

Ejercicios de semáforos

1. En una nave industrial existe una máquina de inyectar que deja cada pieza producida en una cinta transportadora de tamaño limitado. Existe un robot que recoge las piezas de la cinta (una cada vez) y las deja en las cajas de embalaje. Finalmente, tenemos un operario que, de vez en cuando, recoge 3 piezas para realizar el control de calidad, si no hay tres piezas en la cinta lo intentará más tarde. Resuelve el escenario anterior mediante semáforos.

```
#define cinta n
```

```
TSemaforo s, n, e;
```

```
int piezas;
```

```
void inyector() {  
    while(true) {  
        coger_pieza();  
        P(e);  
        P(s);  
        piezas++;  
        inyectar();  
        V(s);  
        V(n);  
    }  
}
```

```
void robot() {  
    while(true) {  
        P(n);  
        P(s);  
        coger_pieza();  
        piezas--;  
        V(s);  
        V(e);  
        embalar();  
    }  
}
```

```
void operario() {  
    while(true) {  
        P(s);  
  
        if(piezas >= 3) {  
            piezas = piezas - 3;  
            V(e); V(e); V(e);  
            control_de_calidad();  
        }  
  
        V(s);  
    }  
}
```

```
void menu() {  
    inicializar(s, 1);  
    inicializar(n, 0);  
    inicializar(e, n);  
    piezas = 0;  
  
    cobegin  
        inyector(); robot(); operario();  
    coend;  
}
```

2. Una tribu de caníbales cenan en comunidad una gran olla que contiene M exploradores cocinados. Cuando un caníbal quiere comer, él mismo se sirve de la olla un explorador, a menos que esté vacía. Si la olla está vacía, el caníbal despierta al cocinero y espera a que éste llene la olla. Desarrollar el código de las acciones de los caníbales y el cocinero usando semáforos.

```
TSemaforo mutex, comer, cocina;  
Int olla, M;
```

```
Void caníbal() {  
    While(true) {  
        P(mutex)  
        If(olla == 0) { V(cocina), P(comer) }  
        Olla--;  
        V(mutex)  
    }  
}
```

```
Void cocinero() {  
    While(true) {  
        P(cocina)  
        Olla = M;  
        V(comer)  
    }  
}
```

```
Void main() {  
    Olla = M;  
    Inicializa(mutex,1);  
    Inicializa(cocina,0);  
    Inicializa(comer,0);  
    Cobegin  
        Caníbal();  
        Cocinero();  
    Coend;  
}
```

3. En un cuartel hay un comedor para 500 soldados. El soldado cuando quiere comer entra en el recinto y coge una bandeja con comida en uno de los 5 mostradores que existen para tal efecto; la bandeja tiene un vaso de agua o un botellín de refresco, si escoge esto último necesita uno de los 50 abridores. Si quiere postre se dirige a uno de los 3 mostradores que lo despachan; Cuando finaliza la comida sale del recinto. Realizar un programa concurrente de forma que utilizando semáforos coordine las tareas de los soldados.

TSemaforo recinto, comida, abridor, postre;

```
Void soldado() {
    Bool babrir, bpostre;

    While(true) {
        P(recinto);
        Entrar_recinto();
        P(comida);
        Coger_bandeja();
        V(comida);
        Cin >> babrir;
        If(babrir) {
            P(abridor);
            Abrir_botellin();
            V(abridor);
        }
        Cin >> bpostre;
        If(bpostre) {
            P(postre);
            Comer_postre();
            V(postre);
        }
        Comer();
        V(recinto);
    }
}
```

```
Void main() {
    Inicializar(recinto,500);
    Inicializar(comida,5);
    Inicializar(abridor,50);
    Inicializar(postre,3);
    Cobegin
        Soldado();
    Coend;
}
```

4. Los directores de un complejo turístico nos piden que regulemos con semáforos el acceso de los turistas a los templos a visitar desde una sala de exposiciones. Para acceder a los templos es necesario esperar en la sala a que venga a buscarlos un guía. El número total de guías es G. Si un visitante quiere acceder a los templos y hay más gente esperando a que venga un guía, deberá hacer cola. La visita es personalizada, es decir, cada guía se lleva sólo a un visitante. Si un guía está disponible y no hay visitantes esperando a la visita de los templos, el guía descansará.

Nota.: Visitante podría ser un semáforo binario.

TSemaforo guía, visitante;

```
Void visitante() {  
    While(true) {  
        V(visitante);  
        P(guia);  
        Seguir_al_guia();  
        Ver_los_templos();  
    }  
}
```

```
Void guía() {  
    While(true) {  
        P(visitante);  
        V(guia);  
        Explicar_templo();  
    }  
}
```

```
Void main() {  
    Inicializar(visitante,0);  
    Inicializar(guía,G);  
    Cobegin  
        Visitante();  
        Guía();  
    Coend;  
}
```