

Es necesario que en la IDT haya una entrada en el índice 33 cuyo offset apunte al handler definido más adelante, su selector de segmento sea uno de código de nivel 0 con RPL 0 y que sus atributos son los correspondientes a un descriptor de interrupciones de 32 bits con DPL de supervisor (0x8E00). También definiremos en el índice 100 una entrada con un offset que apunte a la rutina llamada \_isr100 (todos los handlers siguen la misma convención). El campo segsel tendrá el valor de un selector de código de nivel 0 con RPL 0 y el campo de atributos corresponde a un descriptor de interrupciones de 32 bits con DPL de usuario (0xEE00) para que las tareas puedan llamar al servicio.

```
extern HandleKey
extern EsperarTecla
extern getKey
global _isr33
_isr33:
    pushad

    in al, 0x60
    push eax
    call HandleKey
    add esp, 4

    call pic_finish1 ; Avisar al pic que se recibio la interrupcion
    popad
    iret

global _isr100
_isr100:
    push ebp
    mov ebp, esp
    sub esp, 4

    pushad
    call EsperarTecla
    test ax, ax
    jz .fin

    mov word [sched_task_selector], ax
    jmp far [sched_task_offset]
.fin:
    call getKey
    mov [ebp - 4], eax

    popad
    pop eax
    pop ebp
    iret
```

Modifico la función sched\_next\_task y agrego funciones y variables globales de la siguiente manera:

```
int8_t curr_idx = 0;
uint8_t lastKey;
int8_t pending = -1;
uint8_t keyAvailable = 0;

uint16_t sched_next_task(void) {
    curr_idx = (curr_idx + 1) % 4;
    if(curr_idx == pending){
        curr_idx = (curr_idx + 1) % 4; //salteo la tarea que espera una tecla
    }
    return tasks[curr_idx].idx_gdt;
}

void HandleKey(uint8_t scanCode) {
    if(scanCode && 0x80){
        keyAvailable = 1;
        lastKey = scanCode;
        pending = -1;
    }
}

uint16_t EsperarTecla() {
    if(keyAvailable){
        return 0;
    } else {
        pending = curr_idx;
        return sched_next_task();
    }
}

uint8_t getKey() {
    keyAvailable = 0;
    return lastKey;
}
```

Vale aclarar que el handler del teclado solo reacciona al soltado de las teclas. Quien llame a la función getKey consumirá el último input del teclado.