

Práctica de Laboratorio 4: Interrupciones en MIPS32

Mike Gonzalez y Miguel Monsalves

29 de julio de 2025

1. Introducción

Esta práctica implementa dos sistemas controlados por interrupciones en MIPS32: un buffer circular de teclado con temporizador y un semáforo controlado por pulsador, simulados en MARS.

2. Ejercicio 1: Buffer Circular

2.1. Implementación

```
1 ; C digo completo del Ejercicio 1
```

2.2. Explicación

- **Buffer Circular:** Almacena hasta 256 caracteres con índices `head` y `tail`.
- **Interrupciones:**
 - Teclado: Almacena caracteres en tiempo real (bit 9 de `$cause`)
 - Temporizador: Cada 20 seg imprime el buffer (bit 8 de `$cause`)
- **Registros MMIO:**
 - `0xffff0000`: Control de teclado
 - `0xffff0010/14`: Control de temporizador

3. Ejercicio 2: Semáforo con Pulsador

3.1. Implementación

```
1 ; Código completo del Ejercicio 2
```

3.2. Explicación

- **Estados:**
 1. 0: Verde (espera pulsación)
 2. 1: Espera (20s → amarillo)
 3. 2: Amarillo (10s → rojo)
 4. 3: Rojo (30s → verde)
- **Transiciones:** Controladas por interrupciones de teclado y temporizador.
- **Mensajes:** Se imprimen en cada cambio de estado.

4. Respuestas Teóricas

4.1. Ciclo de Interrupción

1. Ocurre evento (ej: tecla presionada)
2. Procesador termina instrucción actual
3. Guarda EPC y Status
4. Salta a dirección 0x80000180
5. Ejecuta rutina de servicio
6. Restaura contexto y retorna (`eret`)

4.2. Sondeo vs Interrupciones

- **Sondeo:** CPU verifica dispositivos periódicamente (desperdicio de ciclos).
- **Interrupciones:** Dispositivos notifican a CPU (eficiente).

4.3. Ventajas de Interrupciones

- Permite ejecución concurrente
- Reduce consumo energético
- Mejor respuesta a eventos en tiempo real

4.4. Registros MIPS32 para Interrupciones

- `$status`: Habilita interrupciones (bit 0)
- `$cause`: Identifica causa de interrupción
- `$epc`: Guarda dirección de retorno

4.5. Guardar Contexto

Es necesario para evitar:

- Corrupción de registros
- Comportamiento errático al retornar
- Pérdida de datos en programa principal

4.6. Excepciones en MIPS32

- **Situaciones:**
 1. Desbordamiento aritmético
 2. Instrucción no válida
 3. Fallo de página
 4. Acceso a memoria no alineado
- **Etapas del pipeline:**
 - IF: Fallo de fetch
 - ID: Instrucción inválida
 - EX: Desbordamiento
 - MEM: Acceso inválido

5. Conclusiones

Se implementaron exitosamente sistemas controlados por interrupciones, demostrando su eficiencia para gestión de E/S y eventos temporizados. Los códigos funcionan en MARS siguiendo los pasos de configuración indicados.