

GUÍA 7: FLUJO DE INFORMACIÓN

Ejercicio 1:

Se tienen dos procesos p y q . El proceso p está autorizado a leer la clave k , de longitud n (n bits), mientras que el proceso q no está autorizado a leer k .

Se cuenta con el directorio d , inicialmente vacío. Los procesos tienen autorización para leer y escribir en d pero bajo las siguientes condiciones:

- Cuando p lee d , ve todos los archivos en el directorio y puede escribir sus propios archivos.
- Cuando q lee d , no ve ni puede leer archivos creados por p .

Aunque p guarde en d un archivo con nombre k , q no podrá verlo.

Sin embargo:

- si un proceso trata de crear un archivo que ya existe en el directorio, entonces el archivo no es creado y se envía una señal de error al proceso
- si un proceso trata de borrar un archivo que no existe o existe pero pertenece a otro proceso, entonces el archivo no es borrado y se envía una señal de error al proceso.

Verificar que sí es posible que P transmita la clave k a Q a través de D con el siguiente esquema que aprovecha el directorio compartido y las señales de error haciendo un seguimiento para $k="10"$ y $n=2$

Las acciones para P son:

```

00  p: flag:= 0;
01  for i:=1 to n do
      /* wait for q to be ready */
02      if flag = 0
03          then wait until file "huhu" appears in d
04          else wait until file "huhu" disappears;
      /* signal ith bit */
05      If k[i]=1
06          then try to create file "hoho"
07          else try to delete file "hoho";
      /* signal: I have sent */
08      if flag = 0
09          then try to create file "ready"
10              until no error occurs
11          else delete file "ready";
12      invert flag;

```

Las acciones para Q son:

GUÍA 7: FLUJO DE INFORMACIÓN

```

0  q: flag:=0
1  for i:=1 to n do
2      if flag = 0
3          then /* signal: I am ready */
4              create file "huhu" in d
5              /* wait for p to have sent */
6              try to create file "ready"
7              while no error occurs
8                  delete file "ready";
9                  try to create file "ready"
10             else /* signal: I am ready */
11                 delete file "huhu"
12                 /* wait for p to have sent */
13                 try to create file "ready"
14                 until no error occurs;
15                 delete file "ready"
16             /* receive ith bit */
17             try to create file "hoho"
18             if no error occurs
19                 then rec[i]:=0;
20                 delete file "hoho"
21             else rec[i]:=1;
22         invert flag;

```

Ejercicio 2:

Considerar la siguiente función e indicar si fluye información de x a y . Tener en cuenta que el tipo BIT sólo puede tomar el valor 0 ó 1. Justificar.

```

void flowinfo(BIT x, BIT *y)
{
    BIT z;
    *y = 0;
    z = 0;
    if (x == 0)
        z = 1;
    if (z == 0)
        *y = 1;
}

```

Ejercicio 3:

Sean $x, y, z \in \{0, 1\}$. Considera:

```

if ( x == 1 && y == 1)
    z = 1;

```

Considera que z inicialmente vale 0 y que los valores de x e y son equiprobables.

- calcular $H(x)$ y $H(y)$
- calcular $P(z'=0)$ siendo z' el valor de z después de ejecutarse el código anterior
calcular la probabilidad condicional $P(x=i|z'=j)$ y $P(y=i|z'=j)$ para todo $i, j \in \{0,1\}$
- calcular $H(x | z')$ y $H(y | z')$

Ejercicio 4:

Sean $x \in \{0, 1, 2\}$, $y \in \{0, 1\}$. Considera:

```

if ( x > 1)
    y = 1;

```

Sabiendo que $P(x=0) = 0.5$, $P(x=1) = 0.25 = P(x=2)$ y que inicialmente x vale 0.

- calcular entropía $H(x)$
- calcular $H(x | y')$ donde y' es el valor de y después de ejecutar el código anterior.