

Multiprocesadores

Ejercicio 9: Load Link y Store Conditional

García Esteban, Sergio

17-Mayo-2020

Tiempo dedicado (aproximado): 1h

Resumen

Familiarizarse con el uso de las instrucciones Load Link y Store Conditional

Notas generales

El ejercicio puede presentarse de forma individual o en grupos de máximo dos personas. Podéis trabajar en grupos mayores, pero **cada grupo debe elaborar el material a entregar de forma independiente**. Hacednos llegar vuestros ejercicios **en formato pdf** a través de la entrega habilitada en la web de la asignatura (moodle). Incluid vuestro nombre y apellidos en la cabecera del documento y vuestro NIP en el nombre del fichero (ej9_NIP.pdf).

Plazo límite de entrega: lunes 25 de Mayo, 23h59m59s.

Ejercicio

1. Implementar en ensamblador RISC-V las funciones Lock (L) y Unlock (L) para proteger el acceso a una sección crítica mediante consulta continua (busy waiting-spinning) sobre la variable L y utilizando las instrucciones Load Link y Store Conditional

Cuando la variable es igual a 1, hay un proceso dentro de sección crítica, es igual a 0 cuando no hay ninguno. La dirección de la variable pasada por parametro "L" se encuentra en el registro x20.

Lock(L)

```

        addi x12, x0, 1           //copiamos el valor "bloqueado" (1 en este caso)
again:  lr.d x10, (x20)           //load link a la dirección de la variable "L"
        bne x10, x0, again        //si no está libre, volvemos a leer la variable
        sc.d x11, (x20), x12      //si está libre, intentamos escribir el valor
                                   // "bloqueado" con store condicional
        bne x11, x0, again        //si el store no escribe, volvemos a leer la variable

Unlock(L)

        sd x0, (x20)              //escribimos incondicionalmente el valor "desbloqueado"
                                   //(0 en este caso)

```