

## Taller 10 - Interrupciones en 8086

### Objetivos

Aprender y comprender los detalles de la arquitectura 8086 cuando trabajamos con interrupciones.

### Introducción

Las interrupciones son cambios en el flujo de ejecución no causados por otros programas, sino por otros eventos, usualmente relacionados con la entrada/salida. Una vez finalizada la interrupción el flujo de ejecución vuelve al punto donde se detuvo el programa interrumpido.

### Responder las siguientes preguntas

- ¿Qué contexto es salvado por el hardware de x86 al atenderse una interrupción?
- ¿Qué contexto debería salvar el programador? ¿En qué situaciones no es necesario?
- ¿Qué sucede si se produce una interrupción durante la ejecución de un manejador de interrupciones (handler)?. Justifique su respuesta.

### Interrupciones y Puertos de E/S

Considere la letra del taller anterior y la siguiente solución al problema:

```
#define OFF 1
#define ON 2
#define TOL_DEC ...
short v_obj, n_cinta, t_total, t_alarma;
char estado;

void main() {
    out(MOTOR, 0);
    out(DISPLAY_H, 0);
    out(DISPLAY_L, 0);
    estado = OFF;
    // instalar interrupciones
    enable();
    while true {}
}

void hay_boton() {
    char botones = in(BOTONES);
    if (botones & 1) {
        out(DISPLAY_H, 0);
        out(DISPLAY_L, 0);
        out(MOTOR, 0);
        estado = OFF;
    } else {
        if (botones & 2){
            vobj = 50;
        }
        if (botones & 4){
            vobj = 110;
        }
        t_alarma = 0;
        if (estado == OFF)
            t_total = 0;
    }
}
```

```
        estado = ON;
        OUT(MOTOR, vobj << 1 | 1);
    }
}

void cinta() {
    n_cinta++;
}

void timer() {
    short vi; // velocidad instantánea en décimas de km/h
    if (estado == ON) {
        t_total++;
        vi = (10 * n_cinta) / 3;
        // se considera que la velocidad actual es igual a la objetivo si
        // está a menos de una cierta tolerancia
        if (v_obj - TOL_DEC < vi && vi < v_obj + TOL_DEC)
            t_alarma = 0
        else // velocidad fuera de rango
            t_alarma++;

        if (vi > 127) vi = 127; // tope: se muestra hasta 12.7 km/h porque es
                               // lo que permite la representación

        char b_power = vi > 0;
        char b_alarma = (t_alarma > 30? 1: 0)
        out(DISPLAY_H, b_alarma << 15 + m_power << 14 + vi << 7);
        out(DISPLAY_L, t_total);
    }
    n_cinta = 0;
}
```

### **Se pide**

- a) Compilar todas las rutinas de la parte anterior a assembler 8086.
- b) Escriba la secuencia de código que permita instalar un manejador de interrupciones en 8086. ¿Qué cuidados deben tenerse para que se instale correctamente?