleo de hierro con condensadores lo que permitía que fuera casi imposible sobrecargarlo o dañarlo.

El sistema operativo estándar era el CP/M, que permitía ejecutar una gran variedad de programas, la máquina contaba con varias versiones de BÁSICA, por lo que el usuario podía decidir cual utilizar para que la máquina se pudiera adaptar a la cantidad de memoria que se tenía; la versión ampliada: BASIC RML, era muy parecida al BASIC de Microsoft, lo que facilitaba a los usuarios pasar de una plataforma a otra.

Además de BASIC, el 380Z soportaba otros lenguajes como ALGOL y FORTRAN. El uso de estos lenguajes era un punto a su favor, ya que eran populares entre los científicos europeos y estadounidenses, eran y siguen siendo eficaces para realizar cálculos matemáticos complejos, como los que se necesitaban en el diseño estructural.

Retomando la parte de los gráficos, la máquina también tenía un paquete llamado HRG que funcionaba como un archivo de código máquina simple y se podía usar desde cualquier lenguaje, lo cual permitía un control más directo sobre el hardware.

Comparando lo grande que se veía, al menos en la imagen, la diferencia en el hardware que se utilizaba antes al que se usa actualmente es bastante, se ve que era un equipo grande y pesado, con las placas de circuito expuestas y una ventilación muy básica; en contraste, las computadoras que utilizamos actualmente son más compactas, más ligeras y potentes.

Resumiendo, hacer una comparación entre la máquina 380Z y las computadoras actuales fue bastante informativo, ya que se comprenden mejor las limitaciones que tenían años atrás. Considerando las características de potencia, portabilidad y multitareas de los equipos que usamos ayuda a dimensionar que tanto hemos avanzado en esta era digital

El lenguaje ensamblador

Continuando con nuestra introducción al código de lenguaje máquina, analizamos las formas de expresar los programas, desde los números binarios al lenguaje ensamblador.

Las computadoras tienen una forma diferente a la de los humanos de procesar la información. Mientras que en nuestro vocabulario incluimos múltiples signos y símbolos, las computadoras son más “simples” al incluir en su vocabulario únicamente secuencias de unos y ceros (números binarios) para representar de todo: Datos (números, texto, imágenes), instrucciones del programa, direcciones de memoria.

Es por esta razón que los humanos cometen muchos errores a la hora leerlos y escribirlos. Para evitar todos estos problemas, se han adoptado los números hexadecimales para leer, escribir, depurar y programar a bajo nivel. y ¿Por qué hexadecimal y no otros números?

1 dígito hexadecimal = 4 bits

2 dígitos hexadecimales = 1 byte (8 bits)

Un ejemplo de los beneficios de usar hexadecimal y no binarios es el siguiente:

Dirección binaria: 00000000000001010 → 16 bits

En decimal: 10

En hexadecimal: 0x0A

Lo que ve la computadora	Lo que vemos nosotros
00000000 00001010	0x0A
10110011 01100101	0xB365

Hay diferentes formas de acceder a los datos en la memoria:

- Direccionamiento inmediato:

- MOV AX, 5 ; Carga el valor 5 directamente en el registro AX
- Direccionamiento directo:
MOV AX, [1000h] ; Carga en AX el valor almacenado en la dirección 1000h
- Direccionamiento indexado:
MOV AX, [SI + 2] ; tomando a SI como un desplazamiento
- Direccionamiento indirecto:

El lenguaje ensamblador está diseñado para trabajar directamente con la arquitectura del procesador, usa instrucciones mnemónicas (fáciles de recordar) que representan instrucciones de lenguaje máquina (binario).

Las etiquetas son nombres simbólicos que se usan para representar direcciones de memoria, posiciones de instrucciones y para hacer código más legible y mantenible. Con normalidad, un código de lenguaje máquina se imprime como un dump, que es una larga lista de valores hexadecimales.

En lenguaje ensamblador, se hace uso de etiquetas simbólicas que luego el ensamblador traduce a direcciones de memoria reales (hexadecimales). Además, Un opcode (código de operación) es una instrucción en lenguaje máquina, representada por un valor binario (o hexadecimal), que le dice al procesador qué operación realizar.

Referencias

G. Jefferson, R. Ford, S. Tarditti, A. Cuevas. (1984). "Curso práctico del ordenador personal, al micro y el miniordenador". MiComputer, volumen 24. Págs. 478 - 480.