

Análisis del algoritmo de Dekker para tres procesos

Isabel del Pilar Durán Chumillas

1 Algoritmo

```
wantp ← false
wantq ← false
wantr ← false
nProc ← 0
turn ← 1

p:
mientras true
    wantp ← true
    mientras wantq ∧ wantr
        si turn ≠ 1
            wantp ← false
            mientras turn ≠ 1
                await turn 1
            fin_mientras
            wantp ← true
        fin_si
    fin_mientras
    nProc++
    await
    //Seccion critica
    ...
    wantp ← false
    turno ← 2
fin_mientras
```

Para q y r haremos lo mismo cambiando los turnos y condiciones. Podemos ver que la condición de turno ha pasado de $\text{turno} == X$ a $\text{turno} != X$. Esto es por que cuando solo existían dos procesos, el contrario a un turno era directamente, el turno del otro proceso. En esta ocasión existen tres por lo que hay que comprobar que el turno actual sea distinto al del mismo.

Cada proceso debe cambiar la variable turn a uno de los otros tres sin que este

se repita nunca, ya que en este caso uno de ellos jamás tendría acceso.
La salida será aleatoria y siempre con un solo proceso en la zona crítica.