

Gergio De Luca ejer 402 1/2

1. $U[1..N]$
 sem esperarFor[10]; $mutex = 1$; $mutexF = 1$; $esperarUso[10] = (0)$
 queue $U[1..N]$
 queue $FOTOCOP$
 sem enColaU = 0; $U[10] = (0)$
 sem enColaF = 0; $U[10] = (0)$

REGULAR

Procesa $U[1..N]$ {
 1. FOR
 P(mutex);
 $U = U[i]$;
 V(mutex);
 V(enColaU);
 P(esperarFor[i]);
 $FOT = U[i]$;
 P(esperarUso[FOT]);
 V(LibroUso[FOT]);
 FOTOCOP();
 V(Libro[FOT]);



Procesa $FOTOCOP$ {

while true {
 P(mutexF);
 $FOTOCOP = FOTOCOP[i]$;
 V(mutexF);
 V(enColaF);
 P(LibroUso[FOTOCOP]);
 // usando por usuario
 P(Libro[FOTOCOP]);

Procesar empleado()

int idusr, idfor;

while true {

 P(enColausr);

 P(mutexusr);

 idusr = usuarios.pop();

 V(mutexusr);

~~usuarios[idusr] =~~

 P(enColafor);

 P(mutexf);

 idfor = fotocopias.pop();

 V(mutexf);

 usuarios[idusr] = idfor;

 V(esperafor[idusr]);

 V(esperausr[idfor]);

}



Sergio De Luca 6773/1 ese 2 hoja 1/2

```
Procesar clientes [i: 1..N] {  
  var e; resto Ligado, Factura;  
  corralon. esperar(e);  
  escritorio[e]. entregar(Ligado);  
  escritorio[e]. recibir(Factura);  
}
```

BIEN-

```
Procesar empleados [i: 1..4] {  
  resto Ligado, Factura;  
  while (true)  
    corralon. atender(i);  
    escritorio[i]. recibir(Ligado)  
    Factura = MaxFactura();  
    escritorio[i]. entregar(Factura);  
}
```

Monitor corralon {

cola elibres;

cond espera;

var esperando = 0, libres = 0;

procedure esperar (var ide: out) {

if (libres == 0) {

esperando ++;

wait (espera);

} else

libres --;

ide = elibres. pop

procedure proximo (var ide: in) {

push. elibres(ide);

if (esperando > 0) {

esperando --;

signal (espera);

} else libres ++;



Gerardo de Woz 6773/1 eje 2 hoja 2/2

Monitor escritorio [i: 1..4] {

text LIST, FTEXT;

cond aviso, recibe;

bool okplani, OKF;

Procedure entregar (texto LIST: in) {

LIST = LIST;

okplani = true;

signal (aviso);

}

Procedure recibir (texto LIST: out) {

IF (not okplani) wait (aviso);

LIST = LIST;

}

Procedure entregar F (texto F: in) {

FTEXT = F;

OKF = true;

signal (~~ok~~ (recibe));

}

Procedure recibir F (texto F: out) {

IF (not OKF) wait (recibe);

F = FTEXT;

}

