Buenos días Kepa:

Tengo ciertas dudas conceptuales que necesito resolver antes de dar el paso de empezar a programar. No me gustaría hacer cosas sin entender bien lo que estoy haciendo.

En lo que estoy atascado es en todo lo relacionado con el direccionamiento de memoria. Para concretarte mejor, es sobre lo siguiente:

1.- En los procesadores 8086, con un bus de direcciones de 16 bits, la memoria está segmentada en grupos de 64 Kb (2^16 posiciones distintas), pero ¿Cuantos segmentos distintos puede haber?, ¿también 2^16 segmentos?. Si eso es así, ¿porqué entonces nos referirnos a esas posiciones de memoria con 20 bits?. No entiendo porque si los registros de los segmentos son de 16 bits, y los registros para los desplazamientos también son de 16 bits, se utilizan 20 bits para direccionar. Algo se me escapa.

2.- En los procesadores x86 con un bus de direcciones de 32 bits me surge la misma pregunta, es decir, ¿Aquí la memoria está segmentada en grupos de 4Gb (2^32 posiciones distintas)?, ¿Cuantos segmentos distintos puede haber en estos procesadores entonces?, ¿Estaría bien decir que puede haber 2^32 segmentos distintos, cada uno con 2^32 posiciones distintas de memoria?. En estos procesadores los registros para segmentos son de 32 bits, y los desplazamientos son de 32 bits también. ¿Pero, que tiene todo esto que ver con los llamados modos de direccionamiento "real " y "protegido"?.

3.- ¿Yo puedo elegir el segmento de memoria donde quiero empezar a guardar el código, los datos y la pila o me los asigna automáticamente el sistema operativo sin poder intervenir en ese proceso?

4.- Cuando se está programando en ensamblador, ¿hay que indicar en el programa qué modo de direccionamiento hay que usar o se usa el que quieras en cada momento y ya está?.

5.- He visto también que cuando se programa, al menos con ensambladores como TASM o MASM, existen directivas para indicar al ensamblador los segmentos para código, datos y la pila. ¿Pero a mí no me gusta nada eso de las directivas, ya que (a mi modo de ver) volvemos a crear en cierta manera otro lenguaje de más alto nivel que realiza operaciones internas para indicar al compilador del ensamblador no se qué cosas que hacer cuando se escribe una directiva u otra. ¿Hay manera de sustituir esas directivas por código ensamblador puro y evitar estos artificios?.

6.- También tengo una duda que puede parecer algo absurda, y es la siguiente. ¿Cada segmento de memoria tiene asociado un microchip de memoria de la RAM (todo el microchip o parte de él) o no tiene nada que ver?. ¿Pueden existir segmentos de memoria los cuales contengan posiciones de memoria que pertenezcan a 2 o más microchips físicos distintos?. ¿Y en el caso de memoria USB?.

En fin, como puedes ver todo esto me tiene bastante confundido y no encuentro ningún libro o documentación que me pueda resolver estos temas. Disculpa que te moleste por cosas que seguro son muy simples.

¿Podrías preparar una clase para darme respuesta a todas estas cuestiones que te planteo?.

Quedo a la espera de lo que me digas.

Un saludo y gracias por todo.

Mario Cornacchia

629 341 193

cornabatomario@gmail.com