Sincronización a bajo nivel

# Introducción.

Sincronización Condicional vs Exclusión mutua.

Asumimos:

* Lectura y escritura atómicas.
  + **Grano fino:** Hardware
  + **Grano Grueso:** Software
* **Planificador Justo:** te asigna a un proceso un microprocesador.

# Sincronización Condicional

PROGRAM PeticionContinua;

PROCESS Servidor(VAR Peticion : BOOLEAN; VAR x : INTEGER);

BEGIN

WHILE NOT Peticion DO ; (\* ESPERA ACTIVA \*)

WRITELN('Servidor: Peticion atendida, x = ', x);

END;

PROCESS Cliente(VAR Peticion : BOOLEAN; VAR x : INTEGER);

BEGIN

x := 1;

WRITELN('Cliente: Solicito peticion, x = ', x);

Peticion := TRUE;

END;

VAR

x : INTEGER;

Peticion: BOOLEAN;

BEGIN

x := 0;

Peticion := FALSE;

COBEGIN

Servidor(Peticion, x);

Cliente(Peticion, x);

COEND

END.

**Programa Petición Justicia fuerte**

PROCESS Cliente(VAR Peticion : BOOLEAN; VAR x : INTEGER);

BEGIN

REPEAT

Peticion := TRUE;

WRITELN('Solicito Peticion x = ', x);

Peticion := FALSE;

WRITELN('Rescindo Peticion x = ', x);

FOREVER

END;

El proceso servidor puede ejecutarse siempre con Peticion := FALSE

La sincronización condicional no se lleva muy bien con la justicia fuerte.

# Exclusión Mutua

PROCESS P;

BEGIN

REPEAT

Preprotocolo;

SeccionCritica;

PostProtocolo;

SeccionNoCritica;

FOREVER

END;

* **Requisitos:**
  + Exclusión Mutua.
  + Ausencia de Interbloqueo.
  + Ausencia de retrasos innecesarios.
  + Ausencia de inanición.